Performance Studies of File System Design Choices for Two Concurrent Processing Paradigms

LEE, Yong Woo

Ph.D.

University of Edinburgh

1996

Abstract

This thesis studies comparative file access performances in distributed file systems and in shared memory systems. The three major changes in computing practice - computer communication speed growth, computing power growth and transaction size growth - have influenced the file access performance of the two computing paradigms. This study investigates the effect of the three on the file access performance in the two system paradigms using the validated virtual performance models. This study investigates the file access performance of the various design alternatives such as multiple CPUs, multiple disks, multiple networks, multiple file servers, enhanced concurrency, caching, local processing, etc. and discusses the various file system design issues in the two system paradigms in terms of the file access performance. Theoretical limits of the file access performance are investigated in many cases. The effect of the workload characteristics such as workload pattern, workload fluctuation, transaction size, etc. on the file access performance is quantitatively evaluated in the two system paradigms.

This study proposes the virtual server concept based on queueing network theory for the performance modelling and presents virtual server models for the two system paradigms. The models which were used are found to predict the file access performances of the real systems very precisely. A parameterization methodology is proposed to obtain the performance parameters and their values. A workload characterization methodology is proposed which consists of the six steps. Six realistic and representative artificial workloads are obtained. Simulation is used as the main methodology and an analytic modelling approach is used as an auxiliary method to solve the performance models in this research. The simulation results are compared with the analytic solutions case by case to confirm that the two are exactly the same as each other. This study performed the standalone measurement experiments and the real world measurement experiments in the two system paradigms to validate the performance models and the simulation results and to obtain the parameter values.

Acknowledgement

I believe God guided me to this final destination through a long journey. However, the journey to this degree would not have been successful without the help and encouragement of many people.

Professor Sidney Michaelson, my second supervisor, inspired the theme of this thesis, had continuously encouraged the progress, offered me warm and kind advice and arranged the research environment and is now in my mind. I do not forget them at all. I am so sorry that I can not report this degree to Mrs. Kitty Michaelson. Her hospitality, encouragement, warm and humane heart are still soft in my mind. Dr. Alex Wight, my first supervisor made the progress of the journey possible. I have to say that without his patience and sincere guide for this degree I would not have seen the destination. His family, Mrs. Christine Wight and her two daughters offered their warm hospitality whenever I visited Edinburgh during my off-campus period. I do not forget Professor Gordon Plotkin's arranging the study chance as a Ph.D. student in the department, his paying attention to my study, encouraging and advising. I believe that it is my fortune to have met them in Edinburgh.

When I think the research environment in The Computer Science Department, what Isaac Newton, a great physicist, said comes in my mind : I could see very far and very well since I stood on the shoulder of the Giant of The Computer Science Department. I remember the discussions and the talks with Dr. Alan Crawford, Mr. Claudio Calvelli, the roommates and others in the department were helpful to my study and my life in U.K. and the jokes we exchanged got rid of much of the stress. Miss Eleaner Kerse has offered much kind and warm care as a departmental secretary.

This study was partially supported by a joint scholarship program by British Government and Korean Government(Ministry of Science and Technology) and by the CRAY research fund of SERI/KIST. I give my sincere thanks to them for their giving this study opportunity. I appreciate very much the kindness and warmth which Miss Monica Paterson in the British Council in Edinburgh and Miss Sun-Hee Han in the British Council in Korea offered. Mr. Joong-Il Park, the Director of my working division of SERI/KIST until I return to Korea took care of this study so much. Previous Administration Director Mr. Joo-sub Choi helped this study very much. I wish to express my deep gratitude to them. I remember Mr. Sang-whoon Yoon kindly helped my father whenever he visited SERI/KIST while I was outside Korea. I also give my thanks to Dr. Dan-Hyoung Lee, the Senior Director of SERI. I wish to give my sincere thanks to Bank of Scotland that offered me a high level consulting staff position and tried hard to get a work permit for me more than 5 months when I was in financial difficulty. Its humane effort will remain in my mind even though it failed to get it.

Professor Jin-Hyung Kim of the Computer Science Department of KAIST took care of me all the time during my study. I wish to express my deep gratitude to him. Members of the Korean Society in Edinburgh, especially Dr. Soon-Aae Yang and Mr. Nam-Seog Park in the AI Dept and Mr. Sang-Ji Jin in the Computer Science Department took care of the occasional visit of mine in Edinburgh during my off-campus period. Church Minister Sung-Il Choi and his family who lived on the upstair of the flat at Morningside, Professor Song-Cheon Moon of the Computer Science Department of KAIST and his family who handed over the flat at Morningside to us where they had lived before us and Mr. In-Boo Hahn, the President of Korean Society in Scotland when I was in Edinburgh helped our life in Edinburgh. Many kind British people in Edinburgh helped us to manage our life in Edinburgh : especially Mr. Philip McDowell and his family and other nice people in Barclay Church. Mr. Yeon-Rhye Kim, a retired Captain of Korean Navy, and his wife Mrs. Se-Ok Ann and the family of Young-Min Kweon sincerely took care of us during the painful period in Australia.

I could stay outside Korea since my elder sister, Mrs. Hae-Sook Lee, my niece Sng-Mi Lee, Jeong-Wha Lee, Mia Lee, Ji-Young Lee and my nephew Jeong-Hee Lee and Ki-eun Lee took care of my parents instead of us while I and my wife were outside Korea even though they were suffering the pains of losing her husband and their father because of his diabetes. I remember Mia Lee supported my parents, that is, her grand-parents sincerely during the period. I wish to express my deep gratitude to Mrs. Sung-Ja Woo and other neighbours. They took care of my parents sincerely while we were outside Korea. My relative, Captain Yeon-Hong Kyoung who deceased last year and his family gave their warm encouragement. My elder cousin Mr. Park-cheol Kim and Mr. Young-Tak Kim and other relatives gave us warm encouragement. My father-in-law, Mr. Hee-Chul Kim supported this study and encouraged me all the time as a son not as a son-in-law and mother-in-law Mrs. Ng-Nam Park gave us her remote support much.

My tear drops and my heart breaks when I think the period when my father and mother were left behind in Korea while I and my wife stayed outside Korea. Though my father was 85 years old and my mother was suffering the breast cancer, the high blood pressure and the diabetes when I left Korea to start my study in the Edinburgh University, they encouraged my study and never complained. I still remember what my father told me when I left Korea for this study, "Don't worry us and regard your supervisor as me". God and my parents have continuously fed me with the power to break through the difficulties and obstacles. I do not forget those friends and colleagues who helped me even though I do not specify their names here.

I started my study in Edinburgh, went to Sydney, Australia to continue the study, returned to Korea and now I finish it. This extra-ordinary experience drives me to write this lengthy acknowledgement. Recalling the difficulties and obstacles in the journey and thinking the meaning hidden behind them, I and my wife close this acknowledgement.

Declaration

I declare that this thesis was composed by myself, and the work which it describes is my own except where explicitly stated in the text.

LEE, Yong Woo

Table of Contents

5. File Access Performance Evaluation of the Two System Paradigms 147

6. File Access Performance Evaluation of the Design Alternatives in the Two System Paradigms 181

Chapter 1

Introduction

Since Xerox PARC(Palo Alto Research Center) Alto workstation project in early 1970, distributed systems have evolved rapidly. A wide and rapid expansion of the research and development activity in distributed systems has produced a large number of different distributed systems. Recently distributed systems has taken another revolutionary step with the rapidly spreading cluster processing paradigm. According to the dramatic changes in computer technologies, the design of distributed systems has changed a great deal and will continue to change.

Shared memory shared variable systems are now widely used with the help of innovative technological advances in the CPU, the main memory and the secondary disk storage. Sometime a shared memory system is used as the file server of a distributed file system. This is a coming together of these two different paradigms which have attracted great interest.

It may be necessary for us to redesign the distributed file systems or the shared memory systems if the trend in computing practices changes and the underlying technologies advance. The computer communication speed has improved rapidly. The computing power growth has been remarkable. More powerful CPUs and larger capacity memory chips have been introduced regularly. The disk I/O subsystem has also improved slowly but steadily. The data size which users ask computers to process has also steadily increased as the network speed and the computing power have improved. New innovative applications generate a lot of data nowaday and it seems that the data size will grow faster and faster.

All of these drive me to evaluate the file access performances of the various design alternatives in the two system paradigms and to evaluate the effect of the influential changes in computing practice on the file access performance in comparative ways in this research.

1.1 Objectives and Research Problems

This study has the following main objectives.

The first objective is to comparatively evaluate the file access performances of the two system paradigms using currently available systems.
All objectives of this research are pursued in comparative ways in the two different system paradigms, that is, the distributed file systems and the shared memory systems.

The second objective is to explore the file system design issues.
What are the design issues? What are the problems in the file access performances of the two system paradigms and how do we improve the file access performances of the two system paradigms? To answer them, this study evaluates the file access performances of the various design alternatives comparatively in the two different system paradigms. The design alternatives with various caching mechanisms, multiple resources such as multiple CPUs, multiple disks, multiple networks, multiple file servers, etc. are evaluated in terms of the file access performance. Multiple processing using the shared memory systems as the component systems in the distributed file system paradigm is also evaluated in

terms of the file access performance.

The third objective is to evaluate the effect of the changes in computing practice on the file access performance.
What technological advances and changes in computing practice influence the file access performance? How much do they affect the file access performance? The candidates are computer communication speed growth, CPU power growth, disk I/O speed growth, transaction size growth, RPC mechanism enhancement, file system mechanism enhancement, enhancement of the degree of the concurrency during the communication and the disk I/O, etc..

The fourth objective is to quantitatively evaluate the effect of the workload characteristics on the file access performance.
How much do the workload characteristics such as the transaction size, the workload patterns, the workload fluctuation, etc. influence the file access performance in the two system paradigms?

This study seeks answers to the questions which center on the above research objectives. In order to achieve the above research objectives, a number of research problems have to be solved beforehand. Listed below are some of them.

1) How to accurately and efficiently model the two computer system paradigms using queueing network theory?
The performance models should be simple and flexible to allow easy modification and yet accurate.

2) What performance parameters will this study use for the performance models and how are parameter values obtained?
The parameterization methodology should be easy to be performed and should produce accurate parameter values.

3) How to obtain accurate, realistic and representative artificial workloads for the performance models from the real measured workloads in the two system paradigms?

Where can I get real measured data? How do I measure them? How do I prove the measured data are general and useful data? How to process the measured data? How do I prove the constructed artificial workloads are accurate, representative and realistic?

4) How to solve the performance models?

Is the methodology to solve the performance models easy to be used and is the amount of the required effort to get the solutions reasonable? What performance metrics will this study use? Are the solutions precise?

5) How to verify the simulation programs?

It is required to verify that the performance models are correctly implemented into the simulation programs.

6) How to measure the real performance and validate the performance models?

The measurement should be carefully designed to be used for the performance parameterization and for the validation.

1.2 Organization

This dissertation is organized according to the progress of this research.

Chapter 2 presents the taxonomies of the concurrent processing systems and in the taxonomies locates the two system paradigms which are studied in this thesis. The detailed description of the two system paradigms follows. First, the cluster processing paradigm is described. Second, chapter 2 defines what is a distributed system by giving some essential characteristics of distributed systems, presents the classification of distributed systems by surveying the past and present distributed systems and gives the point of view of the future distributed systems using the classification. Third, the shared memory processing paradigm is described. Fourth, chapter 2 describes the file systems which are evaluated in this study.

Three major changes in computing practice which have influenced on the file access performances of the two computing paradigms are discussed. First, the trend of the computer communication speed growth is discussed. For it, the five computer communication generations are defined and past, present and future computer communication networks are classified into generations. Second, the trend of the computing power growth is investigated in three components : the CPU, the memory and the disk. Finally, the trend of the transaction(data) size growth is discussed.

Chapter 3 describes the internal details of the two system paradigms, presents the virtual server performance models for them, describes the parameterization work and explains how I characterize the workloads used in this study from the real measured workloads. What performance metrics this study uses and how this study solves the performance models are also explained.

Chapter 4 describes the real performance measurement work to obtain the performance parameter values and to validate the simulations and the performance models.

Chapter 5 evaluates the file access performances of the two different concurrent processing paradigms and discusses the effect of local processing on the remote file access performance.

Chapter 6 investigates the file access performances of the various design alternatives and the effect of the influential changes in computing practice on the file access performance comparatively in the two system paradigms using the validated virtual performance models. Design issues are also discussed in this chapter in terms of the file access performance.

Chapter 7 evaluates the file access performances of various caching mechanisms in the two different concurrent processing paradigms.

Finally chapter 8 concludes this study by summarizing the major results, highlights the main contribution of this thesis and discusses the remaining research tasks.

Chapter 2

Concurrent Processing Paradigms and Influential Changes in Computing Practices

This chapter presents various taxonomies and locates the target paradigms in them. Table 2.1 shows my classification of the processing paradigms using the mapping concept in Mathematics. The most simple paradigm is mapping one process to one processor exclusively. This is the single processing paradigm. There can be no concurrent processing in the paradigm. The concurrent programming paradigm can be further classified into three processing paradigms. They are the multi-programming paradigm, the multiple processing paradigm(concurrent processing paradigm) and the hybrid form processing paradigm. In the multi-programming paradigm which is also known as the processor sharing paradigm or the time sharing paradigm, many processes share one processor : each process uses the processor during the time quantum given to it. In the multiple processing paradigm, many processors are used at the same time to process many processes and each process exclusively uses one or more processors. Multiple processing can be further classified into two processing paradigms. They are the parallel processing paradigm and the sequentially multiple processing paradigm. In the parallel processing, one process is divided into multiple sub-processes and each sub-process is processed in a different processor. This contrasts with the sequential processing paradigm. In the hybrid form processing paradigm, the multi-programming paradigm and the multiple processing paradigm are used together. Therefore many processes can share a processor as well as one process can use multiple processors. This research does not deal with the parallel processing in the file access operations except when it is explicitly mentioned.

Processing mechanisms			# of the processes	# of the processors
Single processing			one	one
Concurrent Programming	Multiprogramming(*1)		many	one
	Concurrent Processing or Multiple Processing	Parallel Processing (*2)	one	many
		Sequential processing	many	many
	Hybrid form processing		many	many

*1 : Concurrent programming in one processor
*2 : Concurrent programming in one process

Table 2.1 : What is concurrent programming?

Flynn's taxonomy of the computation models in table 2.2 has been widely used in classifying computer systems[FLYNN 72]. It is based on the architectural difference of computer systems. Flynn classifies the Von Neuman model as the SISD(Single Instruction Single Data Stream) computer system. The SIMD(Single Instruction Multiple Data stream) computer systems include vector machines, array machines, and massively parallel machines such as DAP and Connection machines. It is known that as yet no MISD(Multiple Instruction Single Data stream) computer system has appeared.

	Number of the data streams	
Number of the instructions	Single	Multiple
Single	SISD	SIMD
Multiple	MISD	MIMD

Table 2.2 : Flynn's taxonomy.

Most multiprocessor systems and multicomputer systems are classified as the MIMD(Multiple Instruction Multiple Data stream) computer systems. The MIMD computer systems can be further classified into several subclasses. According to the degree of interaction in the main memory and the number of the operating systems to control the entire MIMD system, the MIMD computer systems can be classified into tightly coupled systems and loosely couple systems. The MIMD computer systems can be also classified into shared memory shared variable computer systems, distributed memory message passing computer systems and hybrid form computer systems which have shared memory architecture and use the message passing mechanism[KARP 89]. The supercomputers which have multiple vector processors such as Cray X-MP, Cray Y-MP, Cray 2 and the symmetric MIMD computer systems such as Sequent Symmetry systems are classified into the shared memory machines which use shared variables for interprocess communication and synchronization. The shared memory systems can be further classified into multi-port memory systems, crossbar switch connected systems, shared bus systems, multi-stage network connected systems, etc. according to the used inter-connection method. Since early 1980s, multicomputer architectures have emerged, in which each computer has its own non-shared private memory and uses the message passing mechanism for the interprocess communication and the synchronization. They are called the distributed memory message passing computer systems. The hypercube multi-computers such as NCUBE, FPS and T-series, the transputer based multi-computers such as MEIKO surface systems and non-Von Neumann architectures such as the data flow machines of MIT and Manchester University are examples of distributed memory message passing computer systems. The BBN butterfly is an example of the hybrid form computer systems. Some authors [HOWE etal 87],[BELL 89] classify the hybrid form computer systems into the shared memory computer systems. Johnson[JOHNSON 88] classifies the MIMD computer systems into more complete classes as follows.

1. GMSV(Global Memory Shared Variables) computer systems : same category as the shared memory shared variable computer systems which were explained.
2. DMSV(Distributed Memory Shared Variables) computer systems : new category proposed by [JOHNSON 88]. The systems have distributed memory and use the shared variables for interprocess communication and synchronization.
3. DMMP(Distributed Memory Message Passing) computer systems : same category as the distributed memory message passing computer systems which were explained.
4. GMMP(Global Memory Message Passing) computer systems : same category as the hybrid form computer systems which were explained.

Bell[BELL 89] classifies the distributed systems into the DMMP computer systems. This thesis deals with the GMSV and the DMMP computer systems.

Let's look at some other possible classifications. We can classify the computer systems into centralized systems and distributed systems(decentralized systems). This thesis deals with the two paradigms. According to the computing power, computer systems are often classified into supercomputers, mainframe computers, super-mini or mini-super computers, minicomputers, microcomputers, workstations and personal computers. It is difficult to define the category or the range of each class and one computer system is often classified into different categories or classes according to the classifier's own point of view. Perry et al.[PERRY etal 89] define the supercomputers considering three factors. This definition is usually used by supercomputer architects and engineers. Bell[BELL 89],[BELL 93] defines the supercomputer considering four factors. The common three factors between them are the capability to solve intensively numerical computations, scalar and vector processing speed and price. This thesis covers all classes of computer systems in the classification. According to the usage and the purpose for which the computer systems are best suited, computer systems can be categorized into general purpose computers and special purpose computers. Bell[BELL 89] adds one more category to these two categories. It is the category of the run time defined application specific computers. The multiprogramming computers which can handle various applications at the same time fall into the general purpose computers. The special purpose computers are dedicated for limited purpose or applications. According to the characteristics of the operating system, computers can be classified into batch(background) processing oriented systems, interactive(foreground) processing oriented systems, transaction oriented systems and real time processing systems. IBM MVS systems are an example of the batch processing oriented systems because most of process processing can be done easily in batch mode using JCL(Job Control Language) rather than in interactive mode, even though they support interactive jobs as well. UNIX, IBM VM/CMS, PC operating systems such as DOS and WINDOWS 95 are examples of the interactive job processing oriented systems even though they support batch processing as well. IBM ACP(Airline Control Program) is a typical example of transaction oriented systems which has been used mainly in airline companies. Real time processing systems are mainly used to control machines in real time which require exact on-time operations automatically such as some Hewlett Packard factory machines. This study uses standard UNIX systems and their variants as the target systems. However, this study focuses on both interactive jobs and batch jobs.

2.1 The Cluster Processing Paradigm

Since late 1980s, we have seen intense effort to use a cluster of workstations which are networked together as a virtually single computational resource. We call this kind of computing paradigm the computational cluster or cluster computing or cluster processing. Usually the cluster of workstations are connected via a local area network and the message passing mechanism is commonly used for the inter-process communication. Unlike the traditional distributed systems which will be discussed in detail in section 2.2, usually the cluster of workstations fully maintains the integrity of the participating computer systems. Simply adding some software modules a computer system becomes eligible to be a member of the cluster. Without interrupting the operation of the cluster a member of the cluster can withdraw from the cluster. A member of the cluster can selectively cooperate with other members of the cluster case by case. Heterogeneity is usually allowed in hardware and sometimes in system software. The paradigm is often evaluated as a more advanced paradigm than the traditional distributed system paradigm in many aspects. The following terms has concepts similar to each other. Computational cluster, cluster computing, network-based concurrent computing, Piranha computing, workstation farms, heterogeneous computing, hypercomputing, ensemble computing, meta-system, ultracomputing and virtual heterogeneous computing. Cluster management software such as PVM, P4, Linda, MPI, Condor, DQS, NQS, etc. is readily available in a wide range of computer systems for cluster computing. The computational cluster usually requires a distributed file system for efficient operation. This study can be directly applied to the cluster computing paradigm which can be regarded as a superset of the distributed processing paradigm.

2.2 The Distributed Processing Paradigm

This chapter looks at the definition of the distributed system and the classification of the distributed system. This research is interested in the distributed file systems or the file systems of the distributed systems and the following discussion focuses on them.

2.2.1 Definition

In certain cases, the distinction between the distributed systems and the sophisticated variants of the centralized systems seems ambiguous and it is worthwhile to make clear the definition or the characteristics of the distributed systems.

Lelann[LELANN 81] explains some characteristics of the distributed systems. Here I define the distributed systems as the computer systems which have the characteristics of autonomy(independency), geographical distribution, location transparency(seamlessness) and sharing information and resources.

First, we look at the autonomy characteristic. In the distributed systems, each component system has its own autonomy. By autonomy I mean that each component can be an independent computer system as it wants or as the connection to other system breaks down due to an error or an accident(fault tolerancy) as well as having its own system components such as the processor, the memory, etc..

Second, we look at the geographical distribution characteristic. Most of the distributed systems span the distance which local terminals of centralized systems cannot span[1], typically over LAN(Local Area Network) but a few over WAN(Wide Area Network). This characteristic distinguishes the distributed systems from some loosely coupled MIMD systems.

Third, we look at the location transparency(seamlessness) characteristic. Genuine distributed systems enable the users to share information or resources without distinguishing their locations and the users do not recognize where the service is actually processed for them in the distributed systems. This characteristic distinguishes the distributed systems from the computer systems which integrate the centralized systems together in simple ways.

Fourth, we look at the characteristic of sharing the information and the resources.

(1) They do not span more than 200 feet in most cases.

The distributed systems adopt a typical characteristic and benefit of centralized systems, sharing the information and the resources. The degree of sharing of information and resources varies from a distributed system to a distributed system. The distributed operating systems share everything together, in both information and resources, but the distributed file systems share information and the disk resource.

Keeping all these characteristics together efficiently in a distributed system is very difficult and requires more research endeavor. For example, emphasizing sharing information and resources too much can easily lead to less autonomy(independency) and keeping location transparency through long geographical distance over WAN is not easy at all.

2.2.2 Past, Present and Future Distributed Systems and Their Classification

There have been several forms of system integrations as networking technologies have evolved. This study classifies the forms of the system integrations into four different categories.

- Inter-connected network systems(inter-networked systems),
- Network operating systems,
- Distributed file systems,
- Distributed operating systems.

The inter-networked systems give very low level inter-system services such as sending and receiving e-mails and/or at best transferring files using installed file transfer programs.

In networking operating systems, the component system does not share any information or system resource automatically in seamless(location transparent) manners but manually by users' specifications. In the networking operating systems, the information sharing is possible at the level of file transfer using the installed file transfer program in the worst case or at the level of adjoining file system in the best case. The Newcastle connection system[BROWNBRIDGE 82] uses a kind of adjoining file system. It has the superdirectory above the root directories of all connected machines and a user has to specify the superdirectory of the system which has the required file in order to use it. Hence the adjoining file system is not location transparent(seamless) to users.

In the distributed file systems, information sharing is achieved through the file servers in location transparent(seamless) manner to users. Andrew[MORRIS etal 86], [HAWARD 88] and Coda system[SATYANARAYANAN 90B], [SATYANARAYANAN 90C], CFS(Cambridge File Server), SUN/NFS(Network File Server), etc. fall into this category. Levy and Silbershatz[LEVY etal 90], Satyanarayanan[SATYANARAYANAN 90A] and Svobodova[SVOBODOVA 84] survey the distributed file systems.

In the distributed operating systems, information sharing and resource sharing are achieved completely and seamlessly in location transparent ways. To a user it looks like a single centralized system, that is, a virtually single operating system. In order to distinguish distributed message passing operating systems from other types of distributed operating systems, Chandras[CHANDRAS 90] characterizes fully distributed message passing operating systems as distributed operating systems which have the 6 components : local memory management, global processor management, global process management, global protection scheme, global interprocess communication and distributed storage management. Amoeba, CDCS(Cambridge Distributed Computing System), V system, and Mach are examples of such distributed operating systems. Tanenbaum et al.[TANENBAUM etal 85] survey distributed operating systems.

Here when I say distributed systems, I mean distributed file systems or distributed operating systems because the former two system categories - the inter-networked systems and the network operating systems - do not satisfy the definition of distributed systems.

Information sharing can be currently achieved in 3 ways : no merge at all, adjoining file systems and file servers. If there is no merge of the file systems but there is some information sharing, it is usually achieved through the file transfer program such as uucp or ftp. As explained before, the Newcastle connection system is an example of having adjoining file systems in order to achieve information sharing. Having the file server to support information sharing is the approach of the distributed file system and the most advanced available mechanism to achieve information sharing. The distributed file system looks to users like a single global file system or a single virtual file system.

Current distributed systems can be classified into 4 different architectural models according to the level of each component system. They are the minicomputer model, the workstation model, the processor pool model and the hybrid model. This classification looks similar to [COULORIS etal 88] and [TANENBAUM etal 85], but the definition is different.

In the minicomputer model, the major or target component systems are at the level of minicomputers. LOCUS[POPEK etal 85] was an example of the minicomputer model.2) This study does not classify VAXcluster system[KRONENBERG etal 86] as a minicomputer model, because it covers only

(2) LOCUS was originally developed as an UNIX like distributed operating system written in C in the VAX environment of UCLA, U.S.A.. The project started in early 1970s and the prototype system on PDP-11 was run in 1981. Now it is claimed as a machine independent distributed operating system as a product of LOCUS computing cooperation and classified into the hybrid model of the minicomputer model and the workstation model.

up to 45m using star topology connection(therefore maximum 90m) hence violates the geographical distribution characteristic of the distributed systems.

In the workstation model, the major component systems are at the level of workstations. Most of the distributed file systems fall into this category. They are Andrew and Coda, SUN/NFS, etc.. V distributed operating system[CHERITON etal 83],[CHERITON 84] developed by Stanford University once belonged to this category. Now V system also covers MicroVAX system, and I categorize it into the hybrid model[CHERITON 88].

In the processor pool model, the major component systems are in the form of a processor pool. A processor pool is used by the distributed operating systems as the processor server, motivated by the concept of the file server. The first distributed system in the processor pool model is known to be the CDCS(Cambridge Distributed Computing System) by Cambridge University [NEEDHAM etal 82],[CRAFT 85],[BACON etal 87].[3]

The Amoeba distributed operating system[TANENBAUM etal 85],[TANENBAUM etal 88],[TANENBAUM etal 89],[MULLENDER 89],[MULLENDER etal 90] is an example of the hybrid model which combines the workstation model and the processor pool model.[4] As explained before in this section, LOCUS and V are examples of the hybrid model which combines the minicomputer model and the workstation model[5].

So far this study has classified already developed distributed systems. However, I

[3] In its original processor pool, CDCS had no workstation but a bank of General Automation LSI4 minicomputers and later micro-computers based on M68000 processors with memory to each component processor were added.

[4] Amoeba 4.0 consists of four components. They are workstations, processor pool, specialized servers and gateways for the connection to WAN.

[5] PDP-11 mini-computer systems and SUN workstations can be components in the LOCUS distributed system and SUN workstations and microVAX mini-computers can be components in the V distributed system.

cannot exclude the possibility of exploration of any other architectural model of the distributed systems beyond the classification mentioned above, for example, the mainframe model, the supercomputer model, the graphic processor model, etc. in future.

The distributed systems can be classified into the homogeneous model and the heterogeneous model. Andrew system, Coda system, and SUN/NFS[SANDBERG etal 85] basically belong to the homogeneous model in which each component system is identical or homogeneous. CDCS, V, LOCUS and Amoeba belong to the heterogeneous model. CDCS is a typical example which allows operating system heterogeneity as well as hardware heterogeneity.

According to the topology and the protocol used in networking, current distributed systems can be categorized into two models. They are the model based on Ethernet, a CSMA/CD bus topology LAN and the model based on the Ring topology. Most distributed systems use Ethernet as their LAN. Apollo DOMAIN systems and IBM AIX(IBM version of UNIX)-DS (Distributed System) systems[SAUER etal 87] use token ring based LANs. CDCS and CFS are based on Cambridge ring LAN. Cambridge ring LAN is not a token passing LAN but uses several minipacket slots circulating around ring.

The RPC(Remote Procedure Call) has become the de facto standard for IPC(Inter Processor Communication) in distributed systems. However not all of the distributed systems implement the same RPC. Tay and Ananda[TAY etal 90], [ANANDA etal 93] survey and compare the RPCs in various distributed systems.

2.3 The Shared Memory Processing Paradigm

This study deals with the shared memory processing paradigm which uses shared variables. It belongs to the MIMD paradigm according to Flynn's classification[FLYNN 72] and the GMSV paradigm according to Johnson's classification[JOHNSON 88]. It has shared bus architecture and symmetric property both in the architecture and in the operating system. Sequent symmetry systems are examples of the shared memory processing paradigm. This study considers a computer system which has one processor, for example, a Sun SPARCstation Series Workstation, as a special case of the shared memory processing paradigm, that is, the shared memory processing paradigm with only one processor.

2.4 File Systems

There are many kinds of available file systems. This study deals with UNIX file systems. Many types of file systems are available in current UNIX operating systems. For example, UNIX V 4.2 supports s5(system V file system), ufs(UNIX file system), sfs(secure file system), memfs, vxfs(VERITAS file system), bfs(boot file system), Berkeley file system, etc..[AT&T 94]. The structure of the ufs file system is more complex than that of the s5 file system. The sfs file system is a variant of the ufs file system. The vxfs file system is an extent based high integrity file system. The bfs file system is a special purpose file system which contains all stand-alone programs necessary for boot procedures. The memfs file system is a high performance volatile memory file system which resides in memory and when it is unmounted, the directories and the files disappear. This study deals with commonly used standard file systems among them. The detailed structure and logic of the distributed file system will be explained in section 4.1 and that of the file system of the shared memory will be explained in section 4.3. Any file system that follows the structure and the logic explained in section 4.1 and section 4.3 is the target file system of this study.

2.5 Computer Communication Speed Growth

It is true that the popular use of distributed file systems has influenced the computer communication speed growth. It is also true that the computer communication speed growth has very much influenced the distributed file systems. Therefore this study looks at the trend of the computer communication speed growth in past, present and future computer communication networks.

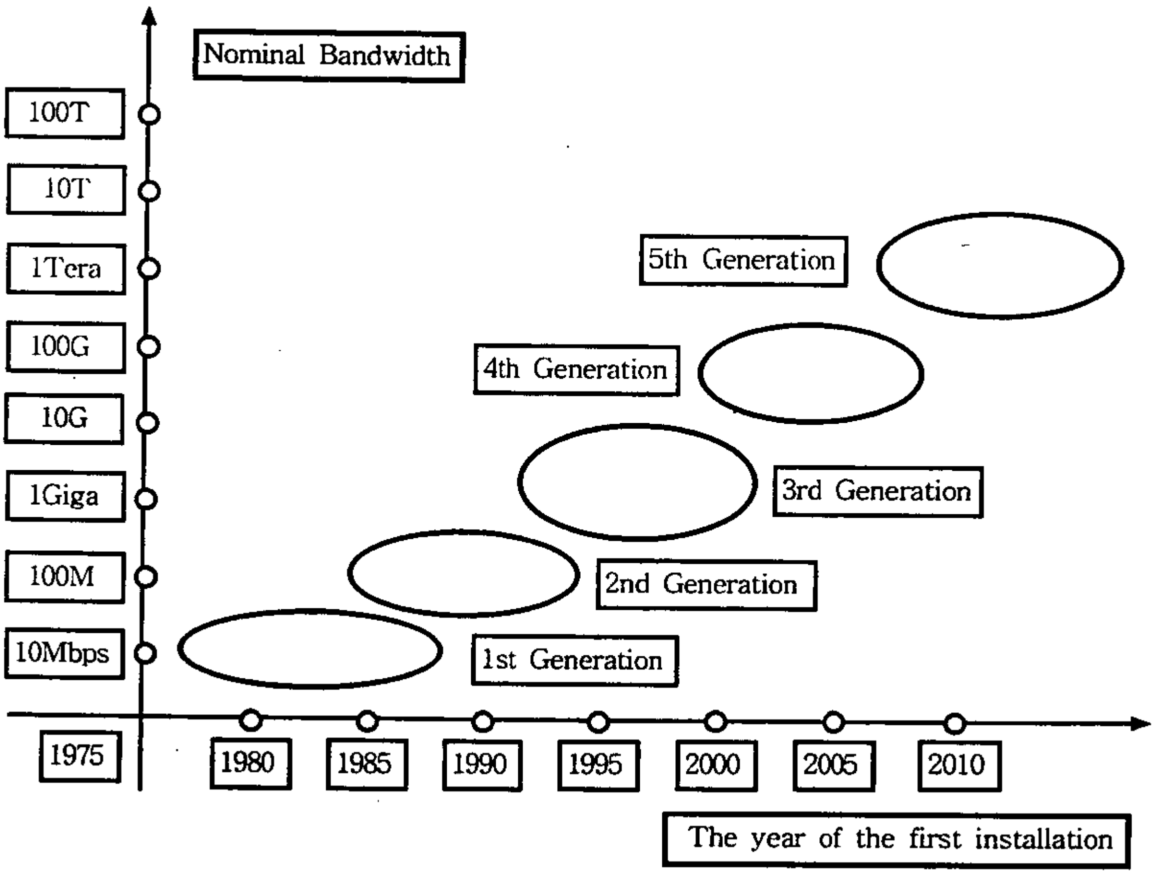


Figure 2.5.1 : The computer communication speed growth

When we say high speed computer communication, we usually think of the range of hundred of Mbps to tens of Gbps nowadays. Three factors are expected to accelerate the computer communication speed growth. First, open system connectivity is expected to accelerate the demand for high speed and high performance computer communication. Second, multi-media services are expected to accelerate the demand for large communication bandwidth. Third, various innovative network services via the Internet such as teleconference, home shopping, remote education, remote medical service, home office service, home banking, etc. are expected to accelerate the demand for the high speed communication network.

Nowadays Internet and WWW(World Wide Web) are very widely used and continue to attract growing attention from all over the world. Therefore the current trend toward WAN based distributed file systems via the Internet with WWW stresses the importance of future research in WAN based distributed file systems even though this study focuses on LAN based distributed file systems.

Below, I classify the computer communication network into five generations mainly according to the speed. Figure 2.5.1 shows the computer communication speed growth.

The first generation computer communication network centers on 10Mbps local area networks such as 10Mbps Ethernet, token-ring local area network, etc.. Mainly text data are manipulated. Stallings[STALLINGS 84] surveys the local area networks which belong to the first generation network. This study very briefly looks at the first generation local area network below since the measurement experiments in chapter 3 and chapter 4 and the baseline distributed file systems in chapter 5, chapter 6 and chapter 7 use the first generation network. Three typical topologies of local networks are star, ring and bus/tree topology : the bus is often treated as a special case of the tree which has only one trunk and no branch. Three kinds of data transfer techniques are currently used : dedicated access, switched access and multiple access. Three typical transmission media used in local networks are twisted pair wire, coaxial cable and optical fiber. There are two typical transmission techniques for local networks. They are baseband and broadband. The baseband technique uses digital signaling and broadband technique uses analog signaling in the range of radio frequency(RF). Current baseband systems can be further classified into coaxial baseband systems and twisted pair baseband systems. The broadband systems can be further classified into FDM(Frequency Division Multiplexing) broadband systems and single channel broadband systems. Many local area networks use bus/tree topologies. Most LANs based on bus/tree topology use the medium access control protocol of CSMA(Carrier Sense Multiple Access)/CD(Collision Detection) which is also referred to as LWT(Listen While Talk) protocol. Ethernet[METCALFE etal 76] uses 1-persistent CSMA/CD protocol. Ethernet was originally developed in 1973, redesigned in the early 1980s and became to be widely used in the mid-1980s. Typical Ethernet uses baseband 50ohms coaxial cable and has the nominal data rate of 10Mbps and standard cable length limit of 500meters. Now 100Mbps Ethernet is commercially available. Many distributed systems use Ethernet as their LANs. HYPERchannel[CHRISTENSEN 79] has the nominal date transfer rate of 50Mbps. It uses a prioritized CSMA(or LBT : Listen Before Talk) protocol. The past and present LANs based on ring topology can be classified into token rings, register insertion rings and slotted rings. Standard IBM token rings have had the data transfer rate of 4Mbps with the signaling rate of 8MHz. On November 1989, IBM began to supply 16Mbps token ring with the signaling rate of 32MHz. OTF(Open Token Foundation), an industry wide consortium has supported IEEE 802.3 based token rings. Other venders have supplied 10Mbps token rings(Proteon and Apollo) and 80Mbps token rings(Proteon). In token ring, there is no limit for the packet size. Cambridge ring LAN is not a token passing LAN but uses several minipacket slots circulating around the ring. Each minipacket has two bytes data and 3 bytes communication overhead : flag bits, source bits and destination bits. The nominal bandwidth of the old Cambridge ring LAN is known to be 10Mbps and effective bandwidth is 4Mbps from the simple calculation of

$$\textit{nominal bandwidth} \times \frac{\textit{data transferred}}{\textit{data transferred} + \textit{communication overhead}}.$$

Token bus rings use ring topology logically and are based on bus/tree topology physically. IEEE 802 committee specifies standards for LANs : IEEE 802.3 for the 1 persistent CSMA/CD, IEEE 802.4 for the token bus, and IEEE 802.5 for the token ring.

The second generation computer communication network centers on 100Mbps local area networks such as 100/200Mbps HDDI, 100Mbps Ethernet, etc. The multi-media service has coincidentally emerged while the second generation network has been commercially available. Abeysundara and Kamal[ABEYSUNDARA etal 91] survey the local area networks which belong to the second generation network. The communication speeds of the following second generation .local area networks are between 50Mbps and 200Mbps. Expressnet, Fastnet, D-Net, Buzz-Net, Tokenless Protocols, Distributed Queue Dual Bus, Z-Net, and X-Net are the bus topology based local area networks. Cambridge Fast Ring and FDDI are the ring topology based local area networks. Hubnet, Collision Avoidance Multiple Broadcast Tree, Tree-Net, and Tinker-tree are star and tree topology based local area networks. Multichannel CSMA Networks, Multihop Networks and Mesh Networks are multi-channel local area networks.

This study very briefly looks at the the FDDI since the measurement experiments in chapter 4 and chapter 5 were performed in a local area network where the FDDI was used as the backbone local area network from floor to floor. FDDI(Fiber Distributed Data Interfaces)[BURR 86],[JOSHI 86],[ROSS 86],[ROSS etal 90],[DAVIDS etal 94] and FDDI-II are local area networks based on token ring mechanism. Two fiber counter-rotating rings are used so that when either one breaks the other can be used as a backup to provide fault tolerancy. They run at the speed of 100Mbps over optical fiber media. FDDI uses multimode fibers because the additional expense of single mode fibers is not needed for networks running at only 100Mbps. Error rate is less than 1 error in 2.5×10^{10} bits. A multi-mode fiber links up to 2km and a single mode fiber links up to at least 60km on a private fiber[ROSS etal 90]. The effective sustained data transfer rate at the data link layer is claimed over 95% of the peak rate of 100Mbps[ROSS etal 90]. The FDDI standard assumes a maximum of 100km and a maximum configuration of 500 nodes on a dual ring[LANG etal 90]. FDDI is originally developed in 1982. Now it is widely used.

The third generation computer communication network offers the network speed of from several Gbps to several tens of Gbps such as Ultra-net, STM-16(2.5Gbps), OC-48(10Gbps) and STM-64(10Gbps). The multi-media services are expected to be mature in the third generation network. As[AS 94] surveys the third generation network. Heidemann et al.[HEIDEMANN etal 91] outline the technologies for the 10 to 40 Gbps networks. FFOL(FDDI Follow-On LAN) is being developed now by the X3T9.5, the Accredited Standards Committee task group. The FFOL is expected to have the data rate of at least 600Mbps, but less than 1.25Gbps.

As[AS 94] surveys the fourth and fifth generation networks and protocols as well. The fourth generation computer communication network centers on hundreds of Gbps networks. The fifth generation computer communication network centers on several Tera-bps networks. Some networks and protocols are claimed to be able to accomodate up to Tera-bps network traffics. They are Photonic star network with random access protocols such as random access, PAC(Protection Against Collision), QUADRO(Queueing Arrivals for Delayed Reception or Routing), token passing protocols and reservation protocols, Photonic bus networks such as AMTRAN, FairNet, RATO-net and EQEB, Photonic ring networks such as PIPELINE and

Photonic mesh networks such as ShuffleNet, WON, MONET, MUltihop-Star, PBNet, Bus-Mesh, network, SIGnet and BlazeNet.

In this research, the baseline distributed file systems use the 10Mbps Ethernet with 100Mbps FDDI as the backbone network. This study analyzes the effect of the communication speed growth on the file access performance in the distributed file systems. That is, this study investigates the file access performance of the distributed file systems while the network speed is gradually increased up to the infinitely fast network, the theoretical limit, beyond the fifth generation network.

2.6 Computing Power Growth

Three major components of the computer systems are the CPU, the memory and the peripheral devices. This study looks at the computing power growth by looking at the growth of the power of each component of the computer systems.

The CPU speed has increased in a factor of 4 improvement every 5 years. In the early 1970s, the CPU speed was around 200Khz. In 1990, the CPU speed was around 50Mhz. In 1995, the CPU speed was near 200Mhz.

The memory chip capacity has improved in a factor of 4 improvement every 3 years. The 1Kbits memory chip was available in the early 1970s, 4Kbits in 1975, 16Kbits in late 1970s, 64Kbits in early 1980s, 256Kbits in 1984, 1Mbits in late 1980s, 4Mbits in 1990, 16Mbits in early 1993, 64Mbits in 1994, 256Mbits in 1994. Samples of 1Gigabits memory chips and samples of 4Gigabits memory chips were presented in 1995. Now 16Gigabits memory chips are being competitively developed. The memory access speed has been also improved during the last 25 years. The capacity and the speed of the cache memory have been also improved.

The disk capacity and the disk I/O speed of the disk have been improved but the disk I/O speed is still much lower than the memory access speed. Wood et al.[WOOD etal 93] investigated the disk trend in terms of the cost and performance. Now some innovative disk I/O subsystems such as RAID disk arrays are available[CHEN etal 94], [GANGER etal 94], [ROSARIO etal 94]. The details of the disk I/O subsystems will be presented in section 4.2.3.

This research explores up to the theoretical limits in both computing power and disk performance, that is, this study explores up to infinitely improved computing power and up to infinitely improved disk speed when this study evaluates the effect on the file access performance of the growth in computing power and in disk speed.

2.7 Transaction Size Growth

The average transaction size is usually larger in a high performance system than in a low performance system. We observe the average transaction size growth when we compare the measured average transaction size of Baker et al.'s work[BAKER etal 91] with that of Ousterhout et al.'s work[OUSTERHOUT etal 85]. There is a 5-6 years time gap between Ousterhout et al.'s work and Baker et al.'s work. When we compare the two measured data, we observe the increase of file I/O traffic rate by a factor of 20 to 30, while the computing power increases by a factor of 200 to 500.[6] This study consider an average transaction size up to 1856kbytes when this study evaluates the file access performance of the two system paradigms. It is 232 time larger than the average 8kbytes transaction size. It means that this study considers the transaction size of up to around 2000 to

[6] Ousterhout et al. measured that the data traffic of between 300bytes and 600bytes per second per an active user, when they define active users as those who caused any file i/o during a 10minutes interval and the data traffic of several thousand bytes per second per an active user, when they define active users as those who caused any file i/o during a 10seconds interval. Baker et al. measured the data traffic of average 8Kbytes per second per an active user in the former case, and the data traffic of average 47Kbytes per second per an active user in the latter case.

5000 times more powerful computer systems than the computer system used by Baker et al., which will be explained in section 4.5.2. It has been observed that every five years the price of computer systems falls 10 times.[BELL 89],[BELL 93] It means after 10(15) years, the price will fall 100(1000) times. Therefore, I expect it will take at least 15 years for us to have popular computer systems which is 2000 to 5000 times as powerful as the popular computing systems in 1991. Therefore, I can say the consideration covers the future computer systems up to at least 15 years from 1991 in terms of the transaction size growth.

2.8 Summary

The target paradigms have been located in the various taxonomies presented. The processing paradigms has been classified using the mapping concept in table 2.1. All the cases except parallel processing and hybrid processing in this classification are dealt with in this study. This study focuses on the MIMD computer systems according to Flynn's taxonomy, the distributed memory message passing computer systems and the shared memory shared variable computer systems according to Karp's taxonomy and the GMSV and DMMP computer systems according to Johnson's taxonomy. This study covers the centralized systems and the distributed systems(decentralized system) and all classes of computer systems in the classification according to computing power.

This study uses standard UNIX systems and their variant as target systems and focuses on both interactive jobs and batch jobs. Commonly used standard file systems are dealt with, which means that any file system which follows the structure and the logic explained in section 4.1 and section 4.3 is the target file system of this study.

My definition of the distributed file system is given with 4 characteristics : the autonomy(independence), the geographical distribution, location transparency(seamlessness) and sharing of information and resources. The forms of system integration are classified into 4 different categories : Inter-connected network systems(inter-networked systems), Network operating systems, Distributed file systems and Distributed operating systems. This study does not deal with the first two categories. According to the level of each component system, current distributed systems are classified into 4 different architectural models : the minicomputer model, the workstation model, the processor pool model and the hybrid model. This study covers all of the four models.

Three major influences on the file access performance of the two computing paradigms have been discussed. They are the computer communication speed growth, the computing power growth and the transaction(data) size growth. Computer communication networks are classified into five generations mainly according to the speed. Detailed explanation about the computer communication mechanism and disk I/O mechanism is given in section 3.2.4 and section 3.2.3 respectively. In section 2.5 and in section 3.2.4, I clearly state that this study focuses on the local area network based distributed file systems.

Chapter 3

File System Performance Modeling and Simulation

This chapter describes what kinds of file systems are studied in this research, what performance models are developed and used, how I find the performance parameters, what kinds of workloads are used for the developed performance models as inputs, how I get the workloads, what I use for the performance metrics and how I solve the developed performance models. Other' related work will be discussed where appropriate. Two different file system paradigms, that is, the distributed file system and the file system of the shared memory system are the target of this study. This study separately models and parameterizes the two paradigms. The internal logics are intensively explained to describe the file systems of the two different systems under study. A more realistic, precise and yet convenient performance modelling method and models based on queueing network theory and the virtual server concept are presented. This study also introduces a unique parameterization method which does not require any sophisticated performance measuring tool. Six representative and realistic workloads are extracted from real measured workloads through a carefully developed workload characterization procedure. The six workloads are used to drive both of the two file system performance models in order to compare the file system access performance of the two different system paradigms. A SLAM II simulation package is used to solve the virtual server models. Analytical methods are also used as auxiliary methods. Careful statistical analysis is applied to the simulation results to verify the correctness of the solutions. Almost all possible performance metrics are used in this study.

Section 3.1 describes the logic and the structure of the distributed file systems of which this study evaluates the performance. Section 3.2 describes the virtual performance models of the distributed file systems, the parameterization procedure for the models and the parameters obtained for the models. Separate models for each component e.g. the client, the file server, the disk I/O subsystem and the network communication facilities are investigated individually in section 3.2.1, section 3.2.2, section 3.2.3 and section 3.2.3. Overall models of the distributed file systems are discussed in section 3.2.5. The virtual performance models are explained in section 3.2.6. The performance parameters of the performance models, the parameterization procedure and others' related works are described in section 3.2.7. Section 3.3 describes the file system of the shared memory multiprocessor system under study. The virtual server model for the file system of the shared memory system is described in section 3.4.1 and the performance parameterization procedure and the parameters are described in section 3.4.2. Section 3.5 describes the workload characterization procedure and the workload used in this study. Section 3.6 discusses the performance metrics and which ones have been used in this research. Section 3.7 explains why I choose simulation as the main method to get the solutions of the models in this study and describes details of the simulation.

3.1. The Distributed File Systems under Study

This section describes the distributed file system which is studied. Every effort was made to keep the distributed file system to be a general one.

Each client of the distributed file system under this study has at least a minimal local disk for the local virtual memory management so that the local paging activity(the virtual memory management activity) is not done globally via the remote file server but is done locally in the local disk of each client. It is worth looking at the reasons in more detail why the local disk at each client is assumed to be in the distributed file system under study. Once disks were expensive devices, produced annoying noise and took considerable space in offices. Now disks are relatively cheap and produces much less noise. Compact and high density disks usually reside inside the bodies of the PCs or the workstations. Then thinking purely from the viewpoint of performance, shall we use the reasonable capacity of the local disk for the client of the distributed file system? This study says yes in the design of the distributed file system. In diskless client systems, every file related activity should consult the remotely located file server. Therefore, the initial system booting and the paging in the client should ask the file server to cooperate via LAN. In diskless client systems, the booting can not be done when either LAN or the file server is not operating. This does not allow the client to act with autonomy1). If either LAN or the file server is not operating, paging to and from the remotely located file server can not be performed at all. Neither this does not allow the clients to act with autonomy. Paging via LAN to and from the file server is reported to produce a lot of bursty traffic through LAN to and from the file server.2) Gusella[GUSELLA 90] measured the diskless workstation traffic on an 10Mbps Ethernet in three different groups separately : the character traffic from a diskless workstation to other machines, the paging traffic between the virtual memory of the diskless workstation and the paging device in the remote file server and the file access traffic from the diskless workstation to the remote file server. He reports that the measured paging traffic reached to, at maximum, 20-25% utilization of the Ethernet during one second intervals between a single

(1) See chapter 2 for the autonomy characteristic of the distributed systems.

(2) "The diskless workstation technology may be doomed to limited development in current LANs." without special arrangements due to the bursting paging traffic[GUSELLA 90].

diskless client workstation and the file server. The diskless workstations were equipped with 4Mbytes main memories which are small nowaday. However I agree with the author's view that larger memory sizes will not decrease the level of the paging traffic over Ethernet and the paging traffic will continue to be a major traffic component in future diskless workstation environments in which each workstation has larger main memory. Because users have a tendency to use their workstations with applications which take full advantage of the increased memory, the sizes of applications will increase as the size of memory increases and the paging traffic will increase as the sizes of applications increase. The sizes of applications are also sensitive to the total system power as well as the main memory size. Lazowska et al.[LAZOWSKA etal 86] report that the ratio of the volume of paging traffic to the volume of file access traffic was one to four in the network of diskless SUN-2s with 2Mbytes main memories. Gusella[GUSELLA 90] reports that it was four to one in the network of diskless SUN-2s and SUN-3s with 4Mbytes main memories. Gusella[GUSELLA 90] explains the reason by giving partial attribution to the fact that "UNIX applications were smaller at that time". If a reasonable capacity of the local disk is used in each client, then the clients can have better autonomy and the clients are no longer troubled by the initial booting traffic and the paging traffic. Some locally important files can be also stored in the local disk so that they are guaranteed to be fetched at any time with faster response time regardless of the operational status of the file server. For these reasons, the local disks are assumed to be provided in the clients of the distributed file systems under study. Therefore, the paging traffic is not considered in the following chapters.

The following part of this section describes the internals of the distributed file systems under study by explaining how the requested data from the clients are processed in the distributed file systems.

The requests are generated in the clients by users and they are processed to be sent to the designated server. The requests depart from the clients, traverse LAN and arrive at the file server. In the file server, the requests receive file services, then responses to the requests are made to be sent to the clients. The responses depart from the file server, traverse LAN and arrive at the clients. The clients process the responses to the users. Below, I explain the internal logic of each part, that is, the client, the network and the file server in more detail by describing how the requests from the clients are processed in each part.

In the client, a user issues a request for reading and/or writing files. The CPU of the client processes the request. If a caching mechanism in the client is used and the wanted file is in the cache of the client, then the request is processed locally. Otherwise, the client makes a request of reading and/or writing the remote file from/to the designated file server. The client builds the request using RPC. Figure 3.1.A shows the RPC mechanism.

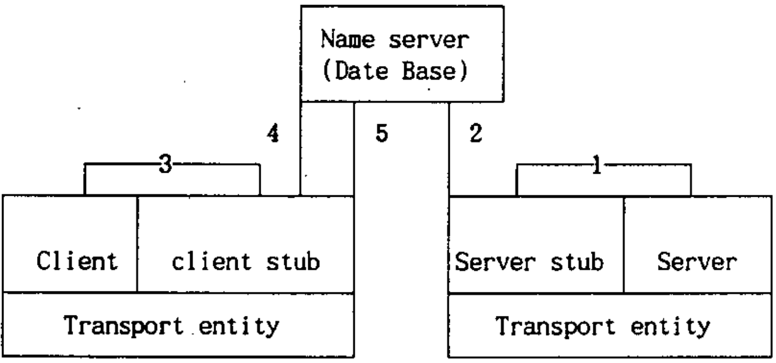


Figure 3.1.A : A RPC mechanism.

When the file server is booted, the file server calls the server stub : an export procedure. The server stub registers with the name server by sending a message containing its name(an ASCII string), its network address and an unique identifier(e.g. a random 32bit integer) : the "naming" procedure. The client calls the client stub : an import procedure. The client stub sends the name of the client and the name of the file server(an ASCII string) to the name server. The name server returns the previously registered network address of the file server and the unique identifier of the file server : the "locating" procedure. The binding procedure consists of the naming procedure and the locating procedure. Subsequent calls do not require the binding any more. The unique identifier is used by the transport entity on the file server machine to determine to which of the file server stubs to give the incoming message. It is also used for the rebind purpose. When the file server reboots after the file server crashes, the file server re-registers with the name server using a new unique identifier number. If the client attempts to communicate with the file server using the old unique identifier, then the client fails to communicate and the client will know a crash happened before. Therefore the client will rebind.

The network interface unit such as the network controller or the network DMA of the client is responsible for sending the request message via LAN to the file server which contains the requested file. In this operation, there can be certain degree of concurrency between the network interface unit of the client and the CPU of the client. This concurrency operation is discussed in detail in section 6.16. After the network communication connection is successfully built between the client and the designated file server, the request message traverses the LAN and arrives at the file server. There can be the operational delay between sending each request message from the client. This delay is called inter-request delay and depends on the characteristics of each distributed file system. The data transmission operations in the network interface via the network are described in detail in section 3.2.4.

The receiving operation in the file server is performed by the network interface unit of the file server. In this operation, there can be a certain degree of concurrency between the network interface unit of the file server and the CPU of the file server. The transferred request is stored in the buffers of the network interface unit. There is a finite number of buffers in the network interface unit and if the buffers are already fully occupied, then incoming request messages are discarded. In this case, the request messages should be retransmitted from the clients. The time spent for the client to retransmit the request message via the LAN to the file server is called retransmission delay time. The buffered request message is sent to the memory of the file server for processing in the file server. The file server fetches the request message and evaluates the request message. Once evaluated, the request is processed in the same way as a local request reading and/or writing local files is processed in the local system. The local processing of the request consists of two distinct operations : the file handling operation and the disk I/O operation.

The file handling operation consists of directory handling, file table lookup, updating file tables, opening files, closing files, etc.. The disk I/O operation consists of disk I/O path setup operation through the disk interface unit, and physical disk I/O operation. The physical I/O operation consists of three major components : the seek, the latency and the transfer. The seek operation is to access the right track of the disk. The latency occurs until the system finds the requested block, that is, when the system puts the requested block under the read/write head. The transfer operation is to read a block of information from the disk to the buffer in the memory or writing it from the buffer in the memory into the disk.

Now the file server makes a response message in response to the request message. The response message is transferred from the memory to the network interface unit for sending. If the finite number of buffers of the network interface unit is already fully occupied, then the file server CPU should wait until the required buffer space is available. This is called requeue delay. The network interface unit of the file server, the CPU of the file server, the network interface unit of the client and the CPU of the client cooperate to setup the communication connection to the client and transfer the queued response message to the client via LAN. The response message departs from the file server, traverses LAN and arrives at the client.

Again in the client, the network interface unit receives the response message in its finite buffers. The received response message is moved to the memory for processing. The client fetches the received response message and evaluates it. Now finally, the pure information or the data processed by the client are sent to the user's window of the client. The user using the client will repeat the above whole life cycle again or do thinking(it is often called as either the user think time or the idle time) or do stand-alone processing(it is often called local processing) for the work in the client.

3.2 Distributed File System Performance Models

Queueing network theory is applied to build the performance model in this study. Why is queueing network theory used? Because there are multiple processes competing each other for the limited system resources in the distributed file systems, queueing and queueing delay become inevitable and it is natural to model the distributed file systems as a network of inter-connected queues. I divide the distributed file systems into 3 parts : the client, the file server and the communication facilities such as the network(LAN) and the network interface unit. This study looks at the performance models of each part and the disk I/O subsystem separately and then the performance models of the whole system. Finally, this study introduces the virtual server models as realistic models.

3.2.1 Models for the Client

The model of the client system naturally depends on the characteristics of the client system. There are usually three kind of client systems : 1) single user single processing systems, 2) single user multiple processing systems, and 3) multiple users multiple processing systems.

MS-DOS based PCs which have Intel 486 processors and Intel Pentium processors are typical examples of the single user single processing systems. Figure 3.2.1.A shows a queueing network model for the single user single processing client systems. There, the CPU is represented as a server without any queue, the disk I/O subsystem is modeled as a server without any queue and the PC screen as a delay server(an infinite server) without any queue. The service time of the screen represents the user think time. Only one process(token) is processed all the time.

Unix based Workstations such as SUN 3, SUN 4 and SUN SPARCstation systems are often used as single user multiprocessing systems[3]. In these systems, a user can have multiple processes through multi-programming using windows or foreground/background processing facilities. They are modelled as figure 3.2.1.B.

Figure 3.2.1.C shows a model of multi-processor workstations. Some current workstations have multi-processors. The multi-user multi-processing systems such as VAX 11/780 systems, Prime EXL320 systems can be also modelled either as figure 3.2.1.B or figure 3.2.1.C. If the systems have multi-processor then they are modelled as figure 3.2.1.C, otherwise, they are modelled as figure 3.2.1.B.

Ferrari et al.[FERRARI etal 83] show another model for VAX 11/780 systems as in figure 3.2.1.D. In the model, a process will (i)use the CPU, (ii)access the disk or display output, (iii)repeat the step(i) and the step(ii) if necessary, (iv)visit the CPU

(3) Multi-programming, but not parallel processing through multi-processor.

and (v)return to the user terminal.

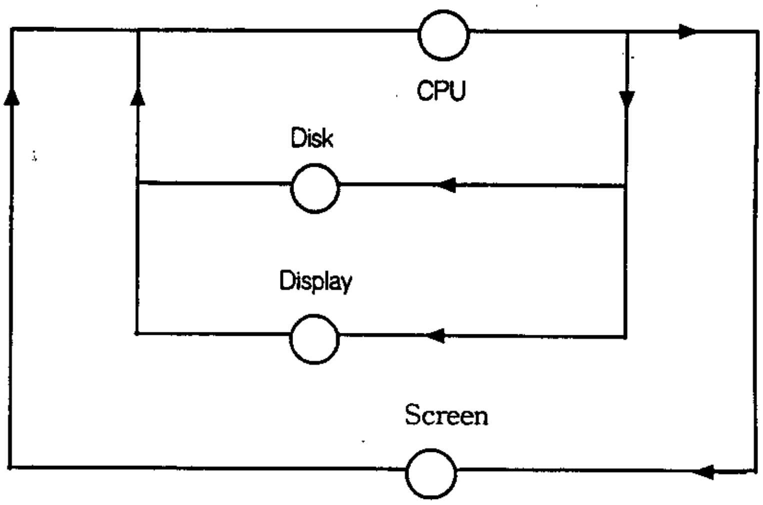


Figure 3.2.1.A : A queueing network model for the single user single processing systems

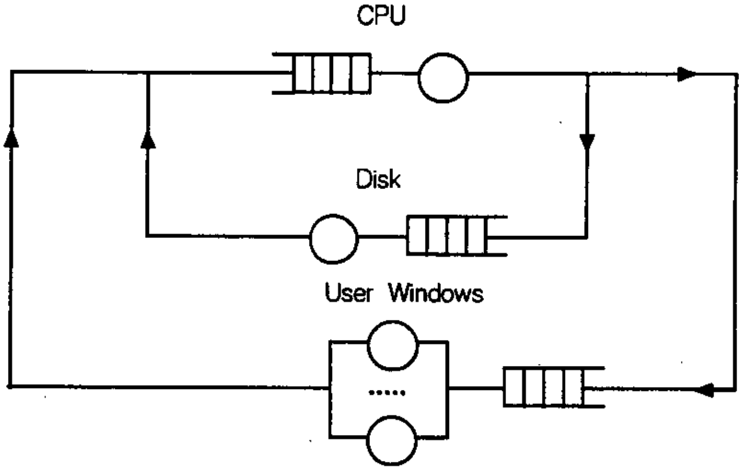


Figure 3.2.1.B : A queueing network model for the single user multi-processing systems

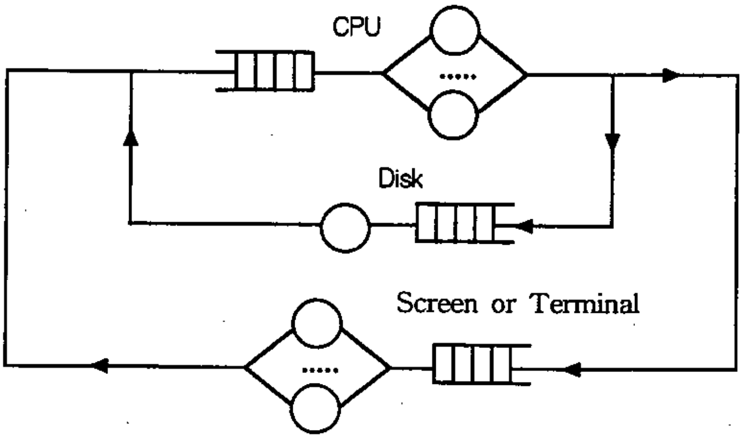


Figure 3.2.1.C : A queueing network model for the multi-processor workstations

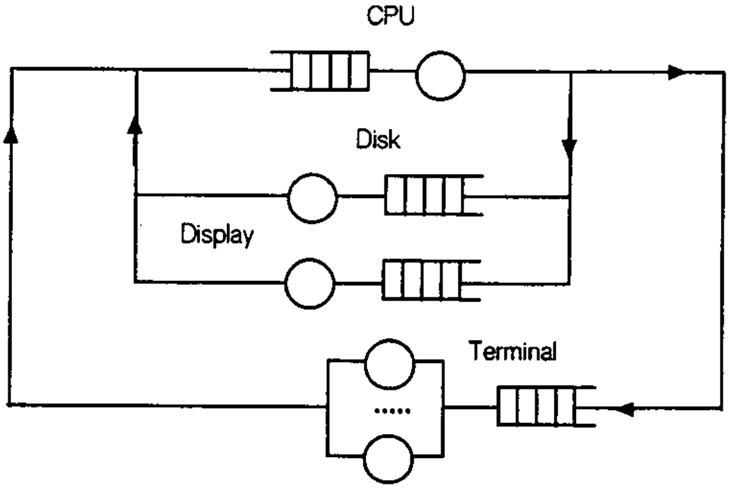


Figure 3.2.1.D : The queueing network model for the VAX 11/780 systems in Ferrari et al.[FERRARI etal 83]

If the above systems are used as the client systems, what do we have to modify in the above models? If diskless systems are used as the client systems, obviously we have to remove the disk servers. In the LOCUS distributed system, the client systems can be the file servers as well and vice versa. In this case, we do not have to modify the models at all.

3.2.2 Models for the File Server

Usually the file server has no user terminal if the usage is fixed as the file server. In this case the corresponding terminal notation should be removed. If the file server is used both as the file server and a client by supporting its own terminals, then we use the above system models as they are.

3.2.3 Models for the Disk I/O Subsystem

I/O operations are observed usually between the memory and the I/O devices, between the I/O devices and the I/O devices and between the CPU and the I/O devices. The I/O subsystem usually consists of the I/O devices, the interface units(control units) and the I/O software.

Three kinds of I/O mechanisms have been widely used since the first introduction of the disk drive storage device in late 1950s. Those are the Programmed I/O(PIO), Direct Memory Access(DMA) and interrupt facilities, and the Channel, an I/O Processor(IOP) in descending order when we consider the amount of the CPU service time spent for the execution of the I/O operations.

The most primitive one among the three I/O mechanisms is the PIO. In PIO, a single character is transferred per an instruction. The CPU must execute an explicit instruction for each and every character read or written. The I/O operations are completely controlled by the CPU. That is, CPU initiates, directs and terminates the I/O operations.

Either memory mapped I/O or I/O mapped I/O is used in the programmed I/O. I/O devices are connected to the I/O ports which are the junctions between the system bus and the I/O devices. In the memory mapped I/O, part of the address space in the main-memory is assigned to the I/O ports. MC68000 microprocessor series once used memory mapped I/O. In the I/O mapped I/O, the I/O address space does not share the main memory. Intel 8085 and 8086 microprocessor series once used the I/O mapped I/O.

The advantage of programmed I/O method is that it requires little I/O hardware. The disadvantages are that the CPU is burdened greatly by polling(testing) and that other I/O operations and the I/O transfer rate depend on the speed of the CPU service, that is, how fast the CPU can test and service an I/O device. This programmed I/O mechanism was widely used till the late 1970s.

DMA is the I/O device that transfers blocks of data to or from the memory by themselves without requiring the intervention of the CPU. The CPU in a computer specifies the I/O device, the memory address where the data are read or written, and the number of bytes(words or characters) to be transmitted. In the DMA mechanism, the CPU initiates the I/O data transfer, the DMA generates the memory addresses and transfers the data as a bus master and the CPU controls the bus master authority among requests. Therefore, the CPU and the DMA interact only when the CPU must yield the control of the system bus to the DMA in response to the requests from the DMA.

Three control mechanisms are possible in DMA to transfer data. First, the DMA transfers a block of data in a time(DMA block transfer). The disadvantage of this control mechanism is that the CPU inactive period is relatively long. Second, in the cycle stealing control mechanism, the DMA interferes with the CPU less by sending one or several data words in a time. Third, the transparent DMA control mechanism guarantees that the DMA does not interrupt the CPU at all since the DMA steals the bus cycles only when the system bus is not actually used by the CPU. The DMA mechanisms require modest hardware complexity(cost) and they have been popular till now in small systems such as most comtemporary workstations.

Channel devices use I/O Processors(IOPs). The IOP is a special purpose computer which has a limited instruction set, so called channel commands, such as read, write, read-backward, skip, rewind, sense, jump, etc.. IOPs are sometimes called Peripheral Processing Units(PPUs). The I/O subsystem has its own CPU, memory and operating system(control program) called I/O supervisor(IOS). Intel 8089 is one chip IOP for intel 8086 microprocessor and its successors. IBM mainframe computers usually use the IOP mechanism. IBM 370 uses the IOS program which resides in the main memory and the CPU activities are required for the IOS to be run. But in IBM 370/XA and its successors, the IOS resides in the memory of the I/O subsystem and it works independently from the CPU activity[CORMIER etal 83],[PADEGS 83]. In the channel mechanism, the communication link between the I/O devices and the main memory is required. The communication link is called as I/O channel. In the Channel mechanism, a separate bus system is used for the I/O channel.

In the PIO mechanism, the CPU controls the I/O device directly. In the DMA mechanism, the CPU is largely freed from the I/O operations. In the IOP mechanism, the CPU can be concurrently operated with the IOP : this is true in IBM 370/XA and its successors. Even for the path setup operation, the CPU does not have to provide service at all. Therefore, in this mechanism, the parameter of the CPU service time for disk I/O disappears.

The disk I/O operations consist of disk I/O path setup operation and physical disk I/O operation as already explained. Physical I/O operation consists of three major components : the seek, the latency and the transfer.

The seek operation is to access the right track and cylinder of the disk. Waters[WATERS 75] estimated the seek time of magnetic disks. Various seek scheduling algorithms have been proposed such as First-Come-First-Served(FCFS) algorithm, Shortest-Seek-Time-First(SSTF) algorithm, SCAN(Elevator) algorithm, Circular-SCAN(C-SCAN) algorithm, LOOK algorithm, C-LOOK algorithm, etc.. Teorey[TEOREY etal 72] compares the performance of some of the algorithms by simulations.

In fixed head disks such as magnetic drums, the disk I/O does not require any seek operation but requires set sector operation. So set sector scheduling is required. If there are more than one track or sector arms in movable head disks, the set sector scheduling is also required. The I/O sequence in channel devices is shown in the Ghant chart of figure 3.2.3.A.

```
A) ¦---(1)---¦-(2)-¦---(3)---¦-(4)-¦------(5)------¦-(6)-¦-----(7)------¦
B) ¦---------¦=====¦---------¦=====¦---------------¦-----¦==============¦
C) ¦---------¦-----¦=========¦-----¦---------------¦====================¦

A : I/O sequence
     (1) An I/O request enqueued
     (2) Send seek command
     (3) Seek,
     (4) Send transfer command,
     (5) Wait for channel,
     (6) Set sector(RPS missing)
     (7) Data Transferred
B : Channel status :  === : connected              --- : disconnected
C : Disk status    :  === : served (in service)    --- : idle
```

Figure 3.2.3.A : The I/O sequence in channel devices

Rhuemmler et al.[RHUEMMLER etal 94] show the Ghant chart for the I/O subsystems which use the DMA mechanism.

In late 1960s and 1970s, the performance of early disk I/O subsystems was analyzed usually using simple mathematical calculations or queuing network models as we can see in the work of [ABATE etal 68],[GOTLIEB & MacEWEN 73],[SKINNER etal 69],[WILHELM 77]. In 1980s, the performance of more complicated disk I/O subsystems was analyzed using queueing network models or simulations as we can see in the work of [BARD 80],[BRANDWAJN 81],[BRANDWAJN 83],[GEIST etal 82],[GOYA etal 84],[HOUTEKAMER 85],[KIM 86],[MAJOR 81]. In late 1980s and early 1990s till now, the performance of special disk I/O subsystems or complex disk I/O subsystems or the disk I/O subsystems combined to special environments were analyzed using queueing network models or simulations as we can see in the work of [ARTIS 94],[BAYLOR etal 94],[DAIGLE etal 90],[RAMAKRISHNAN etal 89].

Most of the studies on the performance of caching use simulations rather than mathematical analysis with queueing network models as we can see in the work of [BAKER etal 91],[OUSTERHOUT etal 85],[RHUEMMLER etal 93],[RHUEMMLER etal 94]. Baker et al.[BAKER etal 91] measured caching activity. Most of studies on the performance of caching investigated the performance of the caching algorithms or cache consistency mechanisms, or factors such as cache size, block size, etc., This study investigates the global effect on the file access performance at given cache hit ratios.

See [CHEN etal 94],[GANGER etal 94] for the details of the disk array such as RAID. See [FEITELSON etal 95],[ROSARIO etal 94],[BAYLOR etal 94] for the details of the parallel I/O subsystems. See [WOOD etal 93],[COLEMAN etal 93] for the trend of DASD(Direct Access Storage Device) evolution. Chen and Patterson[CHEN etal 93] give explanation of various performance metrics for the disk I/O subsystems and present the results of running some popular benchmark programs in the three environments of a DECstation 5000/200 running the Sprite Operating System, a SPARCstation 1+ running Sun Operating System and an HP series 700(model 730) running HP-UX. Rhuemmler and Wilkes[RHUEMMLER etal 93] measured the disk access patterns in UNIX and give good analyzed results as well as some simulation results.

3.2.4 Models for the Network Communication

Application layer	NFS, NIS	
Presentation Layer	XDR	
Session Layer	RPC (Socket)	
Transport Layer	TCP	UDP
Network Layer	IP(Internet protocol)	
Data Link Layer	Ethernet, FDDI, etc.	
Physical Layer	Ethernet, FDDI, etc.	

Figure 3.2.4.A : The SUN/NFS network protocols.

Communication overheads are caused by communication softwares and hardware. The communication overheads are generated in the CPU and the network interface unit of both the host sending requests/responses and the host receiving requests/responses and in the physical network. This section looks at the communication procedure first then discusses the overhead factors in the CPU, the network interface unit and the network. Finally network models will be investigated.

First, let's look at the communication procedure in the distributed environment of SUN/NFS in order to model the network communication procedure later in this section. Figure 3.2.4.A shows SUN/NFS network protocols.

In the SUN workstations, the ISO/OSI model is used. In the SUN workstations, NFS and NIS are put in the application layer. Therefore the file service requests of the clients start from the top layer. Like most UNIX workstations which use the networking codes based on Berkeley UNIX, SUN NFS/RPC usually communicates over the network via the socket interface in the session layer and TCP/IP or UDP/IP in the transport layer and the network layer. The socket interface copies data from the address space of the requesting client into the system buffer and invokes the transport protocol and the network protocol. For a reliable byte stream protocol TCP/IP will be used and for a simple but efficient protocol UDP/IP will be used. UDP/IP provides partial service of TCP/IP. In the case of TCP/IP, the provided services are packetization, error handling such as calculating data checksums(checksumming) and time-out-checking, end to end flow control, congestion control and routing. In the data link layer and the physical layer, LAN protocols such as Ethernet, FDDI, etc. will handle the handed packets. The data link layer creates MAC header(encapsulation), detects and possibly corrects errors that may occur in the physical layer. Finally the physical layer will process physical sending service. It electrically encodes and physically transfers the packets to the receiving node. In the side of the receiver, that is, the file server, similar operations will be performed in the reverse order.

Network interfaces play important roles in the network communication. The internal operations should be analyzed to model the network communication operations correctly. The past, present and possible network interfaces are no network interface, minimal network interfaces with PIO(Programmed I/O), network DMAs(Direct Memory Access), and dedicated communication controllers. For accurate modeling, it is necessary to analyze the data transfer activity on the system bus from an application address space to the network interfaces in the client. The minimal network interface case is looked at first. An application writes a file I/O request into a buffer of its address space, which resides in the host memory, over the system bus[the first system bus access]. The socket code, a protocol providing session layer services, copies the request from the buffer of the user address space in the host memory into a system buffer in the host memory over the system bus. For these operations, a copy of the request in the buffer of the user address space is sent over the system bus to the CPU by the socket code[the second system bus access] and then the request in the CPU is sent over the system bus to the system buffer in the host memory by the socket code[the third system bus access]. The transport protocol reads the request from the system buffer in the host memory into the CPU over the system bus[the fourth system bus access] and calculates the checksum. The data link protocol copies the request from the system buffer to a buffer in the network adaptor over the system bus. For these operations, a copy of the request in the system buffer in the host memory is sent over the system bus to the CPU by the data link protocol[the fifth system bus access] and the request in the CPU is sent over the system bus to the buffer in the network adaptor by the data link protocol[the sixth system bus access]. Therefore the system bus is accessed 5 times in total after an application writes a request into the user address space in order to send the request to the receiving host. In some extra-ordinary implementations, the system bus is accessed more than 6 times for the above operations.

In the immediate primitives[STEENKISTE 94] such as socket interfaces, the buffer area for user data in the host memory is blocked until it is used for retransmission if retransmission should occur. Or the system can alternatively make a retransmission copy of the data as part of the send call. In the minimal network interface, the 4th system bus access for checksum calculation can be performed during(or immediately after) the second system bus access. Thus one system bus access can be saved without modification of the API(Application Programming Interface) and the system bus is accessed 5 times in total including the initial access of the system bus by the application. Further, by using the shared-buffer interface to applications, two more system bus accesses can be saved and the total number of accesses over the system bus becomes three including the access of the system bus by the initial write of the request into the user address space. That is, in the interface applications share the system buffer with system softwares for writing send/receive messages, instead of writing the send/receive messages into the buffers of their own address space. It saves the second system bus access and the third system bus access. In this buffered communication primitives[STEENKISTE 94] such as in Nectar and Firefly, it is not necessary to copy the message for retransmission purpose as in the immediate primitives.

In the DMA network interface the DMA engine is in charge of transferring data between the host memory and the network adapter, while in the minimal network interface with PIO the CPU is in charge of it. In the DMA network interface, the copy operation for the checksum over the system bus is saved and the request is directly transferred from the host memory to the buffers in the network adapter, so that the system bus is accessed 4 times in total. By having the buffers on the network adapter large enough to be used as the system buffer(outboard buffering), the number of accesses over the system bus can be reduced to be two. That is, the application layer transfers the request to the buffer for user data in the host memory and then the data link layer and the DMA engine directly transfer it to the system buffer in the network adapter. In these cases, the operations for the checksum calculation are performed in hardware. The minimum number of system bus accesses in the socket interface is two. The minimum number of system bus accesses can be reduced to be one which is the ultimate possible value. In this case, the requests by the clients are written directly to the buffer in the network adapter. Nectar is an example[COOPER etal 90], [STEENKISTE 94]. More system bus accesses will result in larger system overheads. More bus accesses also cause more contention for the bus bandwidth, more contention for the memory bandwidth and more consumption of the CPU power. All these effects cause larger latency in the network communication and degrade the network performance.

For the performance modeling of the network communication, it is essential to find out what kind of overheads for the network communication operations exist. To find out it, it has to be found out what communication operations are performed where. Communication operations associated with sending and receiving packets in typical UNIX TCP/IP environments can be summarized into 5 groups :

- Group 1 : processing of the transport protocol and processing of the network protocol by the CPU.
- Group 2 : processing of the data link protocol by the CPU and the network interface unit.
- Group 3 : buffer management by the CPU, the system memory and the network interface unit.
- Group 4 : data transmission via the network by the CPU and the network interface unit.
- Group 5 : context switching and interrupt handling by the CPU, the memory and the network interface unit.

The socket call, TCP, IP, interrupt handling, etc. consume the CPU power for the network communication operations. The buffer management operations and the checksumming limit the memory bandwidth. These overheads often make such heavy demands on the resources of contemporary workstations that at maximum only a few tens of Mbits per second can be supported at application level communication even though higher speed networks are used. Reducing network communication overheads has been a key issue in designing host interface for high speed networks since it directly reduces communication latency. Unfortunately it is known that many factors affect the communication overheads and no single source of communication overhead dominates the communication overhead.[CLARK etal 89], [SHROEDER etal 90], [STEENKISTE 94]]. For example, for small size packets, the overhead due to copying buffers is relatively small but for large size packets, this overhead heavily dominates communication latency. The packet size usually grows as the speed of communication goes up. However the trend of communication requires the handling of small packets as well as large packets in the same environment and at the same time. Because of these reasons, considering only a single specific overhead factor or a single specific function for the required communication mechanism is not the right way but all functions in the network interface should be considered. The tendency in current and future communication is to use reliable protocols, powerful network interface hardware, high speed networks and large packet sizes(not true in case of ATM). It is known that cell-based networks like ATM and packet-based networks can be evaluated in similar ways in most cases. A big difference is that pipelining operations can be done with little data in the small uniform packet size of ATM(44-48). In modeling communication operations via networks, it should be considered that in practice different communication interfaces and even different protocols can be used in a host.

Considering the above things which have been explained so far, this study has modelled the communication operations in three components i.e. the operations which use the CPU resource, the operations which use the network interface unit resource and the operations which use the network resource. Each component can be represented as a service center. In the virtual server model, the service center to represent the overheads on the CPU and the service center to represent overheads on the network interface unit have a queue individually. The service center to represent the overheads on the network does not have any queue sometimes or have a queue sometimes. All service centers are represented as virtual service centers and mapped into real resources during simulations.

If the distributed file system is confined to a local area network, then the modelling of the network is relatively simple. Otherwise, that is, if it spans over wide area networks, then the model of the network depends heavily on the network configuration and is very complex. The modelling of wide area networks is beyond the research scope of this thesis. This study focuses on the performance modeling of local area networks since this study focuses on the local area network based distributed file systems. However, compound metropolitan area network models and compound local area network models can be successfully and easily constructed from the local area network models mentioned here as in work by LEE et al.[LEE etal 93], [LEE etal 95].

Currently Ethernet and Token ring are the most popular local area networks and are still expected to spread further. FDDI installation sites are reported to grow rapidly these days and expected to succeed the current position of Ethernet and Token Ring in the end of 1990s. In this study, the performance models of Ethernet and FDDI were constructed and used in the performance models for the distributed file systems. There have been a lot of performance evaluation studies on local area networks especially on Ethernet[SHOCH etal 80],[MARATHE etal 81], Token ring[BUX 89] and FDDI [BHUYAN etal 89], [JAIN 90].

Marathe et al.[MARATHE etal 81] showed that a Last-In-First-Out(LIFO) M/G/1 model with slightly increased service time adequately captures both the mean and the coefficient of variance of the response time in Ethernet. They studied five queueing network models analytically and then compared the result with the simulation output. They are (i)a simple M/M/1 model, (ii)a M/M/1 model with load dependent servers, (iii)a simple M/G/1 model, (iv)a M/G/1 model with increased service time and (v)a multiple regression model. They found the fourth model, the M/G/1 model with slightly increased service time is accurate enough to be used to build higher level models of the network. An adapted model of the fourth model is used as the model for Ethernet in the virtual server models of the distributed file systems, because it is simple, but nevertheless, it is accurate enough to represent Ethernet in the distributed systems, even though it does not model internal mechanism at all such as the back-off algorithm. However, I am not sure that the model is adequate to be used to predict transient or saturated behaviour of Ethernet. Ferrari et al.[FERRARI etal 83] represented Ethernet as a FCFS(First Come First Served) server with an infinite queue. Bester et al.[BESTER etal 84], Goldberg et al.[GOLDBERG etal 83], Lazowsak et al.[LAZOWSKA etal 86] and Ramakrishnan et al.[RAMAKRISHNAN etal 86] represented Ethernet as a service center with a finite queue.

Bhuyan et al.[BHUYAN etal 89] found that a gated M/G/1 and a gated M/G/2 queueing model are accurate enough to represent the performance of FDDI. They compared their analytic results which they had gained through an approximate and uniform analysis with their simulation results. The basic assumptions used to develop the models are (i)the rings have symmetric structures, (ii)the protocols use the non-exhaustive policy which means that when a station receives a token, it does not transmit all messages queued in the station but transmits just one message per token before it passes the token to other station on the ring, (iii)the packets have a fixed size, (iv)each station has an infinite number of the buffers. If FDDI uses class A stations in all stations, then the model leads to a dual walking server model. I adapt the models and use them in the performance models for distributed file systems because I think, it is accurate enough to be used in the performance models of the distributed file systems as far as this study does not violate the basic assumptions of the model.

3.2.5. Models for the Distributed File System

The models for the distributed file systems can be constructed (i)either by simply integrating models of the clients, a model of the network communication and a model of the file server(or server models if the multiple file servers are used) or (ii)adapting the three given component models according to the structures of the distributed file systems and/or the data flow logics of the distributed file systems and/or the workload characteristics. Sometimes the performance model of a distributed file system is developed focused on mainly the life-cycle of the client request. In this approach, some mechanism or part of the physical architecture is often ignored to construct the performance model.

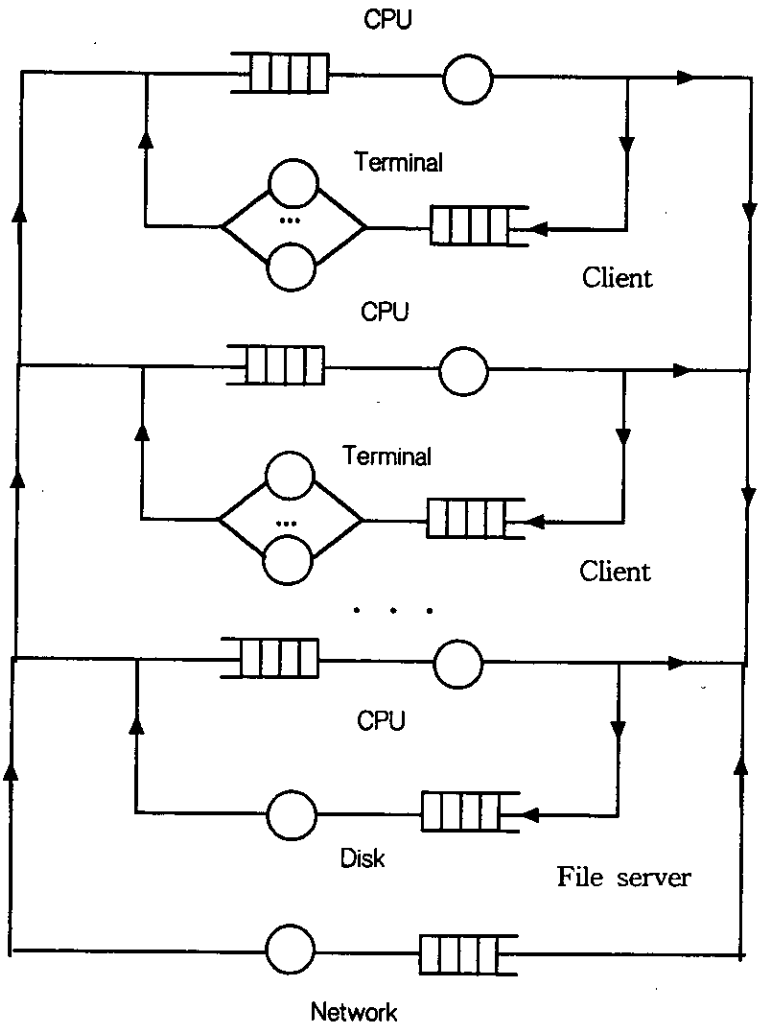


Figure 3.2.5.A : A queueing network model for the distributed system which has the fixed file server and the fixed clients.

The performance models of the distributed file systems in Bester et al.[BESTER etal 84], Ferrari et al.[FERRARI etal 83] and Lazowska et al.[LAZOWSKA etal 86] belong to the first category. In Ferrari's model[FERRARI etal 83], the file server and the clients are fixed in terms of their role. In Bester's model[BESTER etal 84], any system can be either a file server or a client and each system has its own terminals. Figure 3.2.5.A shows a sample distributed system model with the fixed file server and the fixed clients developed in the first approach.

The performance models of the distributed file systems in Perros et al.[PERROS etal 85] and Ramakrishnan et al.[RAMAKRISHNAN etal 86] belong to the second category. In the second approach, the clients are usually modelled simply because the clients just send requests and receive the responses from the file server and contention and queueing at the client nodes is usually negligible. On the other hand, the processing of each request in the file server are modelled in detail because the file server resources are shared by many clients and contention and queueing in the file server usually occur. The virtual server performance models of the distributed file systems belong to the second category since this study built the models by representing the internal logic and following the life cycle of the requests issued in the clients. For the comparison of my models with others in the second category, this study looks at the two models further.

Ramakrishnan et al.[RAMAKRISHNAN etal 86] modelled the clients as two delay servers according to the user behaviour. One delay server represents the think time between program executions. They assumed it to be 10 seconds with the probability of 0.01. That is, the users rarely stop sending requests. The other delay server represents the inter-request delay. They assumed it to be 10msec with the probability of 0.99. That means that in most cases the clients resume sending requests after 10msec. They did not explicitly model the Ethernet. They included the DMA network interface unit as a service center with a finite queue(12 buffers) with 500msec retransmission delay in their model targeting the VAX systems. The requests are transferred from the buffer to the memory of the file server. They distinguished three different file server CPU consumption activities : the request receiving activities including the network interface activity, the pure request processing activities, and the request sending activities including the network interface activity. Those distinct activities are represented by a request receive service center, a pure file service center and a response send service center. The pure file service model is represented by a FCFS service center with the exponential service time distribution for the CPU of the file server and a service center for each disk drive which has its own separate access path.

Perros et al.[PERROS etal 85] developed a performance model for the distributed file system emphasizing the bulk file transfer. A hierarchical model was presented. The high level model is simple. The low level model for the distributed file system represents the disk I/O operation.

3.2.6 The Virtual Server Models

Figure 3.2.6.A, Figure 3.2.6.B, Figure 3.2.6.C, Figure 3.2.6.D, Figure 3.2.6.E, Figure 3.2.6.F and Figure 3.2.6.G show the performance models of distributed file systems used in this study. They represent the internal logics and other details of the distributed file systems which were described in section 3.1. The job flows in the models follow the life cycle of the requests issued in the clients. The virtual server concept is used to model the operations so that each operation and each component are modelled realistically. The virtual server concept enables us to model each operation in reality and expand the developed model to various cases with relatively simple modification. The virtual servers are mapped into real existing resources during simulation. In the file server of figure 3.2.6.A, the CPU is represented by six virtual CPU servers : the request receive virtual CPU server, the request evaluation virtual CPU server, the request processing virtual CPU server, the virtual CPU server for disk I/O, the response build virtual CPU server and the response send virtual CPU server. The six virtual CPU servers with six individual queues are mapped into the CPU server with a queue during simulations. The network interface unit in the file server - the DMA network interface unit - is represented by the two virtual servers : the request receive virtual server of the DMA network interface unit and the response send virtual server of the DMA network interface unit.

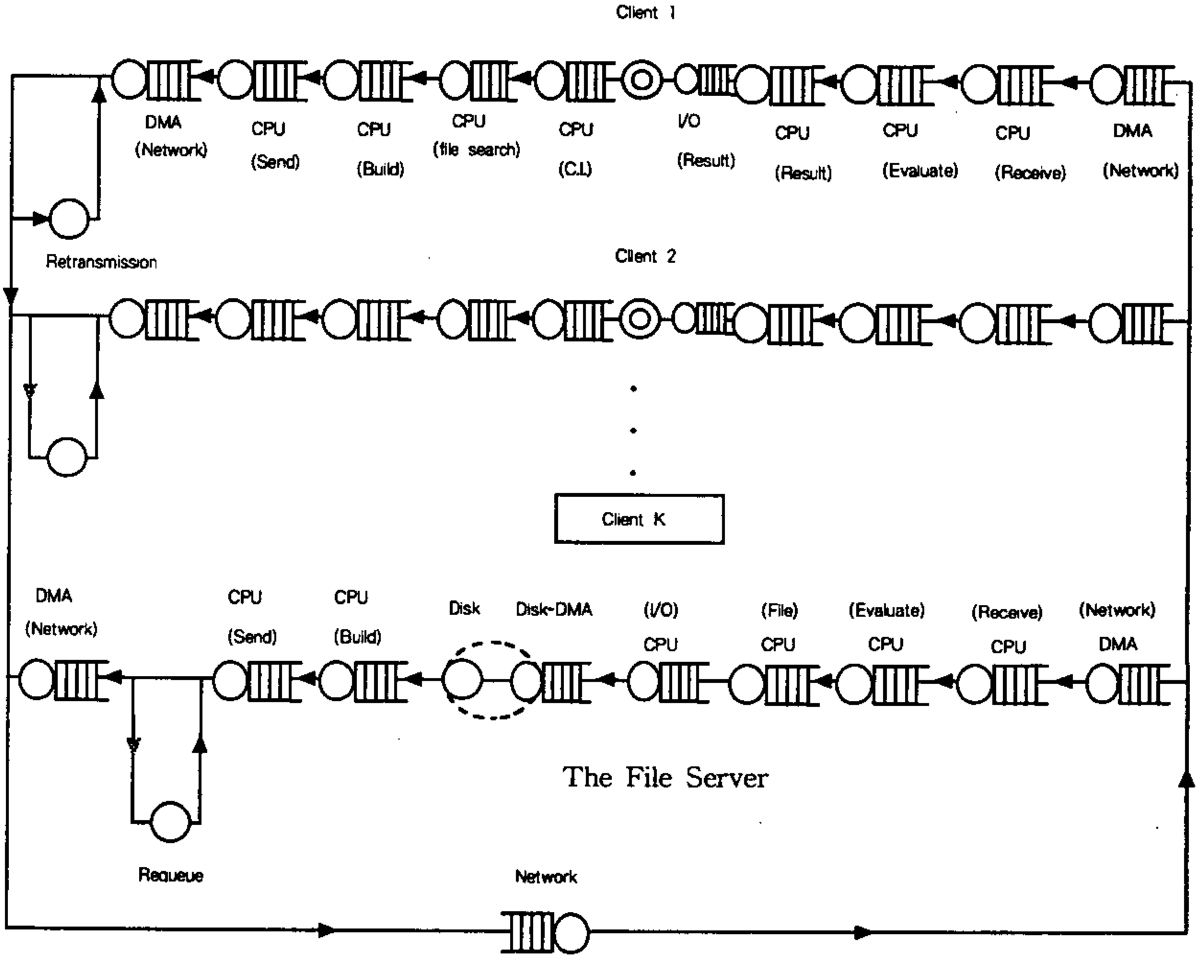


Figure 3.2.6.A : The virtual server model of the distributed file system which has multiple clients and a file server : the baseline case.

The two virtual servers are mapped into a real server of the DMA network interface unit with a queue during simulations. A real server among available real servers is assigned to a virtual server when it is requested by the virtual server and the other virtual servers should wait to acquire the real server until the using(owning) virtual server releases it and it becomes free.

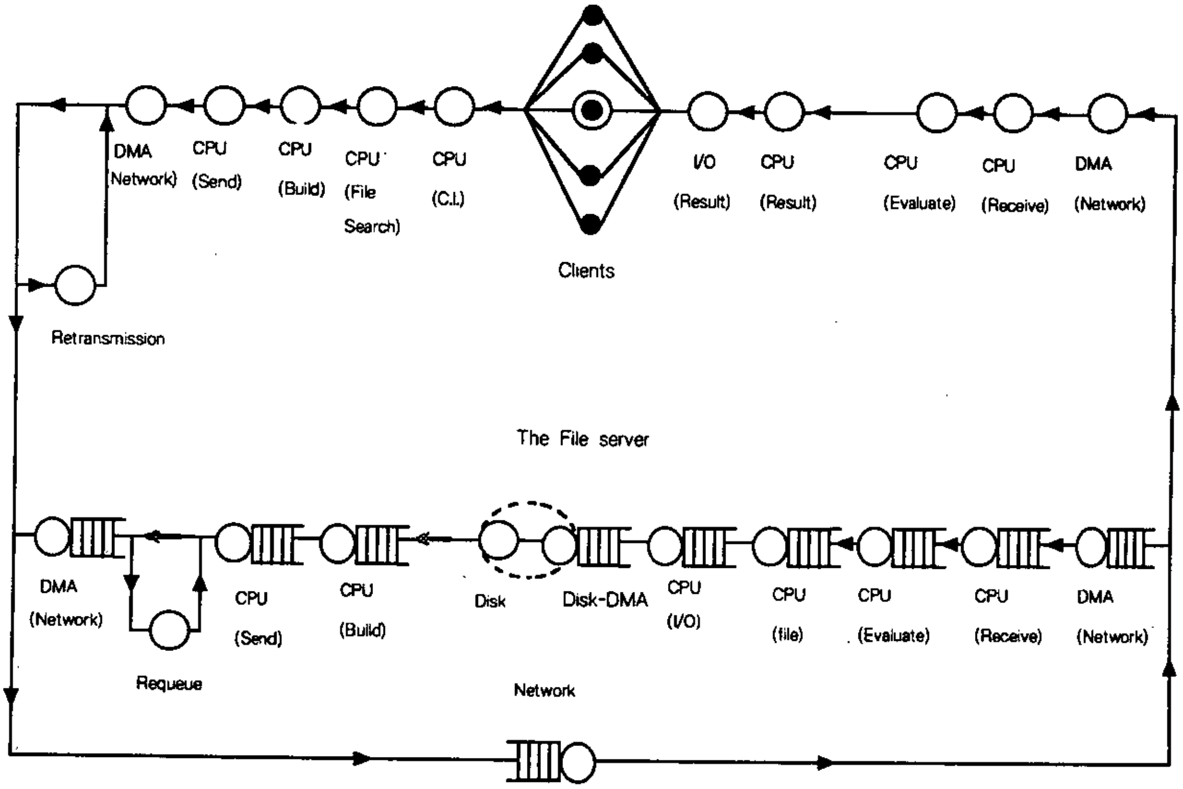


Figure 3.2.6.B : The virtual server model of the distrubuted file system which has multiple clients and a file server : the baseline case.

The quantum sizes of contemporary high performance workstations which use the multiprogramming scheme are usually larger than the service time demands in the virtual servers therefore the virtual server model is close to real environment in terms of modeling accuracy. This virtual server concept was inspired by the virtual memory concept in memory management. See appendix A for the implementation details of the virtual server concept in my SLAM-II simulation program.

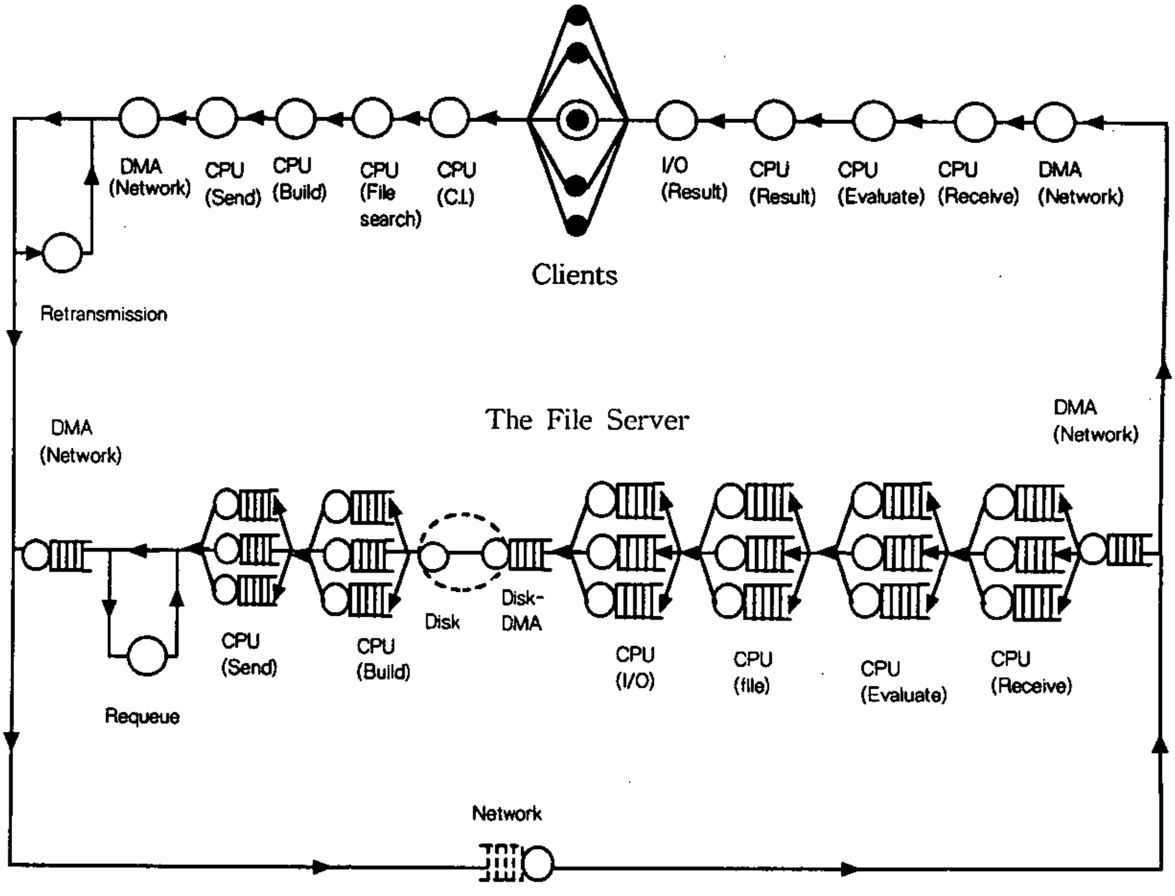


Figure 3.2.6.C : The virtual server model of the distributed file system which has the multiple homogeneous CPUs sharing the memory system in the file server.

In each client of figure 3.2.6.A which shows the virtual server model of the distributed file system with multiple clients and a file server, the model explicitly and separately represents the initial command interpretation service in the CPU of the clients, the CPU service of searching the requested file in the file table of the memory of the client where the request is issued, the request build service in the

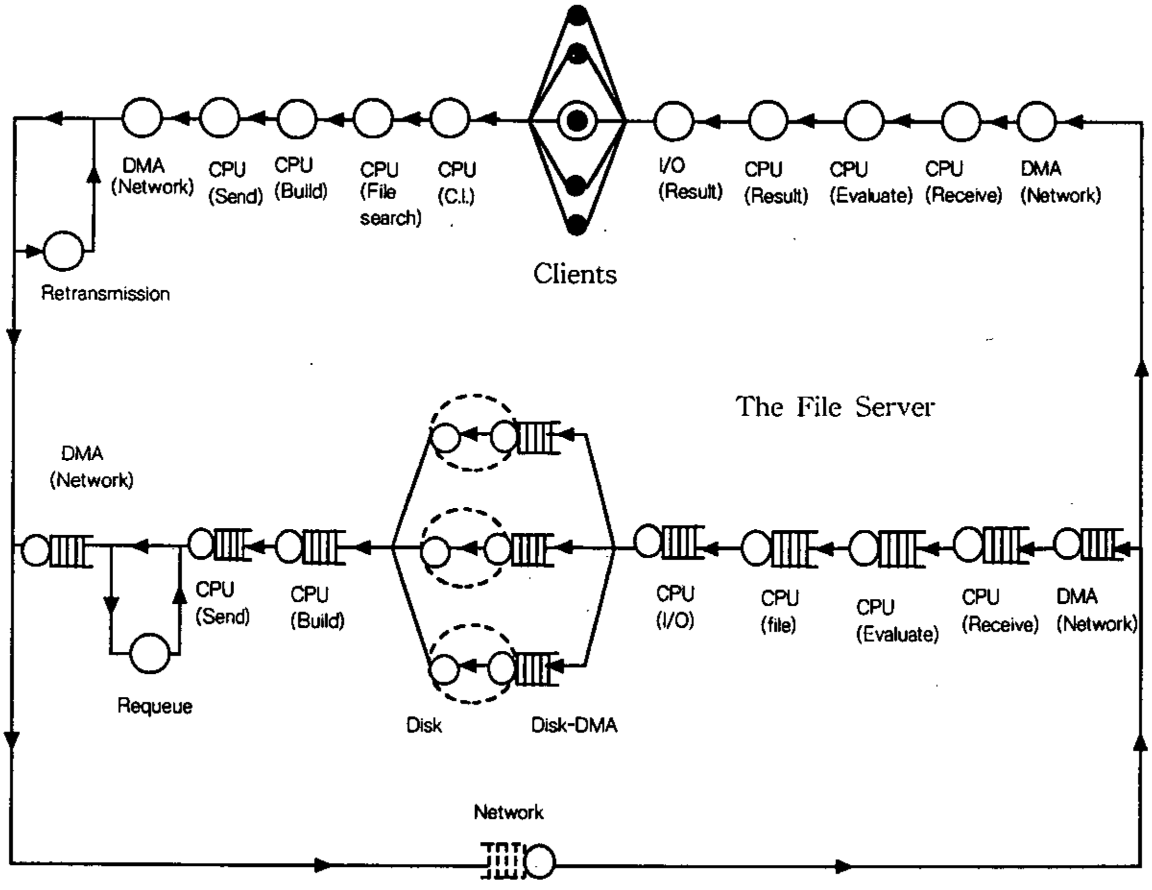


Figure 3.2.6.D : The virtual server model of the distributed file system which has multiple disks and multiple disk interface units in the file server.

CPU, the request send service in the CPU, the request send service in the network interface unit, the response receive service in the CPU, the response receive service in the network interface unit, the response evaluate service in the CPU, the result processing service in the CPU and I/O service to display the result on the screen if necessary.

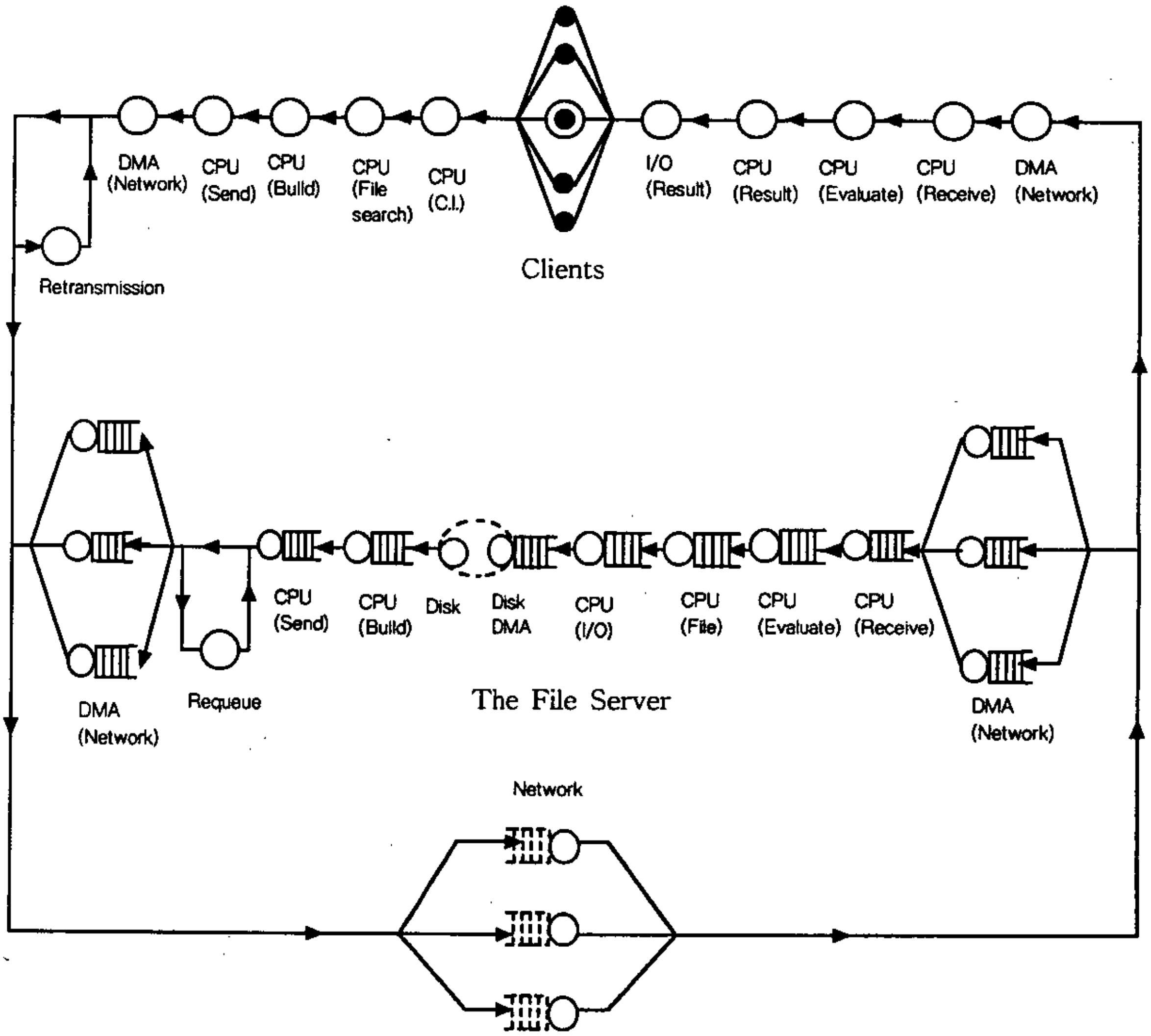


Figure 3.2.6.E : The virtual server model of the distributed file system which has multiple networks with multiple network interface units in the file server.

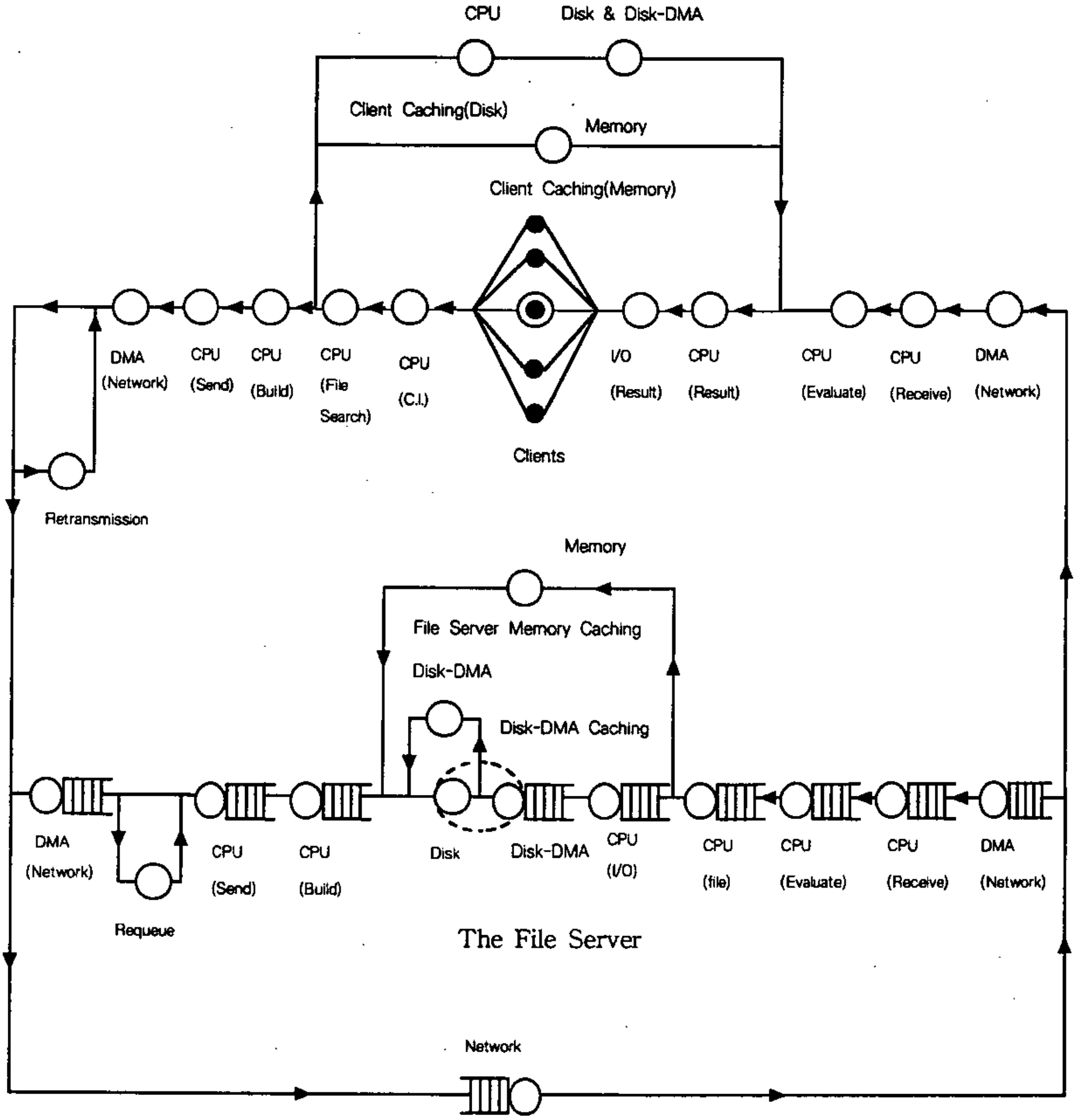


Figure 3.2.6.F: The virtual server model of the distributed file system which represents caching

The retransmission delays can occur if the network is not available due to the collision in transmitting data via Ethernet or if the file server is not available due to the server problem such as crash or rebooting, etc., or if the buffers of the network interface unit of the file server are full.

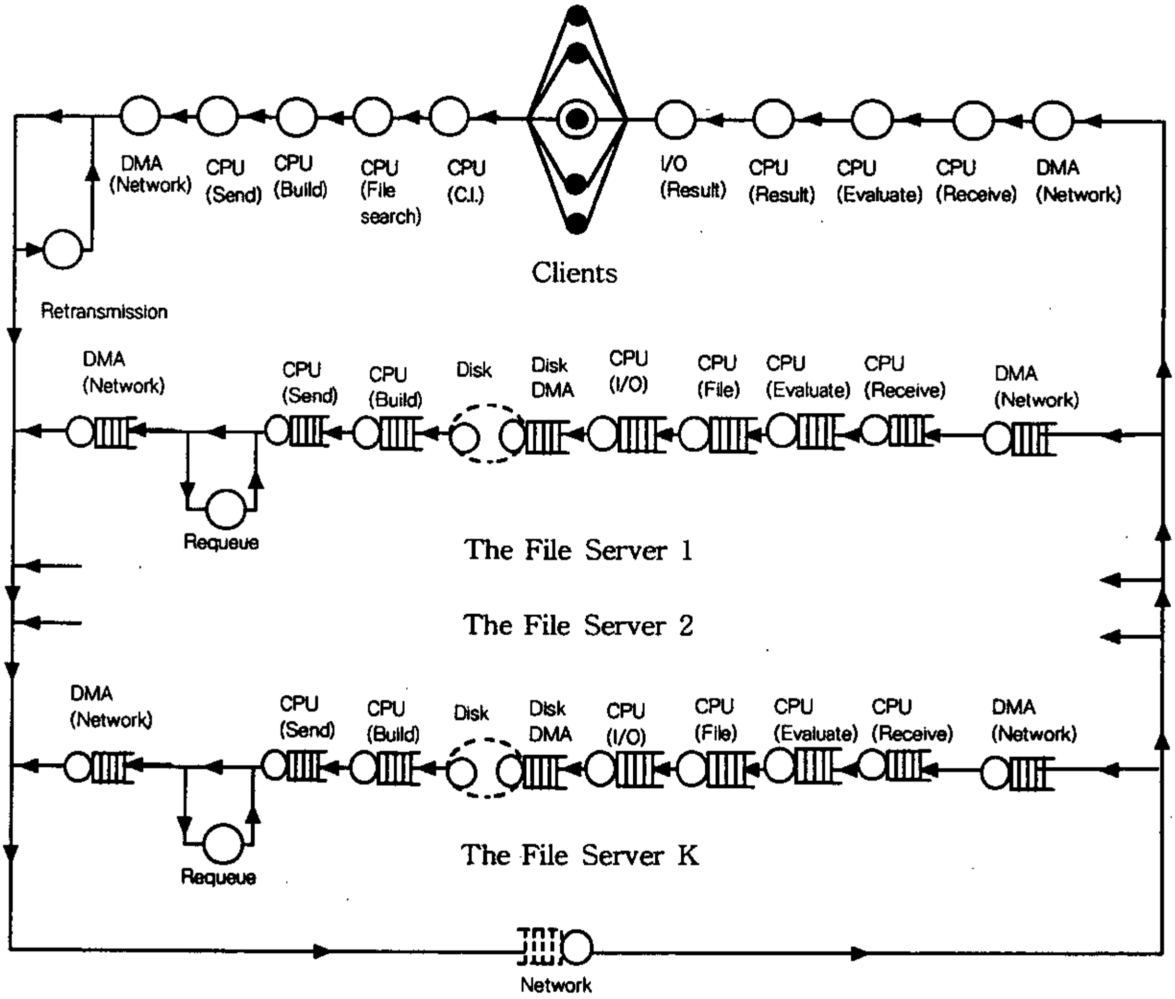


Figure 3.2.6.G : The virtual server model of the distributed file system which has multiple homogeneous file servers.

The buffer full problem can occur only when the incoming data to the network interface unit via network is faster than the outgoing data from the network interface unit to the system buffers in the memory of the file server. I have not observed it during the simulations in case of contemporary SUN workstations such as the SUN SPARCstation 10 workstations and the SUN SPARCstation 470 workstations in 10Mbps Ethernet. It was assumed that the file service activity in each client is so low that the contention for the system resources such as the CPU, the disk, the disk interface unit and the network interface unit is negligible. Thus, figure 3.2.6.A can be drawn as figure 3.2.6.B.

The network transmission service center is represented as a mere delay center or as a service center with a queue in the model. This study uses both models. Before the data transmission, both the network interface unit and the CPU of the client cooperate to do the preprocessing work for data sending, for example, moving data from the memory buffers to the buffers of the network interface unit in the sending site. Then, the network, the network interface unit in the client and the network interface unit in the file server are seized at the same time for the data transmission duration. After the transmission activity, the network interface unit of the client, the network and the network interface unit of the file server are released at the same time. Then, the network interface unit and the CPU of the file server cooperate to do postprocessing work for data receiving, for example, moving the received data in the buffers of the network interface unit into the memory buffers. The internal detail of the operations in the network interface was already explained in section 3.2.4.

In the file server, the model explicitly and separately represents the request receive service in the network interface unit and the CPU, the request evaluation service in the CPU, the file handling service in the CPU, the physical disk I/O service in the CPU, the disk interface unit and the disk, the response build service in the CPU and the response send service in the CPU and the network interface unit.

The response requeue delay in the file server can be represented explicitly as drawn in file server of figure 3.2.6.A. However it is very rare and it occurs only when the speed of the sending data to the client is slower than the speed of the receiving data from the CPU of the file server. The request receive virtual service center of the CPU and that of the network interface unit represent the protocol overhead for the request receive operation. During the postprocessing work period in receiving the request from the client, both the request receive virtual service center of the CPU and the request receive virtual service center of the network interface unit in the file server work together so that they are seized and released at the same time. If any of the two required system resources is unavailable then the other should wait until the unavailable one becomes free and both of them can be seized at the same time. During the preprocessing work period in sending the response to the client, the same mechanism also applies to the response sending virtual service center of the CPU and the response sending virtual service center of the network interface unit in the file server. The request evaluation virtual service center of the CPU represents the interpretation overhead of the RPC requests. The response build virtual service center of the CPU represents the response RPC message build-up overhead. The response send virtual service center of the CPU and the response send virtual service center of the network interface unit represent the communication protocol overhead to send the responses. The details of the operations in the disk I/O subsystem such as the disk path connection, the RPS missing, the rotational positioning, the seek, the data transmission operation, etc. are not represented explicitly as service centers in the model but implicitly in the values of the related parameters and the simulation programs. The disk interface unit and the disk itself are represented as tandem queues so that the disk interface unit is seized first and, until the service in the disk finishes, the seized disk interface unit is not released. The disk interface unit and the CPU cooperate to do the preprocessing work such as the disk I/O path set-up, etc., before starting the disk I/O operations. They also cooperate to do the postprocessing work such as moving data from the buffers of the disk interface unit into the buffers of the memory, etc., after finishing the disk I/O operations. For the cooperation, the service center of the disk interface unit and the virtual service center of the CPU for disk I/O operations are seized and released at the same time. If any of the two required resources is unavailable then the other should wait until the unavailable one becomes free and both of them can be seized at the same time. The buffer capacity of the network interface unit and that of the disk interface unit were set infinite. However it can be set finite if necessary in the models.

Caching is represented explicitly in the model of figure 3.2.6.F. The represented caching are caching in the memories of the clients, caching in the disks of the clients, caching in the memory of the file server and caching in the disk interface unit of the file server.

Figure 3.2.6.C shows the performance model of the multiple homogeneous CPUs sharing the memory system in the file server. A symmetric multiprocessor system with the shared memory mechanism is used as the file server in the figure. They are homogeneous in terms of performance. Considering the bottleneck effect of the shared bus, up to 30CPUs are used in the simulation using the models in this study, according to the prevailing belief that, up to 30CPUs the performance is not usually degraded due to the bottleneck effect of the shared bus. Figure 3.2.6.D shows the performance model of the multiple disks of the file server. Each disk has its own disk interface unit. They are homogeneous in terms of performance. All others remained ths same as figure 3.2.6.A. An unlimited number of disks and disk interface units can be served in the virtual server models, assuming that enough disk paths are guaranteed in terms of the hardware and the software. Figure 3.2.6.E shows the performance model of the multiple networks with the multiple network interface units in the file server. They are homogeneous in terms of performance. All others remained the same as in figure 3.2.6.A. Figure 3.2.6.G shows the performance model when the multiple homogeneous file servers are used. It is assumed that the file replication is done with negligible maintenance expense. In the figure, the possibility to go to a file server is specified by the visiting ratios. If the overhead for maintaining the replicated files consistent in the file servers is negligible, then an infinite number of file servers can be served in the virtual server model.

3.2.7 The Performance Parameters and Parameterization

It is required to parameterize the overhead of each service center to quantify the service demand on each service center. Specially designed measurements were performed repeatedly to get the parameter value of each service center. This section describes how the overheads were measured and the parameter values were obtained.

Specially designed measurements for the parameterization have been performed on 5 workstations all running the SUN UNIX operating system. The 5 workstations are EDLYW3, EDLYW2, KING10, KING470 and EDLYW4. They were networked together via 10Mbps ETHERNET and 100Mbps FDDI. EDLYW3 and KING10 are SUN SPARCstation 10 workstations. Each of them has 32Mbytes main memory, a 36MHz Superscalar SPARC version 8 processor, a 20Kbyte instruction on-chip cache and a 16Kbytes data on-chip cache. Each of them runs the SUN UNIX 4.1.3. The performance is reported to be 101.6MIPS in the SUN internal data published on November 1992(86.1MIPS in the SUN internal data published on May 1992), 20.5MFLOPS in the SUN internal data published on November 1992(10.6MFLOPS in the SUN internal data published on May 1992), 45.2SPECint92, 49.2SPECfp92, 1072SPECrate int92 and 1172SPECrate fp92. The SUN SPARCstation 10 workstation was first announced on May 1992 and first delivered on September 1992[DATAPRO]. EDLYW2 and KING470 are SUN SPARCstation 470 workstations. EDLYW2 has 32Mbytes main memory, a 33MHz 32bit SPARC processor, an integrated floating point co-processor and a 128Kbytes cache memory. The system specification of KING470 is the same as that of EDLYW2. Each of them runs the SUN UNIX O/S 4.1.1. The performance is reported to be 22.6MIPS[DATAPRO], 19.4SPECmarks[DATAPRO]. SUN SPARCstation 470 workstations were first installed on May 1990. EDLYW4 is a SUN 3/60 workstation. It has 4Mbytes main memory, a 20MHz 32bit MC68020 processor, an integrated 20MHz MC6881 floating point co-processor. It runs the SUN UNIX O/S 4.1.1. The performance is reported to be 3MIPS in the SUN internal data. Table 3.2.7.A shows the summarized specifications of the above 5 workstations.

NAME	EDLYW3	KING10	EDLYW2	KING470	EDLYW4
SYSTEM	SUN SPARCstation 10		SUN SPARCstation 470		SUN 3/60
PERFORMANCE	101.6(86.1) MIPS 20.5(10.6) MFLOPS 45.2 SPECint92 49.2 SPECfp92 1072 SPECrate int92 1172 SPECrate fp92		22.6 MIPS 10.4 SPECmarks		3 MIPS
PROCESSOR	36 Mhz superscalar SPARC Version 8 processor		33 Mhz 32 bit SPARC + An integrated floating point co-processor		20Mhz 32bit MC68020 + An integrated 20 Mhz MC6881 floating point co-processor
MEMORY	32 Mega bytes		32 Mega bytes		4 Mega bytes
CACHE	Instruction on chip cache : 20 kbytes Data on chip cache : 16 kbytes		128 kbytes write-back cache		
O.S.	SUN UNIX 4.1.3		SUN UNIX 4.1.1		SUN UNIX 4.1.1
ON MARKET	1992		1990		1982 (?)

Table 3.2.7.A : The summarized specifications of the five workstations used in the measurement for the parameterization.

All of them have their own local disks. EDLYW3 has a 1.05Giga-bytes local disk(MK538FB). The average access time of the MK538FB is 14.56msec for read and

16.06msec for write, the average seek time is 9msec for read and 10.5msec for write and the average latency time is 5.56msec. It has a 256Kbytes multisegmented cache buffer and a SCSI CCS controller. It uses a fast SCSI-II interface which has asynchronous(synchronous) data transfer rate of 4(10)Mbytes per second. The drive configuration is 2036cylinders, 14tracks/cylinder, 72sectors/track and 512bytes/sector. KING10 has a 956Mbytes local disk(ST11200N). The average access time of the ST11200N is 16.06msec for read and 17.56msec for write, the average seek time is 10.5msec for read and 12msec for write and the average latency time is 5.56msec. It has a 256Kbytes multisegmented cache buffer and a SCSI CCS controller. It uses a fast SCSI-II interface which has asynchronous(synchronous) data transfer rate of 4(10)Mbytes per second. Drive configuration is 1730cylinders, 15tracks/cylinder, 72sectors/track and 512bytes/sector.

EDLYW2 has a 670Mbytes local disk. It uses a SCSI interface which has data transfer rate of 1.8Mbytes per second. It has an Emulex MD21 controller. The drive configuration is 1614cylinders, 15tracks/cylinder, 54sectors/track and 512bytes/sector. KING470 has a 670Mbytes local disk which has the same hardware characteristics as EDLYW2.

EDLYW4 has a 327Mbytes local disk(Micropolis). The average access time of Micropolis is 18msec. It has an Emulex MD21 controller. It uses a SCSI-II interface which has data transfer rate of 1.2Mbytes per second. The drive configuration is 1218cylinders, 15tracks/cylinder, 35sectors/track and 512bytes/sector. Table 3.2.7.B shows the summarized characteristics of the local disks of the 5 workstations.

The measurement was deliberately designed so that the value of the individual parameter could be extracted from the measured times of the experiments that were performed in stand-alone mode. Each experiment was repeated 10 times in a measurement - I call it a set of measurements - and the measured values were analyzed to get the mean, the standard deviation, the median and mode from them. I repeated the set of measurements. I constructed sets of linear equations using the measured times, where the variables were the performance parameters shown in table 3.2.7.C.

Name	EDLYW3	KING10	EDLYW2,KING470	EDLYW4
Capacity	1.05 Gbytes	956 Mbytes	670 Mbytes	327 Mbytes
Model	MK538FB	ST11200N		Micropolis
Cache buffer	256 Kbytes Multi-segmented cache buffer	256 Kbytes Multi-segmented cache buffer		
Controller	SCSI CCS Controller	SCSI CCS Controller	Emulex MD21 Controller	Emulex MD21 Controller
Interface	SCSI-II	SCSI-II	SCSI	SCSI-II
Cylinders Tracks/cylinder Sectors/track bytes/sector	2036 14 72 512	1730 15 72 512	1614 15 54 512	1218 15 35 512
Average latency time (msec)	5.56	5.56		-
Average seek time (msec)	9 for read 10.5 for write	10.5 for read 12 for write		
Average access time (msec)	14.56 for read 16.06 for write	16.06 for read 17.56 for write		18
Average transfer time	Aynchronous : 4 Mbytes/sec Synchrounous : 10 Mbytes /sec		1.8 Mbytes/sec	1.2 Mbytes/sec

Table 3.2.7.B : The summarized characteristics of the local disks of the five workstations used in the measurement for the parameterization.

The CPU times and the response times were measured separately so that the CPU time service demand per 1500bytes data transferred and the I/O time service demand per 1500bytes data transferred could be identified separately. The values of some parameters were also directly measured and the measured values were used as guideline values to confirm the accuracy of the extracted values of the parameters.

		File server		SUN 3/60	SUN SPARC 470	SUN SPARC10
		Operation		(msec)	(msec)	(msec)
CPU	c	Command interpretation	f	80.0000	20.000	20.0000
CPU	c	RPC request build	f	3.3300	2.500	1.2500
CPU	c	RPC request send	p	0.1375	0.125	0.1125
I/O	c	Network interface unit	p	5.2625	1.775	0.2875
I/O		Network transmission	p	1.2000	1.200	1.2000
I/O	s	Network interface unit	p	5.2625	1.775	0.2875
CPU	s	RPC request receive	p	0.1375	0.125	0.1125
CPU	s	RPC request evaluation	f	3.3300	2.500	1.2500
CPU	s	File handling	f	20.0000	10.000	5.0000
CPU	s	Disk I/O	p	0.4000	0.150	0.1250
I/O	s	Disk interface unit	f	130.0000	60.000	24.0000
I/O	s	Disk interface unit + Disk I/O	p	4.1200	1.550	1.1250
CPU	s	RPC response build	f	3.3300	2.500	1.2500
CPU	s	RPC response send	p	0.1375	0.125	0.1125
I/O	s	Network interface unit	p	5.2625	1.775	0.2875
I/O		Network transmission	p	1.2000	1.200	1.2000
I/O	c	Network interface unit	p	5.2625	1.775	0.2875
CPU	c	RPC response receive	p	0.1375	0.125	0.1125
CPU	c	RPC response evaluation	f	3.3300	2.500	1.2500
CPU	c	Result processing (cat)	p	0.3500	0.300	0.2500
I/O	c	Result processing (cat)	p	520.0000	100.000	22.0000

- CPU: CPU time, I/O: I/O time, s: server, c: client,
- p: proportional to the data size, f: fixed(constant)
- The values of all parameters proportional to the data size are per 1500bytes data transferred.
- The values of all parameters constant to the data size are per one transaction regardless of the transferred data size.

Table 3.2.7.C : The parameters for the virtual server performance models of the distributed file systems

The built-in functions such as "gettimeofday", ping, spray, etc. were used for the direct measurements. The standard account gathering facilities were used to measure the service time. Caching was deliberately avoided as much as I could. For example, I read and wrote a very large volume of data - 10Mbytes data - after each read/write operation so that the cache would be refreshed each time and the sequence of the experiment was deliberately adjusted so that any read/write had little possibility to occur at an adjacent disk position. Data were spread to the different positions as far as I could so that I could meaningfully compare the measured values with the values of the average access times of the used disks provided by the disk vendors.

The rest of this section describes the procedure of performance parameterization stage by stage in the order that this study progressed.

In the first stage, the values of the CPU time service demand for the disk I/O operation were obtained. For it, I performed a specially designed read-write experiment in stand-alone mode on isolated workstations. The experiment was performed in three classes of SUN workstations : the SUN 3/60 workstation, the SUN SPARCstation 470 workstation and the SUN SPARCstation 10 workstation individually.

The read-write experiment reads a file in the local disk and as a pipelined operation, writes the read data into a file in the local disk at a location different from the location of the read file. It consists of the command interpretation operation, the file handling operation and the disk I/O operation. The consumed CPU time and the response time were measured. The command interpretation operation is interpreted to consume CPU times only. The CPU time consumed for the command interpretation does not vary with the size of the data of the read-write operation. The file handling operation is interpreted to consume CPU time only. In most cases, the requested file table will be in memory already therefore I/O to the disk will rarely happen and the I/O time for searching the file table in the memory is negligible. Thus this interpretation is believed not to diminish the accuracy of the parameterization. The CPU time consumed for the file handling operation is assumed not to vary with the size of the data size of the read-write operation. In reality, disk space fragmentation and file extension might push the consumed CPU time to vary to the size slightly and irregularly. The disk I/O operation consumes both the CPU time and the I/O time. The consumed CPU time consists of a constant portion and a portion proportional to the data size of the read/write operation. This study includes the constant portion in the file handling overhead.

Now we know that in the measured CPU time only the CPU time for disk I/O varies with the data size of the read/write operation. The measured CPU time can be expressed in a linear function of as "y = ax + b" where "x" denotes the size of file, "y" denotes the measured CPU time and "a" and "b" denote constants. The value of "ax" covers the value proportional to the data size and the value of "b" covers the constant value irrespective of the data size. Now I explain how I got the value of "a". The data size was varied from 1500bytes(12Kbits) up to 300Kbytes(2.4Mbits): 1.5, 3, 6, 9, 15, 150, 200, 250 and 300Kbytes and, if necessary, some other sizes and the consumed CPU times were measured at each size. This measurement was repeated in the set of 10 measurements. The measured values were plotted on 2 dimension rectangular coordinate systems and scatter diagrams were made. By applying statistical regression analysis to the values for the curve fitting, I selected the best value of "a"(the slope of the approximating straight line).

This study assumes that the consumed CPU service time of the disk I/O operation for the read is same as that of the write. In reality, the consumed CPU service time for the disk read is different from that of the disk write.

As the value of the CPU time service demand for the disk I/O operations, I got average 4.12msec per 1500bytes data transferred in the SUN 3/60 workstation, 1.55msec per 1500bytes data transferred in the SUN SPARCstation 470 workstation and 1.125msec per 1500bytes data transferred in the SUN SPARCstation 10 workstation as shown in table 3.2.7.C.

In the second stage, I obtained the CPU time service demand of the result processing operation to the window screen where the command had been issued. A read experiment was performed in stand-alone mode on the isolated workstations. The experiment read a file in the local disk and displayed the result on the window screen. The experiment was individually performed in three classes of SUN workstations such as the SUN 3/60 workstation where the SUN window system(sunview) was used, the SUN SPARCstation 470 workstation where the X window system(twm) and the SUN window system were used and the SUN SPARCstation 10 workstation where the X window system and the SUN window system were used.

The consumed CPU time and the response time were measured. By using the measured CPU service times of the previous read-write experiments and the measured CPU service times of these read experiments, I built and solved a set of linear equations to get the CPU time service demand for the result processing operation in this stage, the CPU time service demand of the command interpretation operation in the third stage and the CPU service time demand of the file handling operation in the third stage.

Now let us see these equations in detail. The read operation consists of the command interpretation operation, the file handling operation, the disk I/O operation and the result processing operation to the window as shown in table 3.2.7.D. Table 3.2.7.E shows the operation of the local read-write as explained in the first stage.

Local read		
Sequence	Operation	CPU times (y)
1	Command Interpretation	b1
2	File Processing for local read	b2
3	Disk I/O for local read	(a1 * x)
4	Result Processing	(a2 * x) + b3

Table 3.2.7.D : The sequence of operations for the local read and related CPU time consumed. (a1, a2, b1, b2 : constants, x : the number of 1500bytes packets)

Local read-write		
Sequence	Operation	CPU times (y)
1	Command Interpretation	b1
2	File Processing for local read	b2
3	Disk I/O for local read	(a1 * x)
4	File Processing for local read	b2
5	Disk I/O for local read	(a1 * x)

Table 3.2.7.E : The sequence of operations for the local read-write and related CPU time consumed. (a1, b1, b2 : constants, x : the number of 1500bytes packets)

As explained in the first stage, the measured cpu times can be expressed as "y=ax + b" where "x" denotes the size of file in the number of 1500bytes packets, "y" denotes the measured CPU time and "a" and "b" denote constants. Using this concept, the two tables are used to build the following two linear equations.

(1) The CPU times measured in the local read experiments.

$$y= b1 + b2 + (a1 * x) + (a2 * x) + b3 = (a1 + a2) * x + (b1 + b2 + b3)$$

(2) The CPU times measured in the local read-write experiments.

$$y= b1 + b2 + (a1 * x) + b2 + (a1 * x) = (2a1 * x) + (b1 + 2b2)$$

The result processing operation consists of the portion(b3) which does not vary with the data size and the portion(a2 * x) which is proportional to the data size in both the CPU time and the I/O time. The fixed portion(b3) is assumed to be zero because I interpret that it is negligible in most cases. The following calculations are simple. The proportional portion[(a1 + a2) * x] of the measured CPU service times of the read experiments consists of the CPU time service demands of the disk I/O operation(a1 * x) and the CPU time service demands of the result processing(a2 * x). As in the first stage, the measured values were plotted on 2 dimension rectangular coordinate systems and scatter diagrams were made. By applying statistical regression analysis to the values for the curve fitting, I selected the best value of the slope, i.e., (a1 + a2), of the approximating straight line, that is, the equation (1). In the first stage, the CPU time service demand of the disk I/O operation(a1) was known. Therefore it is straightforward to get the CPU time service demands of the result processing(a2).

Thus in the case of "cat" command, the CPU time service demand of the result processing operation to the window screen where the command had been issued, was obtained to be average 0.35msec per 1500bytes data transferred in the SUN 3/60 workstation, 0.3msec per 1500bytes data transferred in the SUN SPARCstation 470 workstation and 0.25msec per 1500bytes data transferred in the SUN SPARCstation 10 workstation as shown in table 3.2.7.C.

In the third stage, the CPU time service demand of the command interpretation operation and the CPU service demand of the file handling operation were obtained. Since now I know the value of the proportional portion of the consumed CPU time in the equation (1) and the equation (2) of the second stage, the linear equations have two measured CPU time values with two unknown parameters so that it is possible for me to calculate the values of the two parameters. Remember that in the second stage "b3" was assumed to be zero because I interpret that it is negligible in most cases.

In this way, as the CPU time service demand of the command interpretation operation, I got average 80msec for the SUN 3/60 workstation, 20msec for the SUN SPARCstation 470 workstation and 10msec for the SUN SPARCstation 10 workstation, and as the CPU time service demand of the file handling operation, average 20msec in the SUN 3/60 workstation, 10msec in the SUN SPARCstation 470 workstation and 5msec in the SUN SPARCstation 10 workstation as shown in table 3.2.7.C.

In the fourth stage, the CPU time service demand of the send/receive operation in the client and in the file server was obtained. For it, a remote read experiment was performed in stand-alone mode between two interconnected workstations using NFS via Ethernet. In the experiment, a file was read in a remote workstation and the read data were displayed on the window screen in the client workstation. The experiment was individually performed between the SUN SPARCstation 470 workstations where both the X window system and the SUN window system were used, between the SUN SPARCstation 10 workstations where both the X window system and the SUN window system were used. The remote read experiment in the heterogeneous distributed file system was also performed between the SUN 3/60 where the SUN window was used workstation and the SUN SPARCstation 10 workstation where the X window was used.

The remote read consists of the command interpretation operation in the client, the RPC build-up operation in the client, the RPC request send operation in the client, the RPC request receive operation in the file server, the RPC request evaluation operation in the file server, the file handling operation in the file server, the disk I/O operation in the file server, the RPC response build-up operation in the file server, the RPC response send operation in the file server, the RPC response receive operation in the client, the RPC response evaluation operation in the client and the result processing operation to the window in the client as explained in the virtual performance models.

The consumed CPU time and the response time were measured. By using the measured CPU service times of the previous local read experiments and the measured CPU service times of these remote read experiments, I built a set of linear equations to get the CPU times of the communication parameters such as the RPC request send parameter in the client, the RPC request receive parameter in the file server, the RPC response send parameter in the file server and the RPC response receive parameter in the client.

The difference between the CPU service time of the local read and the CPU service time of the remote read consists of the CPU service time of the communication operation and the CPU service time of the RPC related operation. The constant portion, irrespective of the data size, of the CPU service time of the communication operation was assumed to be zero. If it existed, it was included in the RPC response/request build/evaluation service demand. The variable portion, proportional to the data size, of the CPU service time of the communication operation was assumed to be linearly proportional. The measured service time fitted to the linear line very well when the measured values were plotted on 2 dimensional rectangular coordinate systems and a statistical regression analysis for curve fitting was applied to them as in previous stages. It is assumed that the service time demand of the send operation is equal to the service time demand of the receive operation and the service time demand of the send/receive operation in the client is equal to the service time demand of the send/receive operation in the file server. The best fitting slopes of the linear relationship were selected. The differences between these slopes and the slopes of the proportional portion of the measured CPU service time obtained from the local read experiment consist of the CPU time service demands of the request/response send operation or the CPU time service demands of the request/response receive operation in either the client or the file server.

It is average 0.1375msec per 1500bytes data transferred in the SUN 3/60 workstation, 0.125msec per 1500bytes data transferred in the SUN SPARCstation 470 workstation and 0.1125msec per 1500bytes data transferred in the SUN SPARCstation 10 workstation.

The distributed file system which consists of the SUN 3/60 workstation, the SUN 3/60 workstation and the SUN SPARCstation 10 workstation were used for the remote read experiment. I obtained the CPU service time of the request/response send/receive operation in the client/server of the SUN SPARCstation 10 workstation first and then used it to find the CPU service time of the request/response send/receive operation in the client/server of the SUN 3/60 workstation. The measured overhead when the SUN SPARCstation 10 workstation was used as the file server was different from that when it was used as the client. The former case consumed more CPU time than the latter case. The value of the CPU service time of the request/response send/receive operation in the client and the file server of the two cases were obtained separately and they were averaged for the case of the distributed file system which consists of the Sun 3/60 workstations.

In the fifth stage, the CPU time service demand of the request/response build/evaluation operation in the client/server was extracted from the constant portion of the measured CPU service time in the local read experiments of the second stage and the remote read experiments of the fourth stage. In the fourth stage, it was explained that the differences between the service times of the local read experiments and those of the remote read experiments consisted of the communication overhead and the RPC related overhead such as the RPC request build in the client, the RPC request evaluation in the file server, the RPC response build in the file server and the RPC response evaluation in the client. The parameter values of the communication overhead were already found. Therefore only the parameter values of the RPC request/response build/evaluation operation are left unknown. The RPC request/response build/evaluation overhead does not vary with the data size. It is assumed that the overhead of the RPC request build in the client, the overhead of the RPC request evaluation in the file server, the overhead of the RPC response build in the file server and the overhead of the RPC response evaluation in the client are all equal.

The CPU time service demand of the RPC request/response build/evaluation operation in the client and the file server was obtained to be average 3.33msec in the distributed file system which consists of the SUN 3/60 workstations, 2.5msec in the distributed file system which consists of the SUN SPARCstation 470 workstations and 1.25msec in the distributed file system which consists of the SUN SPARCstation 10 workstations.

In the sixth stage, the accuracy of the service demand obtained in the fourth stage and in the fifth stage was improved and verified. For it, a remote write experiment was performed. In the experiment, a file in the remote workstation was read and as a pipelined operation the read data were written into a file either in the local disk or in the remote disk where the location was different from the location of the read file. The experiment was individually performed between the SUN SPARCstation 470 workstations and between the SUN SPARCstation 10 workstations. The remote writing in the heterogeneous distributed file system was also performed between the SUN 3/60 workstation and the SUN SPARCstation 10 workstation.

It is also possible to extract the CPU service time demand of the send/receive operation in the client and the file server and the CPU service time demand of the build/evaluation operation in the client and the file server from the remote write experiments and the local write experiments of the first stage. In this stage, the same procedure as the fourth stage and the fifth stage was used to find out the communication parameter values and the RPC build/evaluation parameter values. This study compared them with those which were obtained in the fourth stage and the fifth stage. It was confirmed that the values of the communication parameters and the values of the RPC build/evaluation parameters which were obtained in this stage had little difference from those obtained in the fourth stage and in the fifth stage.

In the seventh stage, all obtained CPU service demands were used to calculate the CPU service time. Then the calculated CPU service times were compared with the measured ones in all cases one by one and it was confirmed whether the obtained values of the CPU parameters were accurate enough to be accepted. Since in this stage the values of all parameters demanding the CPU time service were obtained, the accuracy of the obtained parameters can be validated. It was found that the amount of the difference between the calculated one and measured one was within 5% in most cases. Now this study is on sound ground to use the obtained parameters for the following stages.

All CPU time service demands have been obtained and validated so far. From the eighth stage, I/O service time demands will be obtained. In the eighth stage, the response times and the CPU service times of the local write experiments were used together so that the I/O time service demand of the disk I/O operation was obtained.

The disk I/O time varies with the I/O data size. As in the previous stages, this study investigates whether the measured I/O service time can be expressed in a linear function such as y=ax+b, where ax covers the portion of I/O service time proportional to the data size. The measured I/O service times were plotted in rectangular coordinate systems and scatter diagrams were made. And by applying a statistical regression analysis to them for the curve fitting, I selected the best fitting slope values of "a". In the local write experiments of the first stage, the only I/O time service demand proportional to the data size is the I/O time service demand of the disk I/O operation and the only CPU time service demand proportional to the data size is the CPU time service demand of the disk I/O operation. I already obtained the CPU time service demand of the disk I/O operation in the first stage. Therefore the proportional portion of the I/O time service demand of the disk I/O operation can be obtained by just getting the difference between the slope and the CPU time service demand of the disk I/O operation.

It was obtained to be average 4.12msec per 1500bytes data transferred in the SUN 3/60 workstation, 1.55msec per 1500bytes data transferred in the SUN SPARCstation 470 workstation and 1.125msec per 1500bytes data transferred in the SUN SPARCstation 10 workstation.

Now the only unknown value, the constant portion of the disk I/O time service demand can be obtained from the sets of equations built with the measured time of the local read-write experiment, since all other values of the required parameters in the local read-write experiment were already known.

The obtained constant portion was average 130msec in the SUN 3/60 workstation, 60msec in the SUN SPARCstation 470 workstation and 24msec in the SUN SPARCstation 10 workstation.

The constant portion of the I/O service time includes the disk path setup time, the initial rotational latency time, the initial seek time, etc.. The proportional portion of the I/O service time mainly consists of the transfer time in case of small and consecutively allocated data. In case of the SUN SPARCstation 10 workstation, the transfer rate of the local disks was 4(10)Mbps in table 3.2.7.B. Therefore the data transfer time is calculated to be 0.0469(0.0188)msec per 1500bytes data transferred. However, the obtained proportional portion from the measurement experiment is much larger than the calculated data transfer time of the SUN SPARCstation 10 workstation. This is also true in the other two workstations.

Why does this happen? The reason is the irregular seek delay and the irregular latency delay. The seek delay and the latency delay are paid just once if the data are small enough to fit into a track and allocated consecutively within the track. Otherwise, the seek delay and the latency delay will be paid more than once and the effect on response time will be irregular. If the size of data is larger than the size of a track/cylinder and the data is allocated consecutively, then additional track change or/and cylinder change(read/write arm movement) between tracks will occur after the track is fully read. If the data is allocated in fragmented disk spaces, then the response time will be affected by additional seek delay and the latency delay due to more complex and irregular arm movement and the track or/and cylinder change activity. In the experiments, no deliberate effort was made to allocate data consecutively in the disk but data were allocated in a natural and standard way according to the given mechanism by vendors as much as possible. Therefore, the measured values of I/O service time parameters can be said to be more realistic than those which are calculated simply using the average seek time, the average latency time and the average transfer rate provided by the disk vendors.

In the ninth stage, the I/O time service demand of the result processing operation was obtained using the measured service time of the local read experiment. The portion proportional to the data size in the I/O time of the local read experiment consists of the I/O time service demand of the disk I/O operation and the I/O time service demand of the result processing operation. The former is already known and if I find the slope of the I/O time of the local read experiment per unit data size, the value of the I/O time demand of the result processing operation can be obtained straightforwardly. It is assumed that the I/O time of the result processing operation in the read experiment is linearly proportional to the data size. A statistical regression analysis was performed to select the best fitting slope. The constant I/O service time portion irrespective of the data size of the result processing is assumed to be zero.

When I used "cat" command in the local read experiment, the obtained I/O time service demand of the result processing operation was average 520msec per 1500bytes data transferred in the SUN 3/60 workstation, 100msec per 1500bytes data transferred in the SUN SPARCstation 470 workstation and 22msec per 1500bytes data transferred in the SUN SPARCstation 10 workstation as in table 3.2.7.C.

In the tenth stage, the I/O time service demand of the network communication was obtained. Only it is unknown in this stage. By applying the statistical regression analysis to the measured response time of the local read-write experiment of the first stage, the best slope of the response time was selected. The difference between the response time of the local read-write experiment and that of the remote read-write experiment consists of the communication overhead and the RPC overhead. The RPC overhead parameters such as the RPC request build, the RPC request evaluation, the RPC response build and the RPC response evaluation were already obtained in the previous stages. The CPU time service demands of the communication parameters such as the RPC request send, the RPC request receive, the RPC response send and the RPC response receive were already obtained as well. Therefore, The I/O time service demand of the network communication operation can be obtained using the two obtained values of the slope. The constant I/O time portion of the communication overhead irrespective of the data size is assumed to be zero. The I/O time portion of the communication overhead proportional to the data size such as the I/O time service demand of the network interface operation and that of network operation is assumed to be linearly proportional to the data size. The nominal speed of Ethernet is known to be 10Mbps. The speed was used to calculate the network transmission time. In this phase, the only unknown parameter value is the I/O time service demand in the network interface unit of both the client and the file server. By assuming that the I/O time service demand of the network interface unit in the sending site is the same as that in the receiving site, I can solve the two simple equations to get the I/O time service demand of each network interface unit.

In the case of Ethernet, the preprocessing time of the communication operation of the network interface unit in the client or the postprocessing time of the communication operation of the network interface unit in the file server was average 5.2625msec per 1500bytes data transferred in the distributed file system which consists of the SUN 3/60 workstations, 1.775msec per 1500bytes data transferred in the distributed file system which consists of the SUN SPARCstation 470 workstations and 0.2875msec per 1500bytes data transferred in the distributed file system which consists of the SUN SPARCstation 10 workstations. The network transmission time of Ethernet was calculated to be 1.2msec per 1500bytes data transferred.

In the eleventh stage, the same procedure as that of the tenth stage was applied to the measured time of the previous local read experiment and the previous remote read experiment so that the accuracies of the service demands of the communication parameters were confirmed.

In the twelfth stage, I confirmed the accuracies of the service demands of the communication parameters and those of RPC parameters by performing two experiments using the "ping" facility and the "spray" facility. A sequence of the "ping" operation and the "spray" operation were performed in stand-alone mode between two interconnected workstations using NFS via ETHERNET. The "Ping" sequence sends the specified number of ICMP ECHO-REQUEST packets to the network hosts and reports the round trip time. The "spray" sequence sends the specified number of one-way stream of packets to the network hosts using RPC and reports the transfer rate and the service time in the CPU time and the response time. The experiments were individually performed between the SUN SPARCstation 470 workstations and between the SUN SPARCstation 10 workstations. The experiments in the heterogeneous distributed file system were also performed between the SUN 3/60 workstation and the SUN SPARCstation 10 workstation.

The sequence of the "ping" test consists of the request send operation and the response receive operation in the client and the request receive operation and the response send operation in the file server. The sequence of the "spray" test consists of the RPC build-up operation in the client, the RPC request send operation in the client, the RPC request receive operation in the file server, and the RPC request evaluation operation in the file server. By using the measured service times of the local read experiment, the remote read experiment, the local write experiment, the remote read-write experiment, the "ping" experiment and the "spray" experiment, I cross-checked the accuracy of the obtained service demands of the communication parameters and that of the RPC parameters.

In the case of Ethernet, the response time of the total communication operations from the client to the file server was measured to be average 25msec per 1500bytes data transferred in the distributed file system which consists of the SUN 3/60 workstations, 10msec per 1500bytes data transferred in the distributed file system which consists of the SUN SPARCstation 470 workstations and 4msec per 1500bytes data transferred in the distributed file system which consists of the SUN SPARCstation 10 workstations.

Table 3.2.7.C shows the parameter values that I obtained from stage 1 to stage 12. A total of 20 parameters were defined and quantified. In the seventh stage, I validated the accuracy of the obtained values of the CPU time related parameters. Now the accuracy of all parameter values can be validated since all were obtained. I used all of the obtained parameter values to calculate the response time of each case and compared it with the measured response time of each case one by one. I found that the amount of difference between the calculated one and measured one was within 5% in most cases. Now this study is on sound ground to use the obtained values of all parameters for the simulation.

So far this study has not used any sophisticated measurement tool and not modified any part of the system softwares such as the operating system and the communication software for the performance measurement for parameterization. However, the values of all parameters have been successfully obtained and they are very precise.

3.3 The File Systems of the Shared Memory Systems under Study

This section describes the file systems of the shared memory systems which are studied in this research. Every effort was made to represent general UNIX file systems. The shared memory systems under study use the shared variable mechanism not the message passing mechanism. They have the shared bus architecture and the symmetric property. Parallel processing in the file system processing such as the parallel file systems is not considered but the multiprocessing is considered in this study. That is, a request is serviced as a whole process unit and is not divided into small pieces for parallel processing either for the data parallelism or the program parallelism.

I describe the internals of the file systems of the shared memory systems under study by describing how the requested data are processed as I did when I described the distributed file systems in section 3.1. In this study, only the requests from the local users are considered, that is, this study only deals with the locally attached terminals so that the communication activity does not exist.

Local users send read requests or/and write requests to the system. The system interprets the requests first. After interpretation, they receive two distinct services : the file handling operation and the disk I/O operation. The file handling operation consists of directory handling, file table lookup, updating file tables, opening files, closing files, etc.. The disk I/O operation consists of disk I/O path setup operation through the disk interface unit, physical disk I/O operation, etc.. The physical I/O operation consist of three major operations : seek operations, set sector operations and transfer operations. The three major operations were already explained in section 3.1. If the request is a write request, the data are buffered to the memory first via the system bus and then written into a disk. And if necessary, the final system message is processed to the user by the result processing mechanism. If the request is a read request, the data are read first from the disk and then buffered to the memory via the system bus. The read data are send to the user screen or only the system message is processed to the user or no action is taken by the result processing mechanism depending on the user request. In the first case, the I/O operation between the memory buffer and the designated screen by the user is performed via the system bus.

3.4 The File System Performance Model of the Shared Memory Systems

This study applies the queueing network theory to build the performance models of the file systems of the shared memory systems as I did in modeling the distributed file systems in section 3.2. The computer system such as the SUN workstation which has only one CPU is considered as a special case of the shared memory systems, that is, the shared memory system which has only one CPU. The virtual server concept is also applied in building the performance models.

3.4.1 The Virtual Server Models

The shared bus can be explicitly represented as a service center and all services from and to the user terminals or the screens go through the service center as in figure 3.4.1.A. Like the local area network of the distributed file systems, the shared bus is a bottleneck point of the shared memory system which has shared bus architecture. This study focuses on comparing the file access performance of the distributed file systems with that of the file systems of the shared memory systems and does not focus especially on the analysis of the traffic of the shared bus. Hence, the bottleneck effect of the shared bus is not explicitly investigated in this study. From this viewpoint, the performance model of figure 3.4.1.B is used in this study. However, considering the bottleneck effect of the shared bus, up to 30CPUs are used during the simulations in the study, according to the prevailed belief that, up to 30CPUs, the performance is not usually degraded due to the bottleneck effect of the shared bus. As assumed in section 3.3, only local users are considered so that the communication cost is not considered at all.

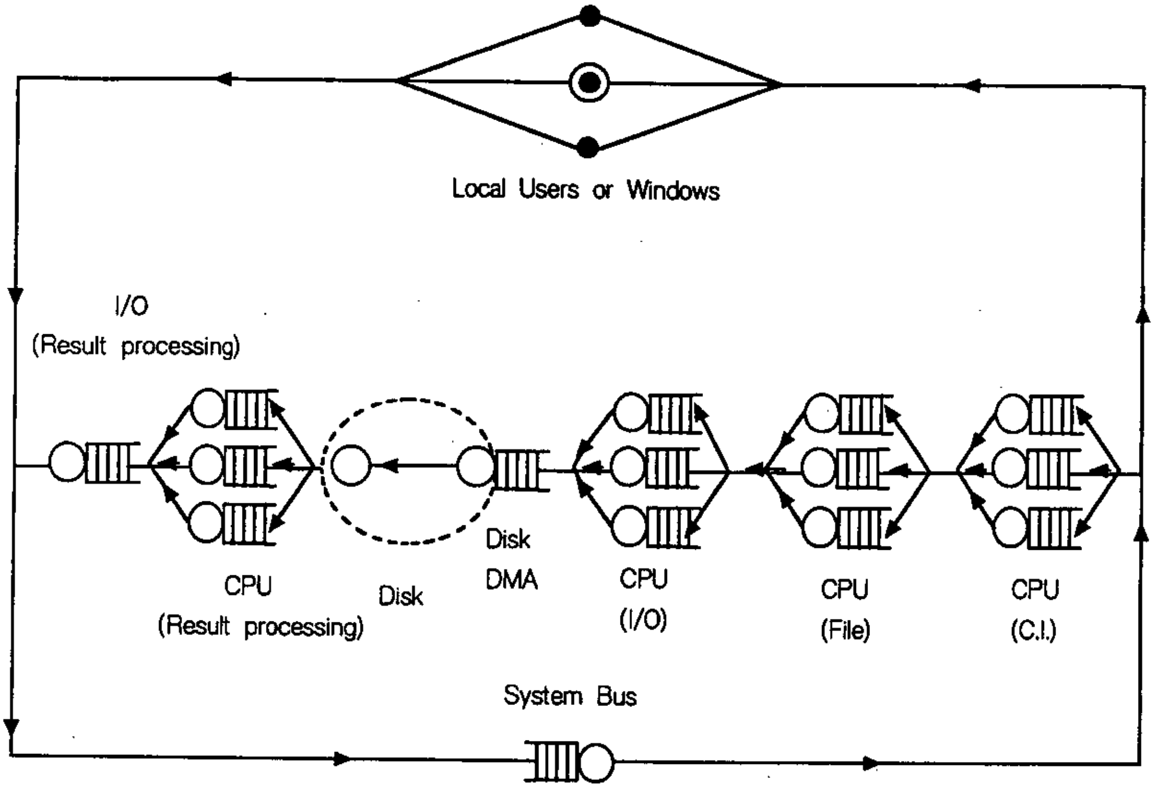


Figure 3.4.1.A : The virtual server model of the shared memory system which represents the system bus as a service center.

In figure 3.4.1.B, the performance model explicitly represents the initial command interpretation service of the CPU, the file processing service of the CPU, the CPU service for the disk I/O operation, the disk I/O service of the disk interface unit and the disk and finally the result processing service of the CPU and the I/O service for the screen display if necessary. As in the performance models of section 3.2.6, the details of the operation in the disk I/O system such as the disk path connection, the RPS missing, the rotational positioning, the seek, the data transmission operation, etc. are not represented explicitly as the service centers in the model but implicitly in the values of parameters and the simulation programs. The disk interface unit and the disk are represented as tandem queues so that the disk interface unit is seized first and, until the service in the disk finishes, the seized interface unit is not released. The disk interface unit and the CPU cooperate to do preprocessing work such as the disk I/O path set-up, etc., before starting the disk I/O operation, and postprocessing such as moving data from the buffers of the disk interface unit into the buffers of the memory, etc., after finishing the disk I/O operation. For the cooperation, the service center of the disk interface unit and the virtual service center of the CPU for the disk I/O operation are seized and released at the same time. If any of the two required resources is unavailable then the other should wait until the unavailable one becomes free and both of them can be seized at the same time.

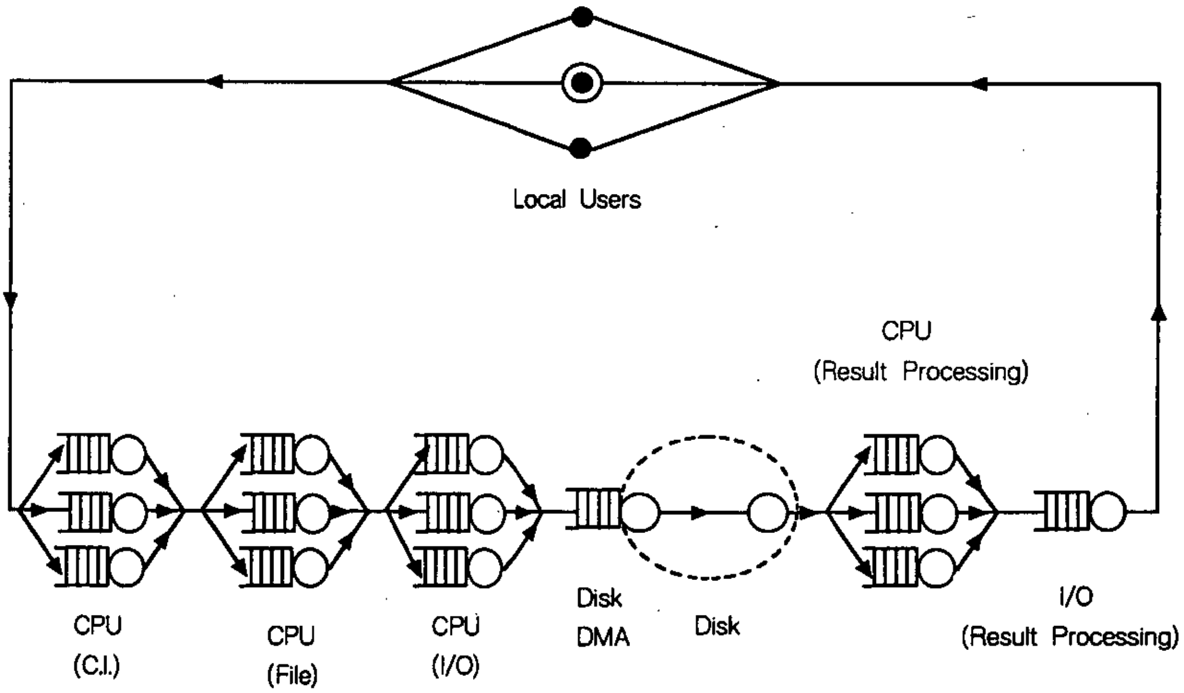


Figure 3.4.1.B: The virtual server model of the shared memory system which does not represent the system bus as a service center.

Caching is represented explicitly in the model. The represented caching are caching in the memory and caching in the disk interface unit. Figure 3.4.1.C shows the caching representation in the model.

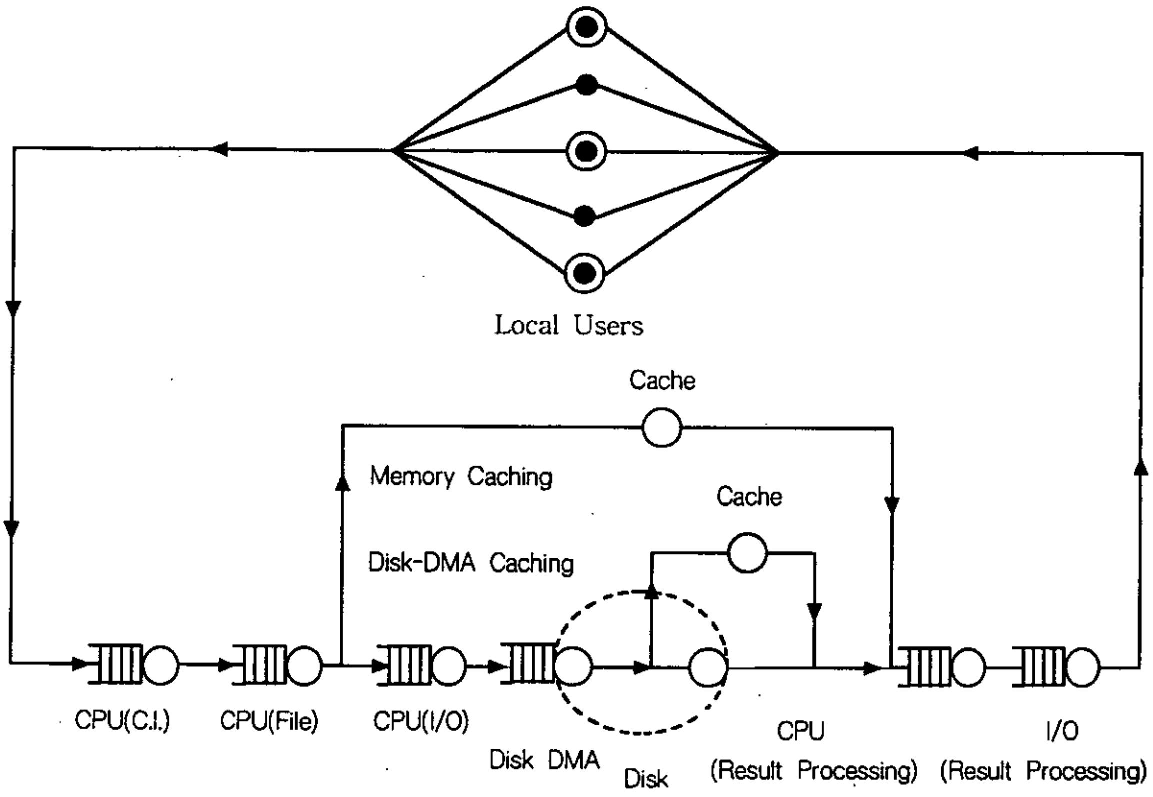


Figure 3.4.1.C : The virtual server model of the shared memory system which represents caching : when the single CPU is used.

Figure 3.4.1.D shows the performance model when the multiple disks and the multiple disk interface units are used. Each disk has its own disk interface unit. They are homogeneous in terms of performance. All others remain the same as figure 3.4.1.B. An infinite number of disks and disk interface units can be served in the model assuming that enough disk paths are guaranteed in terms of the hardware and the software.

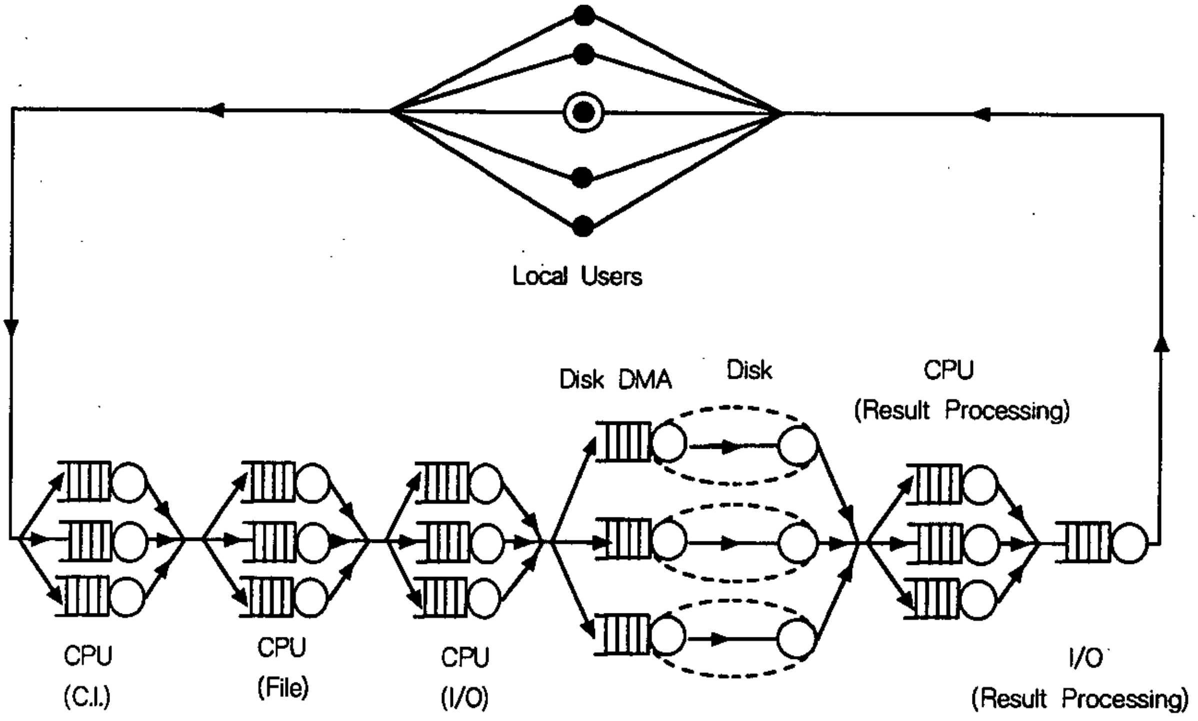


Figure 3.4.1.D : The virtual server model of the shared memory system which has multiple disks and the multiple disk interface units.

3.4.2 Performance Parameters and Parameterization

The specially designed measurement for the parameterization of the distributed file systems which was described in section 3.2.7 was also used for the performance parameterization of the file system of the shared memory system. First, the CPU time service demands were obtained from the measured CPU service times in the experiments. Second, the obtained CPU time service demands were validated. Third, the I/O time service demands were obtained from the measured response times in the experiments. Finally, all obtained service demands were validated. Table 3.4.2.A shows the obtained values of the parameters.

	Operation		SUN 3/60 (msec)	SUN SPARC 470 (msec)	SUN SPARC10 (msec)
CPU	Command interpretation	f	80.00	20.00	20.000
CPU	File handling	f	20.00	10.00	5.000
CPU	Disk I/O	p	0.4	0.15	0.125
I/O	Disk interface unit	f	130.00	60.00	24.000
I/O	Disk interface unit + Disk I/O	p	4.12	1.55	1.125
CPU	Result processing	p	0.35	0.30	0.250
I/O	Result processing	p	520.00	100.00	22.000

- CPU: CPU time, I/O: I/O time
- p: proportional to the data size, f: fixed (constant)
- The values of all parameters proportional to the data size are per 1500bytes data transferred.
- The values of all parameters constant to the data size are per one transaction, regardless of the transferred data size.

Table 3.4.2.A : The parameters for the virtual server performance models of the shared memory systems.

In the first stage, from the first stage of the parameterization procedure of the distributed file system, I found the CPU time service demand of the disk I/O operation. Then, from the second stage of the parameterization of the distributed file system, I found the CPU time service demand of the result processing operation to the window screen where the command had been issued. As the third step, from the third stage of the parameterization of the distributed file system, I found the CPU time service demand of the command interpretation operation and the CPU time service demand of the file handling operation.

In the same way as the seventh stage of the parameterization of the distributed file system, the obtained CPU time service demands were validated and I found that the amount of difference between the calculated one and the measured one was within 5% in most cases.

In the third stage, from the eighth stage of the parameterization of the distributed file system, I found that the disk I/O time service demands : both the constant portions and the proportional portions. Then I found the I/O time service demand of the result processing operation from the ninth stage of the parameterization of the distributed file system.

In the final stage, I used all of the obtained values of the parameters to calculate the response time of each case and compared it with the measured response time of each case one by one. I found that the amount of difference between the calculated one and measured one was within 5% in most cases.

3.5 Workload Characterization and Workload

To drive the developed performance models, artificial workloads are needed. The workload is very important for performance evaluation study. To get the accurate, realistic and representative workload for the developed performance model, I have to gather the real workload from the target system and characterize it. Generally, it is not easy to extract the accurate, realistic and representative artificial workload from the real workload. Section 3.5.1 presents a procedure to extract the accurate, realistic and representative artificial workload from the real workload and how I obtained the artificial workloads used as the inputs to the performance models in this research. Section 3.5.2 describes the artificial workloads.

3.5.1 Workload Characterization

In this section, my workload characterization procedure is introduced. Then this section describes from where I obtained the real workloads and how I characterized the real workloads to make the artificial workloads. Other' related work is discussed where appropriate.

Below, the six steps of my workload characterization procedure are introduced. First, define the objectives and the policies such as (i)whether we do the system independent workload characterization or the system dependent workload characterization, (ii)whether we focus on the interactive workload or the batch workload or both of them, (iii)whether we focus on the remote file access workload or include the local processing activity as well, (iv)whether we focus on the file management workload or the process processing workload, (v)to what degree we consider the statistically significant accuracy, etc..

Second, select the workload characterization parameters. The parameters are usually either system dependent or system independent. The system dependent parameters are based on the amount of the consumed system resource to process the required work. The parameters abstract the physical resource demand from the amount of resource consumed in the system. The system independent parameters are based on the amount of work done in the system. The parameters abstract the logical resource demand, i.e., the work demand from the amount of work done in the system. The workload characterization based on the system dependent(independent) workload parameters produces the system dependent(independent) artificial workload. It can be found that the work demand in the high performance system is greater than that in low performance system. That is, the work demand is somewhat proportional to the performance(speed) and the capacity of the system. We can see this phenomenon in the studies by Baker et al.[BAKER etal 91] and Ousterhout et al.[OUSTERHOUT etal 85] as I explained in section 2.7. Therefore exactly speaking in terms of the computer system scale and the computer system power, there might be no absolutely system independent workload or absolutely system independent workload characterization. However in terms of the workload parameters, there exist the system independent workload or the system independent workload characterization and it is necessary for us to decide whether we do the system independent workload characterization, that is, use the system independent workload or do the system dependent workload characterization, that is, use the system dependent workload.

Third, gather the real workload data. Three methods are available to collect the workload data. The most common and easiest way to get the real workload data is to use the account files and/or the system provided utilities. The performance related packages can be also used. The last method is to use the self developed kernel programs. Also it has to be decided how long we collect the real workload data in order to keep the representativeness.

Fourth, analyze the gathered real workload data in order to obtain the parameter values such as the file size distribution, the ratio of the used access method such as the sequential access to the random access, the ratio of the read operation to the write operation, the CPU usage(demand), the memory usage, the disk I/O traffic, the communication traffic, etc.. For example, in the system dependent workload characterization we find the CPU time, the disk I/O time, the communication time via the network, etc., and in the system independent workload characterization the CPU demand in the unit of program size(number of steps), the number of disk I/O bytes, the number of the transferred packets(bytes) via the network, etc..

Fifth, produce the artificial workload. Statistical methods such as clustering, etc. are often used to produce the artificial workload as in the Calzarossa and Ferrari's work[CALZAROSSA & FERRARI 86], Lee et al.'s work[LEE etal 94] and Smith's work[SMITH 81]. Finally and sixth, calibrate and validate it.

The workload characterization policies of this study based on the above procedure are the following. This study focuses on both the interactive workload and the batch workload, the file management workload, the conventional text data workload and the future workload which contains large scale data as well as the conventional text data. I tried to characterize the workload using the system independent workload parameters in order to feed the system independent inputs to the virtual performance models as much as I can.

The workload characterization work in this study is primarily based on the measured data provided by Baker et al.[BAKER etal 91] and Ousterhout et al.[OUSTERHOUT etal 85] and the data gained from the 1993 International EXPO computer systems which had the integrated heterogeneous file servers including the image file servers with more than 790 clients via compound local area network of FDDI and Ethernet[LEE etal 93], [LEE etal 95]. The measured workload data in the BSD 4.2 UNIX system of the VAX 11/780 systems by Ousterhout et al.[OUSTERHOUT etal 85] and the measured workload data in the SPRITE distributed system of 40 workstations by Baker et al.[BAKER etal 91] for around one year were carefully analyzed and several artificial workloads were abstracted. The abstracted workloads were carefully compared with the analyzed workload data in the 1993 International Exposition Computer System[LEE etal 93], [LEE etal 95]. Then through several calibrations and validations, I finally gained the workloads used in this study. All those steps have been taken in order that the artificial workloads represent the real workloads accurately. In this way, confidence was pursued in the accuracy, the realism, the representativeness and the generality of the artificial workloads.

Ousterhout et al.'s data were taken as a measured data for the file systems of the local shared memory systems and Baker et al.'s data were taken as a measured data for the distributed file systems. The reason to choose them as the base data for the workload characterization is that I believe these data are accurate and representative workload data of general UNIX based file systems in at least two environments.

Ousterhout et al. measured the file I/O traffic of their three VAX 11/780 systems using BSD 4.2 UNIX system in the computer science department of University of California, Berkeley. Lazowska et al.[LAZOWSKA etal 86] measured the file I/O traffic in distributed file systems(diskless workstation environments). The two contemporary works in the two different system paradigms shows the similar file I/O traffic rate. Baker et al. measured the file I/O traffic in the Sprite distributed system where the load was balanced(allows process migration), in the same organization as Ousterhout et al.'s organization.

Lazowska et al. used a batch workload.[4] They did not explain how to get the workload and the internal detail of the workload and therefore I can not check whether it represents the real workload in their environment correctly or not. By a measurement[5], they got 2160Kbytes data traffic and 156seconds local processing time(stand-alone processing time). By a simple calculation, they assumed that the local processing time for the batch workload is 289msec per 4Kbytes request. They also reported the local processing time of 106msec per 4K request for the highly interactive workload by a measurement[6]. They conducted an experiment[7] to find the data traffic volume per active user and got 4Kbytes/second data traffic per an active user. They used this 4Kbytes as the data traffic size of a request and they recalculated every measured data transfer activity in terms of the 4Kbytes transferred. That is, their workload is based on the data unit of the 4Kbytes size. Therefore, a request in their study consists of 4Kbytes data traffic and the local processing time(106msec in case of the highly interactive workload or 289msec in

[4] LAZOWSKA et al. [LAZOWSKA etal 86] : The batch workload consists of "compile/assemble/link sequences for several different compilers and several different source programs".

[5] They measured workload parameters such as local processing time in the clients, and data traffic volume in the idle diskless SUN-2(CPU : MC68010) workstations with SUN/ND(Network Disk), the previous version of SUN/NFS.

[6] They "monitored a number of highly interactive users engaged in software development on the environment" but did not explain the representativeness of their monitoring results.

[7] They supervised a group of software developers on workstations to work hard for 30minutes and measured data traffic volume per an active user.

case of the batch workload) per the 4Kbytes data traffic. They assumed the idle time in the client to be the user think time. They defined active users as those "who caused any file I/O in a second interval". They assumed the remote file access to be 100% sequential access, one seek operation per every two disk operations during the disk I/O operation, the ratio of the read request to the write request to be 3 to 1. They used 4Kbytes and 8Kbytes disk file block size and 1Kbyte and 4Kbytes packet size in the transmission over the local area network.

Ramakrishnan et al.[RAMAKRISHNAN etal 86] characterized their workload as the 151.8Kbytes data traffic per a file copy. There was 10seconds user think time between each user request for a file copy. Each copy consists of 100 requests. The size of the request was 1518bytes which is the maximum packet size of IEEE 802.3 Ethernet. The inter-request time, that is, the processing time between each request in the client was characterized to be 10msec. The client must process the response message received from the file server before sending the next successive request. They did not consider the stand-alone processing time, or the local processing time in the clients in their workload but considered only the remote file access activities. Therefore they guessed that more users than indicated by their model might be supported in actual systems.

PERROS et al.[PERROS etal 85] used the bulk file transfer workload which consists of the requests reading/writing 20Mbytes files. Each request was divided into 128Kbytes sub-requests with the 100msec inter-request delay and each sub-request was further divided into the unit request of 2Kbytes size with zero inter-request delay.

3.5.2 The Workload

This section explains the artificial workloads which this study used to drive the performance models. Table 3.5.2.A shows the used artificial workloads. They were used as the common workloads for the simulation of both the distributed file system and the file system of the shared memory system.

	Transaction size (Kbits / transaction)		Transaction number when the number of active clients = 100 (transactions /sec)	
	Average	Standard Deviation	Average	Standard Deviation
Case 1	64	288	22.75	12.75
Case 2	376	2,144	4.0	3.75
Case 3	405.6	768	22.75	12.75
Case 4	2,528	6,464	4.0	3.75
Case 5	2,528	6,464	22.75	12.75
Case 6	14,852	37,976	4.0	3.75

Table 3.5.2.A : The workloads used in this study

As the normal workload pair, the case 1 workload and the case 2 workload in table 3.5.2.A were used. As the 1st alternative workload pair, the case 3 workload and the case 4 workload in table 3.5.2.A were used. As the second alternative workload pair, the case 5 workload and the case 6 workload in table 3.5.2.A were used. The case 1 workload, the case 3 workload and the case 5 workload represent the steady state workload. They are primarily based on the measurement data over the 10minutes interval by Baker et al.[Baker etal 91].[8] That is, the client which caused any file I/O over the 10minutes interval was considered to be active and the data traffic caused by all active users during the 10minutes interval was

(8) The 40 units of 10MIPS clients workstations with the 24Mbytes to 32Mbytes main memory individually such as the SPARCstation, the SUN 3, the DECstation 3100 and the DECstation 5000 were configured in the Sprite Distributed System of the EECS department of The University of California, Berkely : four file servers were used. Total 70 users were registered : 30 daily and primary users, and 40 frequent and non primary users. The departmental systems were used by the operating system researchers working on the design and the simulation of the new i/o subsystems, the students and the faculty members working on the VLSI circuit design and the parallel processing, the administrators and the graphic researchers.

averaged : I call these workloads as the 10minutes workloads. In the case, they measured average 9.1(the standard deviation is 5.1) active users with the average throughput of 8Kbytes(the standard deviation is 36Kbytes) when 40 client workstations were connected. I interpreted it as the average transaction number of 9.1(the standard deviation is 5.1) per second with the average transaction size of 8Kbytes. During a short measuring period, the caused data traffic rate averaged for the period might be less than the requested data traffic averaged for the period even though the total amount of the caused data traffic should be same as the total amount of the requested data traffic. If the system is measured during a long period and the average system utilization is low, which means low competition on the system resources and little queueing delay, the caused average data traffic rate per second averaged for the long period is close to the requested data traffic averaged for the long period. The measuring period was 24 hours and the measured value was averaged for the period.[Baker etal 91]. Dr. Shiriff, an author of the work[Baker etal 91] confirmed that the system utilization was very low during most of their measuring period. Hence I believe the artificial workloads based on the interpretation have little difference from the real workloads.

The case 2 workload, the case 4 workload and the case 6 workload represent the bursty state workload. They are primarily based on the measurement data over the 10seconds interval by Baker et al.. That is, those who caused any file I/O over the 10seconds interval were considered to be active and the data traffic caused by all active users during the 10seconds interval was averaged : I call these workloads as the 10seconds workloads. In the workload pairs such as the case 1 workload and the case 2 workload, the case 3 workload and the case 4 workload and the case 5 workload and the case 6 workload, the data transfer rate per second of the 10minutes interval workload is slightly smaller than that of the 10seconds workload in the each pair, respectively. In terms of the characteristics of the file I/O traffic, the 10minutes workloads can be interpreted to represent steadiness and the 10seconds workloads represent burstiness. Based on these interpretations, this study used the above 6 workloads to comparatively evaluate the effect of bursty file I/O traffic and steady file I/O traffic on the file system performance of the two different system paradigms.

In the first alternative workload pair, that is, the case 3 workload and the case 4 workload, the mean and the standard deviation of the transaction sizes are adopted from the workloads measured by Baker et al. as they are but the mean and the standard deviation of the transaction rate are adjusted so that the performance results can be compared with those of the normal case.

In the case 5 workload of the second alternative workload pair, the mean and the standard deviation of the transaction sizes are extrapolated from the workloads measured by Baker et al. so that in terms of the ratio the average size of the transactions in the workloads has regular growth all the time.[9] The mean and the standard deviation of transaction sizes of the case 6 workload, the counterpart of the case 5 workload, are obtained by simple calculations[10]. The ratio between the means of the 10minutes workloads and the means of the 10seconds workloads are kept similar all the time.[11] The transaction arrival rates of the second workload pair are adjusted as in table 3.5.2.A so that the performance results can be compared with those of the normal workload pair and those of the first alternative workload pair.

After the representativeness of these 6 workloads was carefully investigated in the very large scale distributed system[LEE etal 93], [LEE etal 95], the sizes of the workloads and the transaction rates of the workloads were accepted as those of

[9] The size of the case 3 workload is 6.338 times as large as the size of the case 1 workload and the size of the case 5 workload is 6.233 times as large as the size of the case 3 workload.

[10] The mean is calculated as 2528Kbits * (376Kbits / 64Kbits) = 14,852Kbits and the standard deviation is calculated as 6464Kbits * (376 Kbits / 64 Kbits) = 37,976Kbits.

[11] The case 1 workload : the case 2 workload = 1 : 5.875. The case 3 workload : the case 4 workload = 1 : 6.21. The case 5 workload : the case 6 workload = 1 : 5.875

the artificial workloads.

In the workloads, the transaction size is assumed to have log-normal distribution so that every possible size of transaction can be generated within the given boundary and runs together or it is assumed to be fixed at the mean value so that the effect of the two different distributions can be compared. For example, the case 2 workload was run in the log-normal distribution with the average of 376Kbits/sec and the standard deviation of 2,144Kbits/sec or as the constant size of 376Kbits/sec. If the normal distribution is used for the transaction size distribution, then I have to cut the negative values among the values generated by the normal distribution. Unfortunately, the portion of the negative values in the given workloads is not negligible but significant due to the relatively large standard deviation values compared with the mean values. Thus, the left cut-off normal distribution gives the right-skewed(positive skewness) normal distribution and the mean and the standard deviation shift to larger values. For example, for the first case workload of which the mean value is 8Kbytes and the standard deviation is 36Kbytes, I found the left cut-off normal distribution without any compensation generates the mean values almost 4 times larger than the specified mean values. Through elaborate tests, I found that most of the measured workload values in Berkely[Baker etal 91], [Ousterhout etal 85] agree remarkably well with the log-normal distribution. If the value of an observed variable is a random proportion of the previously observed values, the log-normal distribution is known to be an appropriated model of the processes.[PRITSKER 84] I think the file access activity of most users has similar characteristics to the above property. That is, I think the value of an observed variable in the file access activity of a user is usually a random proportion of the previously observed values, if the number of the observation is large enough.

As the workloads, the five different transaction sizes were used - 64Kbits, 376Kbits, 405.6Kbits, 2.528Mbits, and 14.853Mbits - so that the transaction size growing trend following the available computing power growth could be investigated. I think the transactions of the average 64Kbits is a typical transaction size of the text data manipulated in contemporary computer systems and the transactions of the average 14.853Mbits is large enough to cover the transactions of the large data manipulated in future(not very far) computer systems. Analyzing the trends in computing practices, I expect the transactions of the average 376Kbits, 405.6Kbits, 2.528Mbits will be common soon.

In the virtual server models, the bulk data are always divided into the requests of which each has constant size of 12,000bits, which is based on the maximum packet size of the IEEE 802.3 Ethernet : the size of the pure transferred data is 1500bytes and the size of the overhead portion is 18bytes.

In the workloads, the transactions are assumed to occur according to the Poisson distributions, that is, the distributions of the inter-arrival times are the exponential distributions or the log-normal distributions or the constant distributions at the mean values. For example, in case of the Poisson distributions, the case 2 workload has the Poisson arrival of the average 3.75transactions/sec when either 100 workstations in the distributed file systems or 100 local users in the shared memory systems are used.

In the Sprite distributed system environment, Baker et al. measured that read-only accesses and write-only accesses were the majority of all accesses and the read-write accesses were the minority of all accesses.[12] Based on these measurements, in the workloads used in this study, I did not consider the read/write access but considered the read-only access and the write-only access. However, my performance models and simulation programs are ready to accept

[12] In the Sprite distributed system environment, Baker et al.[BAKER etal 91] measured that read-only accesses were average 88%(range : 82-94%) of all file accesses, the write-only accesses were average 11%(range : 6-17%) and the read-write accesses were only average 1%(range : 0-1%). The average percentage of each file access pattern among all transferred data was 80%(range : 63-93%) in the read-only, 19%(range : 7-36%) in the write-only and 1%(range : 0-3%) in the read-write.

read/write access without any modification.

In the several VAX/11 780 systems, Ousterhout et al.[OUSTERHOUT etal 85] measured that majority of accesses were whole file accesses[13] and the sequential accesses were the majority accesses and the random accesses were rare.[14] In the Sprite distributed system environment, Baker et al.[BAKER etal 91] measured that the whole file accesses were also the majority and the random file accesses were also rare.[15] Based on these measurements, in the workloads used in this study, only the sequential whole file accesses are considered.

3.6 The Performance Metrics

Typical performance indices are the response time, the queue length, the service time, the waiting time, the resource utilization, etc.. This study measured the response time(the average, the standard deviation, the coefficient of variation, the minimum value and the maximum value and the distribution), the queue length(the average, the standard deviation, the maximum length and the minimum length), the average waiting time, the utilization(the average, the standard deviation and the maximum utilization), the number of the transactions observed (the average, the standard deviation, the coefficient of variation, the minimum value, the maximum value and the distribution) and the inter-arrival time(the average, the standard deviation, the coefficient of variation, the minimum value,

(13) In the several VAX/11 780 systems, Ousterhout et al.[OUSTERHOUT etal 85] measured that "About 70% of all file accesses are whole file transfers, and about 50% of all bytes are transferred in whole file transfers."

(14) The sequential read-only accesses were over 90% among all read-only accesses. The sequential write-only accesses were over 95% among all write-only accesses. The data transferred sequentially were over 65% among all data transferred.

(15) Average 78% of the read-only accesses were the whole file accesses, only average 3% of the read-only accesses were the random file accesses and average 17% of the read-only accesses were other sequential file accesses. Among the data transferred, the average percentage was 89%, 7% and 5% respectively. In the write-only accesses, the access average was 67% in the whole file accesses, only 4% in the random file accesses and 29% in other sequential file accesses. Among the transferred data, the average percentage was 69%, 11% and 19% respectively. All read-write accesses were the random accesses.

the maximum value and the distribution).

3.7 Simulation

The evaluation of the performance models based on the queueing network theory can be done by either the analytic approach or the simulation approach. If we use the analytic approach to solve the queueing network models, there could be two solutions : the exact solutions and the approximated solutions. Compared with the simulation approach, the analytic approaches are relatively cheap to get the solutions, nevertheless effective and flexible to be used for the queueing models but the exact solutions exist for only some cases and the approximated solutions are also limited. The simulation approach can solve almost all cases with the desired accuracy but is relatively expensive in terms of the effort and the modification of models may require relatively high expense.

In the analytic approach, the performance indices are found mathematically. The accuracies of the analytic solutions are known to be within 10% error for the average job throughput and the device utilization and within 30% error for the average response time[LAZOWSKA etal 84]]. The analytic approach is useful only if the solutions can be obtained using a reasonable amount of computations and storages. Exact solutions exist for the product form queueing networks and many computationally efficient and numerically stable algorithms have been proposed to find the exact solutions for them. However, if a product form queueing network is large, it is impossible to get the exact solution due to the unmanageably large number of states and only approximate solutions exist.

Most of performance evaluation studies based on the queueing network theory have produced the analytic solutions. In them, the internal details of the target systems have been often simplified too much. However they got the required analytic solutions with little cost for the time and the storage for the calculation of the solutions. The most of the queueing network models introduced in this thesis also were solved analytically. That is, the performance models of the distributed file systems by Bester et al.[BESTER etal 84], Ferrari et al.[FERRARI etal 83], Goldberg et al.[GOLDBERG etal 83], Lazowska et al.[LAZOWSKA etal 86], Perros et al.[PERROS etal 85] and Ramakrishnan et al.[RAMAKRISHNAN etal 86] and the performance models of the network communication in the local area networks in Bhuyan et al.[BHUYAN etal 89], Bux[BUX 89], Jain[JAIN 90] and Shoch et al.[SHOCH etal 80] were solved analytically.

Analytical techniques can solve only for limited range of features, but simulations can solve vast range of features with the desired accuracy: simulations can solve complex situations which analytical techniques can not. Analytical techniques usually provide the mean values only but simulations can provide estimates of distributions and higher moments. Simulations can solve dynamic or transient behaviours while analytical techniques are usually used to solve static state behaviours. Law et al.[LAW etal 82] give some reasons for the popularity of simulations in detail. Simulations are often used to validate analytic results.

I preferred to use simulations as the primary method to solve the performance models since my models are complex and I want to have precise solutions for the models. However, the analytic approach was sometimes used to solve part of the performance models as a supplementary method. So, a hybrid approach was taken to take advantage of both the simulation and the analytic approach in this study. Shantikumar et al.[SHANTIKUMAR etal 83] survey hybrid techniques.

Two different types of simulations have been widely used in computer performance evaluations : trace driven simulations and stochastic discrete event simulations. In the trace driven simulations, a sequence of trace is first obtained through the measurement of real existing systems and used to drive the simulations. In the simulations, often the models do not have queueing structures.

The advantage of the trace driven simulations is that analysts do not have to construct complicated stochastic workload models. Its disadvantage is that analysts may have difficulties in obtaining good representative traces in practice. In multiprogramming systems, trace driven simulations may yield wrong results due to wrong driven traces. Clark[CLARK 83] describes the difference between measured data and the result of the trace driven simulation for this reason. The trace driven simulation is hardly found in the performance evaluation studies of distributed file systems.

The stochastic discrete event simulation is driven by the sequences of random or pseudorandom numbers with user specified distributions. Occasionally traced data are used in conjunction with random sequences to drive queueing model simulations[SHERMAN etal 72]. The stochastic discrete event simulation has been used widely in performance evaluation studies. First the analyst specifies the model structure. Second the analyst specifies the distributions of the sequence of random or pseudorandom numbers generated by the computer system. Third the analyst drives the model by the sequence of random or pseudorandom numbers generated by the computer system. In the simulations of this study, I use stochastic discrete event simulation methods.

General simulation languages have high level constructs and facilities common to all simulations. They usually offer random number generating facilities, event scheduling facilities, queue management facilities and statistics gathering and reporting facilities. In general purpose simulation languages, there are GPSS[SCHRIBER 74], SIMSCRIPT[KIVIAT etal 73], GASP-IV[PRITSKER 74], SIMULA([DAHL etal 66], [POOLEY 86]), SLAM-II[PRITSKER 84], SIMAN[PEGDEN 86], etc.. There are some general purpose simulation languages which have been developed by the addition of simulation primitives to existing programming languages. They are PASCAL-SIM[OKEEFE 86B], PASSIM[UYENSO etal 80], SIMPAS[BRYANT 80], SIMCAL[MALLOY etal 86], Micro PASSIM[BARNETT 86], SIMTOOLS[SEILA 88] which are based on PASCAL, A*SIM[MELDE etal 88] which is based on ADA, SIMOD[LECUYER etal 87] which is based on Modula-2, CSIM which is based on C, VSIM[CALHOUN etal 87] which is based on C++, TC-PROLOG which is based on PROLOG, etc.. This study uses SLAM-II general simulation language without TESS(a graphical part) facility. SLAM-II has very convenient functions with which I could easily implement the virtual server concept of the performance models into the simulation programs.

As simulation packages for queueing network systems, there are GIST[SINCLAIR etal 86], NETWORK-II.5[GARRISON 87], NUMAS[MUELLER 84], PAWS[PAWS 83, ANDERSON 84], PANACEA[RAMAKRISHNAN etal 82], QNAP[MERLE etal 78], RESQ[SAUER etal 83, KUROSE etal 86], and RESQME[GORDON etal 86]. Sinclair et al.[SINCLAIR etal 86] give a full list of queueing network simulation languages. There are some high level simulation packages for specific computer systems such as SNAP/SHOT[STEWART 79].

The stochastic discrete event simulations are statistical experiments hence their outputs are random samples. The output should be processed carefully through the statistical interpretation. Repeating simulations with statistically different input sequences will produce different output estimates. Therefore sound statistical methods are essential in order to interpret the simulation results correctly. The detailed discussion for these methods can be obtained from the writings of [KLEIJEN 74], [KLEIJEN 75], [LAW etal 82], [LAVENBERG 83], [MACDOUGALL 87].

Considering statistical characteristics of simulations, there are two basic issues. First, simulation analysts should assess random sampling effects in order to assess the accuracy of simulation results. Second, simulation analysts should decide or control the length of simulation run or the number of simulation run if repetition is required. Using the confidence interval, simulation analysts can address these two issues. Through generating the confidence interval, simulation analysts can assess the random sampling effect and accuracy of the simulations. Using the generated confidence interval, they can also control the simulation run length until the output result comes into the desired confidence interval. The narrower the interval, the more confidence can be placed in the estimate.

Lavenberg et al.[LAVENBERG etal 77] and Heidelberger et al.[HEIDELBERGER etal 81] proposed algorithms to control the run length of simulation. The simulation analysts can define the desired accuracy to the algorithms and the simulation model is run according to the algorithms until the specified accuracy is obtained. If the specified accuracy is not obtained within the specified time limit, the simulation is stoped.

Most simulation studies of the queueing network models for computer systems deal with steady state characteristics rather than transient state characteristics. This study deals with the steady state characteristics. In the transient state, the performance simulation results using the performance models of this study may be inaccurate. There are many proposed procedures for generating confidence intervals for steady state characteristics. Autocorrelation and nonstationarity of simulation output sequences hinder the direct application of standard approaches based on IID(Independent and Identically Distributed) observations.

Nonstationarity is due to the model's initial conditions. The mean steady state response time is $\mu = \lim_{n \to \infty} E(X_n)$ where $X = (X_1 ...., X_n)$ is the response time output sequence generated by the simulation. The usual estimate for μ is sample average, that is, $\overline{\mu} = (\frac{1}{N}) \times \sum_{n=1}^{N} X_n$. For small N, $E(X_n) \neq \mu$ or, $E(\overline{\mu}) \neq \mu$, that is, the problem of nonstationarity or problem of initial transiency occurs. The approximately unbiased estimate of μ is a typical approach for dealing with the problem of initial transiency. It has the following 3 steps. First, determine an N_0 such that $E(X_n) \fallingdotseq \mu$ for $N \geq N_0$. Second, delete the observation before N_0. Third, estimate μ such that $\overline{\mu} = \frac{1}{(N-N_0)} \times \sum_{n=N_0+1}^{N} X_n$. Schruben[SCHRUBEN 82] shows statistical tests for stationarity which can be used to test the adequacy of an N_0. I have not gathered the simulation statistics during the 6 seconds from the starting time of the simulation, that is, the simulations results during the initial 6 seconds of simulated time were cut off and discarded in each simulation of mine. It was found that the cutting-off the initial 6 seconds of the simulated time was enough for me to get rid of the nonstationary portions in the simulations.

The problem of autocorrelation is due to the queueing. The waiting time of the next job will be more likely large when the waiting time of a job in a device is large. The central limit theorem does not support correlated observations. In the case of large sample sizes, the expression for the variance of correlated observations is $\sigma^2(\overline{\mu}) \fallingdotseq \frac{\sigma^2(X)}{(N-N_{0)}} \times \sum_{K=-\infty}^{\infty} \rho_K$ (ρ_K is the autocorrelation between X_n and X_{n+K}.) That is, the variance of a correlated sequence $\sigma^2(\overline{\mu})$ is same as the variance of an independent sequence $\frac{\sigma^2(X)}{(N-N_0)}$ times an expansion factor, $\sum_{K=-\infty}^{\infty} \rho_K$, which is the sum of the autocorrelation function or the amount of correlation in the sequence. Normally the expansion factor is positive and often much larger than one in queueing network simulations and it is essential in generating confidence intervals.

When the analysts generate confidence intervals, they can use two approaches to handle the correlation problem : to avoid the correlation and to compensate the correlation by estimation.

Three approaches are known to avoid the correlation. In the first approach, independent replications are used. It is simple but sensitive to the effect of the initial transient and can waste data if simulation analysts discard the transient part from each replication. In the second approach, the batch mean operates on a run[MECHANIC 66],[LAW etal 83]. It has the disadvantage that the selection of an adequate length of blocks(batches) is statistically difficult. In the third approach, regeneration[IGLEHART 78] is used. It is based on the fact that regenerative processes(stochastic sequences) have regeneration points which delimit the sequence into IID random length blocks. In general, computer performance evaluation processes are not regenerative. In some case they have regeneration points but it is usually not enough to generate valid confidence intervals unless the run is quite long. These lack of generality limits the usage of this method. And in some pathological cases, even if the result is acceptable, transients develop too slowly and this method fails. This study uses the same seed values for the random number generation of all the simulations so that the simulations can be regenerative as much as possible. However, when I have to repeat the same simulation, I use a seeding value different from that of the previous run for the random generator each time so that the effect by the specific seed number can be eliminated.

Heidelberger et al.[HEIDELBERGER etal 81] proposes the spectral method, a single run method for estimating the correlation in the sequence. This method has been successful for various empirical computer performance models. Heidelberger et al.[HEIDELBERGER etal 83B] study combining initial transient detection and deletion, confidence interval generation and run length control into an automatic procedure. Schruben[SCHRUBEN 82] tests procedures which combine the transient test and spectral method. Iglehart[IGLEHART 76] and Heidelberger et al.[HEIDELBERGER etal 84] propose techniques of generating confidence intervals quantiles. Law et al.[LAW etal 82] and Schruben[SCHRUBEN 81] give applications of multivariate statistical procedures which place simultaneous confidence intervals on more than one parameter.

I found that the one hour for the run length of simulated time was long enough to keep the simulation results stable in all cases and was long enough to keep the simulation results above 95% of confidence in most cases. When the repetition of simulations was required, usually 10 times of repetition was enough for me to obtain the confident simulation results. I ran the simulation programs for the same period all the time and kept the simulation environment the same all the time by setting the same options in SLAM II control statements so that the simulation results could be compared to each other with better confidence.

3.8 Summary

This chapter has described the logic and the structure of the distributed file systems of which this study evaluates the performance. This study deals with commonly used standard file systems, which means that if any file system follows the structure and the logic, then it is the target file system of this study. Detailed explanation about the latency during the computer communication and during disk I/O has been given. As I stated clearly in section 3.2.4, this study focuses on the local area network based distributed file systems.

The virtual server concept based on queueing network theory has been presented in the performance models of the distributed file systems and in performance models of the shared memory systems.

I have introduced a unique parameterization method which does not require any sophisticated performance measuring tool. The following assumptions were made in the parameterization procedure as explained in section 3.2.7. In the first stage, the file handling operation was interpreted to consume CPU time only, the CPU time consumed for the file handling operation was assumed not to vary to the data size of the read-write operation and it was assumed that the consumed CPU service time of the disk I/O operation for the read is the same as that of the write. In the second stage, the fixed portion of the result processing time both in the CPU time and the I/O time was assumed to be zero. In the fourth stage, the constant portion irrespective of the data size among the CPU service time of the communication operation was assumed to be zero and it was assumed that the service time demand of the send operation is equal to the service time demand of the receive operation and the service time demand of the send/receive operation in the client is equal to the service time demand of the send/receive operation in the file server. In the fifth stage, it was assumed that the overhead of the RPC request build in the client, the overhead of the RPC request evaluation in the file server, the overhead of the RPC response build in the file server and the overhead of the RPC response evaluation in the client are all equal. In the ninth stage, it was assumed that the I/O time of the result processing operation in the read experiment is linearly proportional to the data size and the I/O service time portion constant irrespective of the data size of the result processing was assumed to be zero. In the tenth stage, the I/O time portion of the communication overhead constant irrespective of the data size was assumed to be zero, the I/O time portion of the communication overhead proportional to the data size - the I/O time service demand of the network interface operation and that of the network operation - was assumed to be linearly proportional the data size and the I/O time service demand of the network interface unit in sending was assumed to be same as that in receiving.

Six representative and realistic workloads in three pairs have been extracted from real measured workloads through my carefully developed workload characterization procedure.

I preferred to use simulations as the primary method to solve the performance models since my models are complex and I want to have precise solutions for the models. A SLAM II simulation package has been used to solve the developed virtual server models. However, the analytic approach was sometimes used to solve part of the performance models as a supplementary method. Careful statistical analysis has been applied to the simulation results to verify the correctness of the solutions. Almost all possible performance metrics are used in this study.

Chapter 4

Measurement and Validation

The performance models and the simulation method for the models were described in chapter 3. It is required to verify that the performance models are correctly implemented into the simulation programs[GARZIA 90]. The verification was done when I found out the performance parameters in chapter 3. For the verification, I obtained the analytic solutions for the performance model such as the response time and the CPU time when there is no contention for the system resources using mathematical calculation and compared the solutions with the simulation results. I found that the solutions agree with the results exactly. Therefore, I am sure that the performance models are correctly implemented into the simulation programs.

In order to use the simulation programs with better confidence for the performance evaluation studies in the following chapters, I have to validate the simulation[GARZIA 90]. That is, I have to prove that the simulation accurately predicts the real performance or that the performance result obtained by the simulation agrees with the measured performance result with acceptable confidence. This chapter describes the measurement study for the validation and the measurement study to obtain the performance parameter values. I have already described some of the measurement study to obtain the performance parameter values in chapter 3.

I used various workloads for the validation. I measured the real performance both in the homogeneous distributed systems and the heterogeneous distributed systems.

Section 4.1 describes the methodology of the measurement used in this study. Section 4.2 shows the measured results and compares them with the simulation results respectively in two different system paradigms. In the two system paradigms, I performed two separate groups of experiments to validate the simulation results. The first group of experiments is to measure the file access performance when there is no contention for the system resources and the second group of experiments is to measure the file access performance when there exists contention for the system resources.

4.1 Measurement Methodology

How can we measure the file access performance of the system? Three methods are available. The first method is to use system utilities provided by UNIX systems. The response time, the CPU time, etc. can be collected by the standard UNIX accounting facilities. In most UNIX environments, some performance measurement tools are provided to measure the utilization of the CPU and the disk and the data transfer rate(i.e., number of packets) per second via network as standard utilities. The second method is to use commercially available UNIX performance measurement tools. The third method is develop and implement ones own performance measurement tools or modify the UNIX kernel system.

For the easy reproduction in other environments of what are obtained in this study or to enable me or others to apply easily what is studied in this thesis to other UNIX environments, this study used only the system provided performance tools. They are standard SUN UNIX accounting facilities and standard SUN UNIX performance measurement tools such as "perfmeter", "gettimeofday", "ping", "spray", etc.. So anyone who intends to reproduce what I obtain in this study and apply the study to any other UNIX environment does not have to buy any special performance measurement tool, or does not have to develop any performance measurement tool and implement them into the system or modify the UNIX system at all. The measurement methodology has generality and is easy and simple to use, nevertheless it produces accurate measured values.

All measurement experiments were peformed in dedicated and closed environments. Therefore, no other uninvited users were allowed to use any system component such as the clients, the server and the network in the distributed file systems and in the shared memory systems during the experiments. All measurement experiments were performed according to the predefined scenarios. The predefined scenarios consist of shell scripts. Each predefined scenario was submitted in series in several second interval according to the global clock time and finally after less than 3 minutes, all scenarios ran in each participating client of the distributed file system at the same time. I cut off the measured data during the first 5 minutes or sometime up to 10 minutes to get rid of the performance data during the transient period.

I tried to avoid the caching as much as I could during the experiments of the normal write(read) where no caching was assumed to occur. For example, I scattered the data evenly throughout the disk and whenever I performed the write(read) experiments to measure the file access performance when there was no contention for the system resources, just before the write(read) operation I read a file with 5Mbytes or 10Mbytes meaningless data which was much larger than the size of the system provided cache so that the content of the cache was refreshed with the content of the large file and the cache hit could not occur. However, I still found some caching during writes and much caching during reads. The reason seems to be that I used the same home directory for the data in most cases. The kind of the cache hit which I observed during the measurement was less likely the case that cached data were being used more than once but more likely to be the case that the cache data were being used just once. That means it is a kind of read-ahead and write-back caching since the same data were never accessed in series in the experiments. The details of caching will be discussed in section 7.1.

I measured the starting time, the ending time, the response time and the CPU time both in the distributed file systems and in the shared memory systems using the standard Sun UNIX accounting facilities. I also measured the utilization of the CPU and the disk and the load index in the shared memory systems and the utilization of the CPU and the disk and the load index of the file server and the data transfer rate(number of packets per second) of the network in the distributed file systems, using the standard Sun Perfmeter utilities.

In order to obtain the performance parameter values, I measured the file access performance when there was no contention for system resources both in file servers and clients in the distributed file systems and when there was no contention for system resources in the shared memory systems. In these cases, the inter-arrival time of the request should be larger than the processing time of the request, that is, the response time of the request. I call this the standalone measurement in this study.

In the standalone measurement to obtain the performance parameter values in chapter 3, I performed various performance measurement experiments using the system provided commands such as "cat", "mkdir", "ls","rmdir", ping", "spray", etc.. Let's look at the performance measurement experiments using the "cat" command.

In the measurement of the local write using "cat local_file_1 > local_file_2" command for the shared memory systems, I read a file (local_file_1) in the local disk and as a pipelined operation, wrote the read data into a file(local_file_2) in the local disk at a location different from the location of the read file. The experiment was performed in three classes of Sun workstations such as the Sun

3/60 workstation, the Sun SPARCstation 470 workstation and the Sun SPARCstation 10/30 workstation individually.

In the measurement of the local read using "cat local_file1" command for the shared memory systems, I read a file(local_file_1) in the local disk and displayed the read data on the window screen. The experiment was individually performed in the three classes of Sun workstations such as the Sun 3/60 workstation where the Sun window system, that is, "sunview", was used, the Sun SPARCstation 470 workstation where the X window system, that is, "twm" was used and the Sun SPARCstation 10/30 workstation where both the X window system and the Sun window system were used.

In the measurement of the remote read using "cat remote_file_1" command for the distributed file systems, I read a file(remote_file_1) in the remote disk of the file server and displayed the read data on the window screen of the client which issued the command. The experiment was individually performed between two Sun SPARCstation 10/30 workstations where the X window system and the Sun window system were used and between two Sun SPARCstation 470 workstations where the X window system and the Sun window system were used. The remote read experiments in the heterogeneous distributed file systems were also performed between a Sun SPARCstation 10/30 workstation where the X window system and the Sun window system were used and a Sun 3/60 workstation where the Sun window system was used.

Three different types of remote write experiments were performed in the distributed file systems. In the first type of remote write experiment using "cat remote_file_1 > local_file_1", I read a file(remote_file_1) in the remote disk of the file server and as a pipelined operation wrote the read data into a file(local_file_1) in the local disk of the client. In the second type of remote write experiment using "cat remote_file_1 > remote_file_2", I read a file(remote_file_1) in the remote disk of the file server and as a pipelined operation wrote the read data into a file(remote_file_2) in the remote disk at a location different from the location of the read file. In the third type of remote write experiment using "cat local_file_1 > remote_file_1" command, I read a file(local_file_1) in the local disk of the client and as a pipelined operation wrote the read data into a file(remote_file_1) in the remote disk of the file server. All the three types of experiments were performed in the three different distributed file systems i.e. in the distributed file system which consisted of the Sun SPARCstation 10/30 workstations, in the distributed file system which consisted of the Sun SPARCstation 470 workstations and in the heterogeneous distributed file system which consisted of the Sun SPARCstation 10/30 workstation and the Sun 3/60 workstation.

To match the real environments, the input request arrival rate was varied to reflect the input arrival rate from, for example, 9, 15, ..., 57 clients concurrently using the distributed file systems respectively and to reflect the input request arrival rate from, for example, 9, 15, ..., 57 local users concurrently using the shared memory systems respectively in each experiment. I call these experiments the real world measurement or the live measurement in this study.

The number of the actually participating workstations as the number of clients was varied to be 2, 3, 4, 5, 6, 7, 8 and 9 respectively in the distributed file systems. Two different scenarios were used in the real world measurement of the distributed file systems. In the first scenario, the shell script residing in the window of each client workstation sends each request sequentially, waits until the sent request is completed and as soon as the sent request is completed it sends the next request. Therefore, the actual input arrival rate completely depends on the throughput of the distributed file system. I put up to two or three scenarios or shell scripts in the two or three windows of each client workstation[1] and the

[1] Maximum two or three since we want to ensure the client has no contention for the system resources and therefore the requests in the client have no queueing delay.

number of the participating client workstations was varied.

```
#include <stdio.h>
void main(ac,av)
int ac;
char *av[];
    {
    FILE *fp;
    char *dat=NULL;
    int size;
    int i,j;
    if (ac<3) {
        printf ("writeA [size] [target_filename]\n");
        exit();
        }
    size=atoi(av[1]);
    dat=(char *)malloc(size+1);
    if (!dat) {
        printf ("malloc failure\n");
        exit(1);
        }
    dat[size]=NULL;
    fp=fopen(av[2],"w");
    fwrite(&dat[0],size,1,fp);
    fflush(fp);
    fclose(fp);
    }
```

Figure 4.1.1 : The write program A

In this case, the maximum number of the concurrently arriving input requests is the same as the number of the participating clients multiplied by the number of the shell scripts(windows).

```
#include <stdio.h>
void main(ac,av)
int ac;
char *av[];
    {
    FILE *fp;
    char *dat=NULL;
    int size;
    int i,j;
    if (ac<3) {
        printf ("writeB [size] [target_filename]\n");
        exit();
        }
    size=atoi(av[1]);
    dat=(char *)malloc(size + 1);
    if (!dat) {
        printf ("malloc failure\n");
        exit(1);
        }
    for (i=0;i<size; i++) dat[i]='w';
    dat[size]=NULL;
    fp=fopen(av[2],"w");
    fwrite(&dat[0],size,1,fp);
    fflush(fp);
    fclose(fp);
    }
```

Figure 4.1.2 : The write program B

In the second scenario, the shell script residing in the window of each client workstation sends multiple requests at the same time, and after an instructed time interval, it sends further multiple requests at the same time regardless of the status of the previously sent requests. The shell script repeats the above steps until it is either externally or internally instructed to stop doing it. I put up to two or three shell scripts in each client workstation and the number of the participating client workstations was varied. In this case, the maximum number of the concurrently arriving input requests is same as the number of the participating client workstations multiplied by the number of concurrently submitted requests and the number of shell scripts(windows).

```
#include <stdio.h>
void main(ac,av)
int ac;
char *av[];
    {
    FILE *fp;
    char dat='c';
    int i,j;
    if (ac<3) {
       printf ("writeC [size] [target_filename]\n");
       exit();
       }
    fp=fopen(av[2],"w");
    for (j=0; j<atoi(av[1]); j++) fwrite(&dat,1,1,fp);
    fflush(fp);
    fclose(fp);
    }
```

Figure 4.1.3 : The write program C

In the real world measurement, I used my own read program and write programs. Three kinds of write programs were tested. In the write program A of figure 4.1.1, the content in the memory is written into the disk. In the write program B of figure 4.1.2, first, the memory is written with the character "W" and then the memory content is copied into the disk.

```
#include <stdio.h>
void main(ac,av)
int ac;
char *av[];
        {
        char *data=NULL;
        int size;
        FILE *fp=NULL;
        if (ac<3) {
            printf ("read [size] [source_filename]\n");
            exit();
            }
        size=atoi(av[1]);
        data=(char *)malloc(size+1);
        if (!data) {
            perror ("malloc failure");
            exit();
            }
        fp=fopen(av[2],"r+");
        if (fread(data,size,1,fp)==0) perror("read failure");
        *(data+size)=NULL;
        fclose(fp);
        free(data);
        }
```

Figure 4.1.4 : The read program

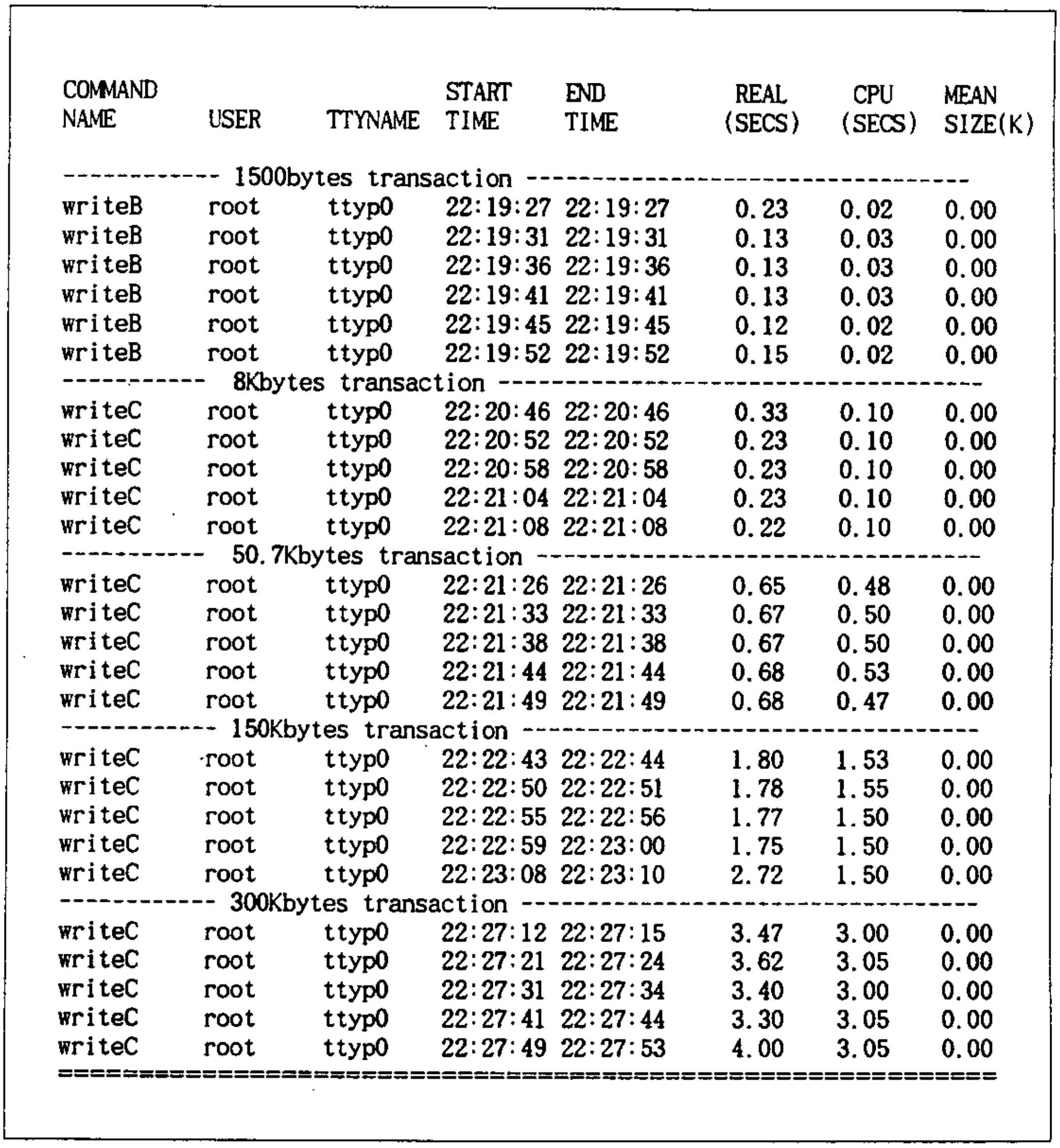
In the write program C of figure 4.1.3, first, the character "c" is directly written into the disk one character by one character. I chose the write program A as the write program. Figure 4.1.4 shows the read program which I used.

```
COMMAND                        START    END              REAL     CPU      MEAN
NAME       USER     TTYNAME    TIME     TIME             (SECS)   (SECS)   SIZE(K)

----------- 1500bytes transaction -----------------------------------
writeC     root     ttyp0      00:20:48 00:20:48         0.15     0.02     0.00
writeC     root     ttyp0      00:20:55 00:20:55         0.12     0.03     0.00
writeC     root     ttyp0      00:20:59 00:20:59         0.12     0.03     0.00
writeC     root     ttyp0      00:21:04 00:21:04         0.12     0.02     0.00
writeC     root     ttyp0      00:21:11 00:21:11         0.13     0.02     0.00
-----------  8Kbytes transaction   ----------------------------------
writeC     root     ttyp0      00:19:23 00:19:23         0.18     0.08     0.00
writeC     root     ttyp0      00:19:32 00:19:32         0.17     0.07     0.00
writeC     root     ttyp0      00:19:40 00:19:40         0.18     0.05     0.00
writeC     root     ttyp0      00:19:45 00:19:45         0.17     0.07     0.00
writeC     root     ttyp0      00:19:50 00:19:50         0.13     0.05     0.00
------------  50.7Kbytes transaction ---------------------------------
writeC     root     ttyp0      00:21:27 00:21:27         0.43     0.30     0.00
writeC     root     ttyp0      00:21:33 00:21:33         0.47     0.27     0.00
writeC     root     ttyp0      00:21:39 00:21:39         0.47     0.28     0.00
writeC     root     ttyp0      00:21:49 00:21:49         0.45     0.28     0.00
writeC     root     ttyp0      00:21:53 00:21:53         0.43     0.30     0.00
------------- 150Kbytes transaction ---------------------------------
writeC     root     ttyp0      00:22:07 00:22:08         1.18     0.80     0.00
writeC     root     ttyp0      00:22:13 00:22:14         1.25     0.82     0.00
writeC     root     ttyp0      00:22:18 00:22:20         2.07     0.83     0.00
writeC     root     ttyp0      00:22:24 00:22:25         1.22     0.85     0.00
writeC     root     ttyp0      00:22:28 00:22:29         1.15     0.82     0.00
---------  300Kbytes transaction ------------------------------------
writeC     root     ttyp0      00:22:41 00:22:43         2.62     1.65     0.00
writeC     root     ttyp0      00:22:48 00:22:50         2.62     1.67     0.00
writeC     root     ttyp0      00:22:54 00:22:56         2.30     1.60     0.00
writeC     root     ttyp0      00:23:02 00:23:04         2.48     1.65     0.00
writeC     root     ttyp0      00:23:08 00:23:10         2.77     1.67     0.00
=====================================================================
```

Table 4.1.1 : Measured response time and CPU time of the remote write program C when there is no contention for the system resources in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Table 4.1.1 and table 4.1.2 show some of the response times and CPU times of the write program C in the distributed file system which consists of the Sun SPARCstation 10 workstations and in the distributed file system which consists of the Sun SPARCstation 470 workstations.



```
COMMAND                        START    END          REAL     CPU      MEAN
NAME       USER     TTYNAME    TIME     TIME         (SECS)   (SECS)   SIZE(K)

------------ 1500bytes transaction ---------------------------------
writeB     root     ttyp0      22:19:27 22:19:27     0.23     0.02     0.00
writeB     root     ttyp0      22:19:31 22:19:31     0.13     0.03     0.00
writeB     root     ttyp0      22:19:36 22:19:36     0.13     0.03     0.00
writeB     root     ttyp0      22:19:41 22:19:41     0.13     0.03     0.00
writeB     root     ttyp0      22:19:45 22:19:45     0.12     0.02     0.00
writeB     root     ttyp0      22:19:52 22:19:52     0.15     0.02     0.00
----------- 8Kbytes transaction ------------------------------------
writeC     root     ttyp0      22:20:46 22:20:46     0.33     0.10     0.00
writeC     root     ttyp0      22:20:52 22:20:52     0.23     0.10     0.00
writeC     root     ttyp0      22:20:58 22:20:58     0.23     0.10     0.00
writeC     root     ttyp0      22:21:04 22:21:04     0.23     0.10     0.00
writeC     root     ttyp0      22:21:08 22:21:08     0.22     0.10     0.00
----------- 50.7Kbytes transaction ---------------------------------
writeC     root     ttyp0      22:21:26 22:21:26     0.65     0.48     0.00
writeC     root     ttyp0      22:21:33 22:21:33     0.67     0.50     0.00
writeC     root     ttyp0      22:21:38 22:21:38     0.67     0.50     0.00
writeC     root     ttyp0      22:21:44 22:21:44     0.68     0.53     0.00
writeC     root     ttyp0      22:21:49 22:21:49     0.68     0.47     0.00
------------ 150Kbytes transaction ---------------------------------
writeC     root     ttyp0      22:22:43 22:22:44     1.80     1.53     0.00
writeC     root     ttyp0      22:22:50 22:22:51     1.78     1.55     0.00
writeC     root     ttyp0      22:22:55 22:22:56     1.77     1.50     0.00
writeC     root     ttyp0      22:22:59 22:23:00     1.75     1.50     0.00
writeC     root     ttyp0      22:23:08 22:23:10     2.72     1.50     0.00
------------ 300Kbytes transaction ---------------------------------
writeC     root     ttyp0      22:27:12 22:27:15     3.47     3.00     0.00
writeC     root     ttyp0      22:27:21 22:27:24     3.62     3.05     0.00
writeC     root     ttyp0      22:27:31 22:27:34     3.40     3.00     0.00
writeC     root     ttyp0      22:27:41 22:27:44     3.30     3.05     0.00
writeC     root     ttyp0      22:27:49 22:27:53     4.00     3.05     0.00
===========================================================================
```

Table 4.1.2 : Measured response time and CPU time of the remote write program C when there is no contention for the system resources in the distributed file system which consists of the Sun SPARCstation 470 workstations.

For the measurement experiments, I have used the five workstations in table 3.2.7.A, a Sun SPARCstation 470 workstation which is same as the king470 in the table, two workstations equivalent to the Sun SPARCstation 1 workstation[2] and a Sorborne workstation[3]. In all experiments, the constant distribution is used for the transaction size.

4.2 Measurement and Validation

In obtaining the performance parameter values, I used moderate measurement values as the representative values for the response time and the CPU time, which means I used in most cases the most frequently observed values or sometimes the average values as the representative values. The distributions of the measured CPU times do not have large standard deviations so that it was not very difficult for me to select the representative values for the parameterization. But the distributions of the measured I/O times have large standard deviations so that it was very difficult to select the representative values for the parameterization. Especially, the disk I/O times show large standard deviations since the disk arm movement and variable latency account for large portions of the disk I/O times and depend on the relative location of each file.

In validating the performance models and the simulation results, I first check whether a simulation value falls into the range of the measured values, that is, it is at least one of the measured values. If it falls into the range, then I evaluate whether the accuracy of the simulation value is acceptable. Further I define that the accuracy of a simulation value is 100% confident if the simulation value is similar to the most frequently observed value, that is, the mode, among the measured values or to the mean of the measured values. In the tables of the

[2] 12.5MIPS(20MHz) or 15.8MIPS(25MHz), 32Mbytes main memory and Panther 1.2Gbytes SCSI drive : 1.3msec average seek time, 8.33msec average latency, 3(5)Mbytes/sec asynchronous(synchronous) SCSI bus transfer rate, 17.4-29.7Mbits/sec disk transfer rate.

[3] 32Mbytes main memory and 670Mbytes disk.

following two sections, I specify the mode value, the most frequently observed value among the measured values if the frequency of the mode value is found to be more than 20% of the total occurrences. Otherwise, I leave it blank. Each read or write file access produces a line of account information. Therefore the total frequency is simply obtained by counting the total number of the lines and the frequency of the mode value is obtained by counting the total number of occurrences of the mode value. Any frequently observed value in the accounting record is a candidate for the mode and is tested to see if the frequency of the mode value is found to be more than 20% of the total occurrences. As shown in the table 4.1.1 and table 4.1.2, the accounting record show the measured response time to the level of 1/100 second.

In each experiment, I used both the write program of figure 4.1.1 and the read program of figure 4.1.4. I did not find any considerable difference between the response time of the read program and that of the write program when there was no contention for the system resources but I found the response time of the read program became smaller than that of the write program as the number of the clients in the three distributed file systems and the number of the local users in the three local systems increased. I experienced much more cache hits in the read experiments than in the write experiments even though I tried to prevent the cache hits occur. This study deals with the measurement results of the write experiments in the following sections of this chapter unless the read experiments are explicitly specified to be dealt with.

4.2.1 The Shared Memory System

In the standalone measurement to obtain the performance parameter values for the shared memory systems in section 3.4.2, I performed various performance measurement experiments using the system provided commands such as "cat", "mkdir", "ls","rmdir", ping", "spray", etc.. In this section I include some of the validation work of the performance measurement experiments using "cat" commands.

Table 4.2.1.1, table 4.2.1.2 and table 4.2.1.3 compare the response time and the CPU time obtained from the virtual performance models when there is no contention for the system resources with the measured response time(system time) and CPU time in the standalone experiment of the local write using "cat local_file_1 > local_file_2" command in the shared memory systems in which I read a file in the local disk and as a pipelined operation, write the read data into a file in the local disk at a location different from the location of the read file respectively in the Sun 3/60 workstation, the Sun SPARCstation 470 workstation and the Sun SPARCstation 10/30 workstation.

Work-	Measurement (msec)						Simulation(msec)	
load	System Time			CPU time			System	CPU
(kbytes)	Min.	Max.	Mode	Min.	Max.	Mode	time	time
1.5	70	230	110	20	30	30	112.3	30.25
15	100	270		20	30	30	135	32.5
150	170	470		30	80	50	360	55
300	220	800	600	70	100	80	610	80

Table 4.2.1.1 : The measured CPU time and the response time in the standalone local write experiment vs. the CPU time and the response time obtained from the simulation when there is no contention for the system resources in the Sun SPARCstation 10/30 workstation.

Work-	Measurement (msec)						Simulation(msec)	
load	System Time			CPU time			System	CPU
(kbytes)	Min.	Max.	Mode	Min.	Max.	Mode	time	time
1.5	120	230	160	30	50	40	163.4	40.3
15	130	230		30	80		194	43
150	180	530		50	100		500	70
300	450	1230	880	70	120	100	840	100

Table 4.2.1.2 : The measured CPU time and the response time in the standalone local write experiment vs. the CPU time and the response time obtained from the simulation when there is no contention for the system resources in the Sun SPARCstation 470 workstation.

Work-load (kbytes)	Measurement (msec)						Simulation(msec)	
	System Time			CPU time			System time	CPU time
	Min.	Max.	Mode	Min.	Max.	Mode.		
1.5	150	530	380	100	150	120	383.04	120.8
15	150	570		100	170		464.4	128
150	570	1530		170	280		1078	200
300	2120	3130	2200	270	370	280	1902	280

Table 4.2.1.3 : The measured CPU time and the response time in the standalone local write experiment vs. the CPU time and the response time obtained from the simulation when there is no contention for the system resources in the Sun 3/60 workstation.

It was observed that both the CPU time and the response time obtained from the simulation when there is no contention for the system resources well agree to the measured CPU time and response time in the standalone local write experiment in the three different systems.

Table 4.2.1.4, table 4.2.1.5 and table 4.2.1.6 compare the response time and the CPU time obtained from the virtual performance models when there is no contention for the system resources with the measured response time and CPU time in the standalone measurement of the local read using "cat local_read_1" command in the shared memory systems in which I read a file in the local disk and display the read data on the window screen respectively in the Sun SPARCstation 10/30 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 workstation.

Work-load (kbytes)	Measurement (msec)						Simulation(msec)	
	System Time			CPU time			System time	CPU time
	Min.	Max.	Mode	Min.	Max.	Mode		
1.5	70	120	100	20	30	20	88.5	25.375
15	70	420		20	50		271.25	28.75
150	1000	2870	2770	20	80		2352.5	62.5
300	1600	5830	4680	50	130	100	4765	100

Table 4.2.1.4 : The measured CPU time and the response time in the standalone local read experiment vs. the CPU time and the response time obtained from the simulation when there is no contention for the system resources in the Sun SPARCstation 10/30 workstation.

Work-load (kbytes)	Measurement (msec)						Simulation(msec)	
	System Time			CPU time			System time	CPU time
	Min.	Max.	Mode	Min.	Max.	Mode		
1.5	50	380	150	20	50	30	161.55	30.45
15	100	1350	1120	30	70		1110	34.5
150	9620	11950	10770	50	120		10215	75
300	20270	22530	21000	70	180	120	20490	120

Table 4.2.1.5 : The measured CPU time and the response time in the standalone local read experiment vs. the CPU time and the response time obtained from the simulation when there is no contention for the system resources in the Sun SPARCstation 470 workstation.

Work-load (kbytes)	Measurement (msec)						Simulation(msec)	
	System Time			CPU time			System time	CPU time
	Min.	Max.	Mode	Min.	Max.	Mode		
1.5	120	6080	450	70	130	100	751.9	100.75
15	1720	31080		100	130	100	5368.2	107.5
150	50770	308070		120	250		52539	175
300	95720	618530	105000	170	280	200	104951	250

Table 4.2.1.6 : The measured CPU time and the response time in the standalone local read experiment vs the CPU time and the response time obtained from the simulation when there is no contention for the system resources in the Sun 3/60 workstation.

It was observed that the CPU time and the response time obtained from the simulation when there is no contention for the system resources agrees well with the measured CPU time and response time in the standalone local read experiments in the three different systems.

As explained in the previous section, three different real world measurement experiments are possible for the shared memory systems. The first is to generate the transactions from one local user using multiple shell scripts. The second is that multiple local users generate the transactions independently and each local user uses one shell script. The third is that multiple local users generate the transactions independently and each local user uses multiple shell scripts. I performed the three experiments with two different shell scripts, giving a total of six different experiments. The first shell script submits transactions sequentially one after the previous one finishes and the second shell script submits multiple transactions at the same time after a specified time-interval repeatedly. In general the third experiment showed the worst response time and the first experiment showed the best response time. In general the response times of the second experiment showed best fitting to the response time of the simulation when I used the workload of which the workload size is constant and the input arrival distribution is the Poisson distribution.

Zero values in the number of the local users mean that the input arrival rate drops to a level where the total number of the arriving transactions is only one during the measurement period. Therefore there exists no contention for the system resources and no queueing delay.

# of local users	Response time (msec)								
	Simulation						Measurement		
	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	55.67	55.67	55.67	55.67	55.67	55.67	30	70	50/70
20	60	58.41	55.67	59.96	56.04	61.11			
40	68	62.24	55.67	65.96	57.63	71.76			
60	78	67.65	55.57	75	62	86.54			
80	96.35	78	55.67	92.82	69.14	113.6	70	130	70
95	124	90	55.67	112	77	151	70	180	70/80
100	138.4	94.37	55.67	121.3	81.01	168.8	70	230	

Table 4.2.1.7 : The response times of the 6 patterns of the 8Kbyte workload obtained from the simulation vs. the response times of the 8Kbytes workload whose size is constant obtained from the real world measurement in the Sun SPARCstation 10 workstation. "a" means the input transaction arrival distribution, "t" means the input transaction size distribution, "n" means the log-normal distribution, "p" means the Poisson distribution and "f" mean the constant distribution(fixed values). "min." means the minimum value and "max." means the maximum value.

# of	Response time (msec)								
local	Simulation						Measurement		
users	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	91.25	91.25	91.25	91.25	91.25	91.25	70	180	70/80
20	113.1	105.9	91.25	125.2	101.1	144	80	230	120
40	200.3	144.7	91.25	210.8	145	301.2	80	350	170
60	1011	439.6	91.25	895.2	526.8	1575	80	1080	430

Table 4.2.1.8 : The response times of the 6 patterns of the 50.7Kbyte workload obtained from the simulation vs. the response times of the 50.7Kbytes workload whose size is constant obtained from the real world measurement in the Sun SPARCstation 10 workstation.

Table 4.2.1.7 compares the response times of the 6 different patterns of the 8Kbytes workload obtained from the simulation with the response times of the 8Kbyte workload, whose transaction size is fixed, obtained from the real world measurement using the write program A of figure 4.1.1 in the Sun SPARCstation 10 workstation. Table 4.2.1.8 compares the response times of the 50.7Kbytes workload and table 4.2.1.9 compares the response times of the 150Kbytes workload.

# of	Response time (msec)								
local	Simulation						Measurement		
users	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	174	174	174	174	174	174	220	420	230
10	235.9	212.2	174	349	262.9	449.2			
15	334.2	254.5	174	538.9	378.8	727.6	220	1570	
20	534.8	343.6	174	971.3	743	1328	220	2070	
25	1247	656.3	174	2514	1695	3680	220	3120	

Table 4.2.1.9 : The response times of the six patterns of the 150Kbyte workload obtained from the simulation vs. the response times of the 150Kbytes workload whose size is constant obtained from the real world measurement in the Sun SPARCstation 10 workstation.

# of	Response time (msec)								
local	Simulation						Measurement		
users	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	99.07	99.07	99.07	99.07	99.07	99.07	50	120	70
5	104	102	99.07	103	100	105			
10	109	106	99.07	107	100	113			
15	116	110	99.07	112	100	121			
20	124.1	115.8	99.07	118.8	100.5	131.7			
25	137	124	99.07	128	105	147			
30	157	135	99.07	140	110	171			
35	187	148	99.07	155	119	208			
40	228.2	162.4	99.07	172.5	130	254.9	50	420	
50	330	200	99.07	220	160	365	50	1800	
60	1819	772.4	99.07	853.4	1819	2555			

Table 4.2.1.10 : The response times of the six patterns of the 8Kbyte workload obtained from the simulation vs. the response times of the 8Kbytes workload whose size is constant obtained from the real world measurement in the Sun SPARCstation 470 workstation.

Table 4.2.1.10 compares the response times of the 8Kbytes workload in the Sun SPARCstation 470 workstation. Table 4.2.1.11 compares the response times of the 50.7Kbytes workload and table 4.2.1.12 compares the response times of the 150Kbytes workload.

# of	Response time (msec)								
local	Simulation						Measurement		
users	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	147.5	147.5	147.5	147.5	147.5	147.5	100	220	120
5	156.4	155	147.5	160.8	150	170			
10	178.1	169.2	147.5	183.1	152.4	209.1			
15	218.2	189	147.5	216.2	162	270			
20	286.9	218.5	147.5	268.3	184.4	366.4			
25	416	275	147.5	365	230	530	120	770	280
30	741.7	405.1	147.5	573	325.9	1141	120	1620	420

Table 4.2.1.11 : The response times of the six patterns of the 50.7Kbyte workload obtained from the simulation vs. the response times of the 50.7Kbytes workload whose size is constant obtained from the real world measurement in the Sun SPARCstation 470 workstation.

# of	Response time (msec)								
local	Simulation						Measurement		
users	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	260	260	260	260	260	260	220	970	260
5	317.9	301	260	403.4	380	508.6			
10	518.5	391	260	712.9	526.4	1014	300	2670	
15	1323	716.9	260	1926	1149	3145	500	4770	1000

Table 4.2.1.12 : The response times of the six patterns of the 150Kbyte workload obtained from the simulation vs. the response times of the 150Kbytes workload whose size is constant obtained from the real world measurement in the Sun SPARCstation 470 workstation.

If the transaction size is large e.g. 150Kbytes, some relatively very large response times were found in the measured response times. I found that in general, as the transaction size increases, the confidence of the simulated response time decreases. In most cases, the average response times of the six workload patterns obtained from the simulations falls within the range of the measured response times.

The measured utilization of the CPU is found to be larger than the simulated utilization of the CPU in most cases. The simulation results for the shared memory system are found to have good confidence in general.

4.2.2 The Distributed File System

In the standalone measurement to obtain the performance parameter values of the distributed file systems explained in section 3.2.7, I performed various performance measurement experiments using the system provided commands such as "cat", "mkdir", "ls","rmdir", ping", "spray", etc.. In this section I include some of the validation work of the performance measurement experiments using the "cat" command.

Table 4.2.2.1, table 4.2.2.2, table 4.2.2.3 and table 4.2.2.4 compare the response time and CPU time obtained from the virtual performance models when there is no contention for the system resources with the measured response time and the measured CPU time in the standalone experiment of the remote write using "cat remote_file_1 > local_file_1" in the distributed file systems, which reads a file in the remote disk of the file server and as a pipelined operation writes the read data into a file in the local disk of the client respectively in the distributed file system which consists of the Sun SPARCstation 10/30 workstations, in the distributed file system which consists of the Sun SPARCstation 470 workstations, in the heterogeneous distributed file system which consists of the file server of the Sun 3/60 workstation and the clients of the Sun SPARCstation 10/30 workstations and in the heterogeneous distributed file system which consists of the file server of the Sun SPARCstation 10/30 workstation and the clients of the Sun 3/60 workstations.

Work-load (kbytes)	Measurement(msec)						Simulation(msec)	
	System Time			CPU time			System time	CPU time
	Min.	Max.	Mode	Min.	Max.	Mode		
1.5	100	170		20	50	40	125.95	35.7
15	130	220		30	80		146.5	39.98
150	320	4480		50	100		709	82.73
300	530	5020	1400	100	170	130	1334	130.23

Table 4.2.2.1 : The measured CPU times and the measured response times in the standalone remote write experiment vs. the CPU times and the response times obtained by the simulation when there is no contention for the system resources in the distributed file system which consists of the Sun SPARCstation 10/30 workstations.

Work-load	Measurement(msec)						Simulation(msec)	
	System Time			CPU time			System	CPU
(kbytes)	Min.	Max.	Mode	Min.	Max.	Mode	time	time
1.5	70	280		30	70	50	103.9	50.8
15	70	370		30	70		316.75	55.75
150	320	2680		70	120		1545.25	105.25
300	450	4720		70	160	160	2910.25	160.25

Table 4.2.2.2 : The measured CPU times and the measured response times in the standalone remote write experiment vs. the CPU times and the response times obtained from the simulation when there is no contention for the system resources in the distributed file system which consists of the Sun SPARCstation 470 workstations.

In table 4.2.2.1, and table 4.2.2.2, we see that both the CPU times and the response times obtained from the simulation when there is no contention for the system resources agree well with the measured CPU times and the measured response times in the standalone remote write experiment in the distributed file system which consists of the Sun SPARCstation 10/30 workstations and the distributed file system which consists of the Sun SPARCstation 470 workstations.

When the standalone remote write experiment is performed in the heterogeneous distributed file system which consists of the Sun 3/60 workstation and the Sun SPARCstation 10/30 workstations, it is found that some CPU times obtained by the simulation are somewhat larger than the range of the measured CPU times and some response times obtained by the simulation are larger than the range of the measured response times. As an explanation, it should be remembered that some parameter values in the sending systems are assumed to be same as those in the receiving systems. By removing this assumption, that is, obtaining each parameter value separately, the simulation values are expected to be within the range of the measured values.

Table 4.2.2.3, table 5.2.2.4, table 5.2.2.5 and table 5.2.2.6 compare the response times and the CPU times obtained from the virtual performance models when there is no contention for the system resources with the measured response times and the measured CPU times in the standalone experiment of the remote read using "cat remote_file_1" command, which reads a file in the remote disk of the file server and displays the read data on the window screen of the client respectively in the distributed file system which consists of the Sun SPARCstation 10/30 workstations, the distributed file system which consists of the Sun SPARCstation 470 workstations, the heterogeneous distributed file system which consists of the file server of the Sun 3/60 workstation and the clients of the Sun SPARCstation 10 workstations and in the heterogeneous distributed file system which consists of the file server of the Sun SPARCstation 10 workstation and the clients of the Sun 3/60 workstations.

Work-load (kbytes)	Measurement(msec)						Simulation(msec)	
	System Time			CPU time			System time	CPU time
	Min.	Max.	Mode	Min.	Max.	Mode		
1.5	70	820	80	20	50	30	81.5	30.825
15	100	830		30	50		384.471	36.225
150	720	2830	2730	70	180		2755.725	90.225
300	1530	5870	5700	70	180	150	5619.225	155

Table 4.2.2.3 : The measured CPU times and the measured response times in the standalone remote read experiment vs. the CPU times and the response times obtained from the simulation when there is no contention for the system resources in the distributed file system which consists of the Sun SPARCstation 10/30 workstations.

Work-load (kbytes)	Measurement(msec)						Simulation(msec)	
	System Time			CPU time			System time	CPU time
	Min.	Max.	Mode	Min.	Max.	Mode		
1.5	70	380		20	70	40	222.5	40.95
15	1110	2180	1170	30	80		1232.75	47.25
150	10250	12280		50	120		11325.25	110.25
300	21030	23830	22000	70	180	180	22560.25	180.25

Table 4.2.2.4 : The measured CPU times and the measured response times in the standalone remote read experiment vs. the CPU times and the response times obtained from the simulation when there is no contention for the system resources in the distributed file system which consists of the Sun SPARCstation 470 workstations.

In table 4.2.2.3 and table 4.2.2.4, it is observed that both the CPU times and the response times obtained from the simulations when there is no contention for the system resources agree well with the measured CPU times and the measured response times in the standalone remote read experiment in the distributed file system which consists of the Sun SPARCstation 10/30 workstations and in the distributed file system which consists of the Sun SPARCstation 470 workstations.

Some response times obtained from the simulation are larger than the response time measured in the standalone remote read experiment in the heterogeneous distributed file system which consists of the Sun SPARCstation 10/30 workstation and the Sun 3/60 workstations. As an explanation, it should be remembered that some parameter values in the sending systems are assumed to be same as those in the receiving systems.

Large queueing delay occurs during the screen display I/O operation in the distributed file systems as well as in the shared memory systems. But no queue is represented for the clients in the simulations because it is assumed that there exists no contention for the system resources in the clients. However the queueing delay can be reflected by directly varying the parameter value instead of representing the queues in the clients of the performance models during the simulations.

As explained in the previous section, three different real world measurement experiments are possible for the distributed file systems. The first is that the multiple shell scripts in one client workstation generate the transactions independently. The second is that multiple client workstations generate the transactions independently and each client workstation uses one shell script. The third is that multiple client workstations generate the transactions independently and each client workstation uses multiple shell scripts. The three experiments were performed with two different shell scripts, therefore total six different experiments were performed. The first shell script submits transactions sequentially one after the previous one finishes and the second shell script submits multiple transactions at the same time after a specified time-interval repeatedly.

The third experiment with the second shell script showed the worst response time and the first experiment with the first shell script showed the best response time. The response times of the second experiment showed best fitting to the response times of the simulation when this study used the workload of which the size is constant and the input arrival distribution is the Poisson distribution.

Zero values in the number of the clients means that the input arrival rate drops to a level where the total number of the arriving transactions is one during the measurement period. Therefore there exists no contention for the system resources and no queueing delay.

# of clients	Response time (msec)								
	Simulation						Measurement		
	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	73.33	73.33	73.33	73.33	73.33	73.33	60	130	80
20	76.89	77.19	73.33	86.42	78.7	92.4			
33	83	79	73.33	99	87.5	110	60	130	80
40	85.25	82.33	73.33	107.8	94	120.6			
57	96	88	73.33	140	120	146	80	270	
60	98.7	89.29	73.33	146.8	126.8	152.8			
73	114	96	73.33	179	153	190	80	350	170
80	122.7	100.5	73.33	198.8	167.3	219.2			
100	170.8	121.6	73.33	287.3	244	334.6			

Table 4.2.2.5 : The response times of the six patterns of the 8Kbyte workload obtained from the simulation vs. the response times of the 8Kbytes workload whose size is constant obtained from the real world measurement in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Table 4.2.2.5 compares the response times of the six different patterns of the 8Kbytes workload obtained from the simulation with the response times of the 8Kbyte workload whose transaction size is fixed obtained from the real world measurement using the write program A of figure 4.1.1 in the distributed file system which consists of the Sun SPARCstation 10 workstations. Table 4.2.2.6 compares the response times of the 50.7Kbytes workload and table 4.2.2.7 compares the response times of the 150Kbytes workload.

# of clients	Response time (msec)								
	Simulation						Measurement		
	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	165.8	165.8	165.8	165.8	165.8	165.8	140	380	
10	200	190	165.8	235.4	220	280			
20	244.6	221.1	165.8	353	282.5	420.2	380	520	420
40	460.8	333.5	202	843.4	672	1099	400	1130	
60	1864	895	202	3574	2796	4503	580	4920	

Table 4.2.2.6 : The response times of the six patterns of the 50.7Kbyte workload obtained from the simulation vs. the response times of the 50.7Kbytes workload whose size is constant obtained from the real world measurement in the distributed file system which consists of the Sun SPARCstation 10 workstations.

# of clients	Response time (msec)								
	Simulation						Measurement		
	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	381	381	381	381	381	381	300	1380	
5	507	440	381	810	685	500			
10	800	614	430	1567	1282	780	500	1670	
15	1287	869	437	2850	2377	1224			
20	2616	1492	445	7167	6153	9332	1120	10430	

Table 4.2.2.7 : The response times of the six patterns of the 150Kbyte workload obtained from the simulation vs. the response times of the 150Kbytes workload whose size is constant obtained from the real world measurement in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Table 4.2.2.8 compares the response times of the six patterns of the 8Kbytes workload obtained from the simulation with the response times of the 8Kbyte workload whose transaction size is fixed(constant) obtained from the real world measurement in the distributed file system which consists of the Sun SPARCstation 470 workstations. Table 4.2.2.9 compares the response times of the 50.7Kbytes workload and table 4.2.2.10 compares the response times of the 150Kbytes workload.

# of clients	Response time (msec)								
	Simulation						Measurement		
	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	140.7	140.7	140.7	140.7	140.7	140.7	100	280	120
10	154	150	140.7	162	153	175			
15	162	156	140.7	178	161	195	120	300	
20	171.9	162.4	140.7	197.6	171	220.2			
25	191	171	140.7	225	187	255			
30	214	182	140.7	260	207	305			
35	248	198	140.7	305	234	375			
40	289.5	217.6	140.7	364.1	268.9	460.6	180	530	

Table 4.2.2.8 : The response times of the six patterns of the 8Kbyte workload obtained from the simulation vs. the response times of the 8Kbytes workload whose size is constant obtained from the real world measurement in the distributed file system which consists of the Sun SPARCstation 470 workstations.

As in the share memory systems, if the transaction size is large e.g. 150Kbytes, some relatively very large response times were found in the measured response times.

The standard deviations of the measured response times and those of the measured CPU times are larger in the distributed file systems than in the shared memory systems. This study found that in general, as the transaction size increases, the confidences of the simulated response times decreases. In most cases, the simulated average response times of the 6 workload patterns fall within the range of the measured response time.

# of clients	Response time (msec)								
	Simulation						Measurement		
	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	331.5	331.5	331.5	331.5	331.5	331.5	320	650	330/ 350
5	390	380	331.5	440	400	510			
10	471	429	331.5	609.6	505	735.3	400	6850	
15	590	495	331.5	890	710	1080			
20	834.2	616.1	331.5	1323	1110	1640			
26	1400	920	331.5	2500	2000	3300			
30	2237	1237	345.6	4193	3257	5963			

Table 4.2.2.9 : The response times of the six patterns of the 50.7Kbyte workload obtained from the simulation vs. the response times of the 50.7Kbytes workload whose size is constant obtained from the real world measurement in the distributed file system which consists of the Sun SPARCstation 470 workstations.

# of clients	Response time (msec)								
	Simulation						Measurement		
	an&tf	ap&tf	af&tf	ap&tn	af&tn	an&tn	min.	max.	mode
0	775	775	775	775	775	775	650	2680	
5	1418	1186	775	2618	2294	3120	800	80180	
10	4399	2570	969.6	8849	6952	15300			

Table 4.2.2.10 : The response times of the six patterns of the 150Kbyte workload obtained from the simulation vs. the response times of the 150Kbytes workload whose size is constant obtained from the real world measurement in the distributed file system which consists of the Sun SPARCstation 470 workstations.

This study found the measured utilization of the CPU of the file server is very similar to the simulated utilization of the CPU : less than 5% deviation. In section 4.2.1, this study observed that the measured utilization of the CPU was larger than the simulated utilization of the CPU in the shared memory systems in most cases. The simulation results in the distributed file systems are found to have good confidence in general.

4.3 Summary

The measurement methodology used in this study has been discussed. This study used only the system provided performance tools. All measurement experiments were peformed in dedicated and closed environments. Two kind of experiments were performed : the standalone measurement and the real world(live) measurement. Shell script based predefined scenarios with either system provided commands("cat" command, etc.) or my own programs for read or write(write program A) were used.

In order to give better understanding of the distribution of the measured values, the mode values with minimum values and maximums values are presented. I specified the mode values only if the frequency of the mode value was found to be more than 20% of the total occurrence. Otherwise I left it blank.

It was observed that both the CPU time and the response time obtained from the simulation when there is no contention for the system resources agree well with the measured CPU time and response time in the standalone measurement experiments both in the shared memory systems and in the distributed file systems.

In real world measurement in both system paradigms, if the transaction size is large e.g. 150Kbytes, some relatively very large response times were found in the measured response times. In general, as the transaction size increases, the confidence of the simulated response time decreases and in most cases, the average response times of the six workload patterns obtained from the simulations falls within the range of the measured response times. In real world measurement, the measured utilization of the CPU is found to be larger than the simulated utilization of the CPU in most cases in the shared memory systems and the measured utilization of the CPU of the file server is very similar to the simulated utilization : less than 5% deviation in the distributed file systems.

The standard deviations of the measured response times and those of the measured CPU times are larger in the distributed file systems than in the shared memory systems. The simulation results for the two system paradigms are found to have good confidence in general.

When the standalone remote write(or read) experiment is performed in some heterogeneous distributed file systems, it is found that some simulation values are somewhat larger than the range of the measured values. For an explanation, it should be remembered that some parameter values in the sending systems are assumed to be same as those in the receiving systems. By removing this assumption, that is, obtaining each parameter value separately, the simulation values are expected to be within the range of the measured values. This remains as a topic for further study as mentioned in chapter 8.

Chapter 5

File Access Performance Evaluation of the Two System Paradigms

Chapter 3 described the virtual server models and performance parameters and chapter 4 validated them in real environments. This chapter comparatively investigates the file access performance of the two system paradigms using the virtual server models.

As the baseline distributed file systems, this study uses the distributed file system which consists of the Sun SPARCstation 10 workstations, the distributed file system which consists of the Sun SPARCstation 470 workstations and the distributed file system which consists of the Sun 3/60 workstations. As the baseline shared memory systems, this study uses the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 workstation. This study uses the six workloads individually in each system. They are the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload.

In the following sections in this chapter, the following conditions hold commonly used unless otherwise specified. Write file access is performed unless read file access is explicitly specified to be performed. No caching occurs unless caching is explicitly specified to occur. The workload pattern of the Poisson distribution for input arrival and the log-normal distribution for input transaction size is used unless the workload pattern is explicitly specified.

5.1 The Effect of Workload

Here this study investigates the effect of the workload characteristics on the file access performance and explains what workload characteristics are chosen for the baseline cases and why they are chosen.

Section 5.1.1 compares the file access performance of the read operations and that of the write operations. Section 5.1.2 investigates the effect of the average transaction size of the workload on the file access performance. Section 5.1.3 investigates the effect of the workload pattern on the file access performance. Section 5.1.4 investigates the effect of the workload fluctuation on the file access performance by comparing the average response time when the steady workload is used and that when the bursty workload is used.

5.1.1 Read and Write

This section compares the file access performance of read and that of write. The baseline performance model of figure 3.4.1.B and the baseline performance parameters of table 3.4.2.A are used for the simulation of the three shared memory systems. The three shared memory systems are the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 Workstation.

The baseline performance model of figure 3.2.6.B and the baseline performance parameters of table 3.2.7.C are used for the simulation of the three distributed file systems. The three distributed file systems are the distributed file system which consists of the Sun SPARCstation 10 workstations, the distributed file system which consists of the Sun SPARCstation 470 workstations and the distributed file system which consists of the Sun 3/60 workstations. The 6 different workloads of table 3.5.2.A are individually used in each system of the both system paradigms.

In the read operation and the write operation, only the processing sequence is different from each other since it is assumed that the request sending operation has the same overhead as the response receive operation in the client and the request sending operation in the client has the same overhead as the response sending operation in the file server and the data reading operation from the disk has the same overhead as the data writing operation on the disk.

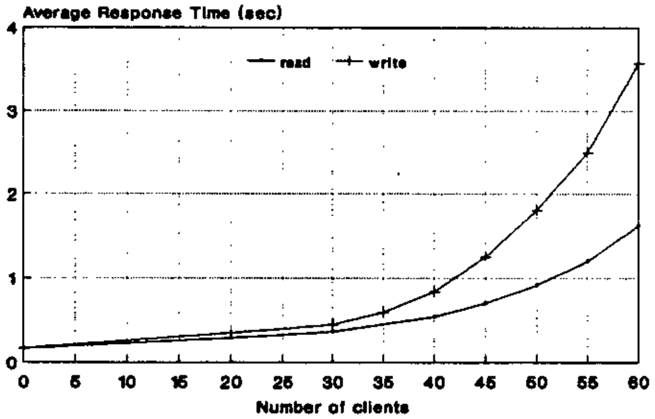


Figure 5.1.1.1 : 50.7Kbytes

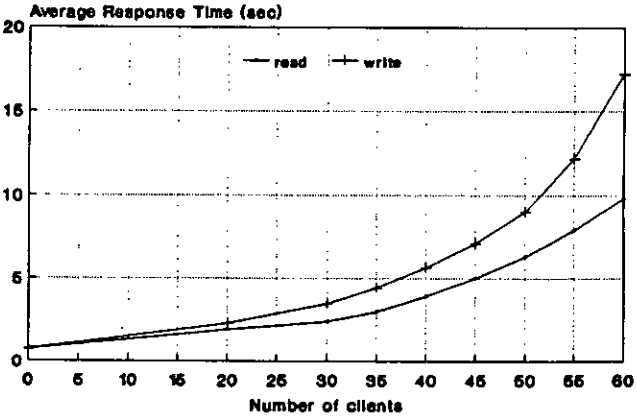


Figure 5.1.1.2 : 316kbytes(B)

The average response time of the read vs. the average response time of the write in the distributed file system which consists of the Sun SPARCstation 10 workstations.

In the shared memory systems, the read and write show the same average response time all the time as we expect. In the distributed file systems, the average response time of write develops faster than the average response time of read as the contention for the system resources grows, that is, the number of clients increases as shown in figure 5.1.1.1 and figure 5.1.1.2. This is due to the correlation effect of the network, the network control unit and the CPU. The growth pattern of average response time is similar all the time. See appendix B for the figures of other cases.

Throughout this chapter, chapter 6 and chapter 7, the write operation is used as the baseline file access operation unless the read operation is specified to be performed. The read operation is less sensitive to the contention for the system resources and in real environments, caching occurs more frequently during reading than during writing.

5.1.2 Transaction Size

This section describes the effect of the transaction size on the file access performance. The transaction size is increased : 8kbytes, 47kbytes, 50.7kbytes, 316kbytes and 1856kbytes. The effect is investigated using the three kinds of systems where the system power and the system organization differ : the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 Workstation. The effect is investigated in two different system paradigms : the distributed file system and the shared memory system. The baseline virtual server model of figure 3.2.6.B, that of figure 3.4.1.B, the baseline performance parameter values of table 3.2.7.C and that of table 3.4.2.A are used. In this section, the transactions are generated according to the Poisson distributions for the arrival and the log-normal distributions for the size.

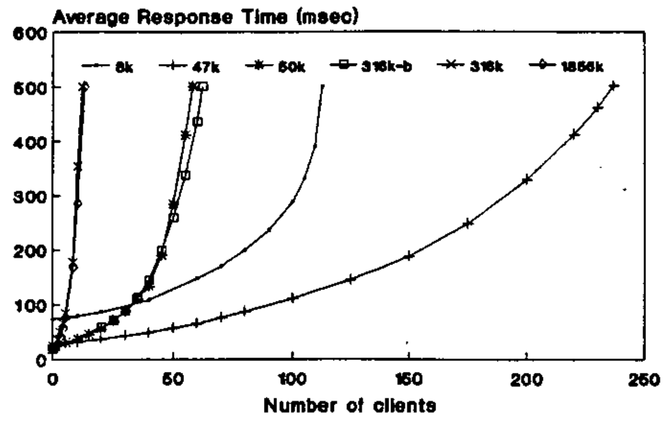


Figure 5.1.2.1 : S10, DFS

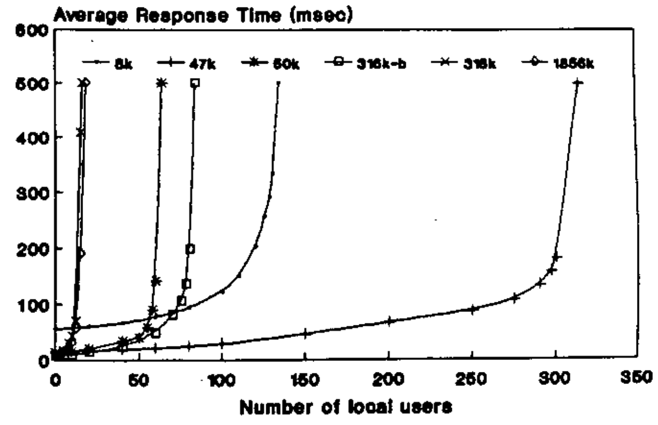


Figure 5.1.2.2 : S10, SMS

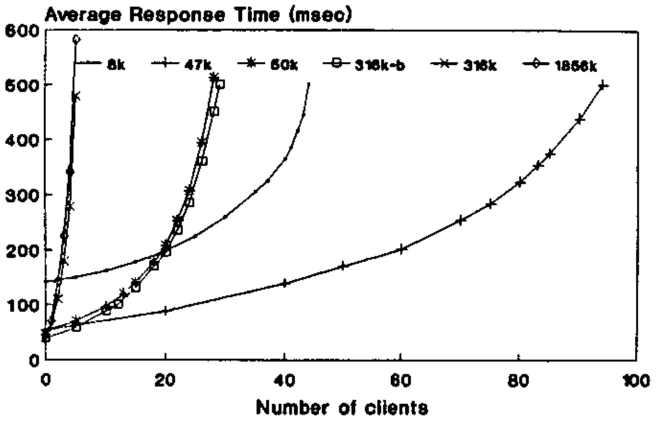


Figure 5.1.2.3 : S470, DFS

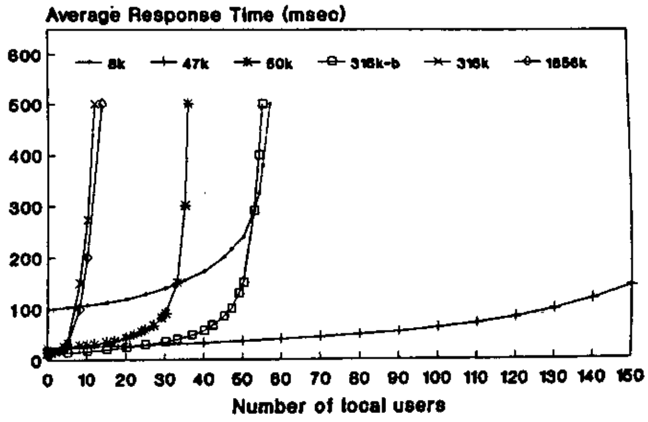


Figure 5.1.2.4 : S470, SMS

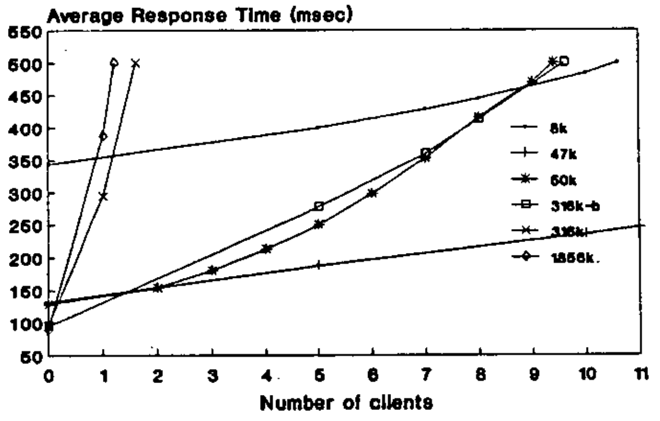


Figure 5.1.2.5 : S360, DFS

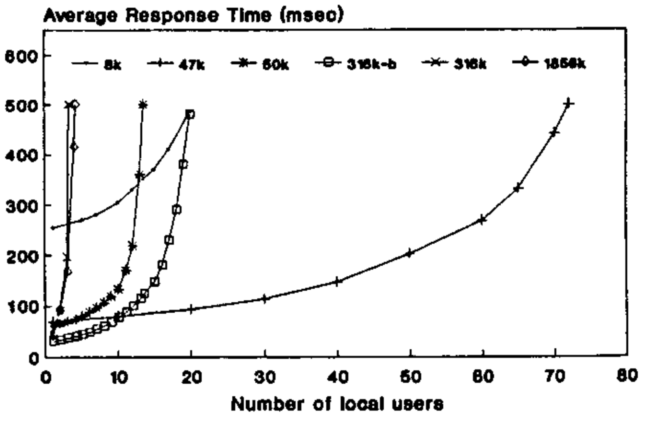


Figure 5.1.2.6 : S360, SMS

The effect of the transaction size on the average response time per 8kbytes data transferred.

Figure 5.1.2.1, figure 5.1.2.3 and figure 5.1.2.5 show the average response time per 8kbytes data transferred of the six workloads in the distributed file system which consists of the Sun SPARCstation 10 workstations, in the distributed file system which consists of the Sun SPARCstation 470 workstations and in the distributed file system which consists of the Sun 3/60 workstations respectively as the number of clients increases gradually.

Figure 5.1.2.2, figure 5.1.2.4 and figure 5.1.2.6 show the average response time per 8kbytes data transferred of the six workloads in the shared memory system of the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 workstation respectively as the number of local users increases gradually.

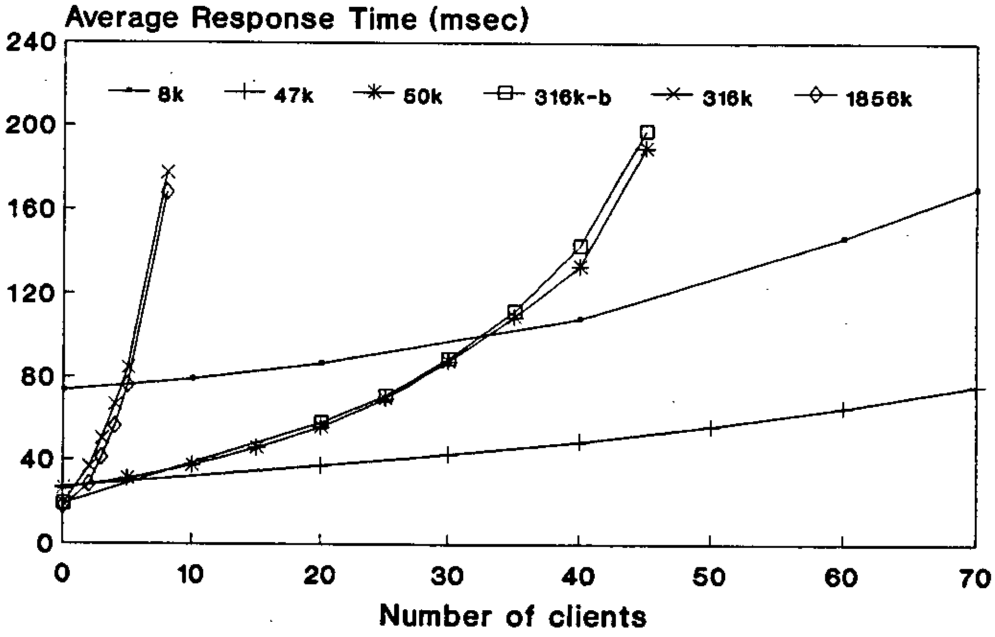


Figure 5.1.2.7 : Zooming of figure 5.1.2.1

As an example, I zoom figure 5.1.2.1 in figure 5.1.2.7, which shows the average response time per 8kbytes data transferred in the distributed file system which consists of the Sun SPARCstation 10 workstations.

The average response time per 8kbyte data transferred of the 1856kbytes workload is much smaller than that of the 8kbytes workload when there is no contention for the system resources. This is due to the amortization of the overheads which are constant to the average transaction size.

In the three shared memory systems, the bursty workloads such as the 47kbytes workload, the 316kbytes(B) workload and the 1856kbytes workload show better average response time per 8kbyte data transferred than the counterpart steady workloads such as the 8kbytes workload, the 50.7kbytes workload and the 316kbytes workload. This is because the amortization effect overwhelms the bursty effect, that is, the effect of the bursty arrivals on the file access performance is less than the effect of the amortization on the file access performance.

In the distributed file systems, the bursty workloads such as the 316kbytes(B) workload and the 1856kbytes workload show better average response time per 8kbytes data transferred than the counterpart steady workloads such as the 50.7kbytes workload and the 316kbytes workload up to a certain number of the clients and beyond the number, the bursty workloads show worse and worse average response time per 8kbytes data transferred than the counterpart steady workloads since now the contention effect overwhelms the amortization effect. This is commonly observed in the three systems.

It is commonly observed in the two different system paradigms that when this study uses the workload pairs such as the 50kbytes workload and the 316kbytes(B) workload, and the 316kbytes workload and the 1856kbytes workload, the gaps between the average response time per 8kbytes data transferred of the steady workloads and that of the bursty workloads are relatively narrow and in the workload pairs of the 8kbytes workload and the 47kbytes workload, the gap is relatively wide. This means that in the workload pair of the 8kbytes workload and the 47kbytes workload, the file access performance is much affected by the amortization and in the two other workload pairs, the amortization has little effect on the file access performance.

Generally the larger the average transaction size is, the better the average response time per 8kbyte data transferred is when there is no contention for the system resources. As the contention increases, that is, the number of clients or the number of local users increases, if the average the transaction size is larger, then the the average response time per 8k data transferred grows more quickly. Therefore, there exist crossing points in the figures.

5.1.3 Workload Pattern

This section describes the effect of the workload pattern on the file access performance. Three different types of arrival distributions and two different types of transaction size distributions are used. The three arrival distributions are the Poisson arrival(the exponential inter-arrival time distribution), the log-normal inter-arrival time distribution and the constant inter--arrival time distribution. The two transaction size distributions are the log-normal distribution and the constant distribution. The total six workload patterns are the followings.

1) The workload pattern which has the Poisson arrival distribution and the log-normal transaction size distribution : ap&tn.
2) The workload pattern which has the Poisson arrival distribution and the constant transaction size distribution : ap&tf.
3) The workload pattern which has the log-normal inter-arrival time distribution and the log-normal transaction size distribution : an&tn.

4) The workload pattern which has the log-normal inter-arrival time distribution and the constant transaction size distribution : an&tf.

5) The workload pattern which has the constant inter-arrival time distribution and the log-normal transaction size distribution : af&tn.

6) The workload pattern which has the constant inter-arrival time distribution and the constant transaction size distribution : af&tf.

The baseline virtual performance model of figure 3.2.6.B and the baseline performance parameter values of table 3.2.7.C are used for the simulation of the distributed file system which consists of the Sun SPARCstation 10 workstations. The baseline virtual performance model of figure 3.4.1.B and the baseline performance parameter values of table 3.4.2.A are used for the shared memory system of the Sun SPARCstation 10 workstation. Only the arrival distribution and the workload size distribution are changed.

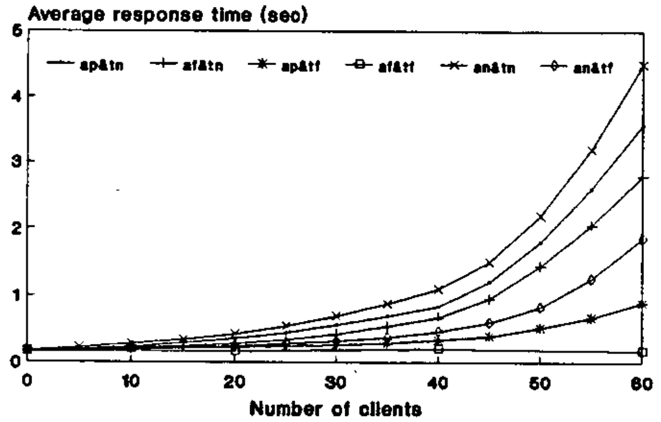


Figure 5.1.3.1 : The effect of the workload pattern on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

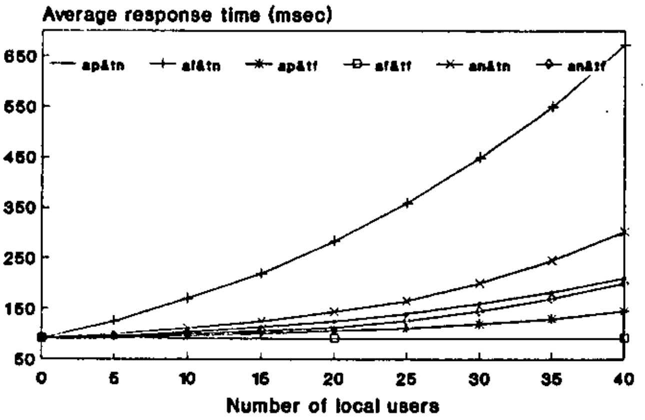


Figure 5.1.3.2 : The effect of the workload pattern on the average response time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

Figure 5.1.3.1 shows the average response times of the six workload patterns when the 50.7kbytes workload is used in the distributed file system which consists of the Sun SPARCstation 10 workstations. Figure 5.1.3.2 shows the average response times of the six workload patterns when the 50.7kbytes workload is used in the shared memory system of the Sun SPARCstation 10 workstation. See appendix B for the figures of other cases.

What was commonly observed is summarized below. The best average response time is always found in the workload pattern of the constant inter-arrival time distribution and the constant transaction size distribution. The workload pattern shows the constant average response time as the number of clients or the number of local users increases whatever workload is used. The worst average response time is always found in the workload pattern of the log-normal inter-arrival time distribution and the log-normal transaction size distribution except for three cases. The workload pattern of the Poisson arrival distribution and the log-normal transaction size distribution shows the second or third worst average response time all the time except for three cases where it takes the position of the worst average response time instead of the workload pattern of the log-normal inter-arrival time distribution and the log-normal transaction size distribution.

It is observed that when steady workloads are used, the workload pattern of the Poisson arrival distribution and the constant transaction size distribution shows worse average response time than the workload pattern of the log-normal inter-arrival time distribution and the constant transaction size distribution but when bursty workloads are used, the converse is true.

By comparing the average response time obtained from simulations with the average response time obtained from measurements, this study finds that the workload pattern of the Poisson arrival distribution and the log-normal transaction size distribution accurately represents the arrival distribution and the transaction size distribution of the real workload, as explained in chapter 4.

5.1.4 Workload Fluctuation

As explained in section 3.5.2, this study interprets the 10minutes workloads as the steady workloads and the 10 seconds workloads as the bursty workloads. Three workload pairs are used in this study. The 8kbytes workload, 50.7kbytes workload and the 316kbytes workload are the 10minutes workloads and the 47kbytes workload, the 316kbytes(B) workload and the 1856kbytes workload are 10 seconds workloads as already explained in chapter 3.

As an example of the bursty workloads, let's look at the 47kbytes workload. We can interpret the 47kbytes data traffic per second as the i/o traffic caused by the series of 8kbytes transactions or one 47kbytes transaction. The former interpretation leads to a fine-grained workload with small inter-arrival times and the latter interpretation leads to a coarse-grained workload with large inter-arrival times because the traffic rate generated should be same in the two interpretations. In the former interpretation, the transaction arrival rate per unit time(second) is the same both in the steady workloads and in the bursty workloads but the arrival distribution is different. That is, the arrival distribution of the bursty workloads follows cluster distributions or group arrival patterns. The cluster distribution or the group arrival pattern means the distribution where the inter-arrival time inside the cluster(the group), that is, the inter-arrival time between the members of the cluster(the group) or intra-cluster-arrival time, is very small and the inter-arrival time between the clusters is very large. In the case of ultimate burstiness, the intra-cluster-arrival time tends to zero in its limit, that is, there is no inter-arrival time gap inside the cluster.

Figure 5.1.4.1 and figure 5.1.4.2 show the average response times per 8kbytes data

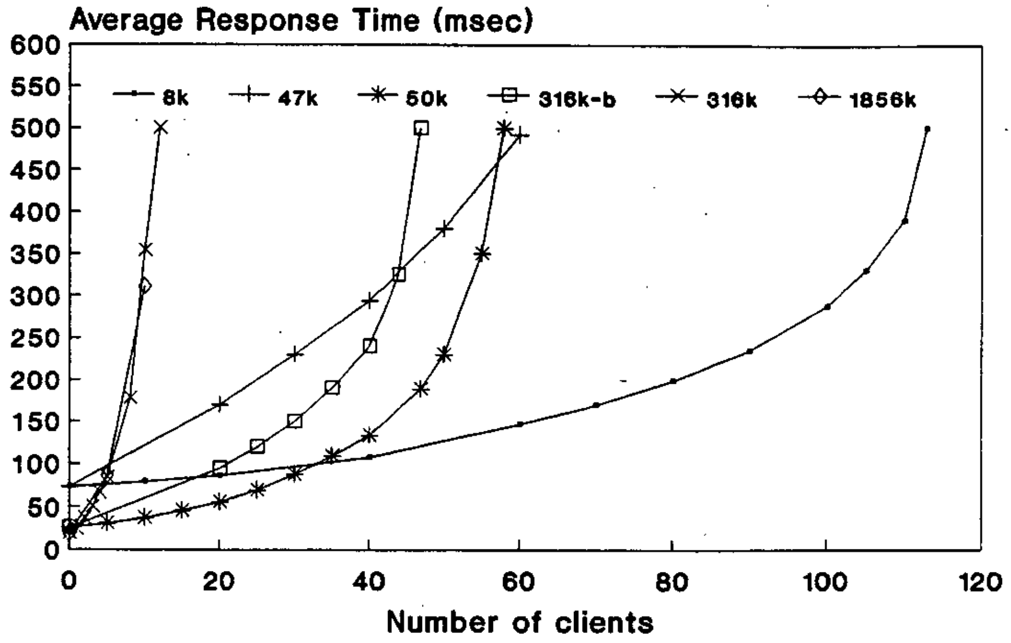


Figure 5.1.4.1 : The effect of the ultimately bursty workloads on the average response times per 8kbytes data transferred in the distributed file system which consists of the Sun SPARCstation 10 workstations.

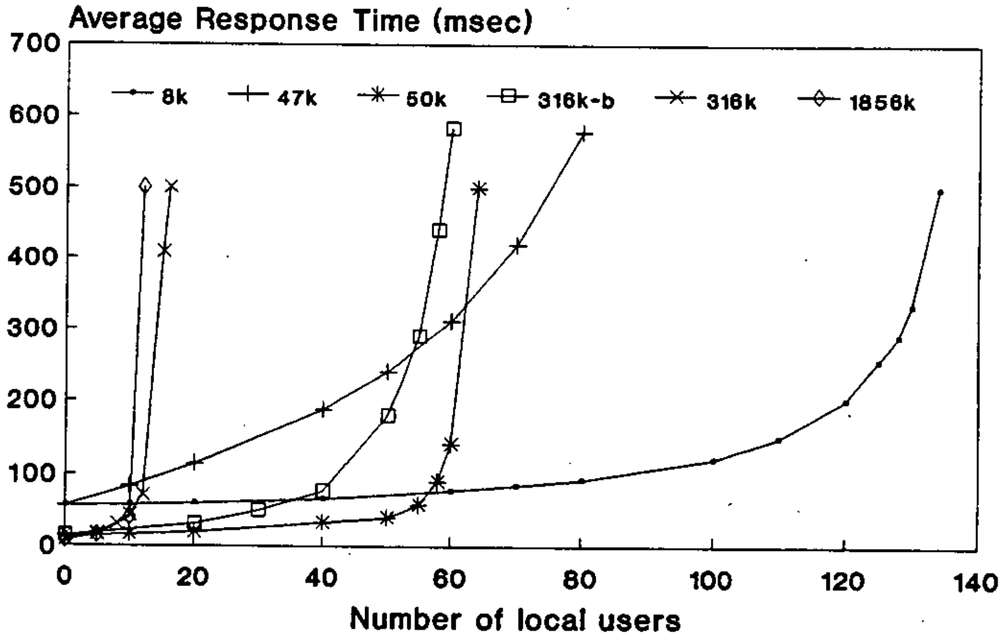


Figure 5.1.4.2 : The effect of the ultimately bursty workloads on the average response times per 8kbytes data transferred in the Sun SPARCstation 10 workstation.

transferred when the ultimately bursty workloads in the former interpretation are used in the distributed file system which consists of the Sun SPARCstation 10 workstations and in the Sun SPARCstation 10 workstation.

This study uses the latter interpretation in the simulations in the following sections. In this case, because of amortization, the average response times of the bursty workloads are smaller than the average response times of the bursty workloads in the former interpretation as comparatively observed in figure 5.1.4.1 and 5.1.4.2 and figure 5.1.2.1 to figure 5.1.2.6.

In section 5.1.3, it was already pointed out that the workload pattern of the Poisson arrival distribution and the constant transaction size distribution shows worse average response time than the workload pattern of the log-normal inter-arrival time distribution and the constant transaction size distribution when the steady workloads are used but the former shows better average response time than the latter when the bursty workloads are used, both in the distributed file systems and in the shared memory systems.

The effect on the file access performance by the workload fluctuation is further explained where appropriate throughout this chapter, chapter 6 and chapter 7 when it is observed.

5.2 Utilization, Congestion and Average Response Time

This section investigates the utilization of each system resource, congestion effect, the effect of each system resource on the average response time. It is found out how many clients or local users can be supported in the baseline environments and which system resource saturates first in each system.

5.2.1 Utilization

Figure 5.2.1, figure 5.2.2, figure 5.2.3 and figure 5.2.4 show the average utilization of the system resources such as the CPU, the disk, the network interface unit and the network respectively in the distributed file system which consists of the Sun SPARCstation 10 workstations. Figure 5.2.5 shows the average utilization of the CPU in the shared memory system of the Sun SPARCstation 10 workstation. Figure 5.2.6, figure 5.2.7 and figure 5.2.8 show the average utilization of the CPU, the disk and the network interface unit respectively in the distributed file system which consists of the Sun SPARCstation 470 workstations. Figure 5.2.9 shows the average utilization of the CPU in the shared memory system of the Sun SPARCstation 470 workstation. Figure 5.2.10, figure 5.2.11 and figure 5.2.12 show the average utilization of the CPU, the disk and the network interface unit respectively in the distributed file system which consists of the Sun 3/60 workstations. Figure 5.2.13 shows the average utilization of the CPU in the shared memory system of the Sun 3/60 workstation.

The figures show the utilization in the theoretical limit. It is observed that if the measured utilization is closer to the theoretical limit, then the file access performance becomes better. When the 6 workloads of which the inter-arrival times are constant and the transaction sizes are also constant are used, the utilizations of the system resources are nearly same as that in the figures. In these cases, the average response time is almost constant as the number of clients increases and the throughput is the best among the 6 workload patterns in section 5.1.3. Each line in the figures is obtained when the system resource of which the line represents the utilization is the major bottleneck point, that is, other system resources in the system have lower utilization than the system resource.

In the figures, the average utilization of the disk i/o subsystem, that is, the disk and disk interface unit in the distributed file system is the same as that in the

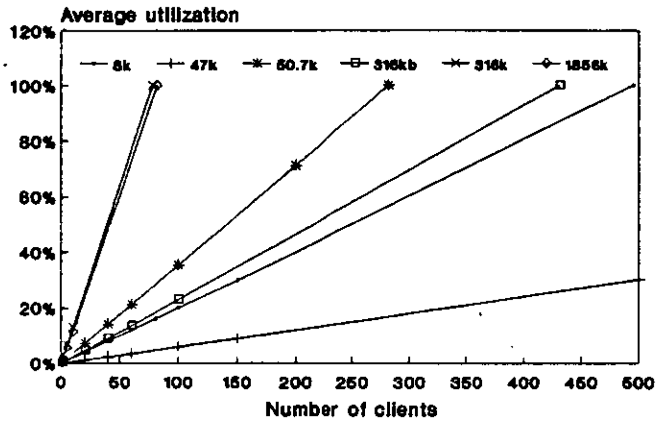


Figure 5.2.1 : S10, DFS, CPU

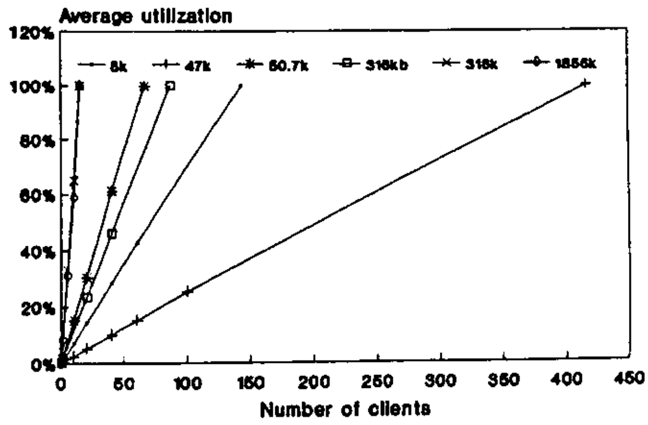


Figure 5.2.2 : S10, DFS, DISK

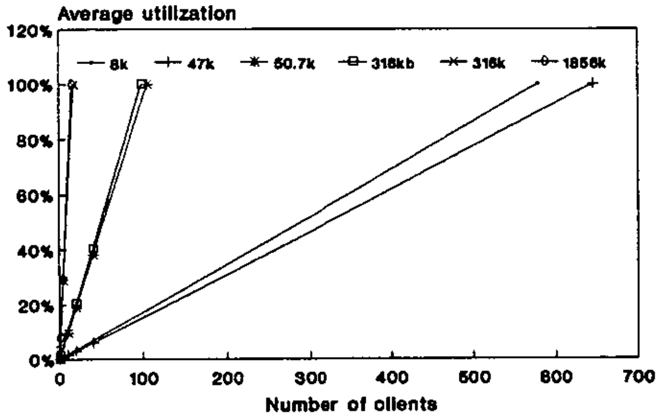


Figure 5.2.3 : S10, DFS, NETWORK

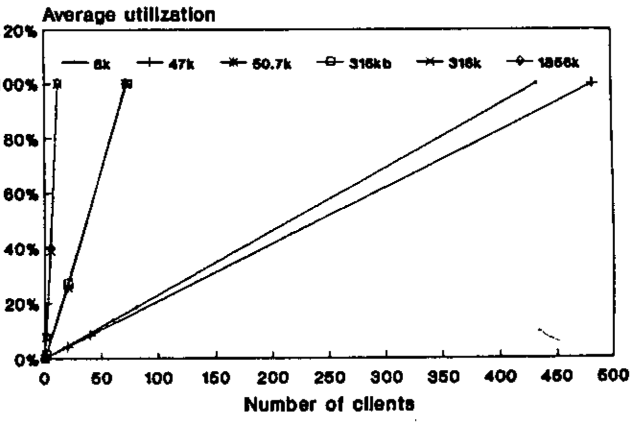


Figure 5.2.4 : S10, DFS, NET-DMA

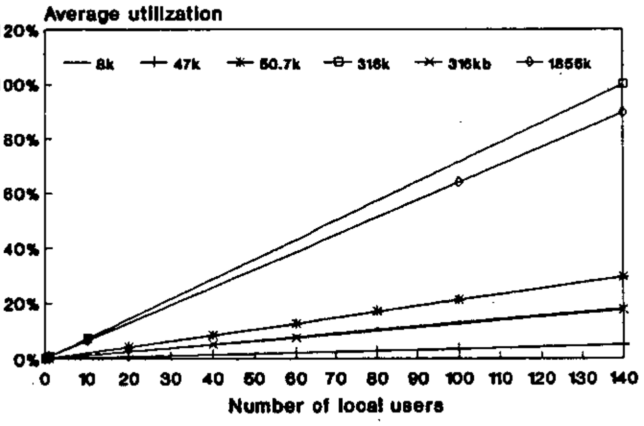


Figure 5.2.5 : S10, SMS, CPU

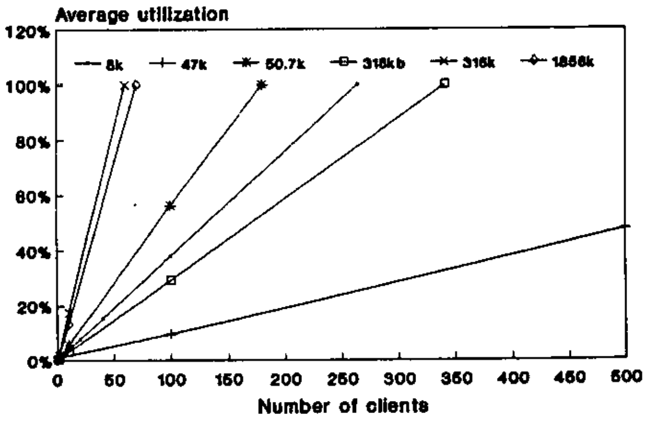


Figure 5.2.6 : S470, DFS, CPU

The average utilization of the system resources.

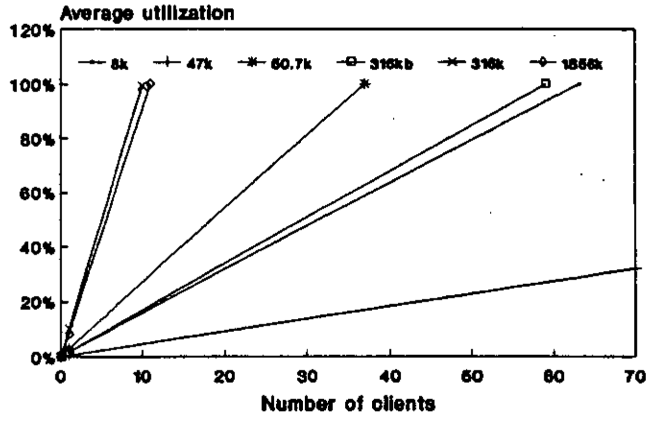


Figure 5.2.7 : S470, DFS, DISK

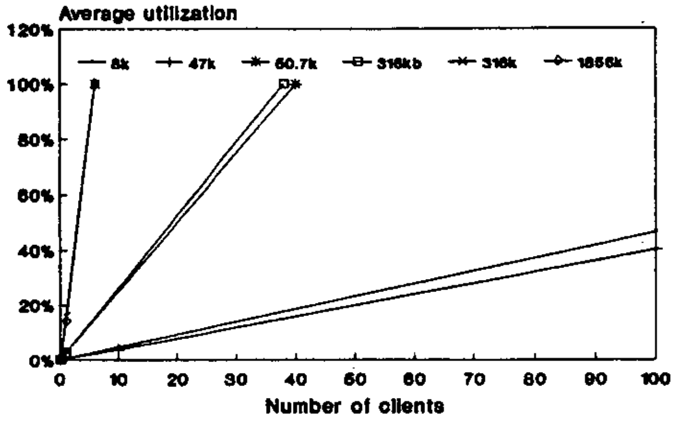


Figure 5.2.8 : S470, DFS, NET-DMA

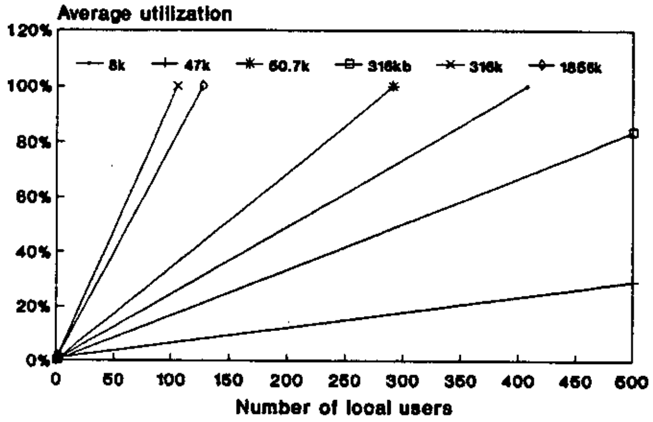


Figure 5.2.9 : S470, SMS, CPU

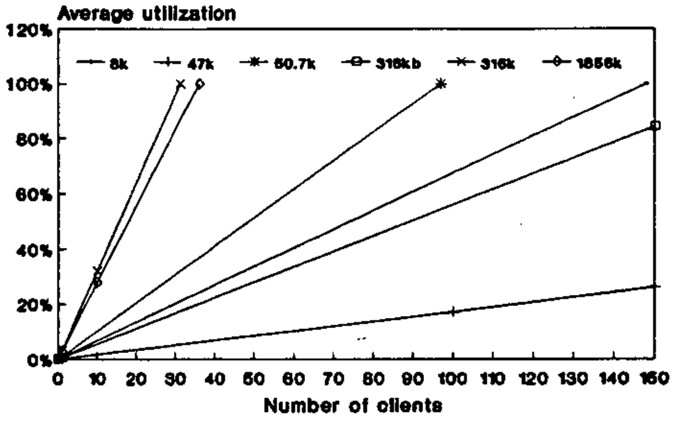


Figure 5.2.10 : S360, DFS, CPU

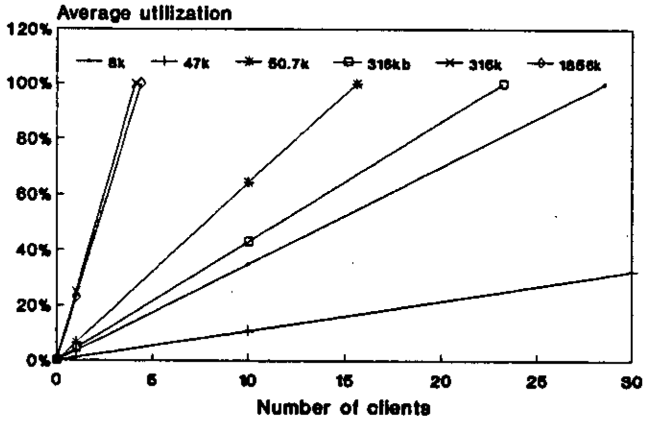


Figure 5.2.11 : S360, DFS, DISK

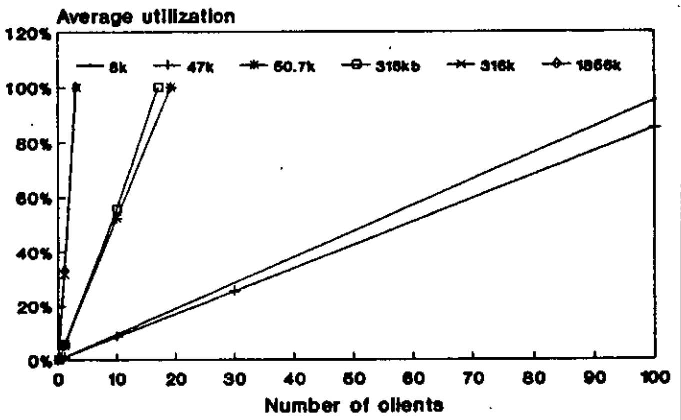


Figure 5.2.12 : S360, DFS, NET-DMA

The average utilization of the system resources.

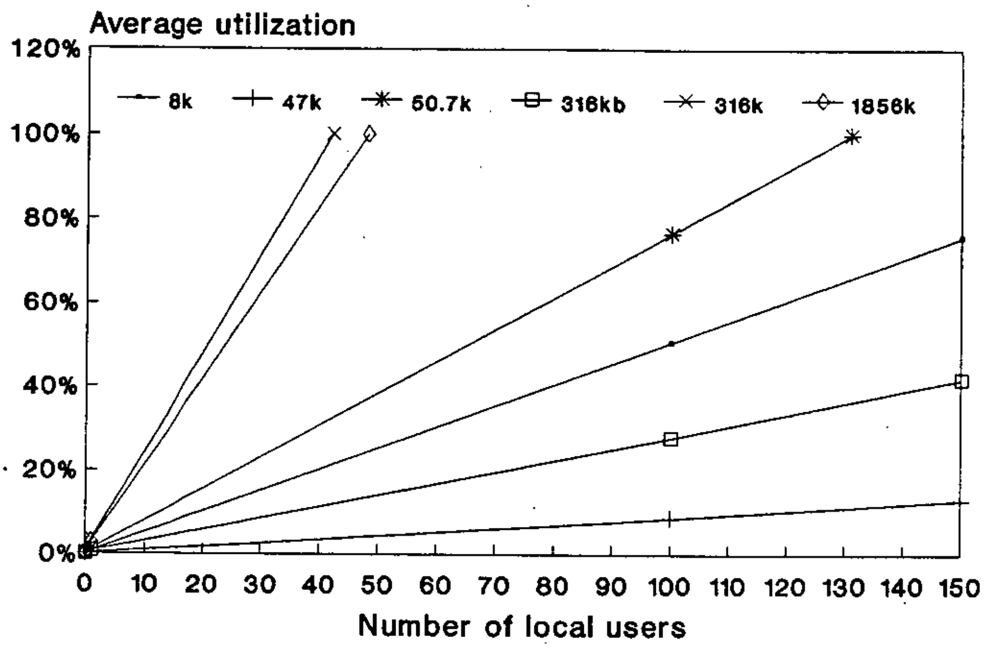


Figure 5.2.13 : The average utilization of the CPU of the Sun 3/60 workstation.

shared memory system. The average utilization of the network should be constant regardless of the system power in the distributed file system, that is, the average utilization of the network in the distributed file system which consists of the Sun SPARCstation 10 workstations is the same as that in the distributed file system which consists of the Sun SPARCstation 470 workstations or the Sun 3/60 workstations since the 10Mbps network is used all the time in the baseline distributed file systems. It varies only with the average transaction size and the number of clients in the distributed file systems.

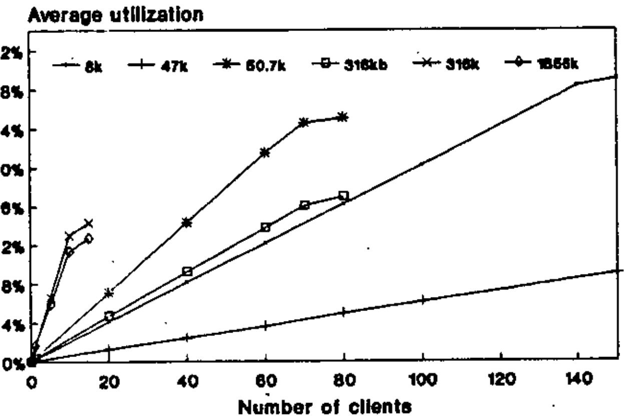


Figure 5.2.14 : S10, DFS, CPU

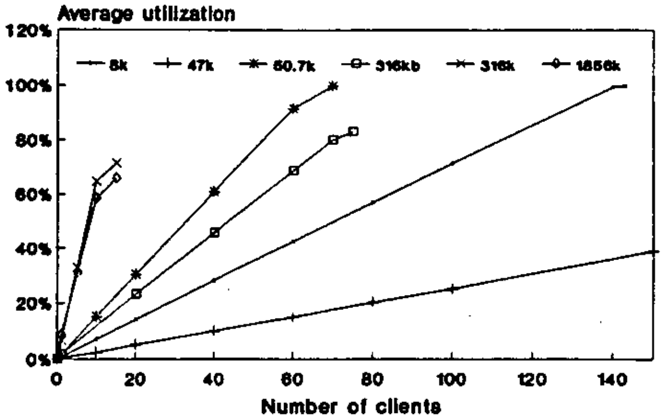


Figure 5.2.15 : S10, DFS, DISK

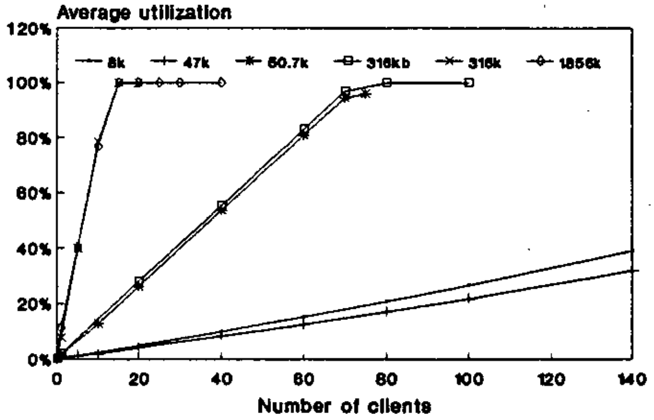


Figure 5.2.16 : S10, DFS, NET-DMA

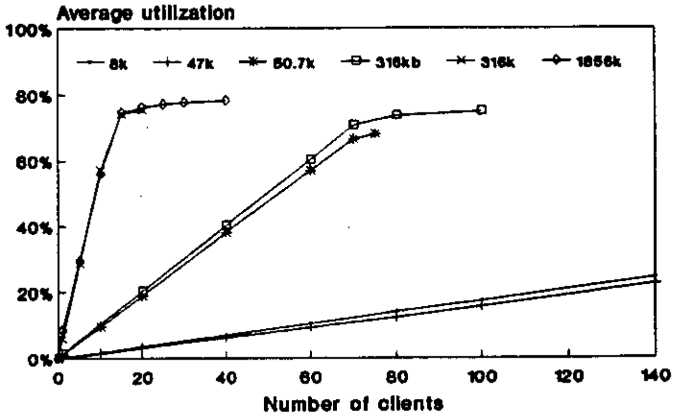


Figure 5.2.17 : S10, DFS, NETWORK

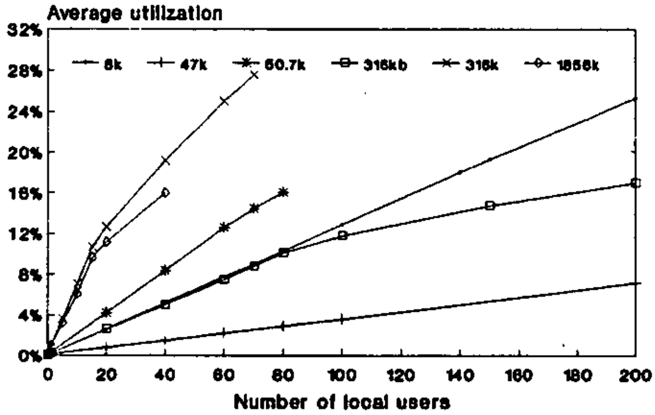


Figure 5.2.18 : S10, SMS, CPU

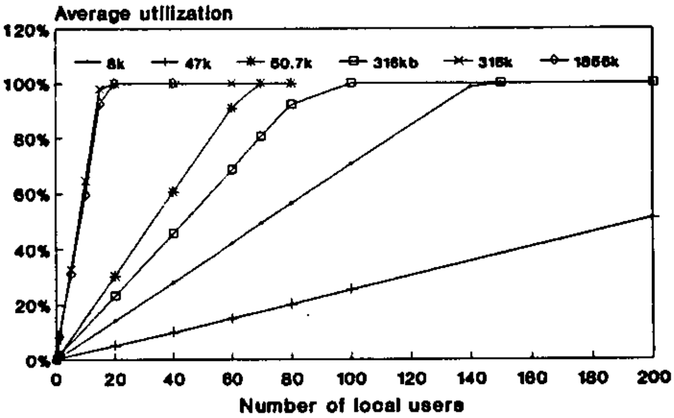


Figure 5.2.19 : S10, SMS, DISK

The simulated average utilization of the system resources.

Figure 5.2.14, figure 5.2.15, figure 5.2.16, figure 5.2.17, figure 5.2.18 and figure 5.2.19 show the simulated average utilization of the system resources of the distributed file system which consists of the Sun SPARCstation 10 workstations and the local shared memory system of the Sun SPARCstation 10 workstation, when the 6 workloads of which the arrival follows the Poisson distribution and the transaction size follows the log-normal distribution are used.

The slopes of the average utilization lines of the CPU, the disk i/o subsystem, the network interface unit and the network are almost straight. If more contention arises in the system, for example when I use the workload of which both the inter-arrival time and the transaction size follow the log-normal distribution, the lines curve below the straight line of the theoretical limit.

5.2.2 Congestion

Table 5.2.2.1 shows the number of clients or the number of local users with which each resource is 100% utilized and which I call the saturation point, in the baseline cases. The numbers are obtained using the theoretical average utilization. For example, the CPU of the Sun SPARCstation 10 workstation can conservatively support up to 495 clients before saturation when the 8kbytes workload is used in the distributed file systems where the other system resources are enhanced to be better so that the CPU of the Sun SPARCstation 10 is the main bottleneck point.

When this study uses the 8kbytes workload or the 50.7kbytes workload among the steady workloads or the 47kbytes workload among the bursty workloads, the disk i/o subsystem is the main bottleneck point which saturates the three baseline distributed file systems. The next bottleneck point is the network control unit of the file server in the three baseline distributed file systems. When this study uses the 316kbytes(B) workload or the 1856kbytes workload among the bursty workloads or the 316kbytes workload among the steady workloads, that is, when this study uses the workloads of which the average transaction size is equal to or larger than 316kbytes, the network control unit is the main bottleneck point which saturates the three baseline distributed file systems even though the capacity of the network is still enough to support more clients. In the three baseline distributed file system, the next bottleneck point is the disk i/o subsystem or the network control unit. In the three baseline shared memory systems, the major bottleneck point is the disk i/o subsystem and the CPU is the next bottleneck point in all 6 workloads.

			8k	47k	50.7k	150k	316kb	316k	1856k
S10	DFS	CPU	495	1662	281	140	431	76	82
		DISK	143	396	66	29	87	15	15
		NET	578	645	105	36	98	17	16
		NET-DMA	433	483	72	27	73	12	12
	SMS	CPU	775	2803	476	250	795	140	156
		DISK	143	396	66	29	87	15	15
S470	DFS	CPU	264	1054	180	103	341	60	70
		DISK	63	220	37	19	59	10	11
		NET	578	645	105	36	98	17	16
		NET-DMA	215	249	40	14	38	6	6
	SMS	CPU	407	1702	291	175	598	105	127
		DISK	63	220	37	19	59	10	11
S360	DFS	CPU	148	573	97	54	178	31	36
		DISK	28	92	15	7	23	4	4
		NET	578	645	105	36	98	17	16
		NET-DMA	105	117	19	6	17	3	3
	SMS	CPU	198	769	131	73	239	42	48
		DISK	28	92	15	7	23	4	4

S10 : The Sun SPARCstation 10 workstation
S470 : The Sun SPARCstation 470 workstation
S360 : The Sun 3/60 workstation

DFS : The distributed file system
SMS : The shared memory system
NET-DMA : The network interface unit

Table 5.2.2.1 : The saturation points in the baseline systems.

Paradigm	Resource	System	Workload Type					
			8k	47k	50.7k	316k(B)	316k	1856k
DFS	CPU	S360	1	1	1	1	1	1
		S470	1.79	1.84	1.86	1.92	1.94	1.95
		S10	3.35	2.91	2.9	2.43	2.46	2.3
SMS	CPU	S360	1	1	1	1	1	1
		S470	2.06	2.22	2.23	2.51	2.5	2.65
		S10	3.92	3.65	3.64	3.33	3.34	3.25
DFS & SMS	DISK	S360	1	1	1	1	1	1
		S470	2.25	2.4	2.47	2.57	2.5	2.75
		S10	5.11	4.31	4.4	3.79	3.75	3.75
DFS	NET-DMA	S360	1	1	1	1	1	1
		S470	2.05	2.13	2.11	2.24	2	2
		S10	4.13	4.13	3.79	4.3	4	4

Table 5.2.2.2 : The ratio of the saturation point of each resource in the three distributed file systems to the saturation point of each resource in the distributed file system which consists of the Sun 3/60 workstations and that in the three shared memory systems to that in the Sun 3/60 workstation.

Table 5.2.2.2 shows the ratio of the saturation point of the system resource such as the CPU, the disk i/o subsystem and the network interface unit among the three distributed file systems and among the three shared memory systems when the 6 workloads are used individually.

The MIPS ratio among the three systems, that is, the ratio of the MIPS of the Sun SPARCstation 10 workstation to the MIPS of the Sun SPARCstation 470 workstation to the MIPS of the Sun 3/60 workstation is 33.4 : 7.34 : 1. The ratios in table 5.2.2.2 are far below the MIPS ratio and different at each resource. The largest ratio is observed in the disk i/o subsystem and the smallest ratio is observed in the CPU.

5.2.3 Average Response Time

This section investigates the effect of each system resource on the average response time when there is no contention for the system resources at all.

Figure 5.2.3.1 to figure 5.2.3.5 show the effect on the average response time of the system resources such as the CPU, the disk i/o subsystem, the network interface unit in the clients, the network and the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations and the effect of the CPU and the disk i/o subsystem on the average response time in the shared memory system of the Sun SPARCstation 10 workstation when the 6 workloads such as the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbyte workload(B), the 316kbytes workload and the 1856kbytes workload are used respectively and there is no contention for the system resources at all. The effect when the 316kbytes workload(B), a bursty state workload, is used is the same as the effect when the 316kbytes workload, the steady state workload, is used.

Figure 5.2.3.6 to figure 5.2.3.10 show the effect on the average response time of the system resources when the Sun SPARCstation 470 workstations are used instead of the Sun SPARCstation 10 workstations in the same environment.

Figure 5.2.3.11 to figure 5.2.3.15 show the effect on the average response time of the system resources when the Sun 3/60 workstations are used instead of the Sun SPARCstation 10 workstations in the same environment.

In the shared memory systems, the percentage of the average CPU time in the total average response time decreases as the average transaction size increases, in other words, the percentage of the average disk i/o time in the total average response time increases as the average transaction size increases. This agrees with our common belief that the file access activity will use the i/o subsystem more heavily than the CPU.

In the client of the distributed file system, the percentage of the average network

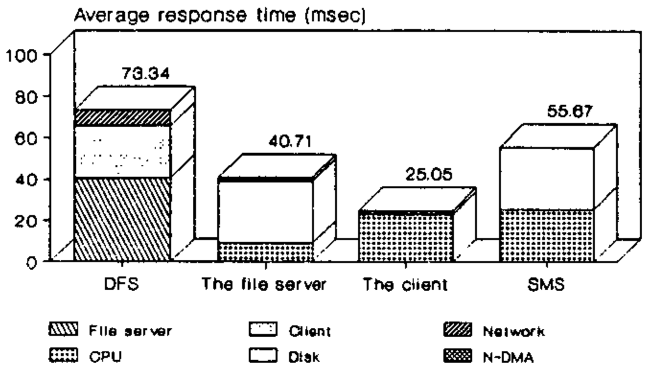


Figure 5.2.3.1 : S10, 8Kbytes

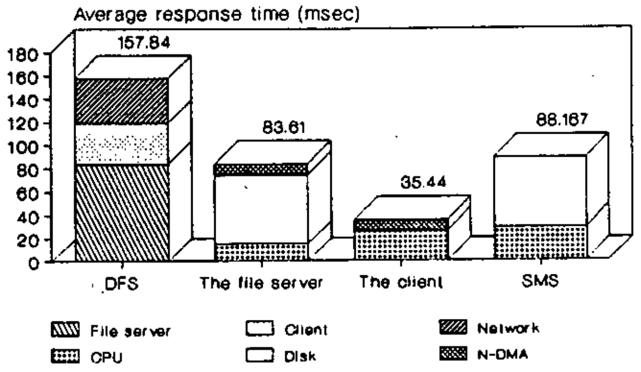


Figure 5.2.3.2 : S10, 47Kbytes

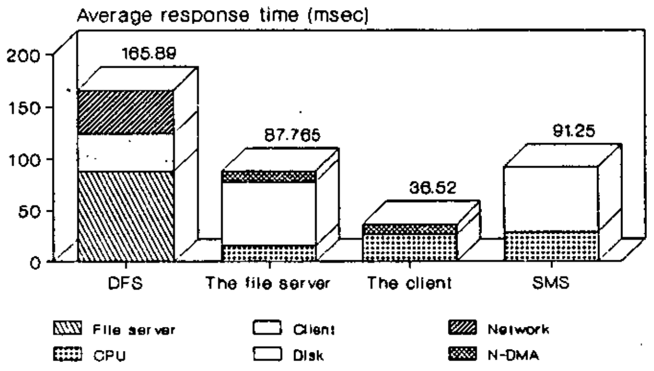


Figure 5.2.3.3 : S10, 50.7Kbytes

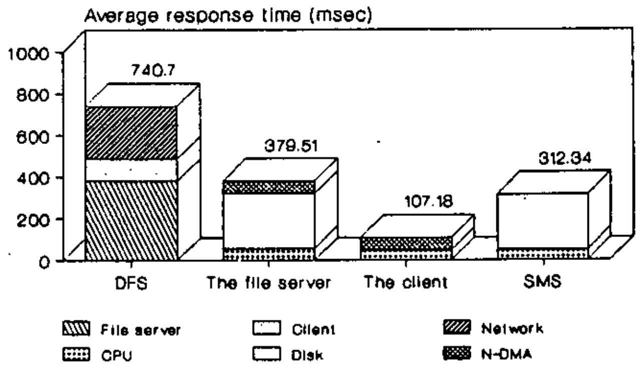


Figure 5.2.3.4 : S10, 316Kbytes

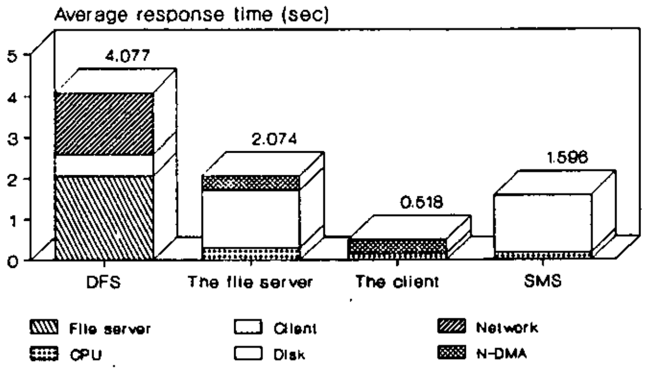


Figure 5.2.3.5 : S10, 1856Kbytes

The map of the average response time when there is no queueing delay in the distributed file system which consists of the Sun SPARCstation 10 workstations and in the Sun SPARCstation 10 workstation. Abbreviation : DFS stands for the distributed file system. SMS stands for the shared memory system. N-DMA stands for the network interface unit.

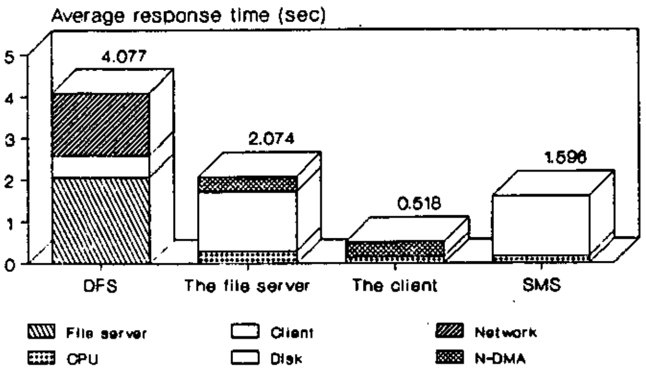


Figure 5.2.3.6 : S470, 8Kbytes

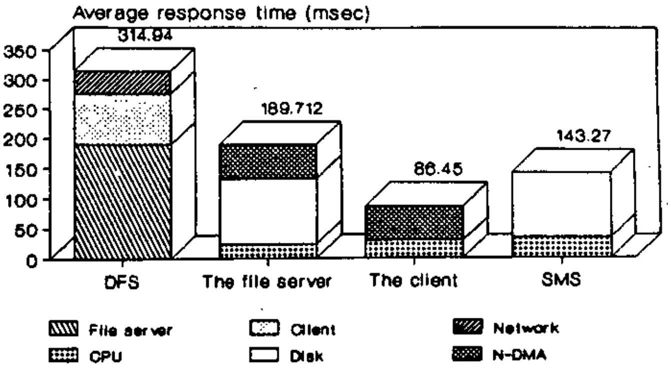


Figure 5.2.3.7 : S470, 47Kbytes

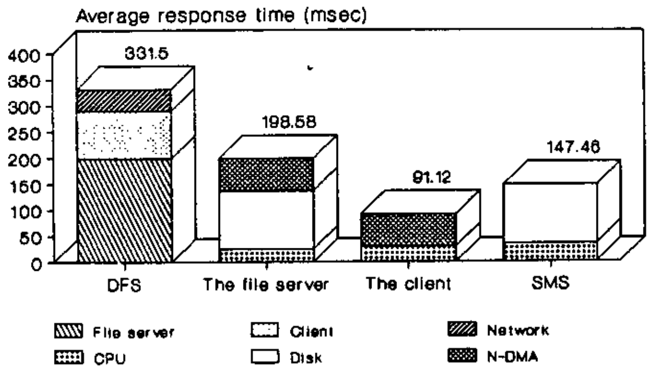


Figure 5.2.3.8 : S470, 50.7Kbytes

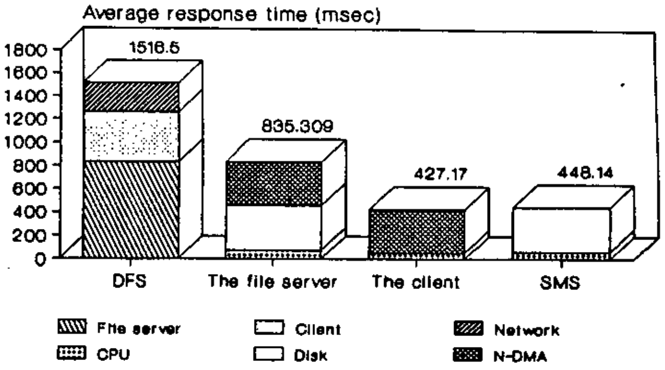


Figure 5.2.3.9 : S470, 316Kbytes

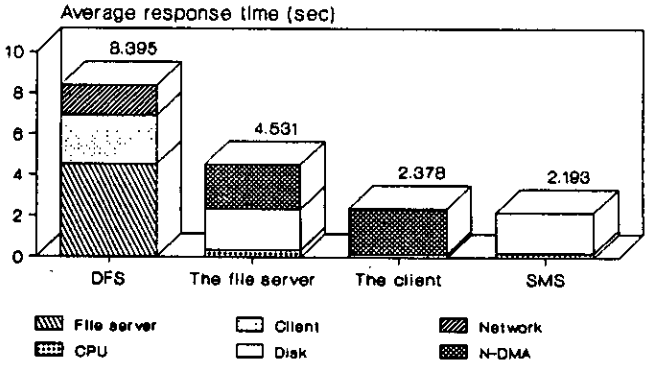


Figure 5.2.3.10 : S470, 1856Kbytes

The map of the average response time when there is no queueing delay in the distributed file system which consists of the Sun SPARCstation 470 workstations and in the Sun SPARCstation 470 workstation.

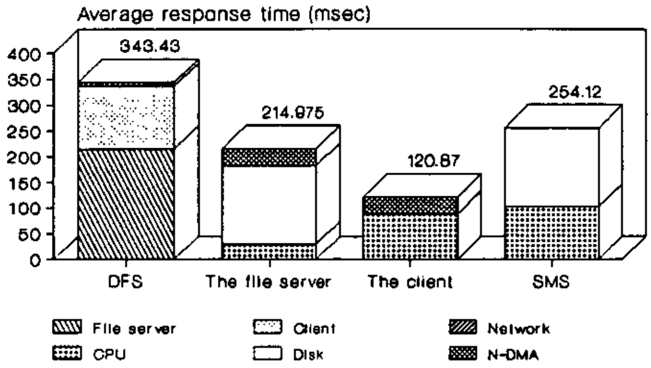


Figure 5.2.3.11 : S3/60, 8Kbytes

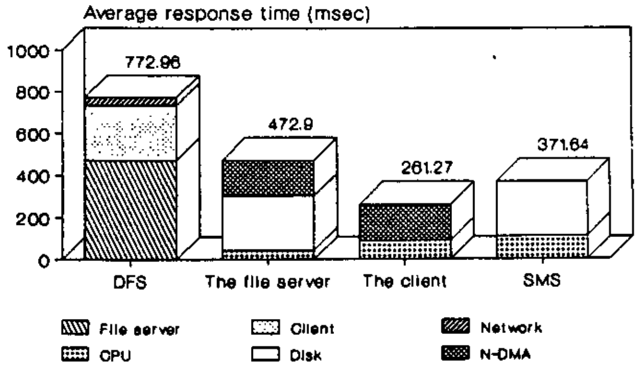


Figure 5.2.3.12 : S3/60, 47Kbytes

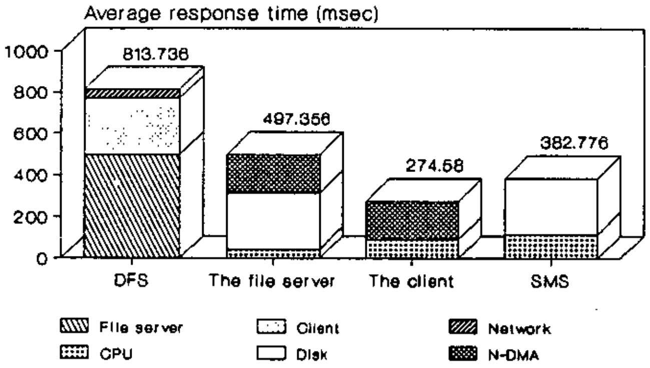


Figure 5.2.3.13 : S3/60, 50.7Kbytes

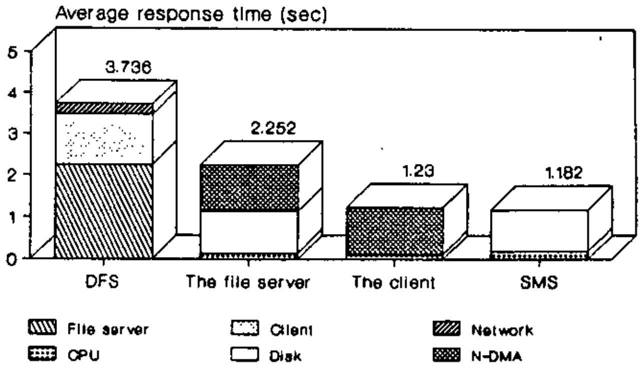


Figure 5.2.3.14 : S3/60, 316Kbytes

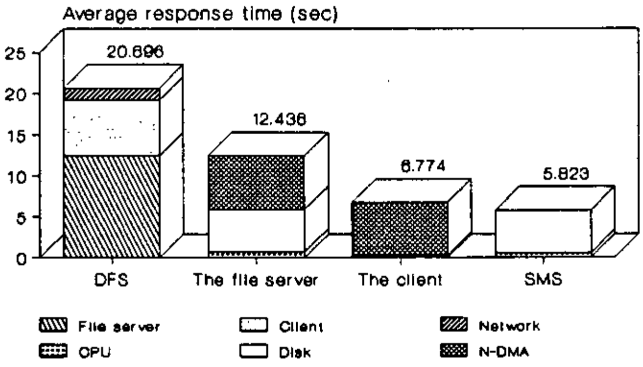


Figure 5.2.3.15 : S3/60, 1856Kbytes

The map of the average response time when there is no queueing delay in the distributed file system which consists of the Sun 3/60 workstations and in the Sun 3/60 workstation.

communication time spent in the network interface unit in the average time spent in the client increases, in other words, the percentage of the average CPU time in the average time spent in the client decreases, as the average transaction size increases. In the client of the distributed file system, the percentage of the average network communication time spent in the network interface unit in the average time spent in the client decreases as the power of the client grows.

In the file server of the distributed file system, the percentage of the average network communication time spent in the network interface unit in the average time spent in the file server increases as the average transaction size increases and the percentage of the average network communication time spent in the network interface unit in the average time spent in the file server decreases as the power of the file server grows.

In the distributed file systems, the percentage of the average data transmission time through the communication network in the total average response time increases as the average transaction size increases or as the power of the component system increases. This means that the main bottleneck point moves to the system resources related to network communication gradually as the power of the component system grows or the average transaction size grows in the distributed file systems.

5.3 The Two System Paradigms

One of the research objectives in this study is to compare the file access performances of the two different system paradigms. This study compares the file access performances of the design alternatives of the distributed file systems and those of the shared memory systems in the following sections. This section compares the file access performances of the baseline distributed file systems and the file access performances of the baseline shared memory systems in the environments which consist of the three different classes of Sun workstations respectively, which are the Sun SPARCstation 10 workstations, the Sun SPARCstation 470 workstations and the Sun 3/60 workstations.

The baseline performance model of figure 3.2.6.B and the baseline performance parameter values of table 3.2.7.C are used for the baseline distributed file systems and the baseline performance model of figure 3.4.1.B and the baseline performance parameter values of table 3.4.2.A are used for the baseline shared memory systems.

Figure 5.3.1 shows the average response time as the number of clients or the number of local users increases when this study uses the 50.7kbytes workload in the environments which consist of the SUN SPARCstation 10 workstations.

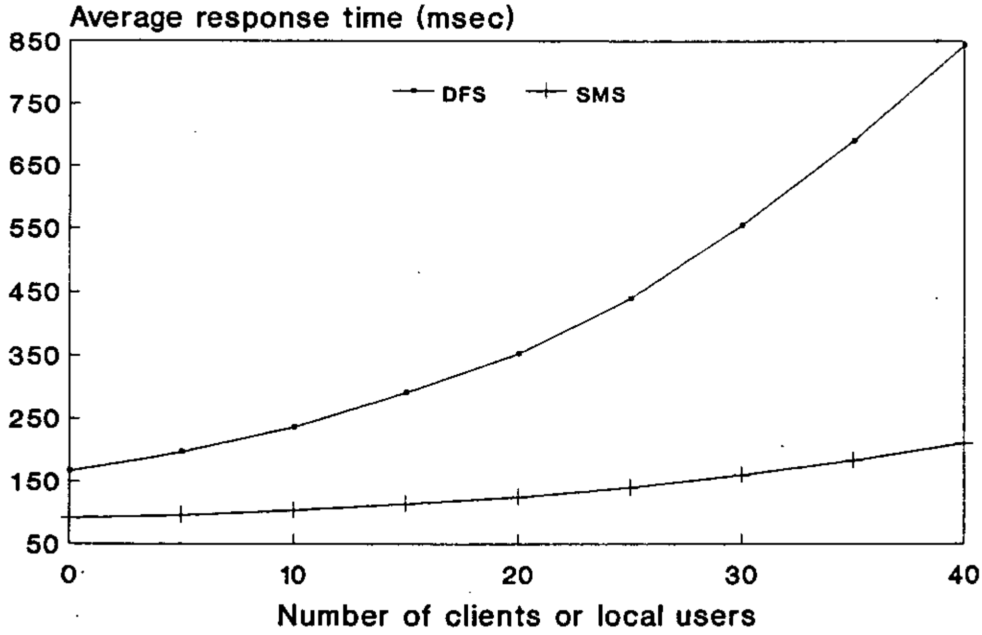


Figure 5.3.1 : The average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations vs. the average response time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

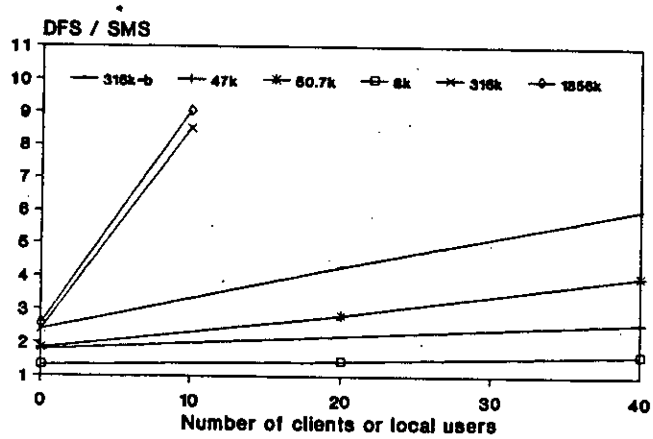


Figure 5.3.2 : S10, DFS/SMS

Figure 5.3.2 : The ratio of the average response time of the distributed file system which consists of the Sun SPARCstation 10 workstations and the average response time of the Sun SPARCstation 10 workstation when the 316kbytes(B) workload is used.

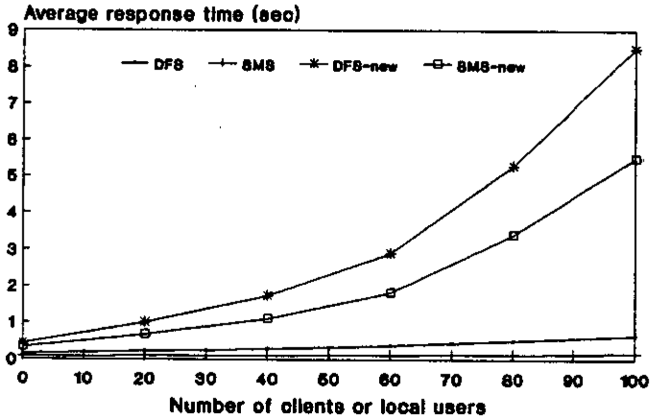


Figure 5.3.3 : S10, 47kbytes

Figure 5.3.3, Figure 5.3.4, Figure 5.3.5 The effect of the maximum burstiness on the average response time in the two system paradigms.

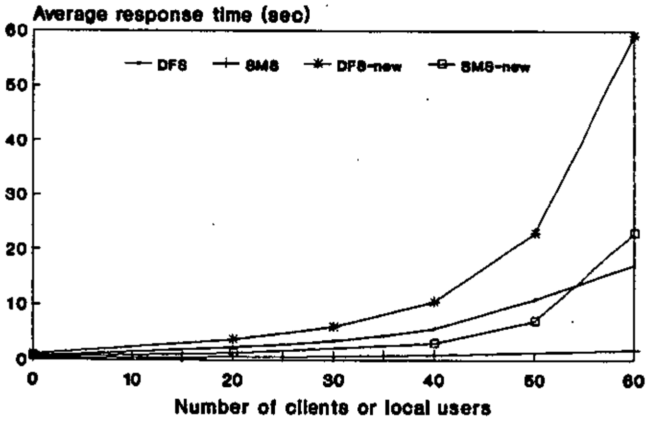


Figure 5.3.4 : S10, 316Kbytes(B)

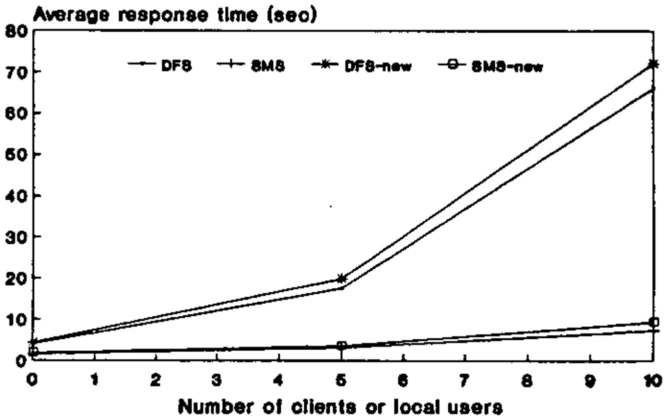


Figure 5.3.5 : S10, 1856Kbytes

The average response times show similar trends as the number of clients or the number of local users increases in all cases of the two different system paradigms when the 6 different workloads are used individually. See appendix B for the figures of other cases.

Figure 5.3.2 shows the ratio of the average response time of the distributed file system to the average response time of the local shared memory system. Generally when the average transaction size grows, the line of the ratio moves upward. The six lines, that is, the ratios of the six workloads show similar trends as the number of clients or the number of local users increases in the two different system paradigms.

Figure 5.3.3, figure 5.3.4 and figure 5.3.5 show the average response time as the number of clients or the number of the users increases when we use the bursty workloads such as the 47kbytes workload, the 316kbytes(B) workload, and the 1856kbytes workload respectively in the environments which consist of the SUN SPARCstation 10 workstations. In the figures, the used workloads have the maximum burstiness. For example, in the case of the 47kbytes workload, this study interprets the 47kbytes data traffic per second as the I/O traffic caused by the series of 8kbytes transactions when the intra-cluster-arrival time tends to zero in its limit, that is, there is no inter-arrival time gap inside the cluster.

5.4 Local Processing

This section investigates the effect of local processing on the file access performance. So far, the local processing has not been considered at all. In a job, some portions might be processed locally, that is, only in the client without receiving any service from the file server. During the local processing, the user might execute the CPU-bound portion or access local files or do both of them. When both the local processing and remote file access are done in a job, the total response time includes the local processing time as well and it may hide the slowness of the remote file access. This section investigates this effect by comparing the ratios of the average response time in the distributed file systems to the average response time in the local system. In figure 5.3.2, the ratio investigated was that between the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations and the average response time in the Sun SPARCstation 10 workstation.

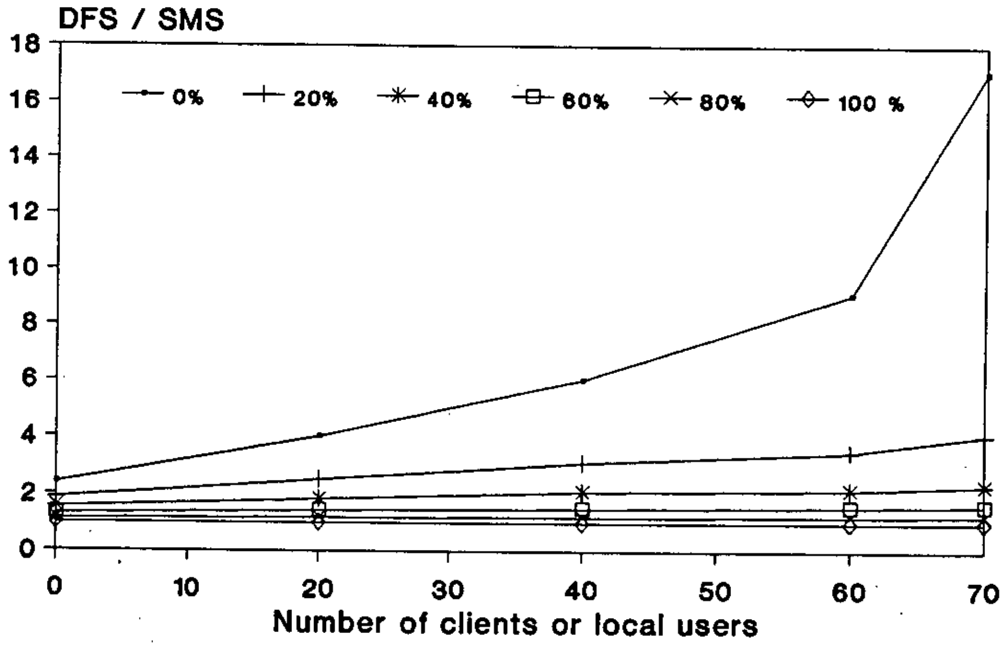


Figure 5.4.1 : The effect of the local processing on the ratio of the average response time of the distributed file system which consists of the Sun SPARCstation 10 workstations to the the average response time of the Sun SPARCstation 10 workstation when the 316Kbytes(B) workload is used.

The effect of local processing on the file access performance in terms of the ratio of the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations to the average response time in the Sun SPARCstation 10 workstation when the 316Kbytes(B) workload is used is shown in figure 5.4.1 as an example. At 0% local processing, the ratio shows the slowness of the average file access time in the distributed file system as it is, compared to the average file access time in the local system. At 20% local processing which means the percentage of the average response time of the remote file access in the total elapsed time of a job in the distributed file system is 80%, the ratio drops greatly. At 100% local processing the ratio is 1.

The effect of local processing on the file access performance is generalized in figure 5.4.2.

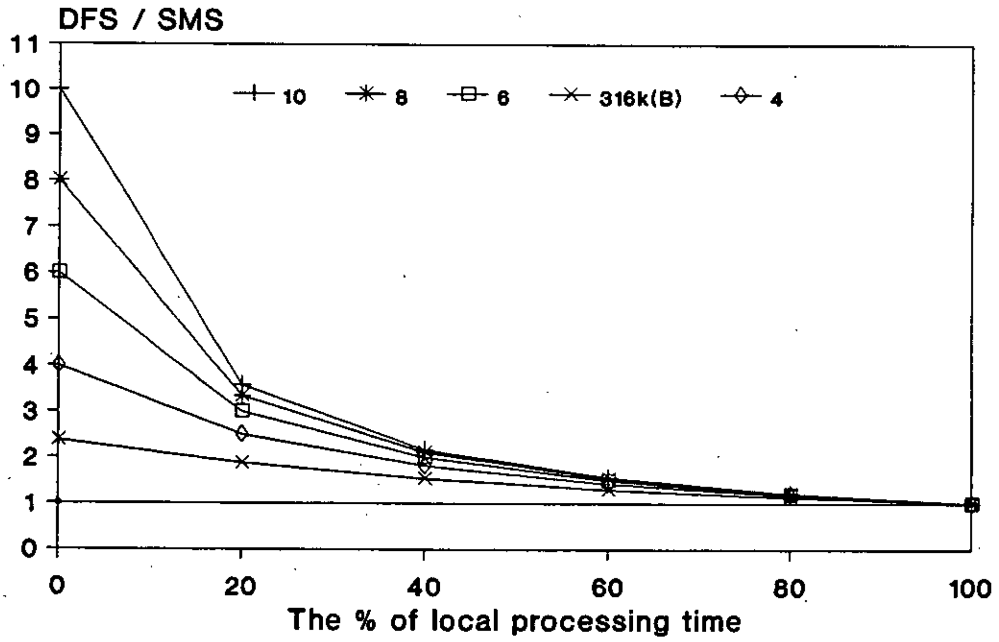


Figure 5.4.2 : The effect of the local processing on the ratio of the average response time of the distributed file system to the the average response time of the local system.

The figure shows the relationship between the percentage of local processing in a job and the ratio of the average response time in the distributed file system to the average response time in the local system. The line is obtained as the following.

$$\text{The ratio} = \frac{\textit{The total elapsed time in the distributed file system}}{\textit{The total elapsed time in the local system}}$$

$$= \frac{\textit{Local processing time} + \textit{The response time of remote file access}}{\textit{Local processsing time} + \textit{The response time of local file access}}$$

If it is assumed that X(1) = The response time of local file access, X(2) = The response time of remote file access, X(3) = Local processing time, then the ratio = $\frac{X(3)+X(2)}{X(3)+X(1)}$. By assuming that A=X(3)/X(2), B=X(2)/X(1), the ratio =

$\frac{A \times X(2)+X(2)}{A \times X(2)+X(2)/B} = \frac{X(2) \times (A+1)}{X(2) \times (A+1/B)} = \frac{B(A+1)}{(AB+1)}$. Therefore, in its limit, if [A] tends to 0, then the ratio tends to B=X(2)/X(1). In its limit, if [A] tends to infinity(∞), then the ratio tends to 1.

In figure 5.4.2, at 100% remote file access, that is, at 0% local processing in a job, the sought ratio becomes the ratio of the average response time of the remote file access in the distributed file system to the average response time of local file access in the local system. At 0% remote file access, that is, at 100% local processing, the ratio becomes one. The line 316k(B) uses the average response times of the 316Kbytes workload in the two system paradigms when there is no contention for the system resources as the initial ratio, i.e., the ratio when there is no local processing. In other lines, 4, 6, 8 and 10 are used as the initial ratios.

From figure 5.4.2, we can see that the ratio quickly decreases. For example, the initial ratio of 10 becomes 3.58 when the local processing time takes 20% of the total processing time in a job. Therefore, the slowness of the remote file access may be hidden from the users when the total response time is given to the users.

5.5 Summary

In the shared memory systems, the read and write show the same average response time all the time as we expect. In the distributed file systems, the average response time of write develops faster than the average response time of read as the contention for the system resources grows, that is, the number of clients increases. Throughout this chapter, chapter 6 and chapter 7, the write operation is used as the baseline file access operation unless the read operation is specified to be performed.

Generally the larger the average transaction size is, the better the average response time per 8kbyte data transferred is when there is no contention for the system resources. As the contention increases, that is, the number of clients or the number of local users increases, if the average the transaction size is larger, then the the average response time per 8k data transferred grows more quickly.

We can interpret bursty workloads in two ways : a fine grained workload with small inter-arrival times and a coarse grained workload with large inter-arrival times. The former shows worse average response time than the latter. This study uses the latter interpretation in the simulations.

If the measured utilization is closer to the theoretical limit, then the file access performance becomes better. In the distributed file systems, either the network interface unit or disk I/O subsystem is the major bottleneck point.

In the file server of the distributed file system, the percentage of the average network communication time spent in the network interface unit in the average time spent in the file server increases as the average transaction size increases and the percentage of the average network communication time spent in the network interface unit in the average time spent in the file server decreases as the power of the file server grows.

The main bottleneck gradually moves to the system resources related to network communication as the power of the component system grows or the average transaction size grows in the distributed file systems.

The average response times show similar trends as the number of clients or the number of local users increases in all cases of the two different system paradigms when the 6 different workloads are used individually.

The slowness of the remote file access may be hidden from the users when the total response time is given to the users.

Chapter 6

File Access Performance Evaluation of the Design Alternatives in the Two System Paradigms

This chapter investigates the file access performance of various design alternatives comparatively in the two system paradigms using the virtual performance models.

In the following sections in this chapter, the following conditions hold commonly used unless otherwise specified. Write file access is performed unless read file access is explicitly specified to be performed. No caching occurs unless caching is explicitly specified to occur. The workload pattern of the Poisson distribution for input arrival and the log-normal distribution for input transaction size is used unless the workload pattern used is specified. When this study enhances any mechanism or improves the power of any system resource or increases the number of any system resource, the Sun SPARCstation 10 workstation is used as the base system for the shared memory system and the distributed file system which consists of the Sun SPARCstation 10 workstations is used as the base distributed file system unless the base system is explicitly specified.

6.1 Multiple CPUs

This section investigates the effect on the file access performance by putting multiple CPUs in each component system of the distributed file systems and in the shared memory systems comparatively. The best multiple processing mechanism is dealt with so that the overhead to maintain multiple CPUs is assumed to be negligible and ignored. By adding CPUs to the file server, the file server system now becomes a shared memory multiprocessor system which uses the shared variable mechanism not the message passing mechanism, has a shared bus architecture and has the symmetric property as explained in section 3.3. No parallelism such as data parallelism or program parallelism is considered in this section. It is assumed that each CPU in the multiple processor system has equal opportunity to process incoming jobs, that is, the probability to be processed in each CPU is the same. The workstation or the system which has only one CPU is regarded as a special case of a shared memory system which has only one CPU. The performance model of figure 3.2.6.C and the baseline performance parameter values of table 3.2.7.C are used for the distributed file systems and the performance model of figure 3.4.1.B and the baseline parameter values of table 3.4.2.A are used for the shared memory systems. The multiple CPUs are represented as the service center which has multiple servers sharing a queue in the figures.

Figure 6.1.1 shows the average response time of the 50.7Kbytes workload in the distributed file system which consists of Sun SPARCstation 10 workstations where the CPUs are added as the number of clients increases gradually. Figure 6.1.2 shows the average response time of the 50.7Kbytes workload when the number of CPUs of the shared memory system is increased to be 2 CPUs, 4 CPUs, 8 CPUs, 10 CPUs, 16 CPUs, 20 CPUs, 24 CPUs, 26 CPUs and 30 CPUs. The base system to which the CPUs are added is the Sun SPARCstation 10 workstation in the figure, as in the distributed file system. See appendix C for the figures of other cases.

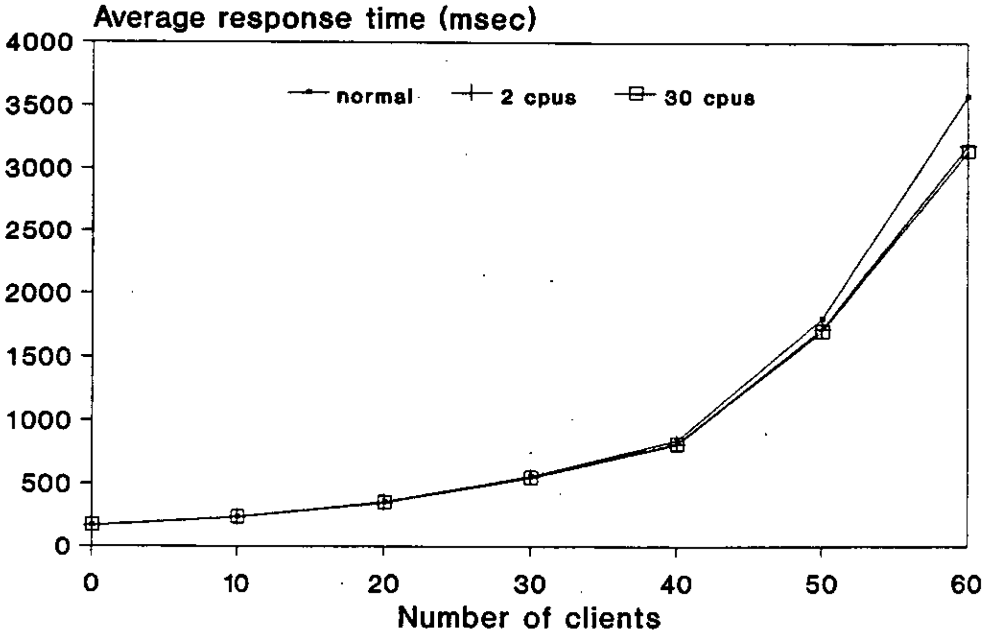


Figure 6.1.1 : The effect of having multiple CPUs on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

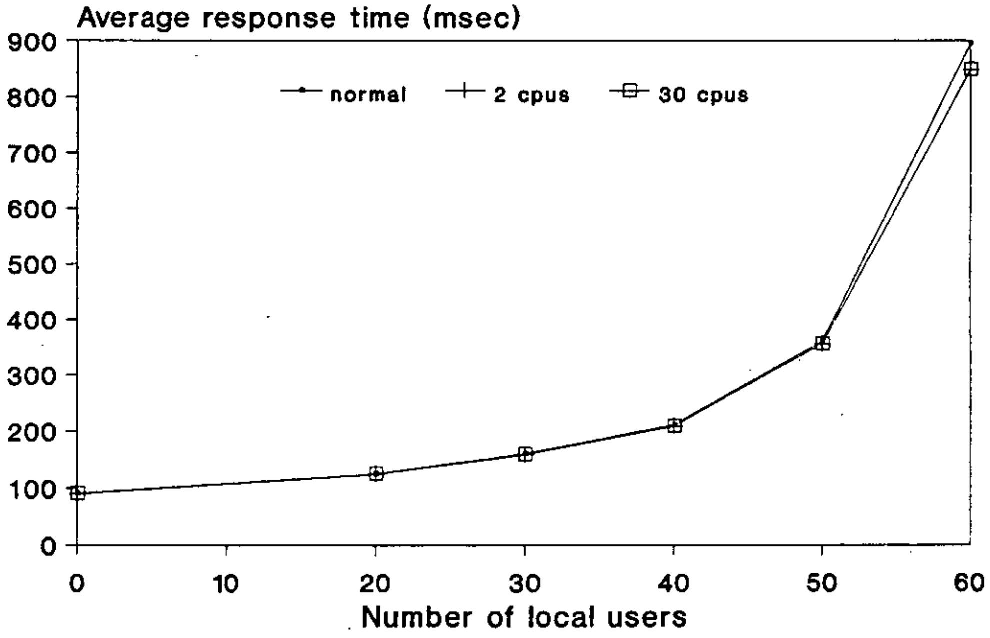


Figure 6.1.2 : The effect of having multiple CPUs on the average response time in the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

The CPU is not the major bottleneck point and the utilization of the CPU is low in both the distributed file systems and the shared memory systems. It means that the contention in the CPU is low. It is observed that the maximum improvement in the average response time by adding CPUs, that is, by getting rid of the queueing delay caused by the contention in the CPU, is small in percentage terms for the average response time of the baseline system in the two system paradigms. Both in the distributed file systems and in the shared memory systems, it is observed that 2 CPUs get rid of most of the queueing delay caused by the contention in the CPU and even though more CPUs are added to a system which already has 4 CPUs, the average response time improves very little regardless of the workload. It should be remembered that it is explained in section 5.2 that the ration of the consumed CPU time to the average response time in both the client and the file server decreases as the workload size grows when there is no contention for the system resources.

6.2 Better CPU

This section investigates the effect on the file access performance when the CPUs of the baseline distributed file system and the CPU of the baseline shared memory system are replaced with better CPUs. The system of which the CPU is replaced with K(2,4,8,...) times more powerful CPU in MIPS or MFLOPS or SPECrate or any other performance benchmarking, does not necessarily produce K(2,4,8,...) times better CPU parameter values. If some processing mechanisms in the baseline systems are enhanced, the values of related CPU parameters might also be reduced. In this study, the K(2,4,8,..) times better CPU means that the values of all CPU parameters are improved to be K(2,4,8,...) times better at the same time, which is a theoretical assumption. The effect on the file access performance when the value of each CPU parameter is improved individually will be investigated in section 6.13. This section investigates the effect on the file access performance when the values of all the CPU parameters are improved together at the same time.

The performance model of figure 3.2.6.B is used for the distributed file systems and the performance model of figure 3.4.1.B is used for the shared memory systems. Both the baseline distributed file system and the baseline shared memory system consist of the Sun SPARCstation 10 workstations. The homogeneity is kept in the distributed file systems by replacing the CPU with the better CPU both in the client and in the file server at the same time.

The following parameters are the CPU parameters in the distributed file system. In the client, they are the command interpretation parameter, the RPC request build parameter and the RPC response evaluation parameter whose values are constant for the transaction size and the request send parameter, the response receive parameter and the result processing parameter whose values are proportional to the transaction size. In the file server, they are the file handling parameter, the RPC request evaluate parameter and the RPC response build parameter whose values are constant for the transaction size and the request receive parameter, the response send parameter and the parameter of the CPU service for the disk I/O of which the values are proportional to the transaction size. The CPU parameters in the shared memory system are the command interpretation parameter and the file handling parameter whose values are constant to the transaction size and the parameter of the CPU service for the disk I/O and the result processing parameter whose values are proportional to the transaction size.

Figure 6.2.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. In the simulations, the CPUs of the baseline distributed file system which consists of the Sun SPARCstation 10 workstations are replaced by the 2 times, 4 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better CPUs. Figure 6.2.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The CPU of the baseline Sun SPARCstation 10 workstation is replaced by a 2 times, 4 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better CPU individually. See appendix C for the figures of other cases.

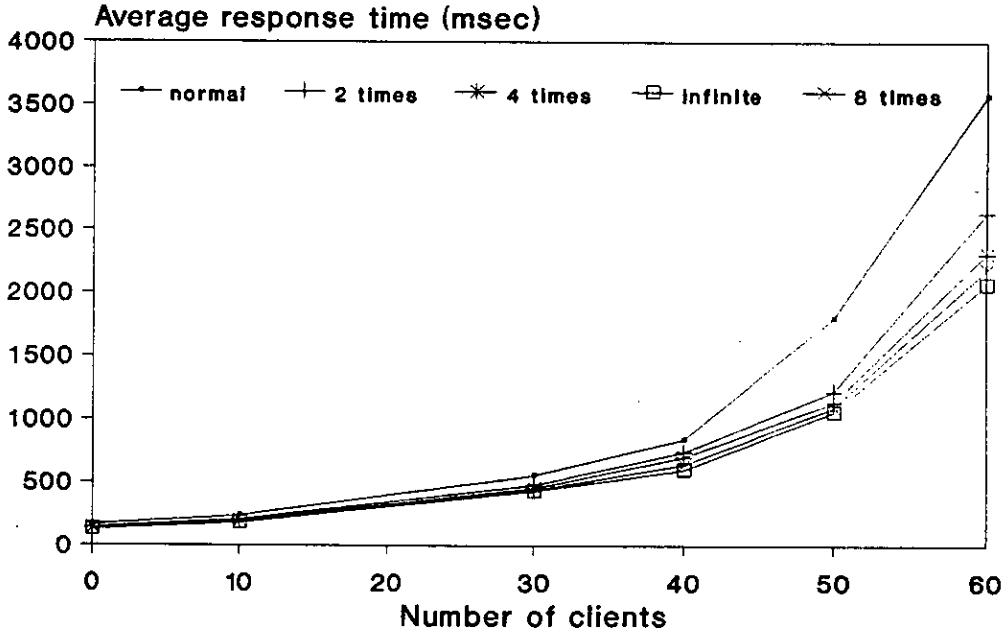


Figure 6.2.1 : The effect of the better CPU on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

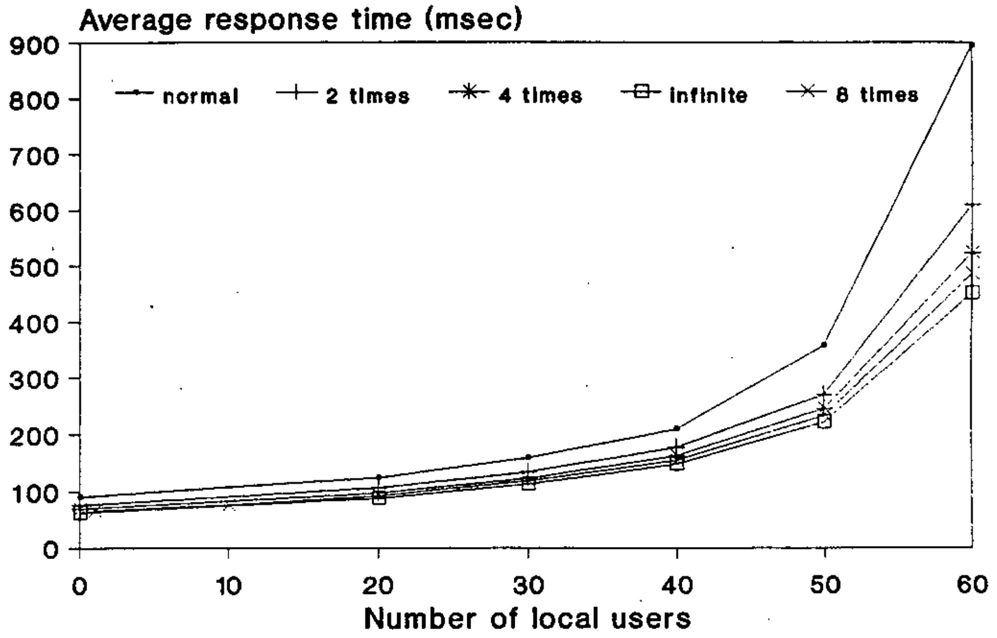


Figure 6.2.2 : The effect of the better CPU on the average response time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

Since the contention in the CPU is low, the overall improvement of the average response time in the distributed file systems and in the shared memory systems is not significant. There are many CPU parameters and the CPU is often called for service but because the amount of service requested is small the contention for the CPU is not high and the utilization of the CPU is low.

It is observed that the system which has a 2 times better CPU produces somewhat better average response time in the two system paradigms but beyond a 4 times better CPU, the average response time of the system improves very little. Similar patterns and characteristics are observed in both system paradigms. In the next section, the file access performance of the better CPU case is compared with that of the equivalent multiple CPUs case in detail.

6.3 Multiple CPUs vs. Better CPU

This section compares the file access performance of the better CPU case and that of the equivalent multiple CPU case. In order to compare them fairly, the improvement of the CPU power is limited to the file server, that is, the CPU of the file server is replaced with the better CPU but not the CPUs in the clients of the distributed file system. Now the distributed file system becomes heterogeneous. In section 6.2, both the CPU of the file server and the CPUs of the clients were replaced with better CPUs to maintain homogeneity.

Table 6.3.1 to table 6.3.5 compare the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the two different cases of the distributed file system as the number of clients increases gradually. The 2 times better CPU case and the 2 CPUs case are compared in the tables.

Table 6.3.6 to table 6.3.10 compare the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the two different cases of the shared memory system as the number of local users increases gradually. The 2 times better CPU case and the 2 CPUs case are compared in the tables.

It is observed that the average response time of the system which has the K(2,4,8,,,,) time better CPU is better than that of equivalent system which has K(2,4,8,...) CPUs both in the distributed file system and in the shared memory system. And as the contention for the system resources of the file server in the distributed file system grows, the difference between the average response time of the better CPU case and that of the equivalent multiple CPUs case becomes larger

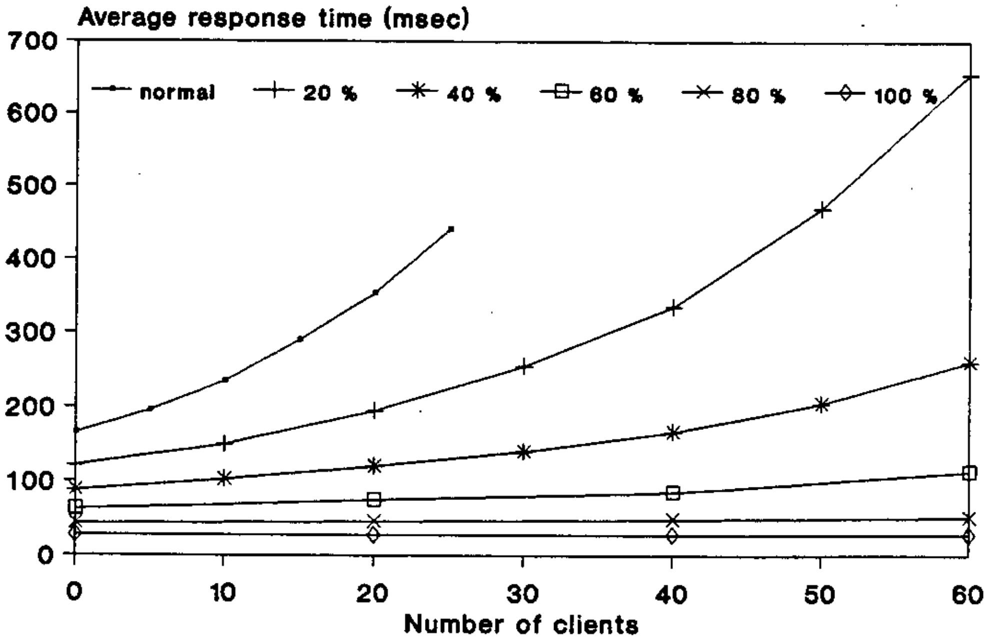


Figure 7.8.1 : The effect on the average response time when we use the combination of caching in the memory of the client, caching in the memory of the file server and caching in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

Figure 7.8.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in the three caches at the same time. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

In the distributed file system, regular improvement in the average response time is observed as the cache hit rate increases since all queueing delays gradually disappear at the same rate as the cache hit rate increases. The saturation point increases significantly as the cache hit rate increases.

7.8 Combination of Caching in the Memory of the Client, Caching in the Memory of the File Server and Caching in the Disk Interface Unit of the File Server

This section investigates the effect on file access performance when we use the combination of caching in the memory of the client, caching in the memory of the file server and caching in the disk interface unit of the file server in the distributed file system.

In this combination, the requests from the client are screened first by the cache in the memory of the client, second by the cache in the memory of the file server and third and last by the cache in the disk interface unit of the file server. If the requested data are in the memory of the client, then the data are fetched for the response and the remaining operations are bypassed. Therefore the network communication cost and all costs in the file server are saved as explained in section 7.3. The utilization of the CPU, the disk interface unit, the disk and the network interface unit of the file server and the network are reduced. If the requested data are not found in the memory of the client but found in the memory of the file server, then the cost of all disk I/O operations are saved as explained in section 7.1. The utilization of the CPU, the disk interface unit and the disk of the file server are reduced. If the requested data are not found in the cache in memory of the client and not in the cache in the memory of the file server but found in the cache in the disk interface unit of the file server, then the cost of the operations for I/O in the disk interface unit and the disk is saved as explained in section 7.2. The utilization of the disk interface unit and the disk of the file server are reduced.

in general. This is also observed in the shared memory system.

The better CPU cases use theoretically better CPUs which improve the values of all CPU parameters at the same time. If this study were tp compare the system where the CPU is replaced with the K(2,4,8,...) time more powerful CPU in MIPs, MFLOPS, etc., with the system where the K(2,4,8,...) CPUs are used in both system paradigms, the difference between the average response time of the better CPU case and that of the equivalent multiple CPUs case would be much less and even the multiple CPUs case might be better than the better CPU case in the average response time.

	The number of clients					
	0	20	40	60	80	100
2 CPUs	73.33	86	106.4	144	194.8	278.1
2 Times Better CPU	57.28	69.09	88.07	123.7	170	244.7

Table 6-3-1 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the distributed file system which consists of the Sun SPARCstation 10 workstations when the 8Kbytes workload is used.

	The number of clients							
	0	20	40	60	80	100	200	300
2 CPUs	157.8	217.5	285.8	381.8	507.8	644.1	2286	5984
2 Times Better CPU	150.27	204.47	267.97	354.67	474.27	597.37	2090.1	5254.1

Table 6-3-2 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the distributed file system which consists of the Sun SPARCstation 10 workstations when the 47Kbytes workload is used.

	The number of clients			
	0	20	40	60
2 CPUs	165.8	347.9	815.7	3181
2 Times Better CPU	157.88	317.2	753.31	2647.3

Table 6-3-3 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the distributed file system which consists of the Sun SPARCstation 10 workstations when the 50.7Kbytes workload is used.

	The number of clients			
	0	20	40	60
2 CPUs	740.7	2265	5541	15790
2 Times Better CPU	711.9	2105	5097.2	14094

Table 6-3-4 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the distributed file system which consists of the Sun SPARCstation 10 workstations when the 316kytes(B) workload is used.

	The number of clients			The number of clients		
	0	5	10	0	5	10
2 CPUs	740.7	3274	13260	4078	17400	62700
2 Times Better CPU	711.875	2650	11973.16	3927.25	16010.95	57200.95

Table 6-3-5 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the distributed file system which consists of the Sun SPARCstation 10 workstations when the 316Kbytes workload is used and the 1856Kbytes workload is used.

	The number of clients					
	0	20	40	60	80	100
2 CPUs	55.67	59.72	65.39	76.11	91.04	118.1
2 Times Better CPU	42.835	46.72	52.13	61.2	76.07	100.9

Table 6-3-6 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the Sun SPARCstation 10 workstation when the 8Kbytes workload is used.

	The number of clients							
	0	20	40	60	80	100	200	300
2 CPUs	88.17	97.55	110.7	125.6	144.1	169	390.2	1028
2 Times Better CPU	73.712	82.1	94.14	107.5	123.4	146	337.4	847.9

Table 6-3-7 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the Sun SPARCstation 10 workstation when the 47Kbytes workload is used.

	The number of clients			
	0	20	40	60
2 CPUs	91.25	124.8	209	848.9
2 Times Better CPU	76.64	106.9	179	607.4

Table 6-3-8 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the Sun SPARCstation 10 workstation when the 50.7Kbytes workload is used.

	The number of clients			
	0	20	40	60
2 CPUs	312.3	535.5	935	1886
2 Times Better CPU	288.5	484.9	825.3	1579

Table 6-3-9 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the Sun SPARCstation 10 workstation when the 316Kbytes(B) workload is used.

	The number of clients			The number of clients		
	0	5	10	0	5	10
2 CPUs	312.3	666.9	1645	1596	3152	7325
2 Times Better CPU	283.475	597.8	1392	1506.17	2868	6302

Table 6-3-10 : The average response time(msec) of the two times better CPU case vs. the average response time of the two CPUs case in the Sun SPARCstation 10 workstation when the 316Kbytes workload is used and when the 1856Kbytes workload is used.

6.4 Multiple Disk I/O Subsystems

This section investigates the effect on the file access performance when multiple disks and multiple disk interface units are used both in the distributed file system and in the shared memory system comparatively. In both system paradigms, the Sun SPARCstation 10 workstations are used.

The performance model of figure 3.2.6.D and the baseline performance parameter values of table 3.2.7.C are used for the distributed file system and the performance model of figure 3.4.1.D and the baseline performance parameter values of table 3.4.2.A are used for the shared memory system. The multiple disks and multiple disk interface units are represented as multiple tandem servers which share a queue in the performance models. Each disk interface unit is assumed to receive I/O requests with equal opportunity since the multiple disk I/O subsystems are assumed to have the symmetric property. The overhead to manage the multiple disks and the multiple disk interface units is assumed to be negligible, which means this study considers the theoretical limit.

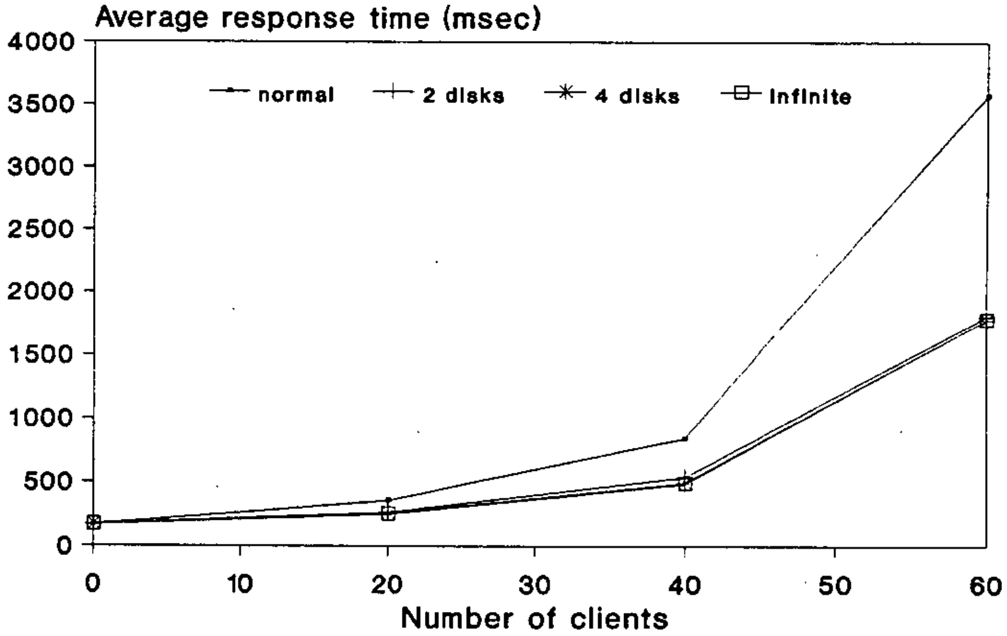


Figure 6.4.1 : The effect on the average response time of having multiple disk I/O subsystems in the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

Figure 6.4.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The number of disks and the number of disk interface unit in the file server are increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity at the same time. Except the disk and the disk interface unit in the file server, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. Figure 6.4.2 shows the average response time of the 50.7Kbytes workload respectively in the shared memory system as the number of local users increases gradually. The number of disks and the number of disk interface units are increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity at the same time. Except the disks and the disk interface units, all others are kept the same as the baseline Sun SPARCstation 10 workstation. See appendix C for the figures of other cases.

Since the contention in the disk I/O subsystem is high and the disk I/O subsystem is one of the two major bottleneck points, the overall improvement of the average response time both in the distributed file system and in the shared memory system is significant. It is observed that the average response time significantly improves in the system which has 2 disks and 2 disk interface units and in the system which has 4 disks and 4 disk interface units the average response time still improves but the improved amount of the average response time is not twice as much as that in the system which has 2 disks and 2 disk interface units in both system paradigms. In the system which has 4 disks and 4 disk interface units, most of contention in the disk I/O subsystem disappears and the network interface unit, the next busiest resource, now becomes the major bottleneck point and dominates the queueing delay. Therefore, putting more than 4 disks and 4 disk interface units in the file server of the baseline distributed file system is not efficient in terms of the performance/cost. In the system which has multiple disks and multiple disk interface units, the saturation point, that is, the maximum supportable number of clients, does not significantly increase since the saturation point of the network interface unit is a little larger than that of the disk I/O subsystem.

In the shared memory system, when the system has more than 4 disks and 4 disk interface units, the bottleneck point is now the CPU of which the saturation point is very large. Therefore, as disk and disk interface unit are added to the baseline system one by one, the saturation point almost linearly increases.

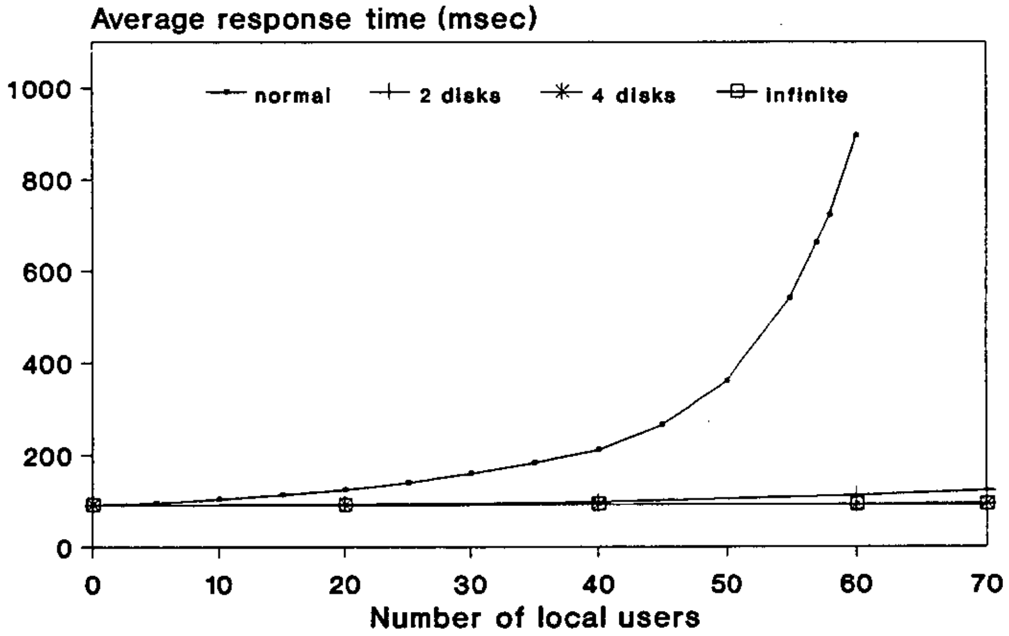


Figure 6.4.2 : The effect on the average response time of having multiple disk I/O subsystems in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

6.5 Better Disk I/O Subsystem

6.5.1 Reduced Disk I/O Time

This section investigates the effect on the file access performance when only the disk I/O time is improved comparatively in the two system paradigms. What can improve the disk I/O time? Faster disks, disk arrays, striping mechanism, disk interface units which have faster data transfer rates, etc. can improve the disk I/O time. See the work by Wood and Hodges[WOOD etal 93] for the trend of DASD performance. This section does not investigate in detail the methods to reduce the disk I/O time but investigates the effect on the file access performance when the disk I/O time is improved in the two different system paradigms comparatively.

The disk I/O time has not improved as much as the system power has increased as we can see in table 3.2.7.C. The ratio of the disk I/O time which is constant for transaction size in the Sun SPARCstation 10 workstation to the disk I/O time which is constant for transaction size in the Sun SPARCstation 470 to the disk I/O time which is constant for transaction size in the Sun 3/60 workstation is 1:3:6 and the ratio of the disk I/O time which is proportional to the transaction size is 1 : 1.37 : 3.67. They are far below the inverse of the power ratio in MIPS of the three systems, which is 1 : 7.34 : 33.87.

Figure 6.5.1.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. Both the constant portion and the proportional portion of the disk I/O time are improved to be 2, 4, 8, 10, 16, 20, 30, 100, 1000 times and infinitely faster. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The baseline performance model of figure 3.2.6.B is used for the distributed file system. See appendix C for the figures of other cases.

Figure 6.5.1.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. Both the constant portion and the proportional portion of the disk I/O time are improved to be 2, 4, 8, 10, 16, 20, 30, 100, 1000 times and infinitely faster. Except for these, all others are kept the same as the baseline Sun SPARCstation 10 workstation. The baseline performance model of figure 3.4.1.B is used for the shared memory system. See appendix C for the figures of other cases.

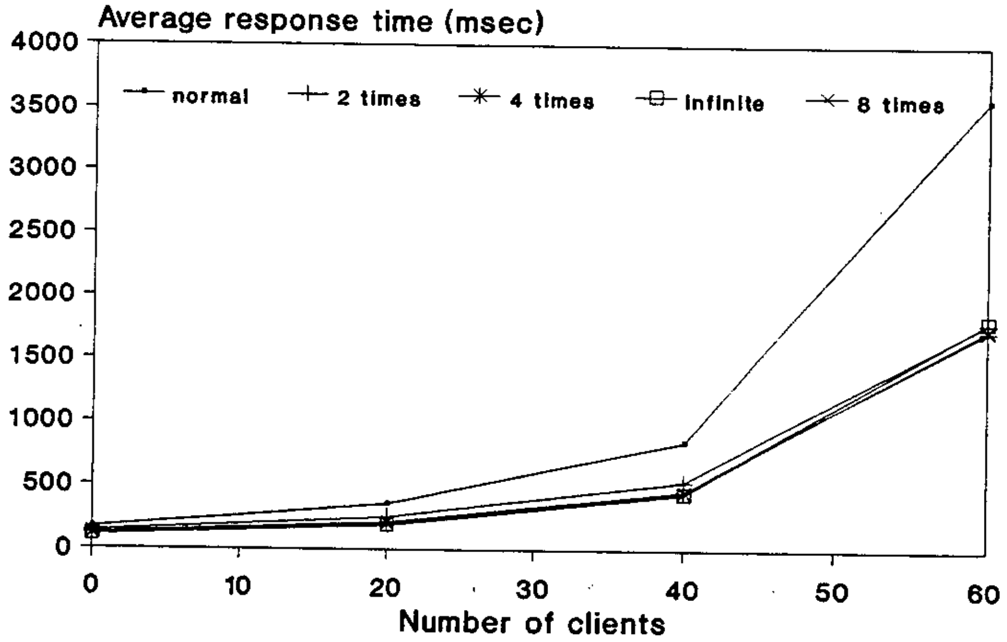


Figure 6.5.1.1 : The effect of having the better disk I/O subsystem on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

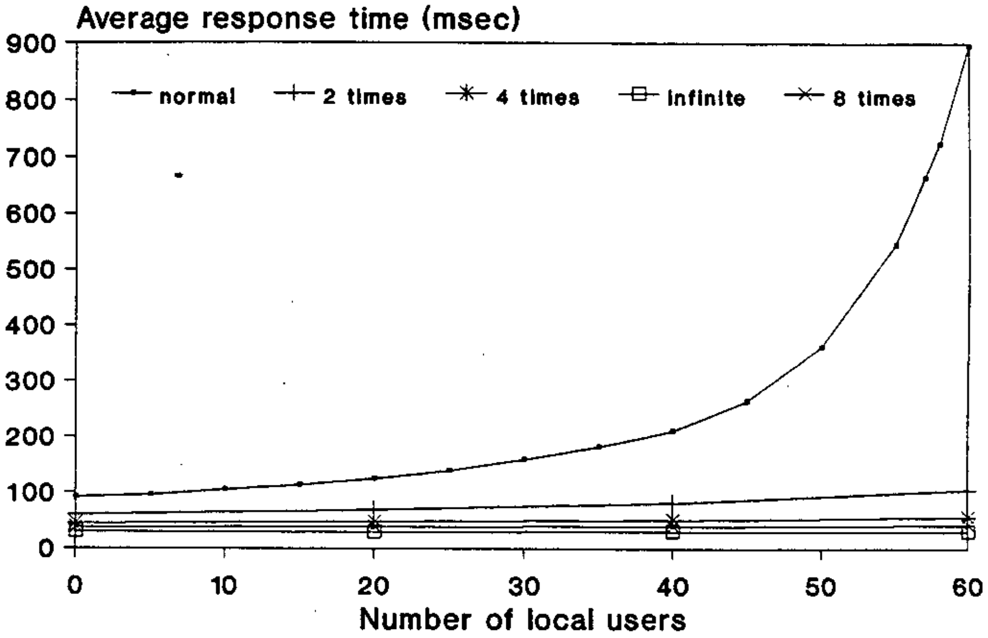


Figure 6.5.1.2 : The effect of having the better disk I/O subsystem on the average response time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

The overall improvement in the average response time in both the distributed file system and the shared memory system is significant. It is observed that the 2 times faster disk I/O improves the average response time significantly in both system paradigms. When the speed of the disk I/O is doubled each time, it is found that the improvement rate of the average response time decreases gradually in the distributed file system, that is, the average response time does not linearly improve in the distributed file system. Even though the disk I/O time improves further beyond 8 times faster in the distributed file system, the average response time does not improve further.

In the shared memory system, when the speed of the disk I/O is doubled each time, it is found that the average response time almost linearly improves until the CPU becomes the major bottleneck point compared to the distributed file system. The six different workloads produce similar patterns for the average response time in both system paradigms.

6.5.2 Other improvements

The path setup, the disk connection, the interference for data transfer, etc. require CPU service. There are two kinds of disk I/O overheads : the disk I/O service time and the CPU service time for disk I/O. The previous section investigated the effect of improving the disk I/O service time on the file access performance. All other improvements in the disk I/O operations which lead to the improvement of the CPU service time are covered in this section. What can reduce the CPU service time for disk I/O? This section does not investigate how to reduce the CPU service time for disk I/O but investigates the effect on the file access performance comparatively in the two different system paradigms when the CPU service time for disk I/O is improved.

The ratio of the CPU service time for disk I/O parameter value in the Sun SPARCstation 10 workstation to that in the Sun SPARCstation 470 workstation to that in the Sun 3/60 workstation is 1: 1.2 : 3.2. They are far below the inverse of the power ratio in MIPS of the three systems, which is 1 : 7.34 : 33.87.

Figure 6.5.2.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. In each figure, the CPU service time parameter value for disk I/O is improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun

SPARCstation 10 workstations. The baseline performance model of figure 3.2.6.B is used for the distributed file system. Figure 6.5.2.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The CPU service time parameter value for disk I/O is improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline Sun SPARCstation 10 workstation. The baseline performance model of figure 3.4.1.B is used for the shared memory system. See appendix C for the figures of other cases.

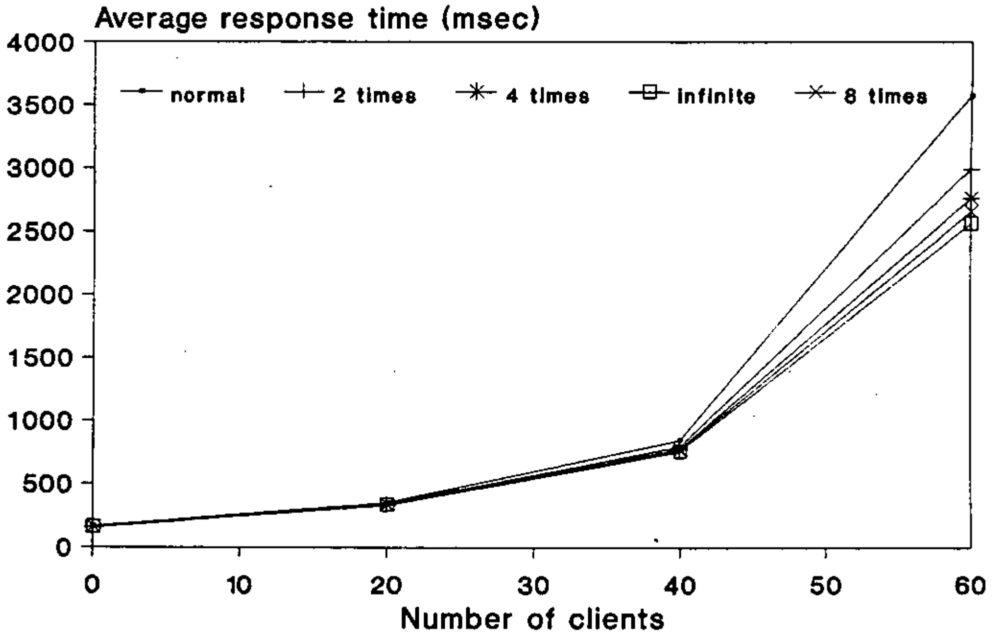


Figure 6.5.2.1 : The effect of the improved CPU service time for disk I/O on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

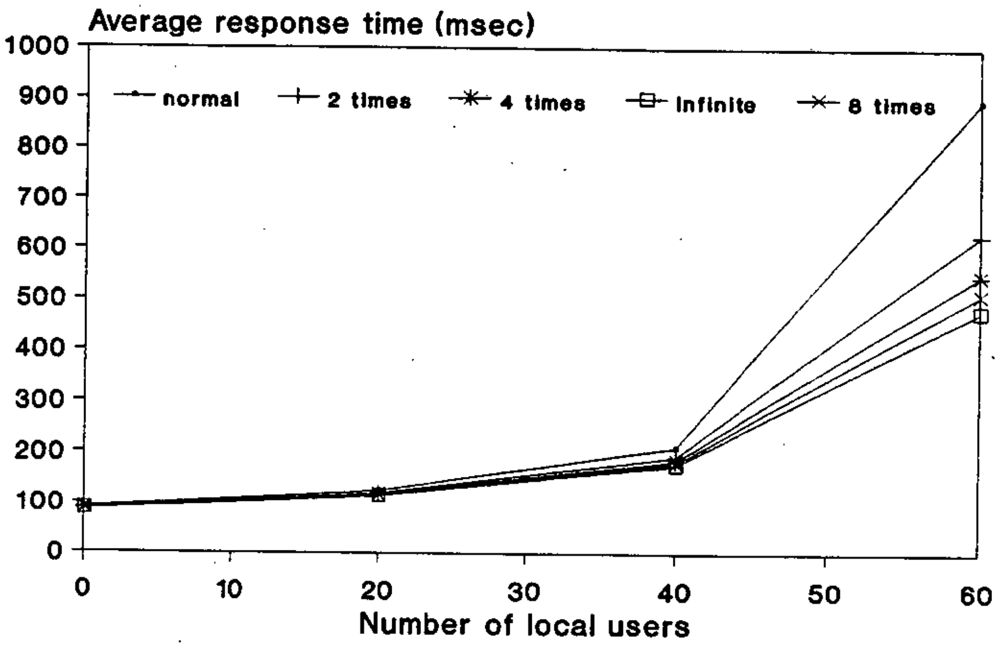


Figure 6.5.2.2 : The effect of the improved CPU service time for disk I/O on the average response time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

The overall improvement of the average response time in the distributed file system and in the shared memory system is not significant, as we expect. It has to be recalled that in section 6.2 it was already found that when the values of all CPU parameters were improved in the baseline systems, the average response time does not improve significantly. The CPU service time parameter for disk I/O is one of the CPU parameters. If we further improve the parameter value which was already improved to be 4 times better, then the improvement in the average response time is trivial. This is observed both in the distributed file system and in the shared memory system. The six workloads produce similar patterns for the average response times in both system paradigms.

6.5.3 All Improvements at the Same Time

This section investigates the effect on the file access performance comparatively in the two different system paradigms when all parameters values for disk I/O are improved at the same time. The parameters for disk I/O are the CPU service time parameter for disk I/O and the disk I/O parameter as explained earlier.

Figure 6.5.3.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The values of all parameters for disk I/O are improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The baseline performance model of figure 3.2.6.B is used for the distributed file system. See appendix C for the figures of other cases.

Figure 6.5.3.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The values of all parameters for disk I/O are improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline Sun SPARCstation 10 workstation. The baseline performance model of figure 3.4.2.B is used for the shared memory system. See appendix C for the figures of other cases.

Since the contention in the disk I/O subsystem is high and the disk I/O subsystem is one of the major bottleneck points, the overall improvement of the average response time in both the distributed file system and the shared memory system is significant.

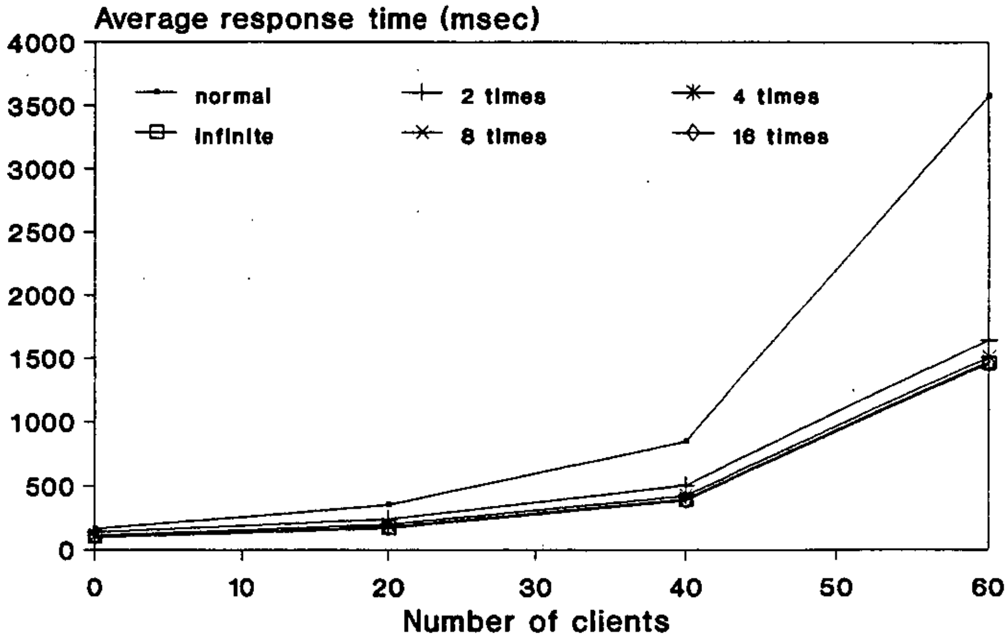


Figure 6.5.3.1 : The effect on the average response time when the values of all parameters for disk I/O are improved at the same time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

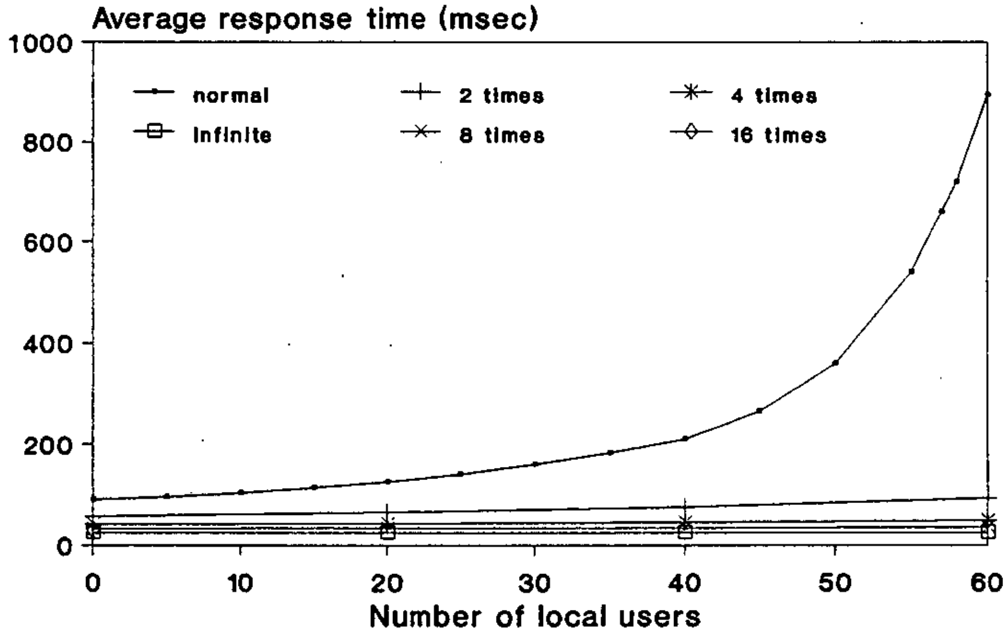


Figure 6.5.3.2 : The effect on the average response time when the values of all parameters for disk I/O are improved at the same time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

It is observed that in the 2 times better cases the average response time improves significantly but the improvement rate of the average response time decreases as the degree of improvement increases in the distributed file system even though the average response time improves as far as all the parameter values for disk I/O improve. If we further improve the parameter values which were already improved to be 8 times better, the average response time improves very little in the distributed file system. The reason is because the 8 times better case already gets rid of the most of the contention for the disk I/O subsystem and the network interface unit, one of the busiest resources, now becomes the major bottleneck point and dominates the queueing delay in the distributed file system. The saturation point does not significantly increase or does not increase at all since the saturation point of the network interface unit is a little larger or a little smaller than that of the disk I/O subsystem according to the workload.

In the baseline shared memory system, when we continue to double all parameter values of the disk I/O each time, almost linear improvement rate of the average response time is observed until the CPU becomes the major bottleneck point in contrast to the distributed file system. Except for this characteristic, the average response time in the shared memory system follows the same pattern as that in the distributed file system.

The six different workloads produce similar patterns for the average response times in both system paradigms.

6.6 Multiple Disk I/O Subsystems vs. Better Disk I/O Subsystem

This section compares the file access performance of the faster disk I/O subsystem cases of section 6.5.1 and the file access performance of the equivalent multiple disk I/O subsystem cases of section 6.4 in detail. The CPU service time for disk I/O is kept unchanged and only the disk I/O time in the disk I/O subsystem is improved.

Figure 6.6.1 to figure 6.6.6 compare the average response time in the distributed file system where the disk I/O time is improved to be two times faster and the average response time in the distributed file system which has two disks and two disk interface units when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively and the number of clients increases gradually.

Figure 6.6.7 to figure 6.6.12 compare the average response time in the distributed file system where the disk I/O time is improved to be four times faster and the average response time in the distributed file system which has four disks and four disk interface units when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively and the number of clients increases gradually.

Figure 6.6.13 to figure 6.6.18 compare the average response time in the shared memory system where the disk I/O time is improved to be two times faster and the average response time in the shared memory system which has two disks and two disk interface units when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively and the number of local users increases gradually.

Figure 6.6.19 to figure 6.6.24 compare the average response time in the shared memory systems where the disk I/O time is improved to be four times

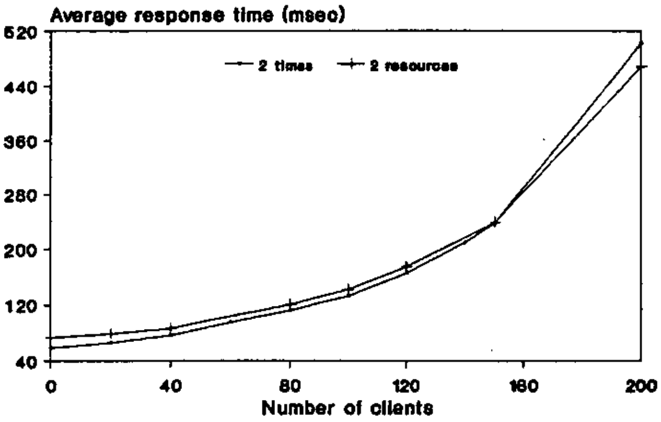


Figure 6.6.1 : 8Kbytes

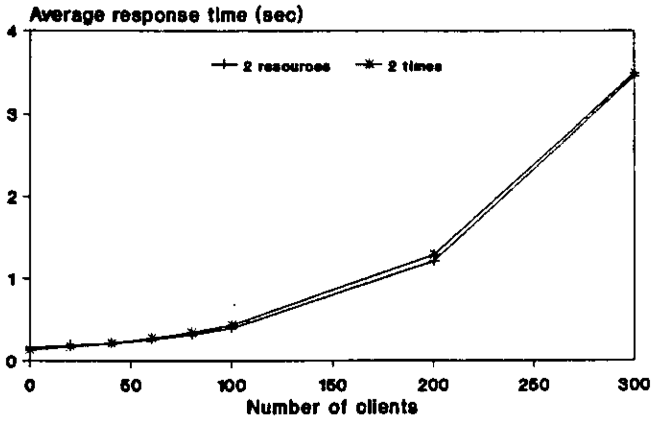


Figure 6.6.2 : 47Kbytes

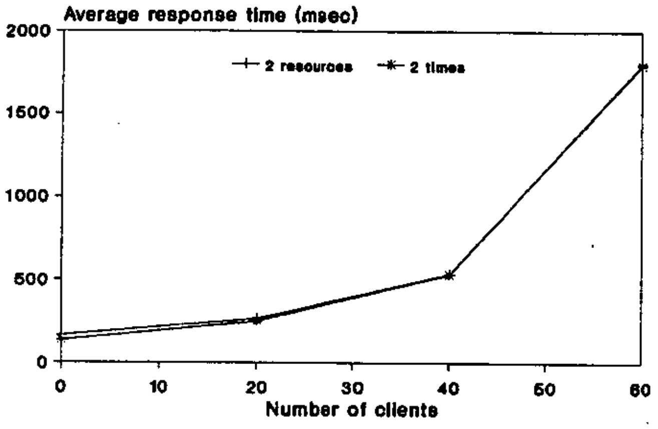


Figure 6.6.3 : 50.7Kbytes

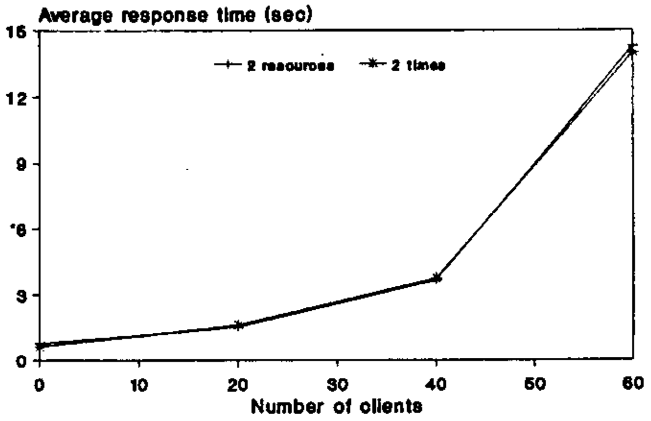


Figure 6.6.4 : 316Kbytes(B)

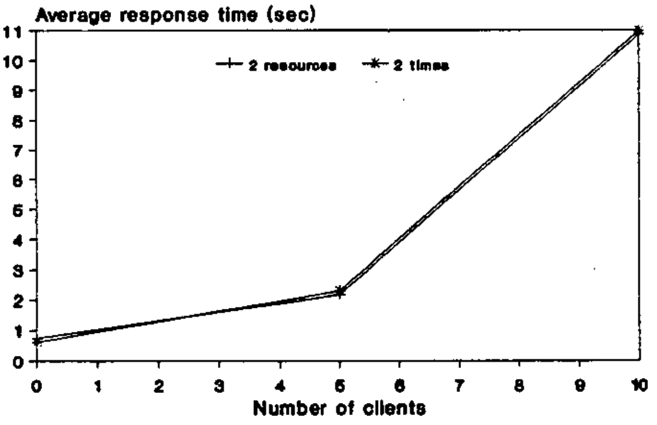


Figure 6.6.5 : 316Kbytes

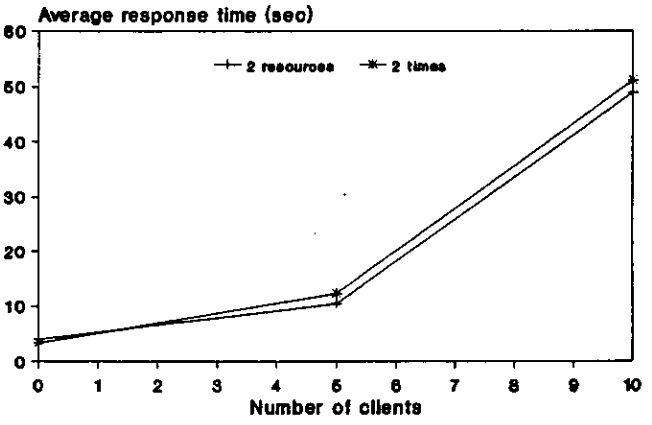


Figure 6.6.6 : 1856Kbytes

The average response time of the case of the two times better disk I/O time vs. the average response time of the case of the two disk I/O subsystems in the distributed file system which consists of the Sun SPARCstation 10 workstations.

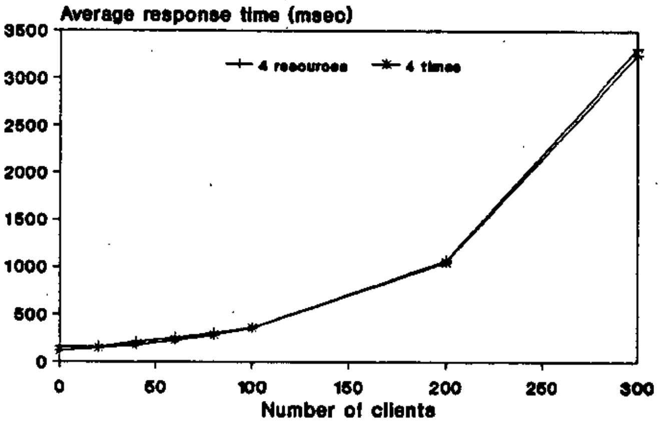


Figure 6.6.7 : 8Kbytes

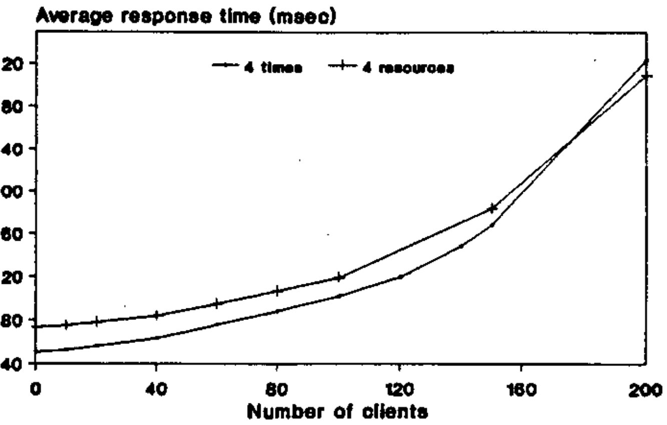


Figure 6.6.8 : 47Kbytes

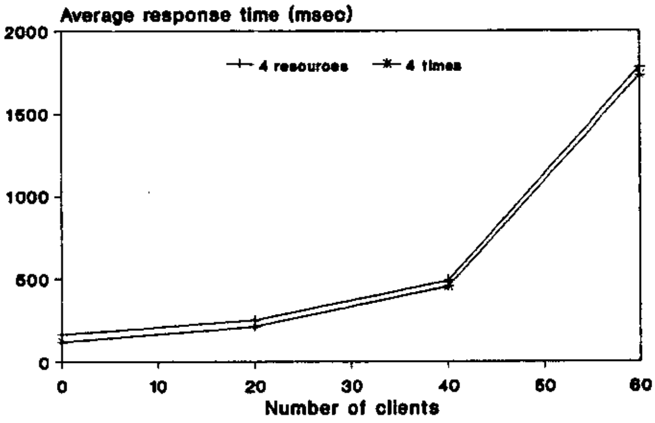


Figure 6.6.9 : 50.7Kbytes

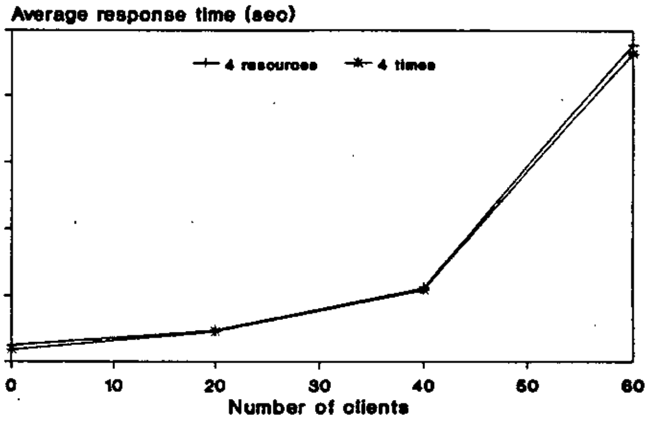


Figure 6.6.10 : 316Kbytes(B)

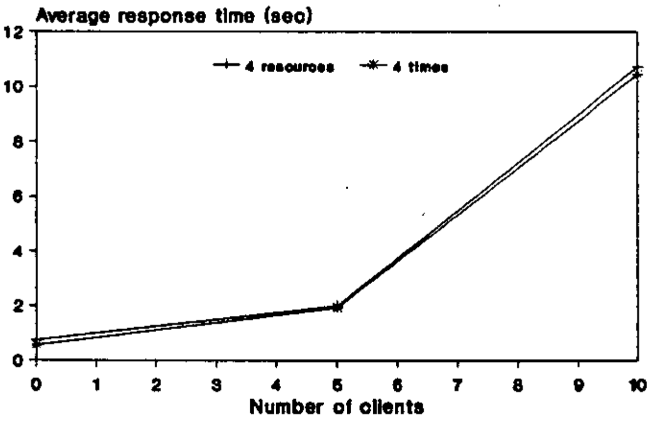


Figure 6.6.11 : 316Kbytes

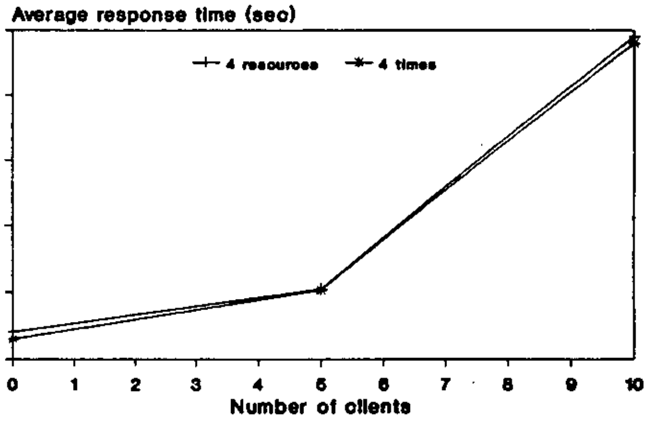


Figure 6.6.12 : 1856Kbytes

The average response time of the case of the four times better disk I/O time vs. the average response time of the case of the four disk I/O subsystems in the distributed file system which consists of the Sun SPARCstation 10 workstations.

faster and the average response time in the shared memory system which has four disks and four disk interface units when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively and the number of local users increases gradually.

It should be remembered that it was observed in section 6.4 that beyond 4 disks and 4 disk interface units the average response time improved very little since the contention for the disk I/O subsystem almost disappeared with 4 disks and 4 disk interface units but in section 6.5.1 the average response time improved as far as the disk I/O time improved. Therefore the gap between the two average response time increases when we use more than 4 disks and 4 disk interface units and when we improve the disk I/O time to be more than 4 times better. Thus the figures of the above two cases are enough for us to compare the file access performance in all cases.

In the figures to compare the system where the disk I/O time is improved to be two times faster and the system which uses the two disks and the two disk interface units, the two average response time curves have one crossing point except when the 316Kbytes(B) workload is used. When there is no contention for the system resources, the average response time in the system where the disk I/O time is improved to be 2 times faster is always smaller than that in the system which has 2 disks and 2 disk interface units. The average response time in the system where the disk I/O time is improved to be two times faster grows more quickly than that in the system which has 2 disks and 2 disk interface units as the contention grows in both system paradigms. This means the average response time in the system where the disk I/O time is improved to be two times faster is more sensitive to the number of clients than that in the system which has 2 disks and 2 disk interface units in both system paradigms.

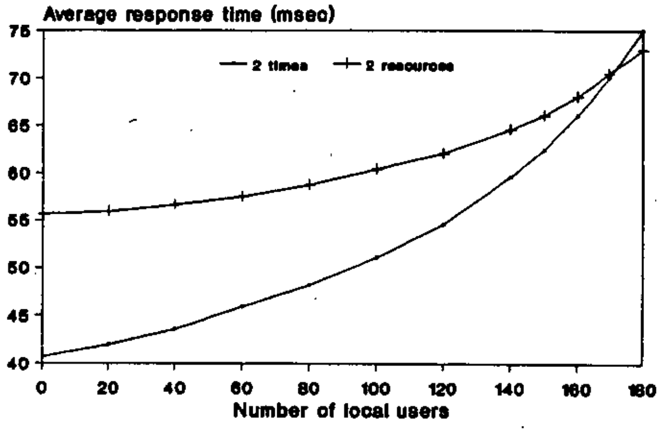


Figure 6.6.13 : 8Kbytes

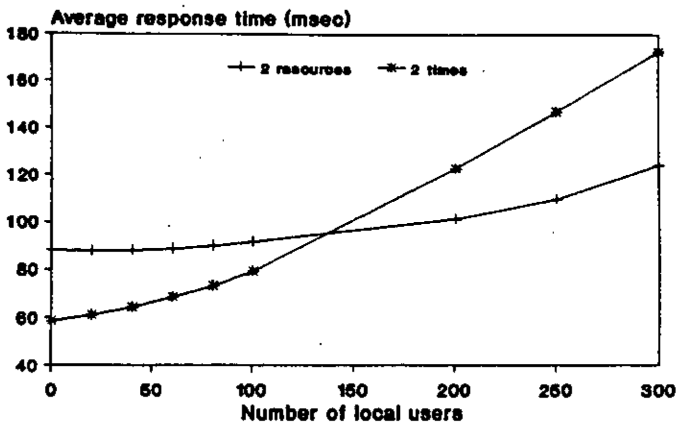


Figure 6.6.14 : 47Kbytes

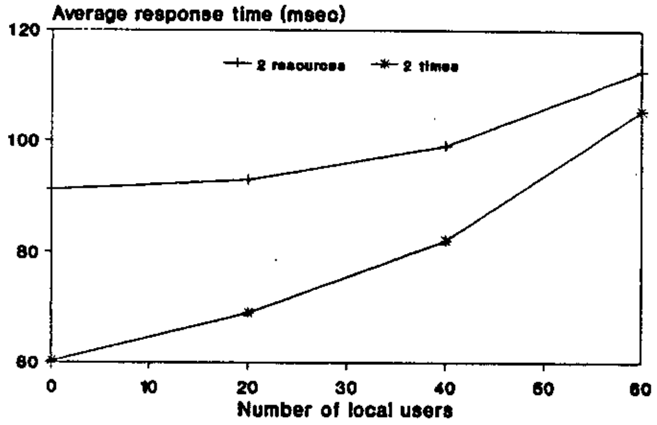


Figure 6.6.15 : 50.7Kbytes

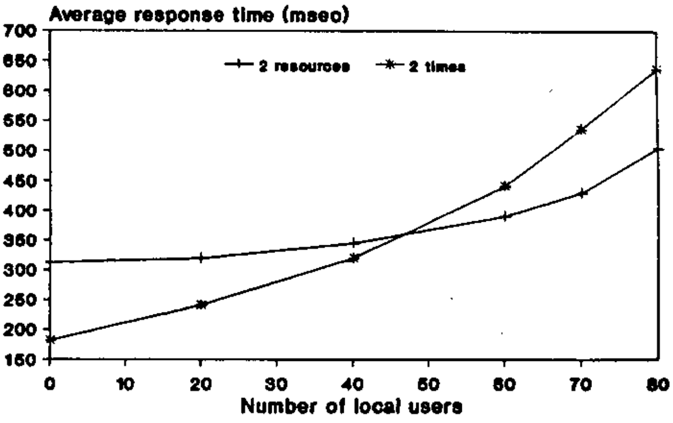


Figure 6.6.16 : 316Kbytes(B)

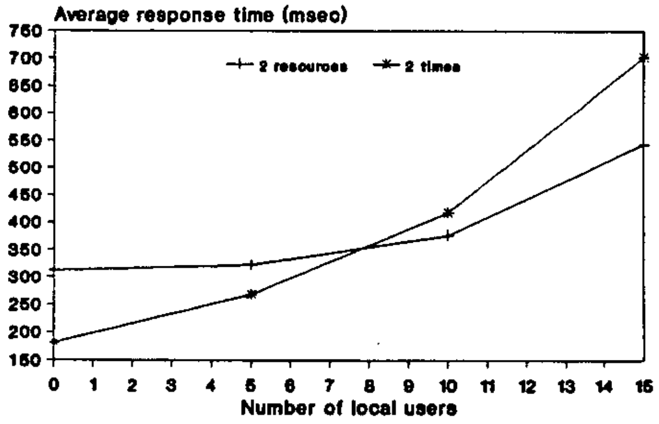


Figure 6.6.17 : 316Kbytes

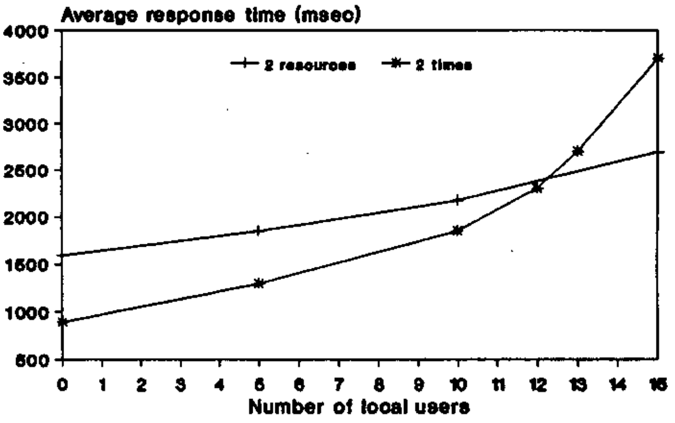


Figure 6.6.18 : 1856Kbytes

The average response time of the case of the two times better disk I/O time vs. the average response time of the case of the two disk I/O subsystems in the Sun SPARCstation 10 workstation.

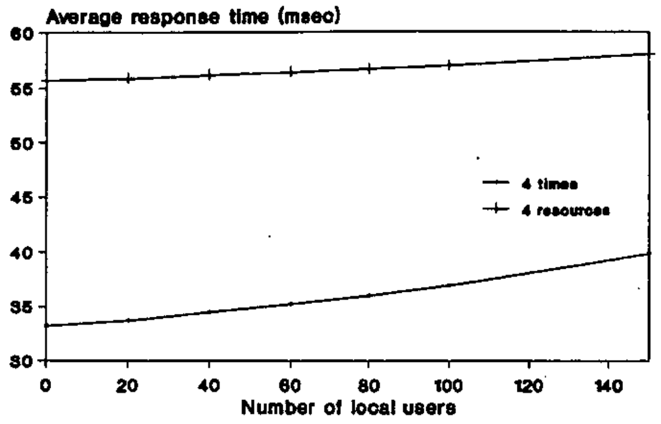


Figure 6.6.19 : 8Kbytes

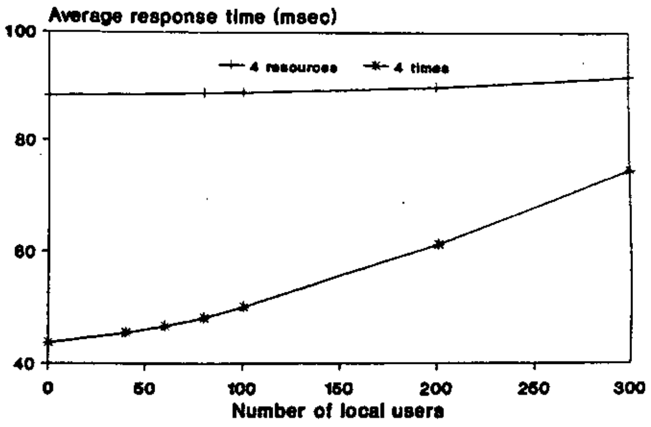


Figure 6.6.20 : 47Kbytes

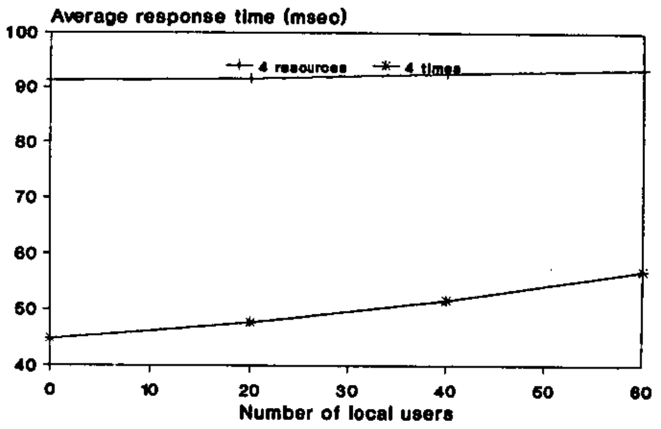


Figure 6.6.21 : 50.7Kbytes

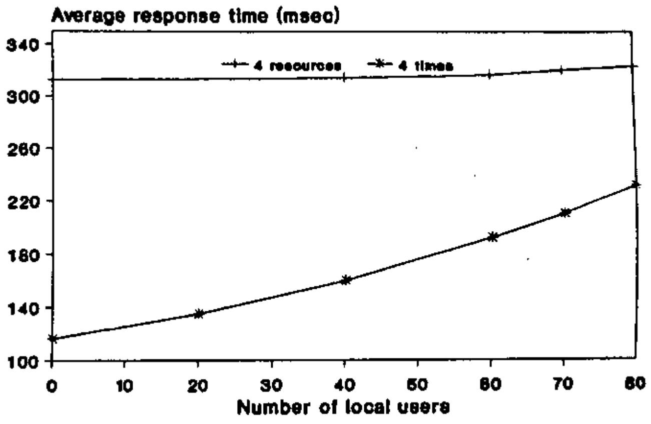


Figure 6.6.22 : 316Kbytes(B)

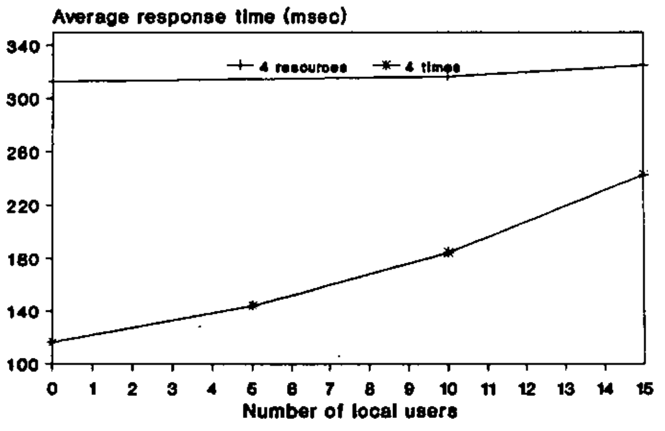


Figure 6.6.23 : 316Kbytes

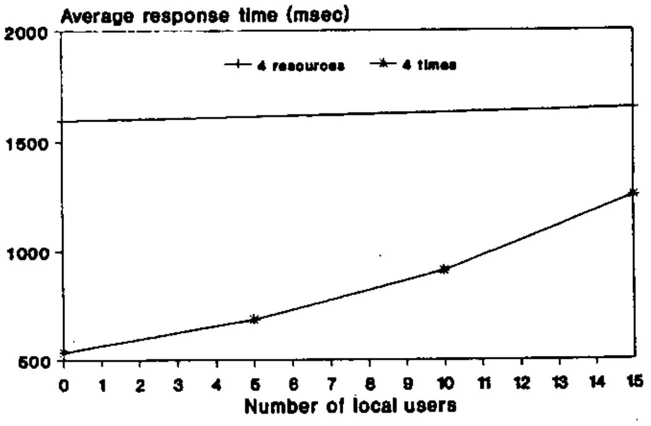


Figure 6.6.24 : 1856Kbytes

The average response time of the case of the four times better disk I/O time vs. the average response time of the case of the four disk I/O subsystems in the Sun SPARCstation 10 workstation.

Only the average response time of the 316Kbytes(B) workload has 2 intersecting points in the distributed file system. When the 316Kbytes(B) workload is supplied, until the first crossing point, the faster case shows a better average response time than the multiple case and from the first crossing point to the second crossing point, the order is reversed and after the second crossing point, the order is again reversed and the order becomes the same as the order before the first crossing point.

In the figures to compare the system where the disk I/O time is improved to be four times faster and the system which uses four disks and four disk interface units, all six workloads have one crossing point before saturation even though the crossing point is not shown in the scale of the figure for the 8Kbytes workload and in the scale of the figure for the 47Kbytes workload. When there is no contention for the system resources, the average response time in the system where the disk I/O time is improved to be 4 times faster is always smaller than that in the system which has 4 disks and 4 disk interface units. The average response time in the system where the disk I/O time is improved to be 4 times faster grows more quickly than that of the system which has 4 disks and 4 disk interface units as the contention grows in both system paradigms.

As the disk I/O speed and the number of disks and the number of disk interface units increase, the two lines of the average response time cross with more clients in the distributed file system or with more local users in the shared memory system. This means the system which has the faster disk and the system which has multiple disks and multiple disk interface units becomes less sensitive to the number of clients or the number of local users as the disk I/O speed and the number of disks and disk interface units increase. As the average transaction size increases, the two lines of the average response time cross with fewer clients in the distributed file system or with fewer local users in the local shared memory system. This means the system which has the faster disk is more sensitive to the average transaction size than the system which has multiple disks and multiple disk interface units.

Generally the six workloads show similar patterns except that the average response time of the 318Kbytes(B) workload has two crossing points in the distributed file system.

6.7 Multiple Networks and Multiple Network Interface Units

This section investigates the effect on the file access performance when multiple networks and multiple network interface units are used in the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations.

The performance model of figure 3.2.6.E and the baseline performance parameter values of table 3.2.7.C are used for the distributed file system. The multiple networks and the multiple network interface units are represented as multiple servers which share a queue in the performance model. Each network interface unit is assumed to receive the RPC requests and the RPC responses with equal opportunity since the multiple networks and the multiple network interface units are assumed to have the symmetric property. The overhead to manage the multiple networks and the multiple network interface units is assumed to be negligible, which means this study considers the theoretical limit.

Figure 6.7.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. Both the number of networks and the number of network interface units in the file server are increased to be K(2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity). Except for the number of networks and the number of network interface units in the file server, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix C for the figures of other cases.

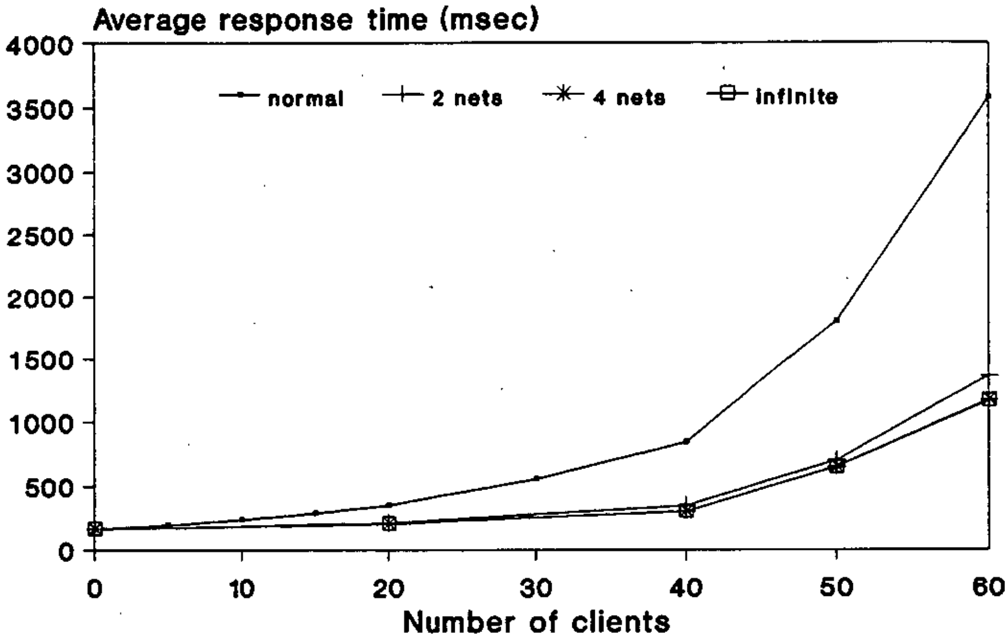


Figure 6.7.1 : The effect on the average response time of having multiple networks and multiple network interface units in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

Since the network interface unit is one of the major bottleneck points[1], the overall improvement in the average response time in the distributed file system is significant. When 2 networks and 2 network interface units are used, the average

[1] The table 6.2.1 shows that the disk i/o subsystem is the busiest bottleneck point and the network interface unit is the next busiest bottleneck point when we use the 8kbytes workload or the 47kbytes workload or the 50.7kbytes workload and the network interface unit is the major bottleneck point and the disk i/o subsystem is the next busiest bottleneck point when we use the 316kbytes(B) workload or the 316kbytes workload or the 1856kbytes workload in the baseline distributed file system which consists of the Sun SparcStation 10 workstations.

response time improves greatly. When 4 networks and 4 network interface units are used, the average response time improves a little further but less than twice the amount improved when 2 networks and 2 network interface units are used. When 4 multiple networks and 4 network interface units are used, most of the contention for the networks and the network interface units disappears and the disk I/O subsystem, previously one of the major bottleneck points, now becomes the major bottleneck point and dominates the queueing delay. Therefore, if more networks and network interface units are added to the system which already has 4 networks and 4 network interface units in the system, the average response time improves very little and it is not effective in terms of cost/performance. Even when an infinite number of networks and an infinite number of network interface units are used in the baseline distributed file system which consists of the Sun SPARCstation 10 workstations, the saturation point increases a little or does not increase at all since the saturation point of the disk I/O subsystem is a little larger or smaller than that of network interface units, depending on the workload as table 5.2.1 shows.

It should be noticed that the network interface unit is always saturated before the network. No notable change is observed in the pattern of the average response time as the workload size increases.

6.8 Faster Network Communication

6.8.1 Faster Network

This section investigates the effect on the file access performance when the network transmission speed is improved.

Figure 6.8.1.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The baseline performance model of figure 3.2.6.B is used and only the network transmission speed is improved to be 2(20Mbps), 5(50Mbps), 10(100Mbps), 50(500Mbps), 100(1Gbps), 1000(10Gbps) times and infinitely faster. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The network retransmission delay may be adjusted when the transmission speed is changed. See appendix C for the figures of other cases.

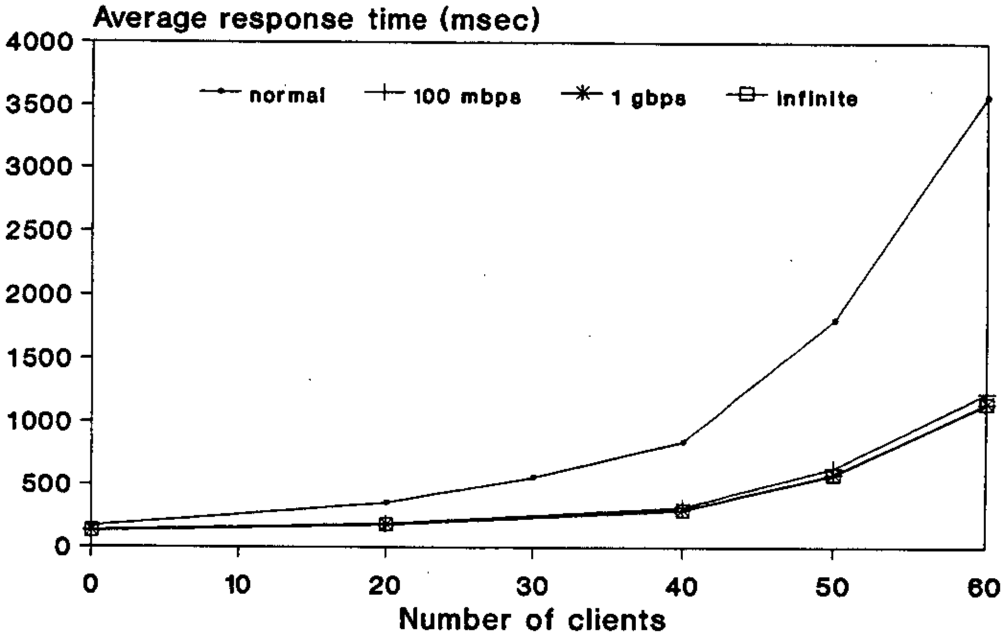


Figure 6.8.1.1 : The effect of having a faster network on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

The overall improvement of the average response time in the distributed file system is significant. When the 10 times faster(100Mbps) network is used, the average response time improves significantly. If we improve the network speed further beyond 10 times faster, the average response time improves a little, which means that most of the contention for the network disappears with a 100Mbps network in the environment. The 100Mbps network speed is now offered by 100Mbps Ethernet, 100Mbps FDDI, etc.. It is found that the utilization of the network interface unit is much reduced, therefore the contention for it in the file server is much reduced as the network speed increases since the busy period of the network interface unit during data transmission is reduced as the network speed increases. No notable change is observed in the pattern of the average response time as the workload size increases. In the simulation, the network is seized during the transmission of the whole transaction data without any intervention. Therefore, the queueing delay due to the contention for the network in real environments might be less than what was observed in this study. This also applies to the disk I/O subsystem and the network interface unit.

6.8.2 Better Network Interface Unit

This section investigates the effect on the file access performance when the performance of the network interface unit is improved. The parameter value of the I/O time for the request send operation and that for the response receive operation in the network interface units of the clients and that for the request receive operation and that for the response send operation in the network interface unit of the file server are improved. It is notable that the ratio among the parameter value of the distributed file system, which consists of the Sun SPARCstation 10 workstations, the parameter value of the distributed file system, which consists of the Sun SPARCstation 470 workstations, and the parameter value of the distributed file system, which consists of the Sun 3/60 workstations, in table 3.2.7.C is 1: 6.18 : 18.31, which is relatively close to the inverse of the MIPS ratio in the three component systems, 1 : 7.34 : 33.87, compared with the ratios of the other parameters.

Figure 6.8.2.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The baseline performance model of figure 3.2.6.B is used and the I/O time for the request send operation and that for the response receive operation in the network interface units of the clients and that for the request receive operation and that for the response send operation in the network interface unit of the file server are improved to be 2 times, 4 times, 6 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix C for the figures of other cases.

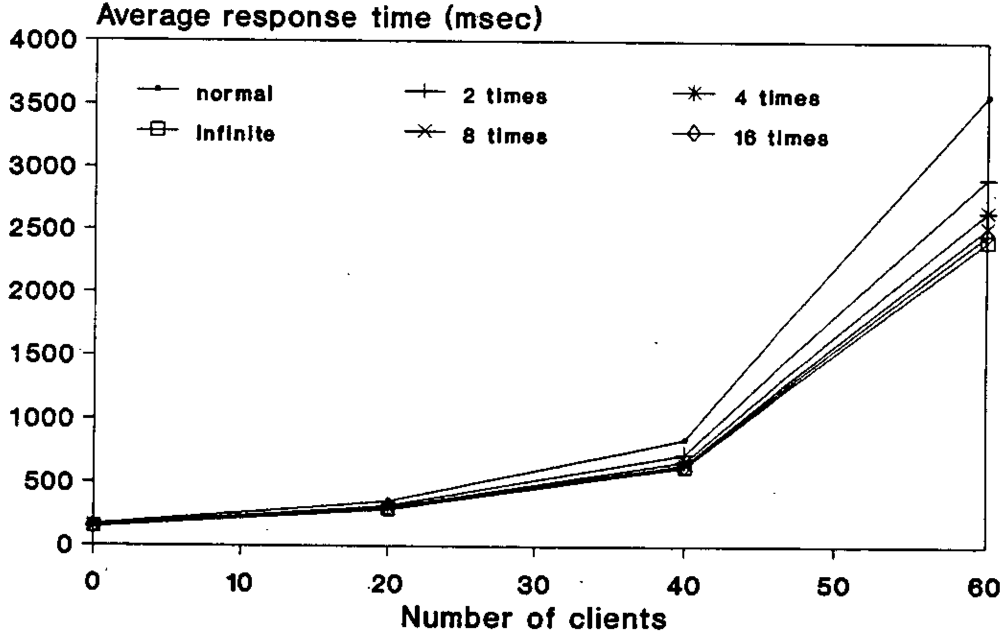


Figure 6.8.2.1 : The effect on the average response time of having the better network interface units in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

The overall improvement of the average response time in the distributed file system is significant. It is observed that the average response time improves as far as the parameter values are improved. However, if the parameter values are further improved when they are already 16 times better, the improved amount of the average response time is trivial. It means that the contention for the network interface unit almost disappears when the parameter values are improved to be 16 times better. No notable change is observed in the pattern of the average response time as the average transaction size increases.

6.8.3 Enhanced Communication Mechanism

This section investigates the effect on the file access performance when the communication mechanism is enhanced. Better mechanisms in the communication software and in the communication hardware can reduce the CPU service time for the network communication as explained in section 3.2.4. Better communication mechanisms might reduce the I/O time for the network communication correspondingly as well. This section investigates the effect on the file access performance when only the CPU service time for the network communication is reduced.

This study changes the CPU service time for the network communication both in the clients and in the file server at the same time in order to maintain the homogeneity in the distributed file system. CPU time is consumed to setup the communication path, to move the transaction data between the memory buffer and the buffer of the network interface unit, to handle the interrupt by the network interface unit, etc.. The ratio among the parameter value of the distributed file system which consists of the Sun SPARCstation 10 workstations, that of the distributed file system which consists of the Sun SPARCstation 470 workstations and that of the distributed file system which consists of the Sun 3/60 workstations in table 3.2.7.C is 1 : 1.12 : 1.23, far below the inverse of the MIPS ratio in the three systems, 1 : 7.34 : 33.87.

Figure 6.8.3.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The baseline performance model of figure 3.2.6.B is used and the CPU time for the request send operation and that for the response receive operation in the clients and that for the request receive operation and that for the response send operation in the file server are improved to be 2 times, 4 times, 6 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix C for the figures of other cases.

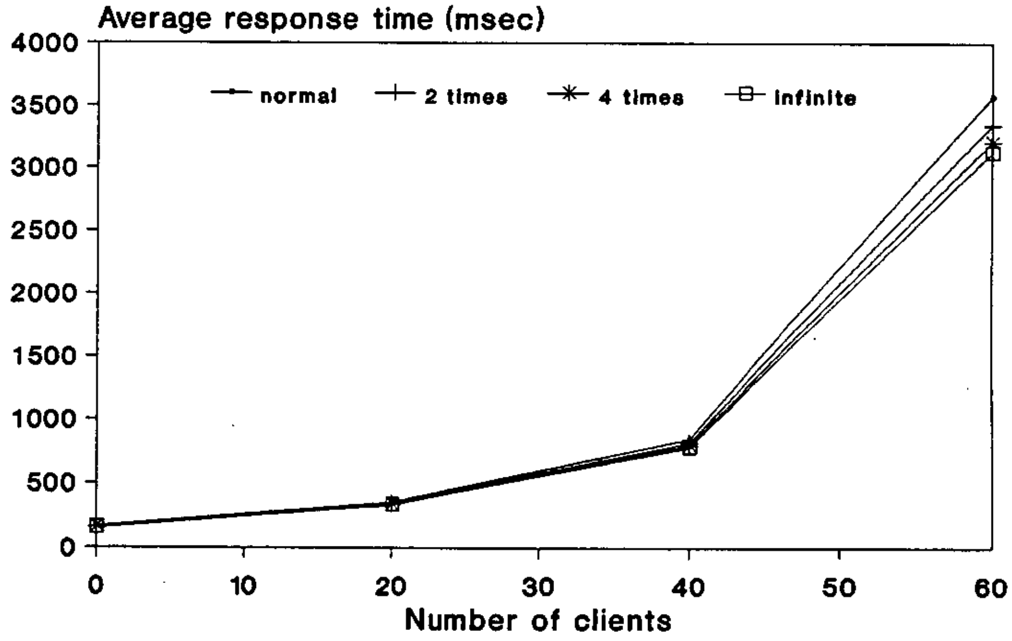


Figure 6.8.3.1 : The effect of having the better communication mechanism on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

The overall improvement of the average response time in the distributed file system is small. It should be remembered that the effect on the file access performance by all CPU parameters together was found to be small in section 6.1. The effect on the file access performance investigated in this section can not be larger than that. No notable change is observed in the pattern of the average response time as the average transaction size increases.

6.8.4 All Improvements at the Same Time

This section investigates the effect on the file access performance when the performance factors investigated in section 6.8.1, section 6.8.2 and section 6.8.3 are considered at the same time.

The parameters for the network communications considered in this section are the parameters of the network transmission, the parameter of the I/O time for the network communication and the parameters of the CPU service time for the network communication.

Figure 6.8.4.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The baseline performance model of figure 3.2.6.B is used and the values of all parameters for the network communication are improved to be 2 times, 4 times, 6 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The network speed is set to be 50Mbps not 40Mbps for the 4 times better case and 100Mbps for the 8 times better case and for the 16 times better case. In all other cases, the degree of improvement is kept the same for all parameters. See appendix C for the figures of other cases.

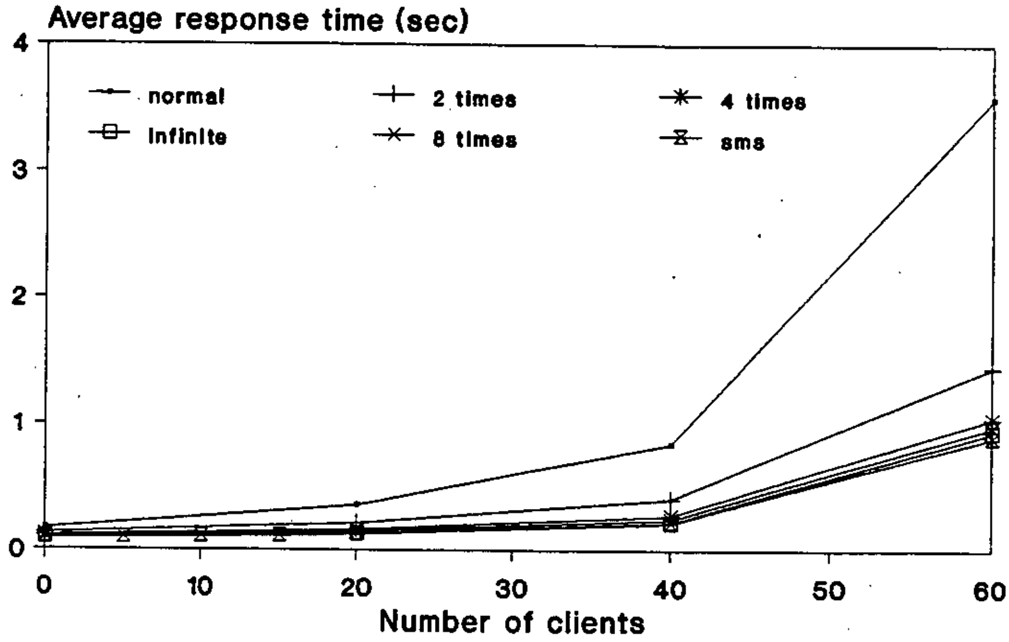


Figure 6.8.4.1 : The effect of the better communication on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

The overall improvement of the average response time in the distributed file system is significant. The average response time further improves even though the amount of improvement is getting smaller and smaller as the degree of improvement in the parameter values is doubled. When we improve the parameter values further in the distributed file system where the values were already improved to be 16 times better, then the further improved amount becomes trivial. It means that the queueing delay due to the contention during the network communication almost disappears when the parameter values are improved to be 16 times better.

When all communication parameter values are improved to be infinitely better, then the average response time of the distributed file system is almost the same as the average response time of the baseline shared memory system since now the only difference between the file access overheads of the two system paradigms is the RPC related overhead which is small and constant for the average transaction size. When all communication parameter values are improved to be 16 times better in the baseline distributed file system which consists of the Sun SPARCstation 10 workstations, the average response time of the distributed file system becomes very close to the average response time of the baseline shared memory system as we see in figure 6.8.4.1. Even with 10 times better communication parameters, the baseline distributed file system show the file access performance close to the baseline shared memory system.

6.9 Multiple Networks vs. Better Network

This section compares the file access performance of the distributed file system which uses a faster network and a better network interface unit in the file server and the distributed file system which uses multiple networks and multiple network interface units.

In order to compare them fairly, this study does not modify the clients at all but replaces the network with a faster network and the network interface unit of the file server with a better network interface unit in the distributed file system. Now the distributed file system becomes heterogeneous. In section 6.8.1, section 6.8.2, section 6.8.3 and section 6.8.4, the related parameter values were changed both in the file server and in the clients to maintain homogeneity.

Figure 6.9.1 to figure 6.9.6 compare the average response time of the distributed file system which uses the 2 times faster network and the 2 times better network interface unit in the file server and the average response time of the distributed file system which uses the 2 networks and the 2 network interface units in the file server when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively and the number of clients increases gradually.

Figure 6.9.7 to figure 6.9.12 compare the average response time of the distributed file system which uses a 4 times faster network and a 4 times better network interface unit in the file server and the average response time of the distributed file system which uses 4 networks and 4 network interface units in the file server when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively and the number of clients increases gradually.

Figure 6.9.13 to figure 6.9.18 compare the average response time of the distributed file system which uses a 10 times faster network and a 10 times better network interface unit in the file server and the average response time of the distributed file system which uses the 10 networks and the 10 network interface units in the file server when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively and the number of clients increases gradually.

When there is no contention for the system resources, the average response time in the distributed file system which uses the faster network and the better network interface unit in the file server is always smaller than that in the distributed file system which has the multiple networks and the multiple network interface units in the file server. The average response time in the distributed file system which uses the faster network and the better network interface unit in the file server develops more quickly than the average response time in the distributed file system which has the multiple networks and the multiple network interface units

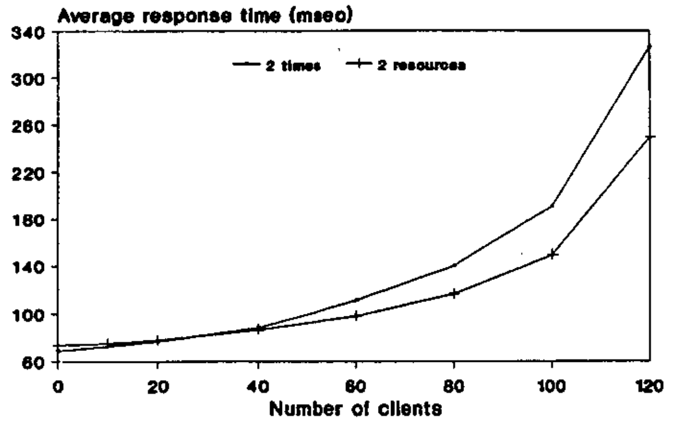


Figure 6.9.1 : 8Kbytes

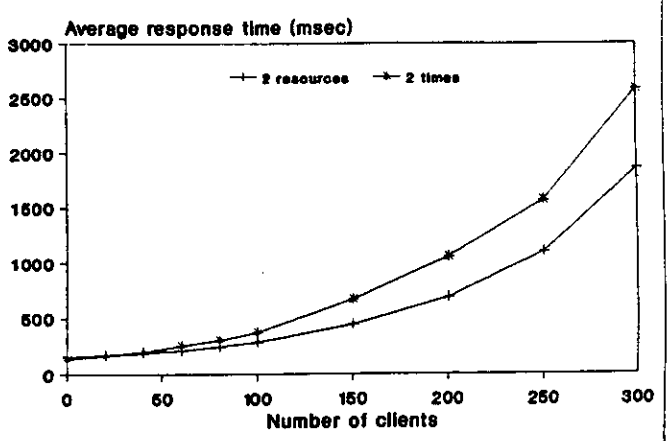


Figure 6.9.2 : 47Kbytes

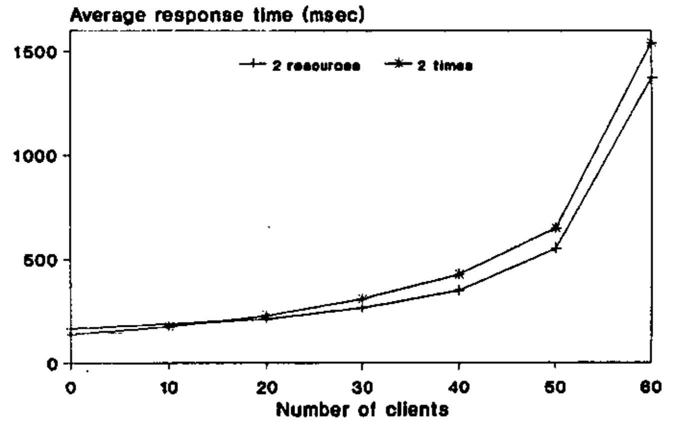


Figure 6.9.3 : 50.7Kbytes

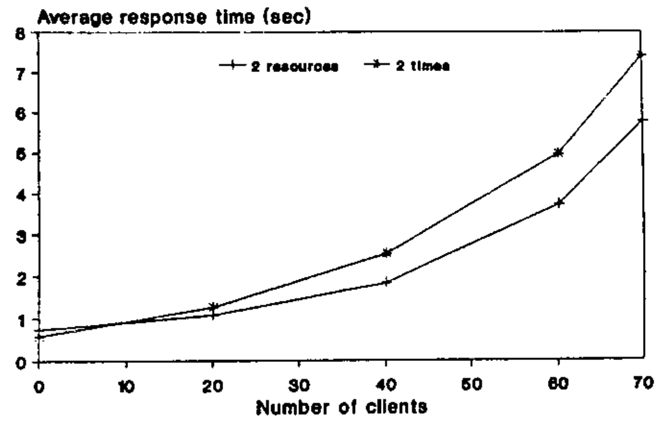


Figure 6.9.4 : 316Kbytes(B)

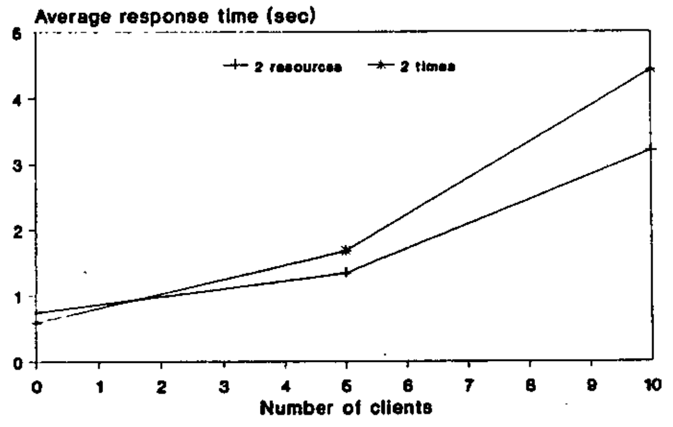


Figure 6.9.5 : 316Kbytes

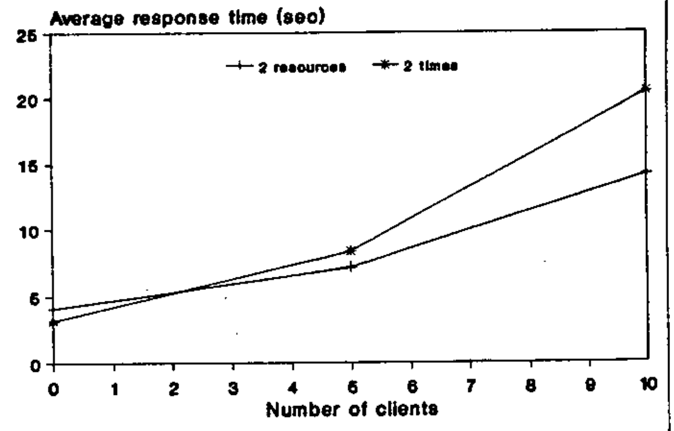


Figure 6.9.6 : 1856Kbytes

The average response time of the case of having the 2 times faster network and the 2 times better network interface unit vs. the average response time of the case of having the 2 networks and the 2 network interface units in the distributed file system which consists of the Sun SPARCstation 10 workstations.

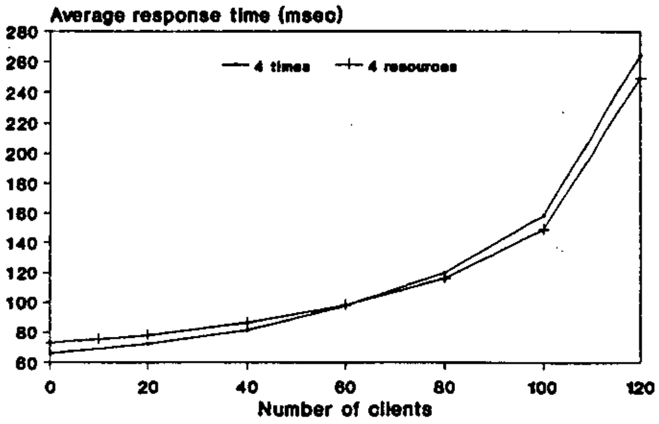


Figure 6.9.7 : 8Kbytes

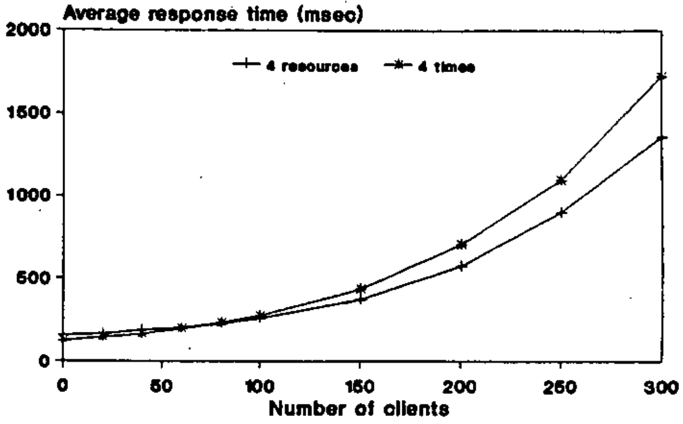


Figure 6.9.8 : 47Kbytes

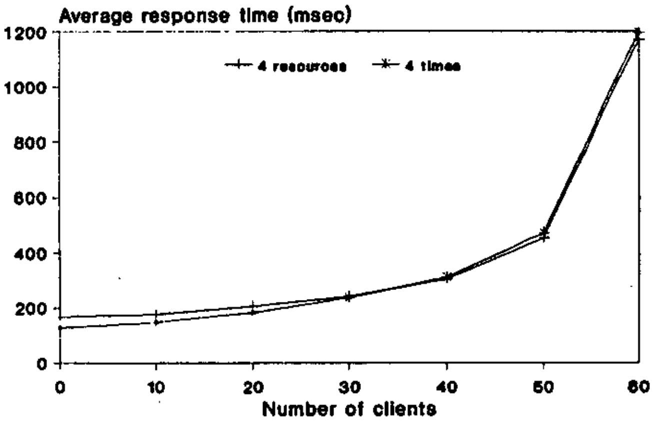


Figure 6.9.9 : 50.7Kbytes

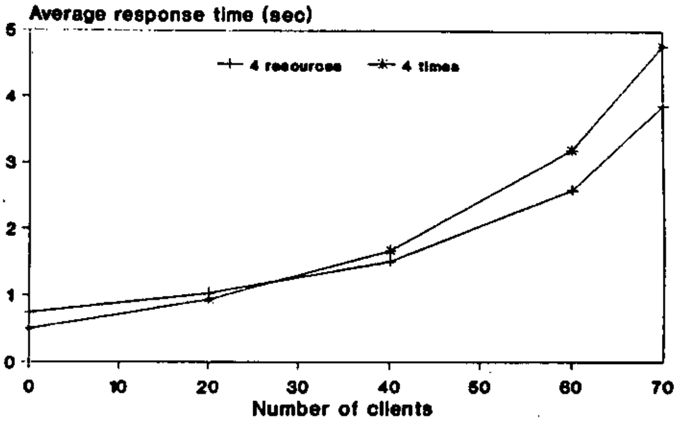


Figure 6.9.10 : 316Kbytes(B)

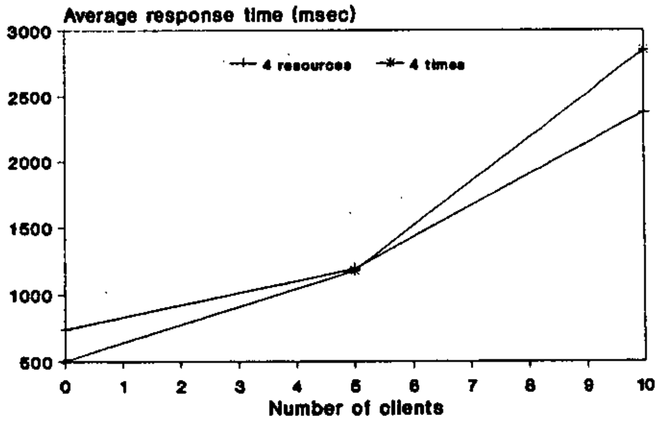


Figure 6.9.11 : 316Kbytes

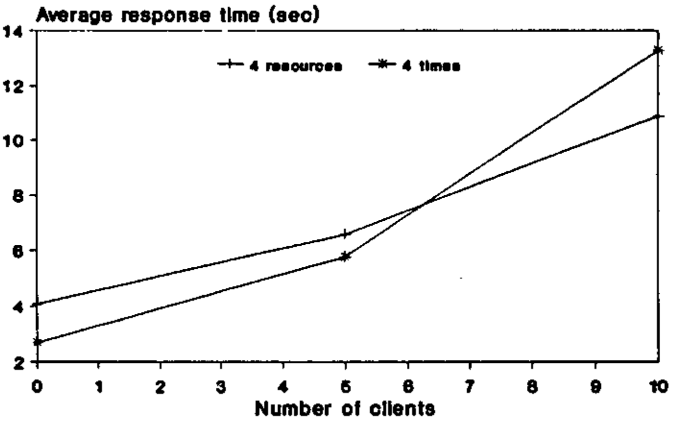


Figure 6.9.12 : 1856Kbytes

The average response time of the case of having the 4 times faster network and the 4 times better network interface unit vs. the average response time of the case of having the 4 networks and the 4 network interface units in the distributed file system which consists of the Sun SPARCstation 10 workstations.

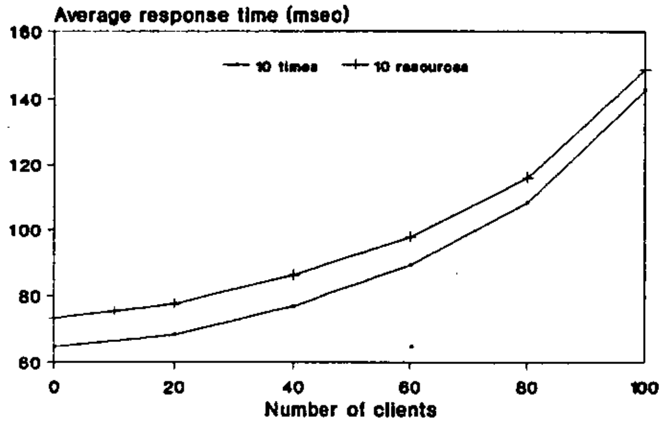


Figure 6.9.13 : 8Kbytes

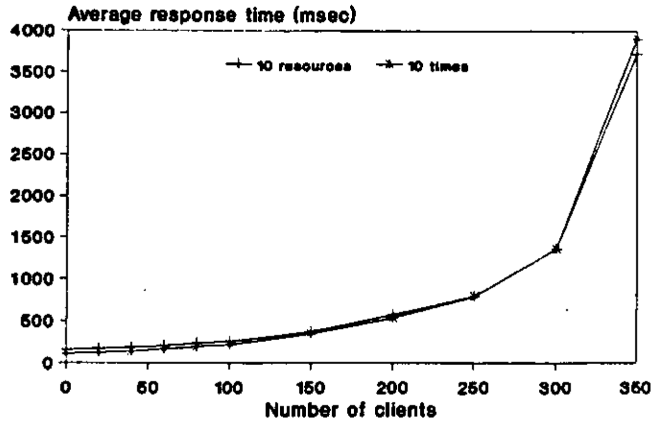


Figure 6.9.14 : 47Kbytes

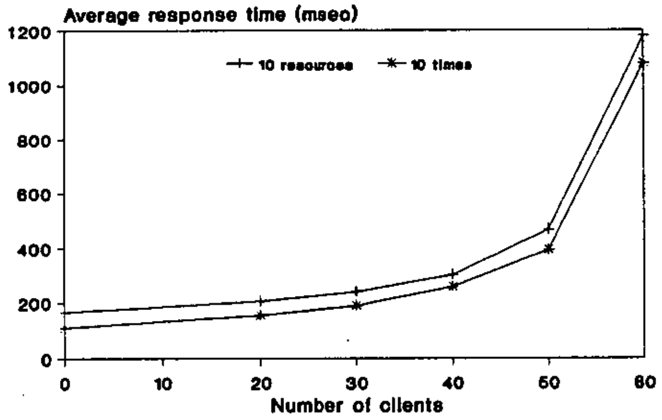


Figure 6.9.15 : 50.7Kbytes

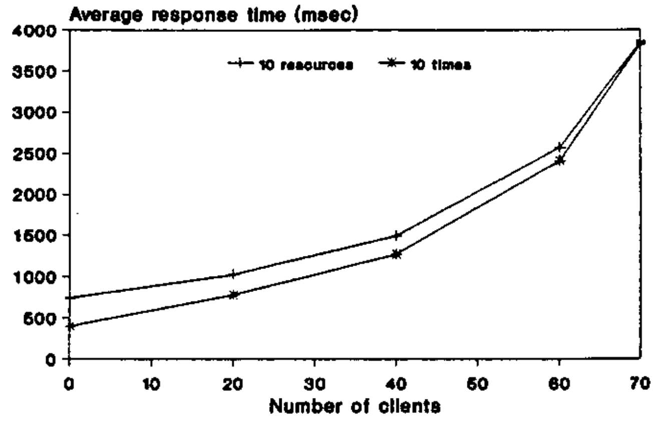


Figure 6.9.16 : 316Kbytes(B)

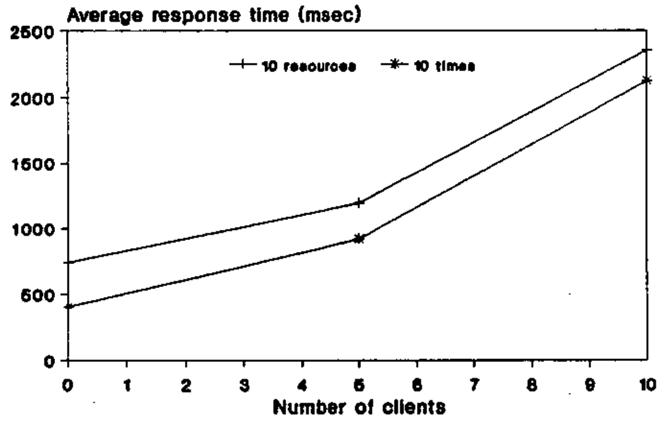


Figure 6.9.17 : 316Kbytes

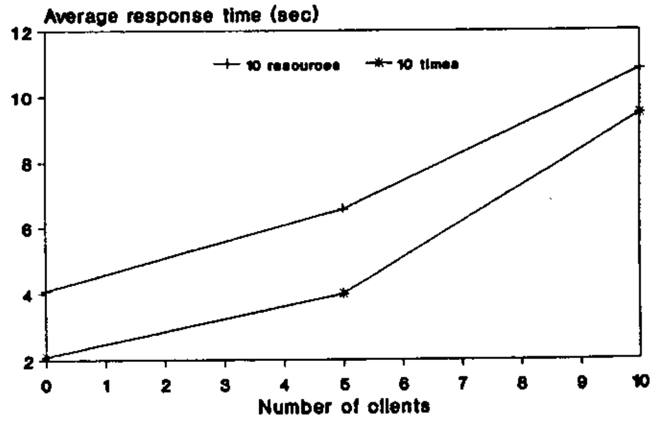


Figure 6.9.18 : 1856Kbytes

The average response time of the case of having the 10 times faster network and the 10 times better network interface unit vs. the average response time of the case of having the 10 networks and the 10 network interface units in the distributed file system which consists of the Sun SPARCstation 10 workstations.

in the file server. Therefore, the two lines of the average response time cross once in the figures. This happens since the average response time of the distributed file system which has multiple networks and multiple network interface units in the file server is less sensitive to the number of clients than the average response time of the distributed file system which has the faster network and the better network interface unit in the file server.

Let's see where the two lines cross. First we look at figure 6.9.1 to figure 6.9.2. The two lines cross at around 30 clients when the 8Kbytes workload is used, at around 30 clients when the 40.7Kbytes workload is used, at around 15 clients when the 50.7Kbytes workload is used, at around 15 clients when the 316Kbytes(B) workload is used, at around 1.5 clients when the 316Kbytes workload is used and at around 2.2 clients when the 1856Kbytes workload is used. It is found that as the workload size increases, the two lines cross earlier, that is, with fewer clients.2) This happens since the average response time of the distributed file system which has multiple networks and multiple network interface units in the file server is less sensitive to the average transaction size than the average response time of the distributed file system which has the faster network and the better network interface unit in the file server.

Let's look at figure 6.9.7 to figure 6.9.12. In figure 6.9.7 to figure 6.9.12, where the degree of multiplicity and the degree of improvement is 4, the two lines cross at around 60 clients when the 8Kbytes workload is used, at around 60 clients when the 40.7Kbytes workload is used, at around 40 clients when the 50.7Kbytes workload is used, at around 33 clients when the 316Kbytes workload is used, at around 5.2 clients when the 316Kbytes workload is used and at around 6.2 clients when the 1856Kbytes workload is used. In the figures, as the degree of

(2) This is true among the steady state workloads or among the bursty state workloads. But it is not true across the steady state workloads and bursty state workloads. For example, it is not true when we compare the crossing point when we use the 316kbytes workload and the crossing point when we use the 1856kbytes workload.

multiplicity and the degree of improvement increases, the two lines of the average response time cross with more clients or with more contention.

It is notable that the number of clients where the two lines cross when the degree of multiplicity and the degree of improvement is 4 is more than two times as large as the crossing point when the degree of mutiplicity and the degree of improvement is 2. The improvement is getting smaller and smaller as the degree is doubled each time. Generally the six workloads show similar patterns for the average response times.

6.10 Other Enhancements

6.10.1 Enhanced File System Mechanism

This section comparatively investigates the effect on the file access performance when the file system mechanism is enhanced both in the distributed file system and in the shared memory system.

When the file system mechanism is enhanced, the CPU service time for the file handling operations such as directory handling, file table lookup, updating file tables, opening files, closing files, etc. is reduced. This section analyzes the effect on the file access performance when the CPU service time for the file handling operations is improved. It does not matter whether it is directly improved by the enhancement of the file system mechanism or indirectly improved by any other or complex enhancement. For example, in the case of parallel file systems, if the parallel file system enhances the file system mechanism and therefore improves the CPU service time, then the effect is also analyzed in this section and if it improves disk I/O time then the effect was already analyzed in section 6.5.

The overhead from the file system mechanism in the file server is 20msec in the distributed file system which consists of the Sun 3/60 workstations, 10msec in the distributed file system which consists of the Sun SPARCstation 470 workstations and 5msec in the distributed file system which consists of the Sun SPARCstation 10 workstations. The ratio is 4 : 2 : 1 while the MIPS ratio in the three component systems is 1 : 7.34 : 33.87.

Let's look at the effect on the file access performance of the overhead of the file system mechanism in the file server when the average transaction size of the workload is increased in the environment where there is no contention for the system resources. First it is looked at in the distributed file system. When the 8Kbytes workload is used, the overhead of the file system mechanism takes 5.8% of the average response time in the distributed file system which consists of the Sun 3/60 workstations, 7.2% of the average response time in the distributed file system which consists of the Sun SPARCstation 470 workstations and 6.9% of the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations. When the 1856k bytes workload is used, the overhead of the file system mechanism takes 0.13%, 0.12% and 0.13% in the three distributed file systems respectively. It is found that as the average transaction size of the workload increases, the effect on the average response time decreases and becomes trivial. This is due to amortization since the overhead of the file system mechanism does not vary with the transaction size.

Now let's look at the effect on the file access performance when the average transaction size of the workload is increased in the shared memory system where there is no contention for the system resources. The overhead of the file system mechanism in the shared memory system is same as that in the distributed file system. When the 8Kbyte workload is used, the overhead of the file system mechanism takes 7.9% of the average response time in the Sun 3/60 workstation, 10.1% of the average response time in the Sun SPARCstation 470 workstation and

9% of the average response time in the Sun SPARCstation 10 workstation. When the 1856k bytes workload is used, the overhead of the file system mechanism takes 0.32%, 0.46% and 0.32% in the three distributed file systems respectively. It is found that as the average transaction size of the workload increases, the effect on the average response time decreases and becomes trivial as in the distributed file system..

It was found that as the number of clients increased, the effect of the parameter of the file processing mechanism decreased further and became trivial. It was observed that the average response time improved very little when the parameter value was improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better respectively and the 8Kbytes workload, the 47bytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload were used respectively in the distributed file system which consisted of the Sun SPARCstation 10 workstations. This dissertation does not include these figures since the performance effect is trivial.

It was observed that the average response time improved very little when the parameter value was improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better respectively and the 8Kbytes workload, the 47bytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload were used respectively in the Sun SPARCstation 10 workstation. This dissertation does not include these figures since the performance effect is trivial.

Since the parameter is a CPU service time parameter, the effect of the parameter on the file access performance should be always smaller than the effect of all CPU service time parameters on the file access performance which was already investigated in Section 6.2.

Since the overhead by the file system mechanism in the shared memory system is the same as that in the distributed file system, the effect on the file access performance in the shared memory system is larger than that in the distributed file system even though the effect is trivial in both paradigms.

6.10.2 Enhanced RPC Mechanism

This section investigates the effect on the file access performance when the RPC mechanism is enhanced in the distributed file system. For the detailed investigation of various RPC mechanisms, refer to the papers of [ANANDA etal 93],[TAY etal 90] which survey the RPC mechanisms.

Four performance parameters are related to the RPC mechanism. They are the parameter of the RPC build operation in the client, the parameter of the RPC evaluation operation in the file server, the parameter of the RPC build operation in the file server, the parameter of the RPC evaluation operation in the client. All the parameters belong to the CPU parameters. Therefore, when the RPC mechanism is enhanced, the CPU service time for the RPC operations is reduced. This section investigates the effect on the file access performance when the parameter values are improved in the distributed file system.

The total RPC overhead both in the file server and in the clients is 13.32msec in the distributed file system which consists of the Sun 3/60 workstations, 10msec in the distributed file system which consists of the Sun SPARCstation 470 workstations and 5msec in the distributed file system which consists of the Sun SPARCstation 10 workstations. The ratio is 2.67 : 2 : 1 while the MIPS ratio in the three component systems is 1 : 7.34 : 33.87.

Let's look at the effect of the RPC overhead on the file access performance when the average transaction size of the workload is increased in the distributed file system where there is no contention for the system resources. When the 8Kbytes workload is used, the total RPC overhead takes 3.9% of the average response time in the distributed file system which consists of the Sun 3/60 workstations, 7.2% of the average response time in the distributed file system which consists of the Sun SPARCstation 470 workstations and 6.9% of the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations. When the 1856k bytes workload is used, the RPC overhead takes 0.13%, 0.12% and 0.13% in the three distributed file systems respectively. It is found that as the average transaction size of the workload increases, the effect on the average response time decreases and becomes trivial. This is due to amortization since the RPC overhead does not vary with the transaction size. As the number of clients increases, the effect of the RPC parameters on the file access performance decreases further and becomes trivial.

It was observed that the average response time improved very little when the values of all RPC parameters were improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better respectively and the 8Kbytes workload, the 47bytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload were used respectively in the distributed file system which consisted of the Sun SPARCstation 10 workstations. This dissertation does not include these figures since the performance effect is trivial.

Since the parameters belong to the CPU service time parameters, the effect of the parameters on the file access performance should be always smaller than the effect of all CPU service time parameters on the file access performance which was already investigated in Section 6.2. The pattern of the effect on the file access performance by enhancing the RPC mechanism is similar to that by enhancing the file system mechanism.

6.10.3 Enhanced Command Interpretation Mechanism

This section comparatively investigates the effect on the file access performance when the command interpretation mechanism is enhanced both in the distributed file system and the shared memory system.

The parameter of the command interpretation operation is one of the CPU parameters. When the command interpretation is enhanced, the CPU service time for the command interpretation operation is reduced. This section investigates the effect on the file access performance when the parameter value is improved in both system paradigms.

The overhead of the command interpretation operation is 80msec in the distributed file system which consists of the Sun 3/60 workstations, 20msec in the distributed file system which consists of the Sun SPARCstation 470 workstations and 20msec in the distributed file system which consists of the Sun SPARCstation 10 workstations. The ratio is 4 : 1 : 1 while the MIPS ratio in the three component systems is 1 : 7.34 : 33.87.

Let's look at the effect on the file access performance of the overhead of the command interpretation operation when the average transaction size of the workload is increased in the environment where there is no contention for the system resources. First it is looked at in the distributed file system. When the 8Kbytes workload is used, the overhead takes 23.4% of the average response time in the distributed file system which consists of the Sun 3/60 workstations, 15.3% of the average response time in the distributed file system which consists of the Sun SPARCstation 470 workstations and 27.3% of the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations. When the 1856k bytes workload is used, the overhead of the command interpretation operation takes 0.5%, 0.24% and 0.5% of the average response time in the three distributed file systems respectively. It is found that as the average transaction size of the workload increases, the effect on the average response time decreases and becomes trivial. This is due to amortization since the overhead does not vary with the transaction size.

Now let's look at the effect of the overhead of the command interpretation operation on the file access performance when the average transaction size of the workload is increased in the baseline shared memory system where there is no contention for the system resources. The overhead in the shared memory system is the same as that in the distributed file system. When the 8Kbytes workload is used, the overhead takes 31.5% of the average response time in the Sun 3/60 workstation, 20.2% of the average response time in the Sun SPARCstation 470 workstation and 35.6% of the average response time in the Sun SPARCstation 10 workstation. When the 1856k bytes workload is used, the overhead takes 1.3%, 0.92% and 1.26% of the average response time in the three systems respectively. It is found that as the average transaction size of the workload increases, the effect on the average response time decreases and becomes trivial as in the distributed file system.

As the number of clients increases, the effect on the file access performance of the parameter quickly decreases to be trivial in the distributed file system because the command interpretation overhead is paid by the clients and has nothing to do with the queueing delay in the file server. But in the shared memory system, as the number of local users increases, the effect on the file access performance of the parameter increases due to the queueing delay.

It was observed that the average response time improved by the same amount as the decreased amount of the command interpretation overhead all the time when the parameter value was improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better respectively and the 8Kbytes workload, the 47bytes workload, the

50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload were used respectively in the distributed file system which consisted of the Sun SPARCstation 10 workstations. For example, the 10msec(50%) improvement in the command interpretation overhead always leads to 10msec improvement of the average response time regardless of the workload used and the number of clients. Therefore, the relative effect on the average response time becomes smaller when a workload with larger average transaction size or more clients is used even though the effect is significant when the 8Kbytes workload is used and there is very low contention in the system.

The effect on the average response time is a little larger in the shared memory system than in distributed file system since the overhead in the shared memory system is the same as that in the distributed file system and the overhead contributes to the queueing delay in the shared memory system unlike in the distributed file system.

Since the parameter is also one of the CPU service time parameters, the effect on the file access performance of the parameter should be always smaller than the effect on the file access performance of all CPU service time parameters which were already investigated in Section 6.2. This dissertation does not include these figures here.

6.10.4 Enhanced Screen Display Mechanism

If the read data are required to be displayed on the user screen or the designated window, then the screen display mechanism comes into paly and the additional overhead for the result processing for it should be paid. This section comparatively investigates the effect on the file access performance when the screen display overhead is improved both in the distributed file system and in the shared memory system.

The overhead is paid by the clients and has nothing to do with the queueing delay in the file server. The I/O time due to the screen display operation is 520msec in the the Sun 3/60 workstation, 100msec in the Sun SPARCstation 470 workstation and 22msec in the Sun SPARCstation 10 workstation. The ratio is 23.7 : 4.6 : 1 while the MIPS ratio in the three systems is 1 : 7.34 : 33.87. The value is proportional to the size of the transaction and therefore the effect of the overhead on the average response time overwhelms the other effects as the average size of the transaction increases.

Table 6.10.4.1 shows the overhead when the six workloads are used and there is no contention for the system resources in the Sun SPARCstation 10 workstation, in the Sun SPARCstation 470 workstation and in the Sun 3/60 workstation. The effect is considerable when the 1856Kbytes workload is used.

	I/O time (msec)			CPU time (msec)		
Workload	s360	s470	s10	s360	s470	s10
8k	2773.4	533.4	117.4	1.87	1.6	1.34
47.5k	16328	3133.4	689.4	10.97	9.4	7.84
50k	17576	3380	743.6	11.83	10.14	8.45
316k	109546.7	21000.7	4634.7	73.74	63.2	52.67
1856k	643413.4	123733.4	27221.4	433.07	371.2	309.34

Table 6.10.4.1 : The screen display overhead when there is no contention for the system resources.

In the distributed file system, as the number of clients increases, the effect on the file access performance of the parameter decreases because the overhead is paid by the clients and has nothing to do with the queueing delay in the file server. Since no queueing delay is correlated with the overhead, it is straight-forward to find out the average response time by simple calculations. The relative effect on the average response time becomes smaller when more clients use the system. However, the overhead is so large that it dominates the average response time. The effect on the average response time is larger in the shared memory system than in the distributed file system since the overhead in the shared memory systems is same as that in the distributed file system and the overhead contributes to the queueing delay in the shared memory system unlike in the distributed file system. This dissertation does not include the figures of the average response time when we improve the value of the screen display parameter in both system paradigms since I think the response time is too large to be considered when the 6 workloads are used.

6.11 Multiple Resources in the System

This section investigates the effect on the file access performance when more CPUs, more disks and more disk interface units are added at the same time in the file server of the distributed file system, that is, when multiple CPUs, multiple disks and multiple disk interface units are used all together in the file server of the distributed file system. This section also comparatively investigates the effect on the file access performance when multiple CPUs, multiple disks and multiple disk interface units are used in the shared memory system. The effect on the file access performance when multiple CPUs are used and the effect on the file access performance when multiple disks and multiple disk interface units are used were investigated in section 6.1 and in section 6.4 respectively. This section investigates the effect of the combination on the file access performance.

As the base system to which more system resources are added, the Sun SPARCstation 10 workstations are used in both system paradigms. The performance model of figure 3.2.6.C and figure 3.2.6.D and the baseline performance parameter values of table 3.2.7.C are used for the distributed file system and the performance model of figure 3.4.1.B and the baseline performance parameter values of table 3.4.2.A are used for the shared memory system. Each group of the multiple resources is represented as multiple servers which share a queue in the performance models. Each service center is assumed to serve with equal opportunity since each group of the multiple resources is assumed to have the symmetric property. The overhead to manage the multiple resources is assumed to be negligible, which means this study considers the theoretical limit.

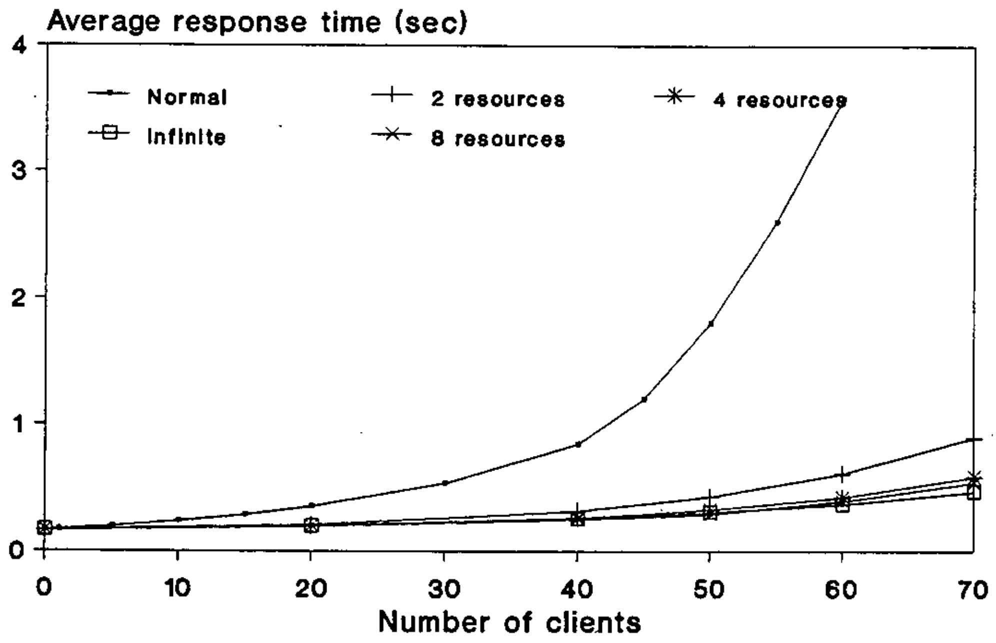


Figure 6.11.1 : The effect on the average response time of having multiple resources in the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

Figure 6.11.1 shows the average response time of the 50.7Kbytes workload in the distributed file system which consists of the Sun SPARCstation 10 workstations as the number of clients increases gradually. The number of resources in the file server is increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity. Except for these, all others are kept the same as the baseline distributed file system. Figure 6.11.2 shows the average response time of the 50.7Kbytes workload in the shared memory system of the Sun SPARCstation 10 workstation as the number of local users increases gradually. The number of resources is increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity. Except for these, all others are kept the same as the baseline shared memory system. See appendix C for the figures of other cases.

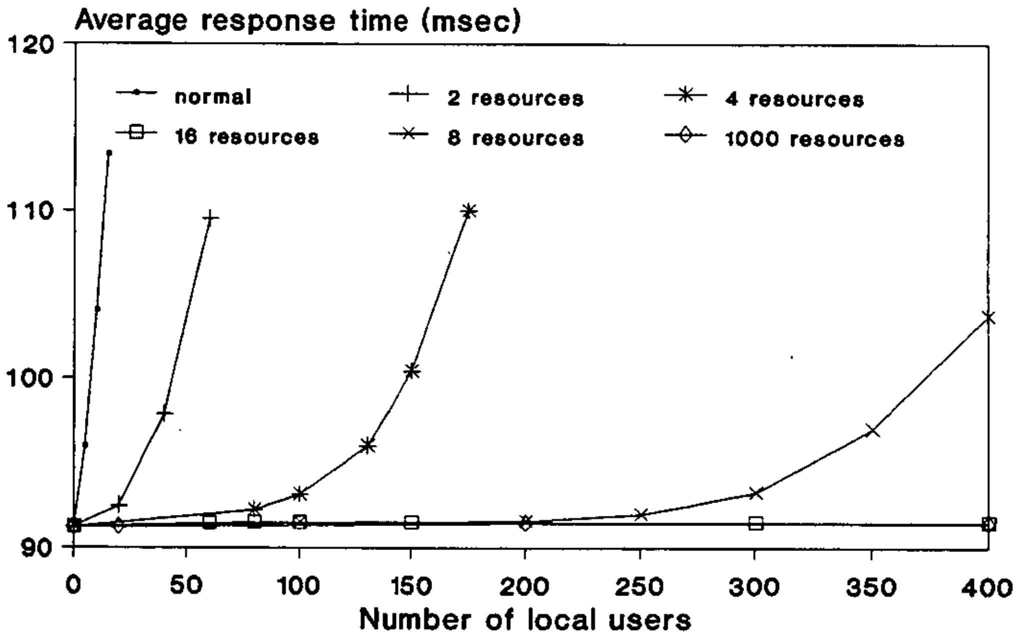


Figure 6.11.2 : The effect on the average response time of having multiple resources in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

It is observed that the distributed file system which has 2 resources improves the average response time most efficiently so the best performance/cost can be obtained in the environment and the contention for the system resources almost disappears in the distributed file system which has 4 resources. Therefore, putting more resources to the distributed file system which has 4 resources already improves the average response time little.

No notable change is observed in the pattern of the average response time as the average transaction size increases. Neither is any notable difference observed between the patterns for the average response times when steady workloads are used and the patterns for the average response times when bursty workloads are used.

The notable difference between the figures for the distributed file system and the figures for the shared memory system is that in the figures for the distributed file system the number of clients which saturates the distributed file system does not increase much since the 10Mbps network remains as the major bottleneck point, even though the overall improvement of the average response time is significant, but in the shared memory system the number of local users which saturates the shared memory system increases almost linearly as the degree of multiplicity increases.

6.12 Better System

This section investigates the effect on the file access performance comparatively when better systems are used in the distributed file system and in the shared memory system, for example, when the Sun SPARCstation 10 workstations are replaced with better component systems in the baseline distributed file system and in the baseline shared memory system. In this case, all the performance parameters both in the file server and in the clients of the distributed file system except the parameters of the network communication in table 3.2.7.C are improved at the same time and all the performance parameters in the shared memory system in table 3.4.2.A are improved at the same time.

The effect on the file access performance when the performance parameters are improved separately one by one or group by group, were already investigated in previous sections. This section investigates the effect of combinations on the file access performance. As the base system where all parameter values are improved at the same time, the Sun SPARCstation 10 workstation is used in the two system paradigms. The baseline performance model of figure 3.2.6.B and the modified performance parameter values based on table 3.2.7.C are used for the distributed file system and the baseline performance model of figure 3.4.1.B. and the modified performance parameter values based on table 3.4.2.A are used for the shared memory system.

Figure 6.12.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The values of all parameters except the network transmission speed in table 3.2.7.C are improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix C for the figures of other cases.

We observe that the distributed file system where all parameter values except the parameter of the network speed are improved to be 2 times better shows the best performance/cost. Until the degree of improvement reaches eight, the average response time improves by a reasonable amount. No notable change is observed in the pattern of the average response time as the average transaction size increases.

Neither is any notable difference observed between the patterns for the average response times when steady workloads are used and the patterns for the average response times when bursty workloads are used.

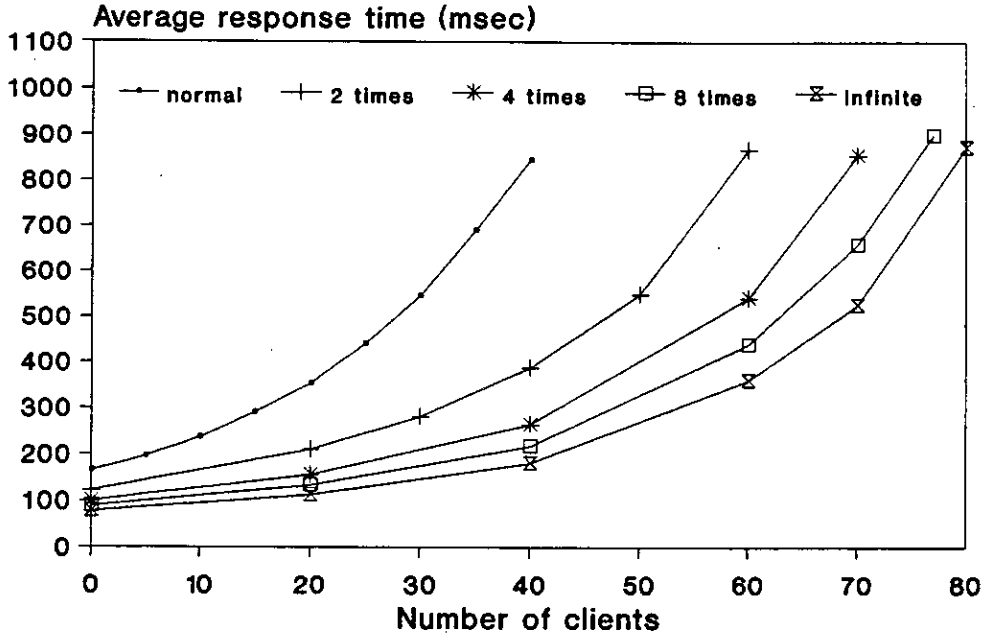


Figure 6.12.1 : The effect of the better system on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

Figure 6.12.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The values of all parameters are improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for these, all others are kept the same as the baseline Sun SPARCstation 10 workstation. See appendix C for the figures of other cases.

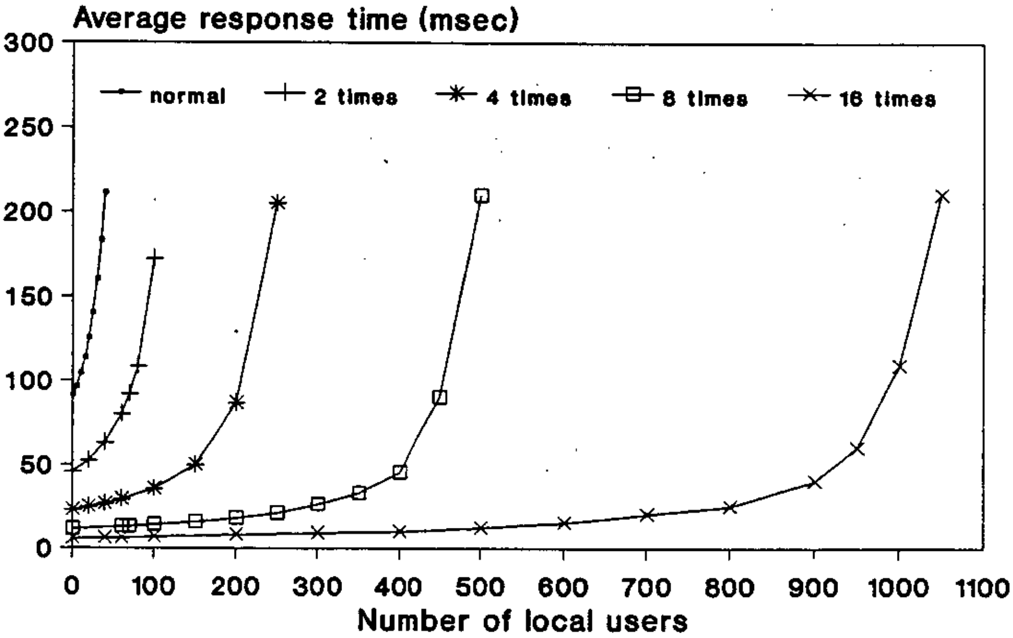


Figure 6.12.2 : The effect of the better system on the average response time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

In the figures for the shared memory system, the regular improvement in the average response time is observed as the degree of improvement increases unlike in the figures for the distributed file system. No notable change is observed in the pattern of the average response time as the average transaction size increases like in the distributed file system. Neither is any notable difference observed between the patterns for the average response times when steady workloads are used and the patterns for the average response times when bursty workloads are used as in the distributed file system.

The notable difference between the figures for the distributed file system and the figures for the shared memory system is that in the figures for the distributed file system, the saturation point does not increase much since the 10Mbps network remains as the major bottleneck point, even though the overall improvement of the average response time is significant, but in the shared memory system the saturation point increases almost linearly as the degree of improvement increases since the parameter values of the bottleneck resource improve at the same time.

Figure 6.12.3 to figure 6.12.8 compare the average response times of the three shared memory systems when the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are used respectively. The three shared memory systems are the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 workstation.

In the figures, let's investigate what relationship exists between the MIPS value and the file access performance. In order to look at the accuracy of the MIPS values of the computer systems used in this study, the confidence of the MIPS value of a system is defined as the following. The MIPS value of the computer system is normalized to the MIPS value of a baseline computer system and the average response time of the computer system is normalized to the average response times of the baseline computer system. If the inverse of the normalized MIPS value is the same as the normalized average response time of the file access request when there is no contention for the system resources, then the confidence of the normalized MIPS value in the file access performance is defined to be 100%. If the inverse of the normalized MIPS value is not found in the normalized average response times until the system saturates due to contention, then the confidence of the normalized MIPS value in the file access performance is defined

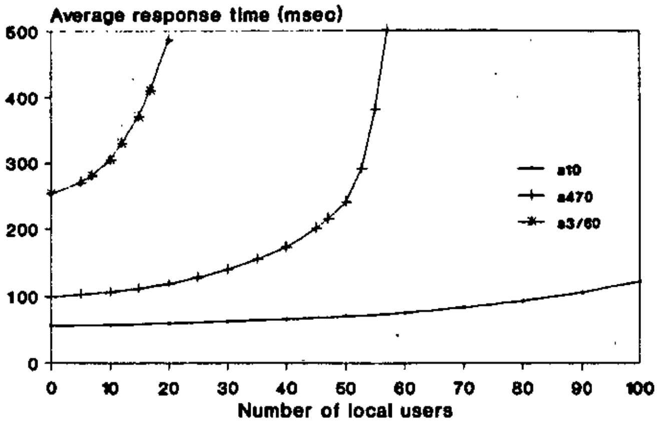


Figure 6.12.3 : 8Kbytes

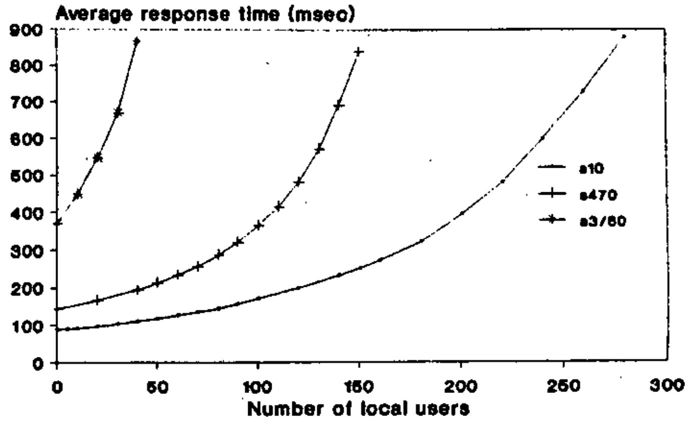


Figure 6.12.4 : 47Kbytes

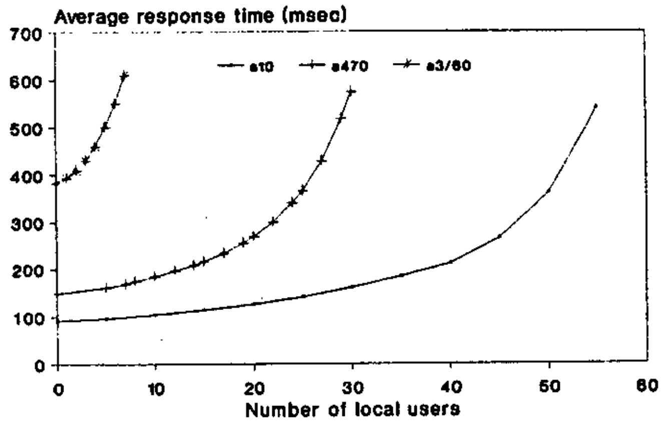


Figure 6.12.5 : 50.7Kbytes

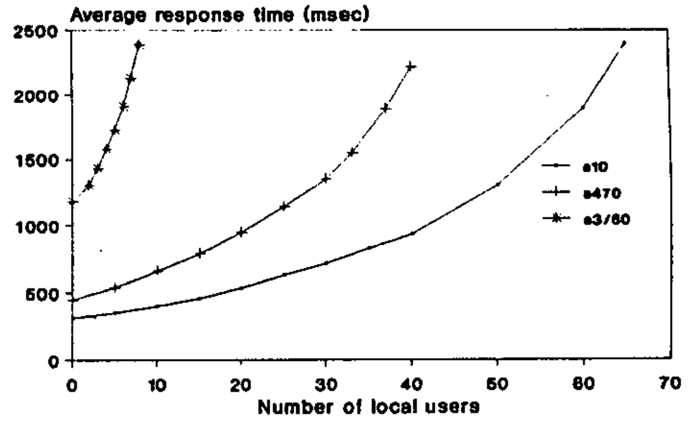


Figure 6.12.6 : 316Kbytes(B)

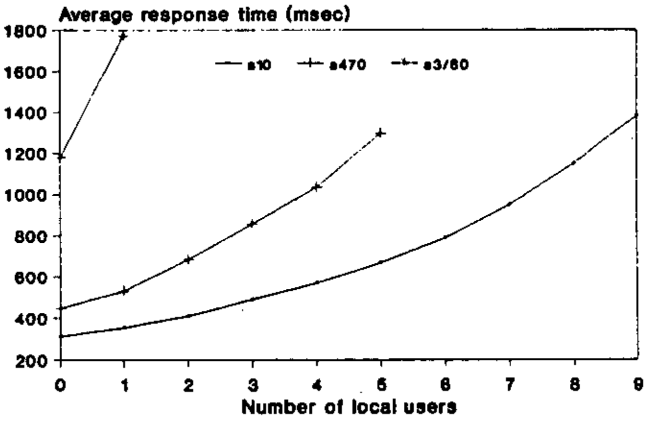


Figure 6.12.7 : 316Kbytes

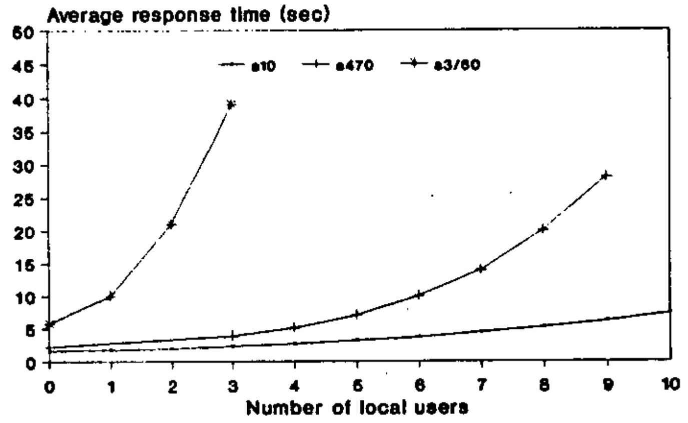


Figure 6.12.8 : 1856Kbytes

The average response time in the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 workstation.

to be 0%.

If the inverse of the normalized MIPS value is found in the normalized average response times when the contention for the system resources is acceptable or below the acceptable level, that is, the utilization of system resources is acceptable, then the confidence of the normalized MIPS value in the file access performance is defined to be acceptable. Otherwise, the confidence of the normalized MIPS value in the file access performance is defined to be unacceptable. For the acceptable level of the utilization, this study uses what the rule of thumb in computing practice commonly tells. According to the rule of thumb in computing practice, the utilization of the disk I/O subsystem and the utilization of the communication facilities are recommended not to exceed an average 40% to 50% while the utilization of the CPU is not limited up to 100% for acceptable performance.

	s3/60	s470	s10	2 times	4 times	8 times
8k	4.57	1.78	1	0.5	0.25	0.125
47k	4.22	1.63	1	0.5	0.25	0.125
50.7k	4.2	1.63	1	0.5	0.25	0.125
316k	3.79	1.44	1	0.5	0.25	0.125
1856k	3.65	1.38	1	0.5	0.25	0.125

Table 6.12.1 : The average response time in the three shared memory systems when there is no contention for the system resources, normalized to the response time in the Sun SPARCstation 10 workstation when there is no contention for the system resources.

Now let's look at the figures. The MIPS ratio among the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 workstation is 33.87 : 7.34 : 1. The ratios of the average response times of the shared memory systems to the Sun SPARCstation 10 workstation when there is no contention for the system resources is shown in table 6.12.1. It is found that the confidence of the MIPS ratio among the three systems in the file access performance is never

100% whatever workload is used.

Let's look at the average response time in the figures when contention for the system resources exists. First we look at the average response time when the 8Kbytes workload is used. With 55 local users where the utilization of the CPU and the disk I/O subsystem are 13.6% and 87.4% respectively, the average response time of the Sun SPARCstation 470 workstation normalized to that of the Sun SPARCstation 10 workstation is 5.3 which is a little larger than the inverse of the normalized MIPS value(4.62). At near the saturation point, the average response time of the Sun 3/60 workstation normalized to that of the Sun SPARCstation 10 workstation becomes the same as the MIPS ratio of the Sun SPARCstation 10 workstation to the Sun 3/60 workstation. Therefore the confidence of the normalized MIPS values of the three systems in the file access performance is said to be low or unacceptable when the 8Kbytes workload is used.

Second we look at the average response time when the 47Kbytes workload is used in the figures. At 150 local users where the utilization of the CPU and the disk I/O subsystem are 8.9% and 68.2% respectively, the normalized average response time of the Sun SPARCstation 470 workstation is 3.36 which is smaller than the inverse of the normalized MIPS value(4.62). At 80 local users where the utilization of the CPU and the disk I/O subsystem are 10.5% and 87.2% respectively, the normalized average response time of the Sun 3/60 workstation is 34.25 which is larger than the inverse of the normalized MIPS value(33.87). Therefore the confidence of the normalized MIPS values of the three systems in the file access performance is said to be low or unacceptable when the 47Kbytes workload is used.

Third, we look at the average response time when the 50Kbytes workload is used, when the 316Kbytes(B) workload is used, when the 316Kbytes workload is used and when the 1856Kbytes workload is used in the figures. In all figures, at near the saturation point, the normalized average response time becomes the same as the inverse of the normalized MIPS value. Therefore the confidence of the normalized MIPS values of the three systems in the file access performance is said to be low or unacceptable when the 50Kbytes workload is used, when the 316Kbytes(B) workload is used, when the 316Kbytes workload is used and when the 1856Kbytes workload is used.

From the investigation, it is concluded that the confidence of the normalized MIPS values of the three systems in the file access performance is low or unacceptable regardless of the workload used.

Figure 6.12.9 to figure 6.12.14 compare the average response time of the three distributed file systems when the systems are supplied with the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively. The three distributed file systems are the distributed file system which consists of the Sun SPARCstation 10 workstations, the distributed file system which consists of the Sun SPARCstation 470 workstations and the distributed file system which consists of the Sun 3/60 workstations.

	s3/60	s470	s10	2 times	4 times	8 times
8k	4.7	2	1	0.73	0.59	0.52
47k	4.9	2	1	0.74	0.61	0.54
50.7k	4.91	2	1	0.74	0.61	0.54
316k	5.04	2.05	1	0.75	0.62	0.56
1856k	5.08	2.06	1	0.75	0.62	0.56

Table 6.12.2 : The normalized average response time in the distributed file systems when there is no contention for the system resources.

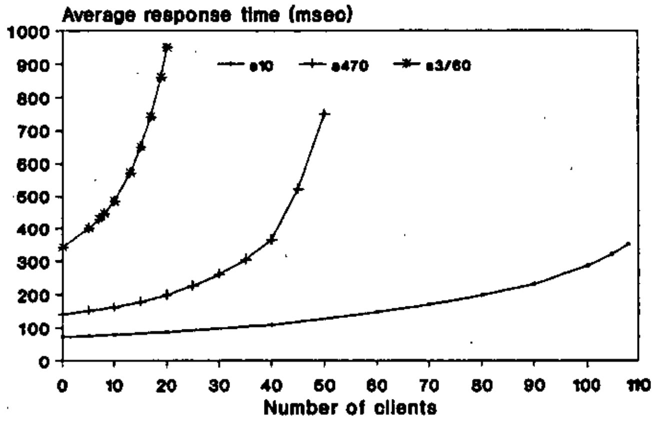


Figure 6.12.9 : 8Kbytes

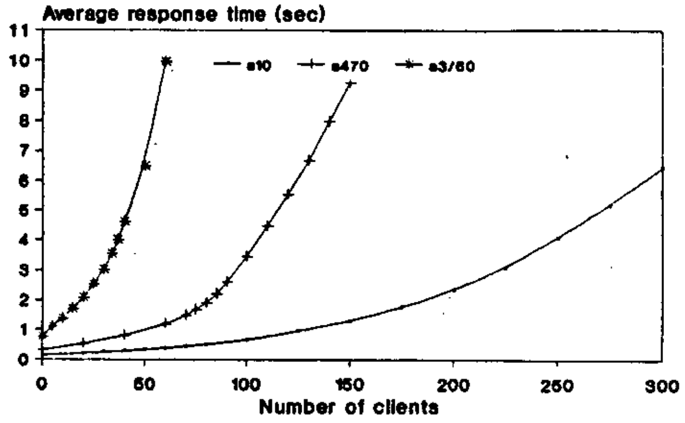


Figure 6.12.10 : 47Kbytes

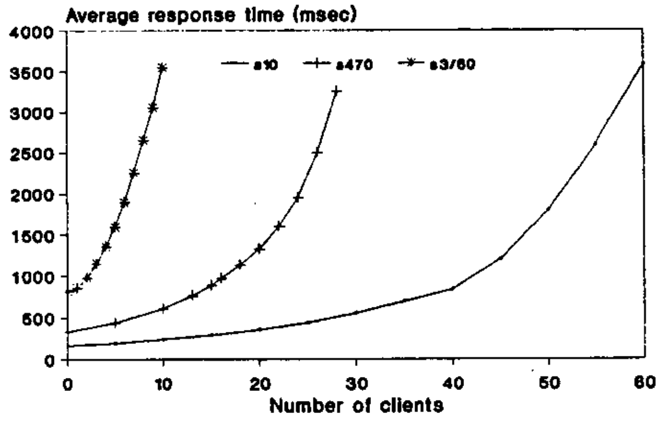


Figure 6.12.11 : 50.7Kbytes

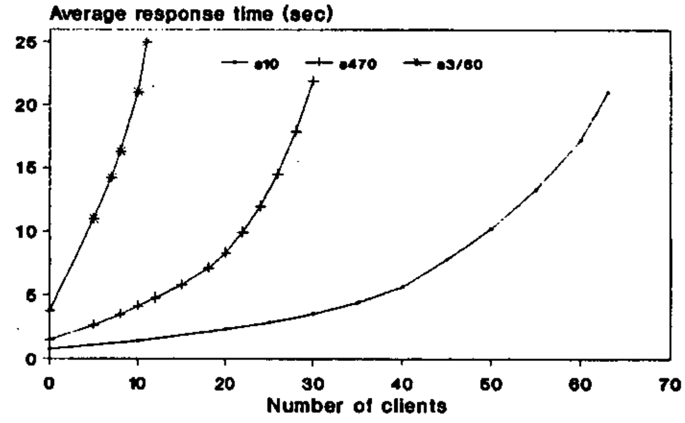


Figure 6.12.12 : 316Kbytes(B)

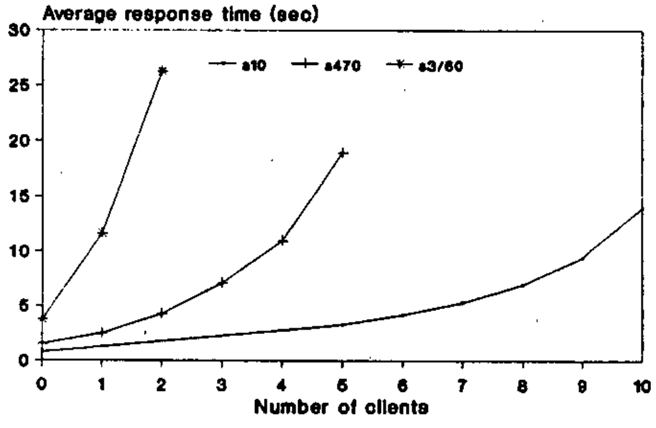


Figure 6.12.13 : 316Kbytes

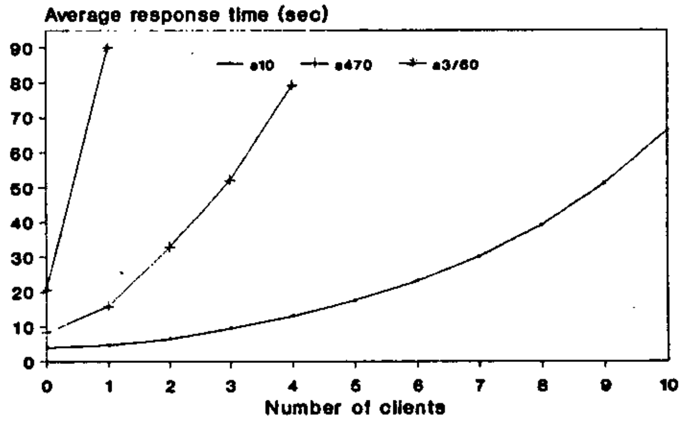


Figure 6.12.14 : 1856Kbytes

The average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations, the distributed file system which consists of the Sun SPARCstation 470 workstations and the distributed file system which consists of the Sun 3/60 workstation.

Let's investigate what relationship exists between the MIPS value and the file access performance in the figures for the distributed file systems as we did in the shared memory systems previously. First, we investigate it when there is no contention for the system resources. Table 6.12.2 shows the average response time of the distributed file systems when there is no contention for the system resources normalized to the average response time of the baseline distributed file system which consists of the Sun SPARCstation 10 workstations when there is no contention for the system resources. It is found that the confidence of the normalized MIPS values of the component systems in the file access performance is never 100% whatever workload is used when there is no contention for the system resources.

Second we investigate the relationship when the contention for the system resources exists. Let's look at the average response time when the 8Kbytes workload is used in the figures. At 45 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 17.1%, 71.5%, 21.3% and 7.8% respectively, the average response time in the distributed file system which consists of the Sun SPARCstation 470 workstations normalized to the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations is 4.62 which is the inverse of the normalized MIPS value(4.62). At near the saturation point, the average response time of the distributed file system which consists of the Sun 3/60 workstations normalized to the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations becomes the same as the inverse of the normalized MIPS value. Therefore the confidence of the normalized MIPS values of the component systems of the three distributed file systems in the file access performance is said to be low or unacceptable when the 8Kbytes workload is used.

Let's look at the average response time when the 47Kbytes workload is used. At

100 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 9.5%, 45.5%, 40.2% and 15.6% respectively, the normalized average response time of the distributed file system which consists of the Sun SPARCstation 470 workstations is 5.29 which is larger than the inverse of the normalized MIPS value(4.62). At 70 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 12.3%, 76.1%, 60% and 10.9% respectively, the normalized average response time of the distributed file system which consists of the Sun 3/60 workstations is 38.21 which is larger than the inverse of the normalized MIPS value(33.87). Therefore the confidence of the MIPS ratio between the component systems of the former two distributed file systems in the file access performance is said to be high and acceptable but the confidence of the normalized MIPS ratio between the component systems of the latter two distributed file systems is said to be low or unacceptable when the 47Kbytes workload is used.

Let's look at the average response time when the 50.7Kbytes workload is used. At 25 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 13.9%, 67.6%, 62.5% and 23.9% respectively, the normalized average response time of the distributed file system which consists of the Sun SPARCstation 470 workstations is 5 which is larger than the inverse of the normalized MIPS value(4.62). At near the saturation point, the normalized average response time of the distributed file system which consists of the Sun 3/60 workstations becomes the same as the inverse of the normalized MIPS value. Therefore the confidence of the normalized MIPS values of the three component systems in the file access performance is said to be low or unacceptable when the 50Kbytes workload is used.

Let's look at the the average response time when the 316Kbytes workload(B) is used. At 25 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 7.4%, 42.4%, 65.8% and 25.6%

respectively, the normalized average response time of the distributed file system which consists of the Sun SPARCstation 470 workstations is 4.65 which is slightly larger than the inverse of the normalized MIPS value(4.62). The normalized average response time of the distributed file system which consists of the Sun 3/60 workstations become never the same as the inverse of the normalized MIPS value. Therefore the confidence of the normalized MIPS ratio between the Sun SPARCstation 10 workstation and the Sun SPARCstation 470 workstation is said to be low or unacceptable in the file access performance of the two distributed file systems and the confidence of the MIPS ratio between the Sun SPARCstation 10 workstation and the Sun 3/60 workstation is said to be 0% in the file access performances of the two distributed file systems when the 316Kbytes(B) workload is used.

Let's look at the average response time when the 316Kbytes workload is used. At 4 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 6.7%, 40%, 66.7% and 23.6% respectively, the normalized average response time of the distributed file system which consists of the Sun SPARCstation 470 workstations is 4.19 which is smaller than the inverse of the normalized MIPS value(4.62). The normalized average response time of distributed file system which consists of the Sun 3/60 workstations becomes the same as the inverse of the normalized MIPS value near the saturation point. Therefore the confidence of the normalized MIPS values of the three component systems in the file access performance is said to be low or unacceptable when the 316Kbytes workload is used.

Let's look at the average response time when the 1856Kbytes workload is used. At 2 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 2.9%, 18.2%, 33.4% and 12.5% respectively, the normalized average response time of the distributed file system which consists of the Sun SPARCstation 470 workstation is 5.04 which is larger than the inverse of the normalized MIPS value(4.62). At 2 clients where the utilization of the CPU, the disk I/O subsystem, the network interface unit and the network is 5.6%, 50%, 66.7% and 12.5% respectively, the normalized average response time of the distributed file system which consists of the Sun 3/60 workstations is 45.05 which is larger than the inverse of the normalized MIPS value(33.87). Therefore the confidence of the MIPS ratio between the Sun SPARCstation 10 workstation and the Sun SPARCstation 470 workstation is said to be high or acceptable in the file access performances of the two distributed file systems and the confidence of the MIPS ratio between the Sun SPARCstation 10 workstation and the Sun 3/60 workstation is said to be low or unacceptable in the file access performances of the two distributed file systems when the 1856Kbytes workload is used.

From the investigation, it is observed that in the distributed file system which consists of the Sun SPARCstation 10 workstations and the distributed file system which consists of the Sun SPARCstation 470 workstations, the confidence of the normalized MIPS values is high or acceptable when the 47Kbytes workload or the 1856Kbytes workload is used and low or unacceptable when any one of the other four workloads is used. The normalized MIPS value of the distributed file system which consists of the Sun 3/60 workstations is observed to have low or zero confidence in file access performance. Generally, the confidence of the normalized MIPS values in file access performance is observed to be better in the distributed file systems than in the shared memory systems.

6.13 Multiple File Servers

This section investigates the effect on the file access performance when the distributed file system has multiple file servers. Files are assumed to be replicated in the file servers and the file replication overhead in the multiple file servers is assumed to be negligible, which is the best theoretical case. The file servers are assumed to be homogeneous. The performance model of figure 3.2.6.G and the baseline performance parameter values of table 3.2.7.C are used for the distributed file system which consists of the Sun SPARCstation 10 workstations. In the model, each file server is assumed to serve the incoming requests with equal opportunity.

Figure 6.13.1 to figure 6.13.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the number of file servers is increased to be 2, 4, 6, 8, 10, 14, 16, 20, 24, and 27. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure 6.13.7 and figure 6.13.8 show the average response time of the 50.7Kbytes workload and the 316Kbytes(B) workload respectively when a 100Mbps network is used instead of a 10Mbps network and figure 6.13.9 and figure 6.13.10 show the average response time of the 316Kbytes workload and the 1856Kbytes workload respectively when a 1Gbps network is used instead of a 10Mbps network.

It is observed that when 2 file servers are used the distributed file system shows the best performance/cost and the improved amount of the average response time between when 4 file servers are used and when 27 file servers are used is same as that between when 2 file servers are used and when 4 file servers are used. This is due to the network speed limit. Therefore, it is efficient in terms of performance/cost to use up to 4 file servers in the 10Mbps LAN environment.

Let's check it when the 47Kbytes workload is used. Within 500msec average response time, the distributed file system which has one file server supports up to 80 clients, two file severs up to 120 clients, four file servers up to 160 clients, six file servers up to 180 clients, eight file servers up to 185 clients, 10 file servers up

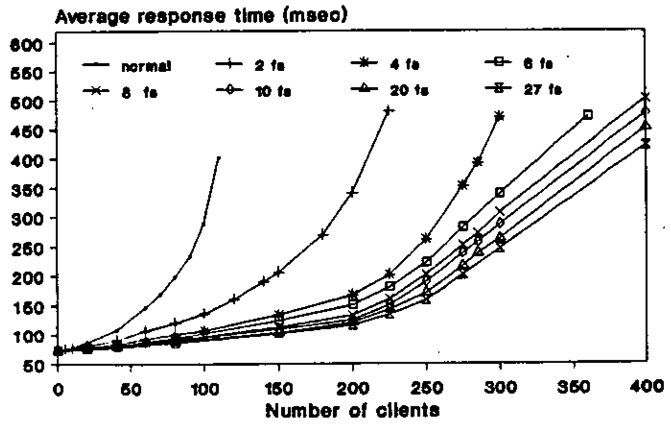


Figure 6.13.1 : 8Kbytes

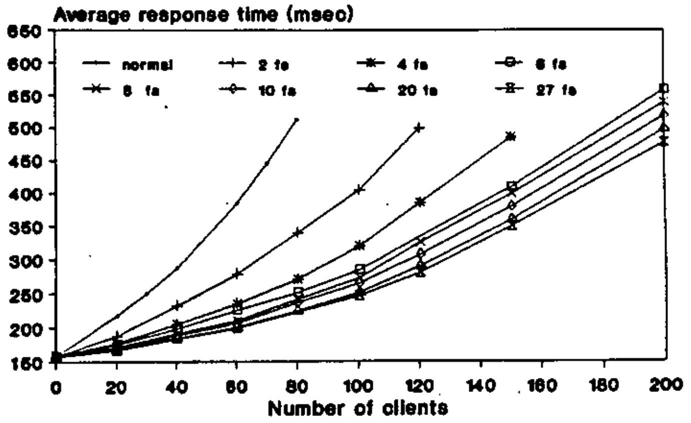


Figure 6.13.2 : 47Kbytes

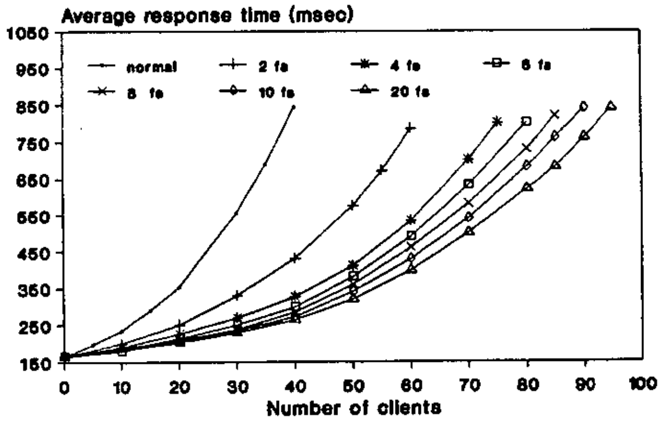


Figure 6.13.3 : 50.7Kbytes

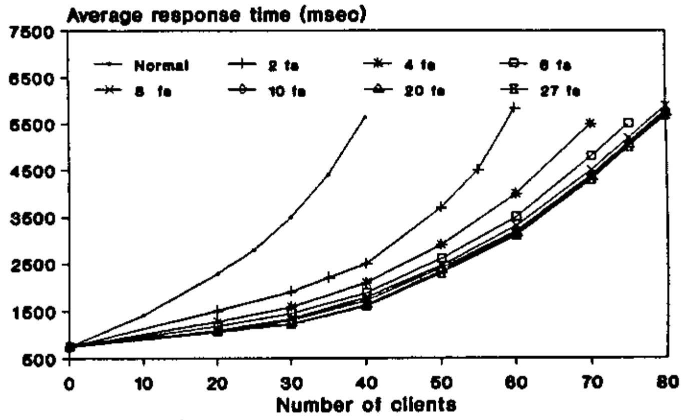


Figure 6.13.4 : 316Kbytes(B)

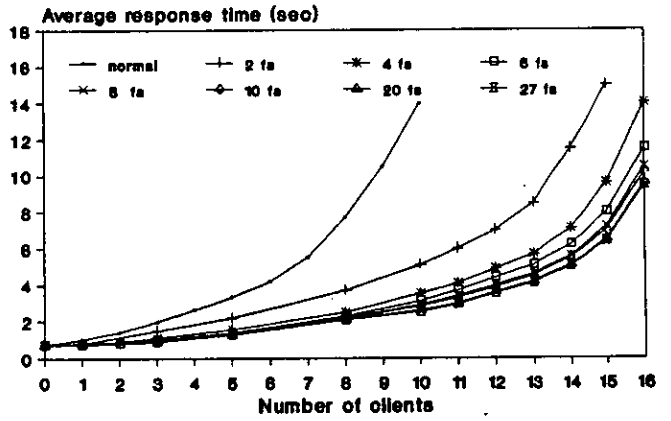


Figure 6.13.5 : 316Kbytes

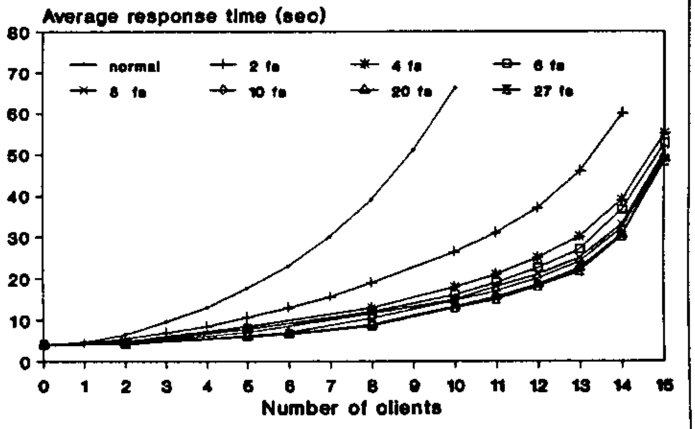


Figure 6.13.6 : 1856Kbytes

The effect of having multiple file servers on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations via the 10Mbps LAN.

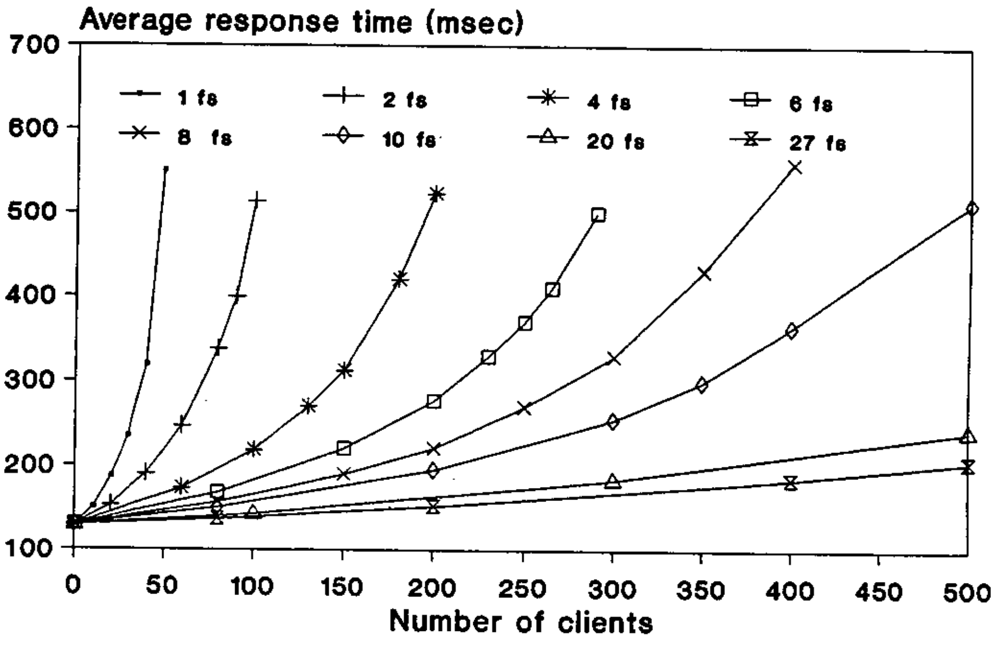


Figure 6.13.7 : 50.7Kbytes, 100Mbps

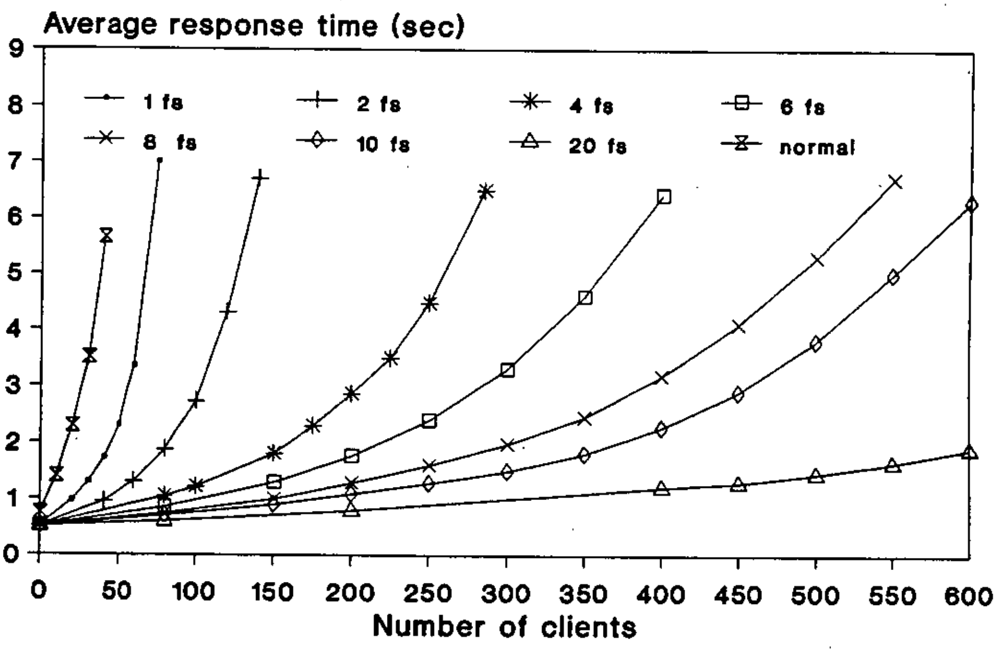


Figure 6.13.8 : 316Kbytes(B), 100Mbps

The effect of having multiple file servers on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations via the 100Mbps LAN.

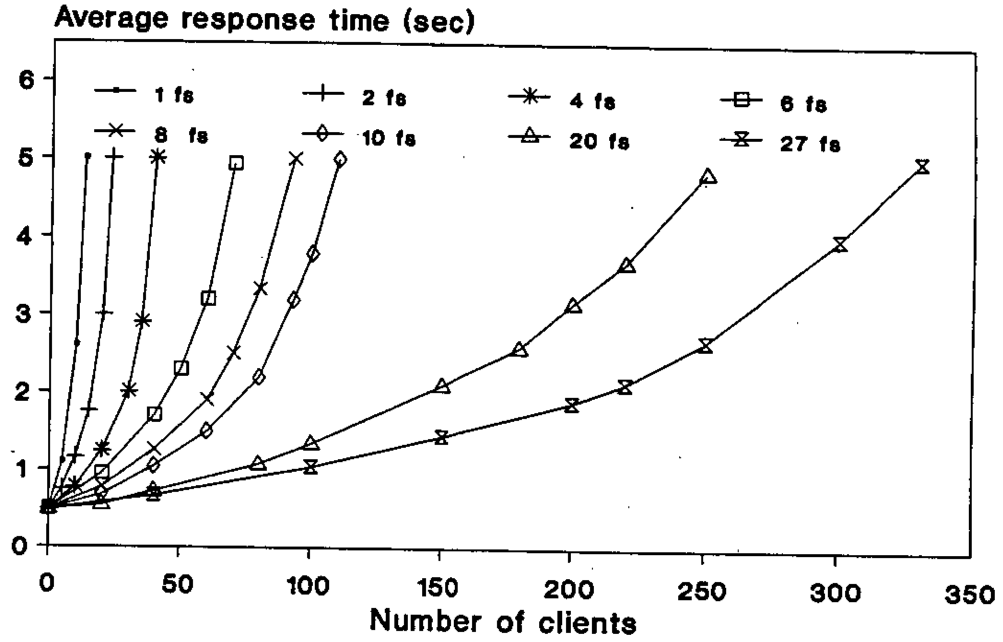


Figure 6.13.9 : 316Kbytes, 1Gbps

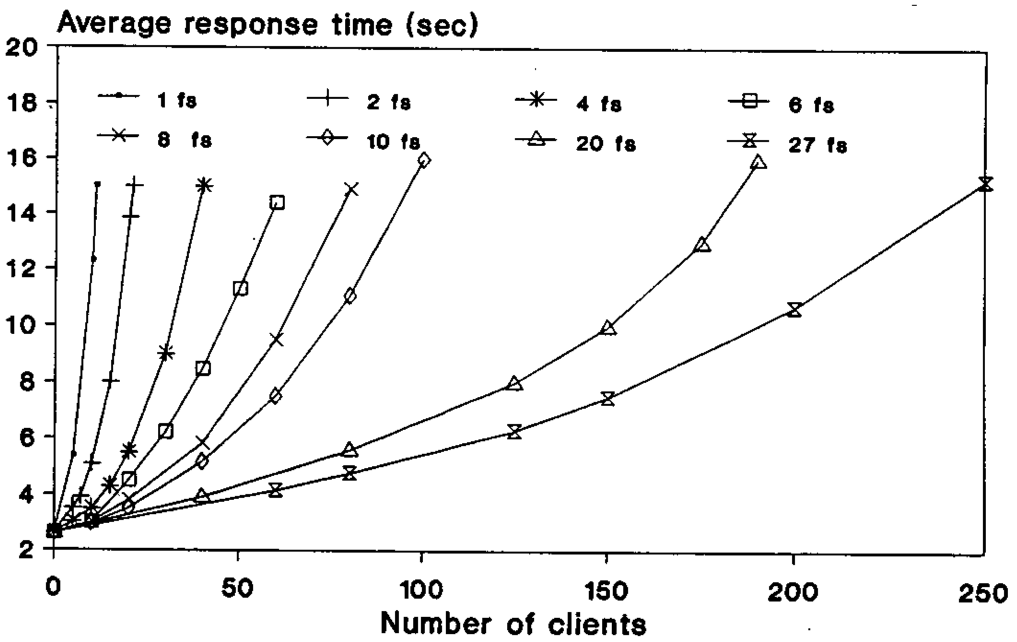


Figure 6.13.10 : 1856Kbytes, 1Gbps

The effect of having multiple file servers on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations via the 1Gbps LAN.

to 195 clients, 20 file servers up to 200 clients and 27 file servers up to 210 clients.

Let's check it when the 50Kbytes workload is used. Within 750msec average response time, the distributed file system which has one file server supports up to 35 clients, two file severs up to 60 clients, four file servers up to 73 clients, 6 file servers up to 78 clients, eight file servers up to 82 clients and 10 file servers up to 85 clients and 20 file servers up to 90 clients.

Let's check it when the 316Kbytes(B) workload is used. Within 3.5 second average response time, the distributed file system which has one file server supports up to 30 clients, two file severs up to 50 clients, four file servers up to 55 clients, six file servers up to 60 clients and 27 file servers up to 64 clients.

No notable change is observed in the pattern of the average response time as the average transaction size increases. Neither is any notable difference observed in the patterns for the average response times when steady workloads are used and those when bursty workloads are used.

When a 100Mbps network is used for the 50.7Kbytes workload and the 316Kbytes(B) workload, it is observed that the average response time improves more evenly than when a 10Mbps network is used as we expect. The average response times of the two workloads are within 3 seconds, which is generally known as the maximum response time the users can wait even though they do not have patiency. The average response time of the 50.7Kbytes workload is within 1 second up to a reasonable number of clients. In this sense, I think that a 100Mbps network is desirable for the distributed file system which has multiple file servers when one of the two workloads is used. No notable change is observed in the pattern of the average response time as the average transaction size increases. Neither is any notable difference observed between the patterns for the average response times when steady workloads are used and the patterns for the average response times when bursty workloads are used.

When a 1Gbps network is used for the 316Kbytes workload and the 1856Kbytes workload, it is observed that up to 27 file servers, the average response time improves almost linearly. It improves much more evenly than when we use a 10Mbps network, as we expect. The average response times of the two workloads are within 3 seconds up to a reasonable number of clients. The figures show that a 1Gbps network is desirable for the environment which has multiple file servers when one of the two workloads is used. No notable change is observed in the pattern of the average response time as the average transaction size increases. Neither is any notable difference observed between the patterns for the average response times when steady workloads are used and the patterns for the average response times when bursty workloads are used.

6.14 Multiple Resources in the File Server vs. Better File Server vs. Multiple File Servers.

This section compares the file access performance of the distributed file system which has multiple resources in the file server, that which has a better file server and that which has multiple file servers. In order to compare them fairly, only the file server is changed in this section. Therefore, the heterogeneous distributed file system which has the better file server in this section is different from the homogeneous distributed file system which has the better file server and the better clients in section 6.12. In order to compare them fairly, I put multiple network interface units as well as multiple CPUs, multiple disks and multiple disk interface units in the file server for the multiple resources case. However, putting more than two network interface units in the file server does not improve the system performance further since up to two network interface units are utilized unless multiple networks are provided. That is, one of the two network interface units is used for the incoming data from the clients and the other is used for the outgoing data to the clients. In this section, the enhancement of the distributed file system is done based on the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure 6.14.1 to figure 6.14.6 compare the average response when the system has two CPUs, two disks, two disk interface units and two network interface units at the same time in the file server, when the system has a two times better file server and when the system has two file servers. The two times better file server means all performance parameters in the file server are improved to be two times better. The average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are shown respectively as the number of clients increases gradually in the figures.

In each figure of all 6 workloads, there exists at least one crossing point. The first crossing point occurs between the multiple resources case and the better system case since the average response time of the better system case develops faster than that of the multiple resources case. This mean the average response time of the distributed file system which has the better file server is more sensitive to the number of clients than the average response time of the distributed file system whose file server has multiple resources.

The crossing point occurs at 73 clients when the 8Kbytes workload is used, at 30 clients when the 47Kbytes workload is used, at 20 clients when the 50.7Kbytes workload is used, at 10 clients when the 316Kbytes(B) workload is used, at 1.5 clients when the 316Kbytes workload is used and at 1.5 clients when the 1856Kbytes workload is used. It is observed that the crossing point occurs at fewer clients as the average transaction size increases.

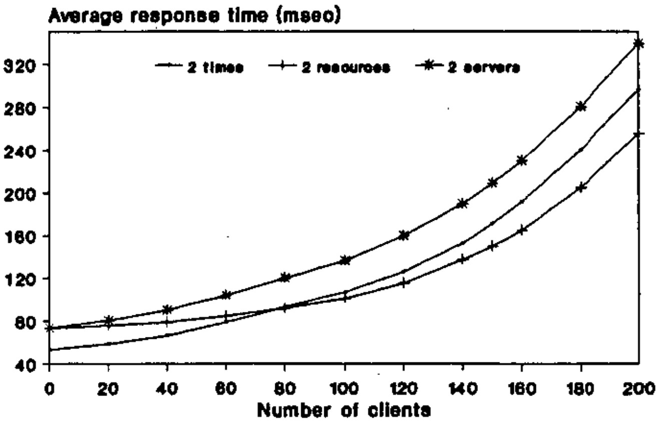


Figure 6.14.1 : 8Kbytes

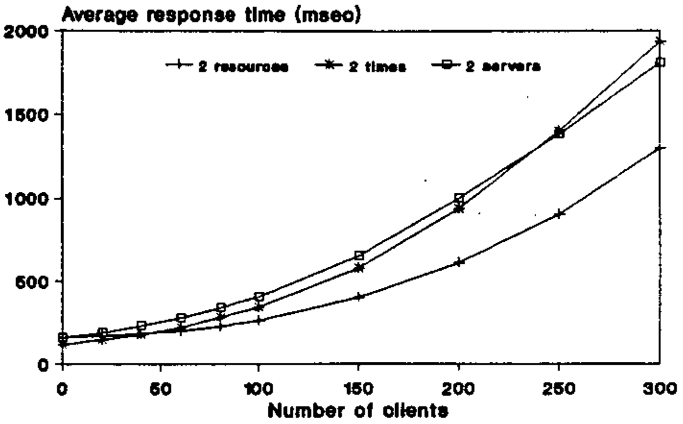


Figure 6.14.2 : 47Kbytes

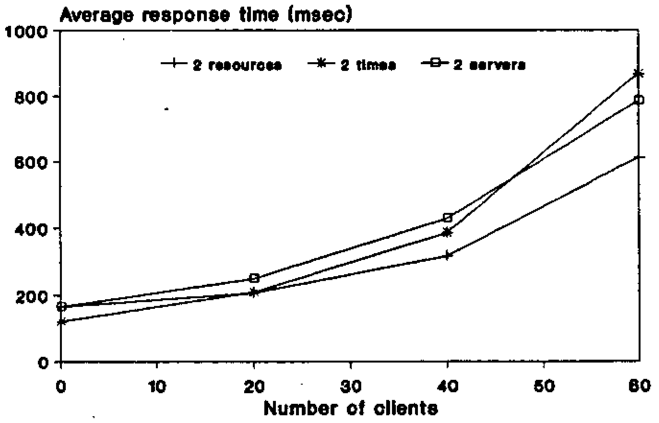


Figure 6.14.3 : 50.7Kbytes

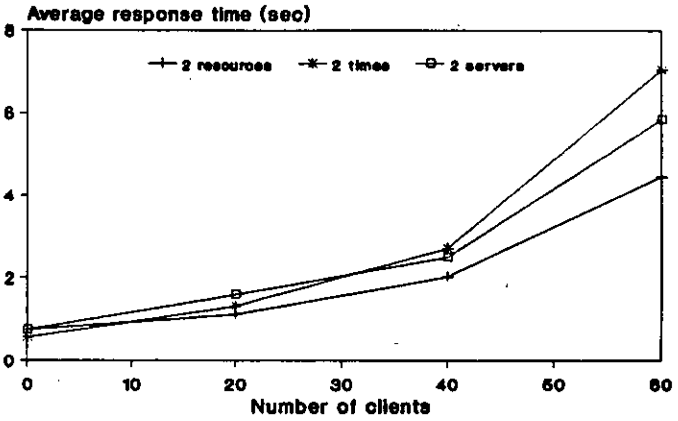


Figure 6.14.4 : 316Kbytes(B)

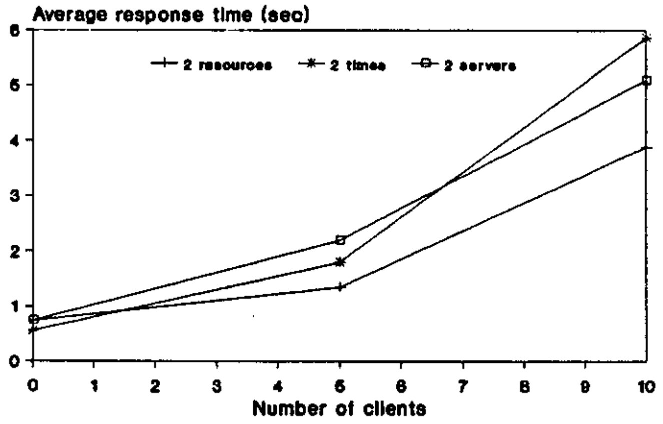


Figure 6.14.5 : 316Kbytes

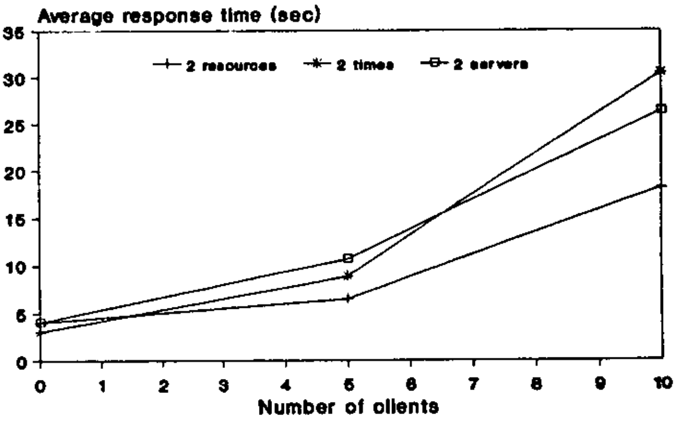


Figure 6.14.6 : 1856Kbytes

The average response time of the two resources case vs. the average response time of the two times better case vs. the average response time of the two file servers case in the distributed file system which consists of the Sun SPARCstation 10 workstations.

It is notable that except in figure 6.14.1, that is, when the 8Kbytes workload is used, in each figure, there exist two crossing points. At the first crossing point, the line of the average response time of the better file server case intersects the line of the average response time of the multiple resources case and at the second crossing point, the line of the better file server case intersects the line of the multiple file servers case. Therefore beyond the second crossing point, the average response time of the better file server case becomes the worst. This means the average response time of the distributed file system which has the better file server is most sensitive to the number of clients among the three cases.

Figure 6.14.7 to figure 6.14.12 compare the average response when the system has 4 CPUs, 4 disks, 4 disk interface units and 4 network interface units at the same time in the file server, when the system has a 4 times better file server and when the system has 4 file servers. The average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are shown respectively as the number of clients increases gradually in the figures.

In each figure of all 6 workloads, there exists at least one crossing point as in figure 6.14.1 to figure 6.14.6. The first crossing point occurs between the multiple resources case and the better file server case since the average response time of the better system case develops faster than that of the multiple file servers case.

We also see that the crossing point occurs at fewer clients as the average transaction size increases. This happens since the average response time of the distributed file system which has the better file server is more sensitive to the average transaction size than the average response time of the distributed file system whose file server has multiple resources.

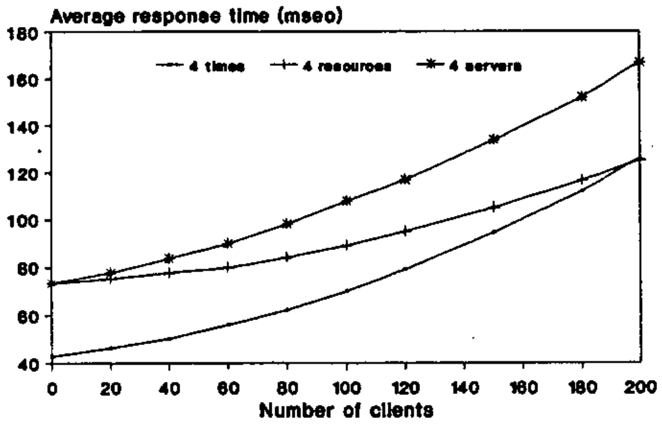


Figure 6.14.7 : 8Kbytes

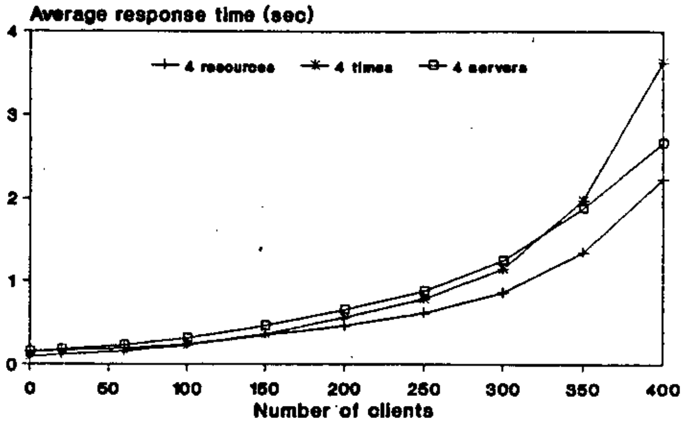


Figure 6.14.8 : 47Kbytes

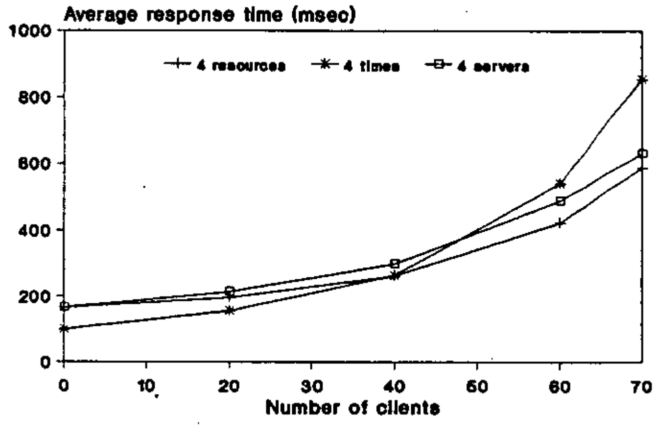


Figure 6.14.9 : 50.7Kbytes

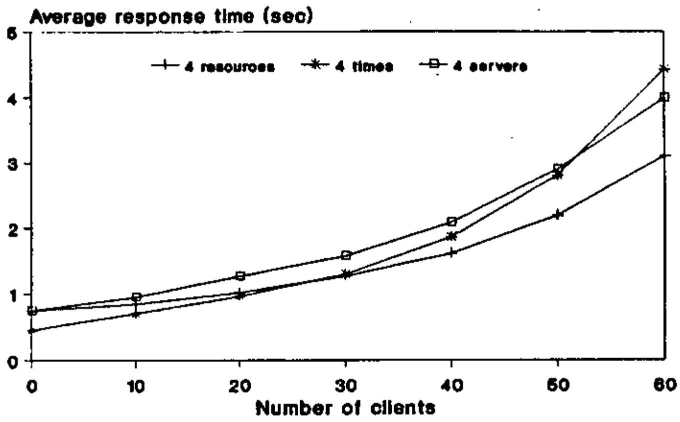


Figure 6.14.10 : 316Kbytes(B)

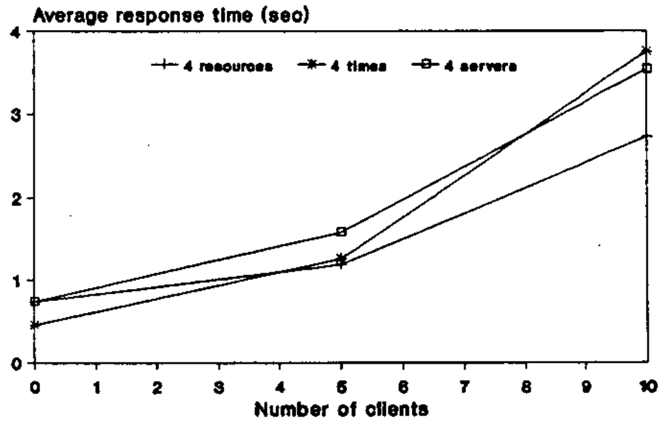


Figure 6.14.11 : 316Kbytes

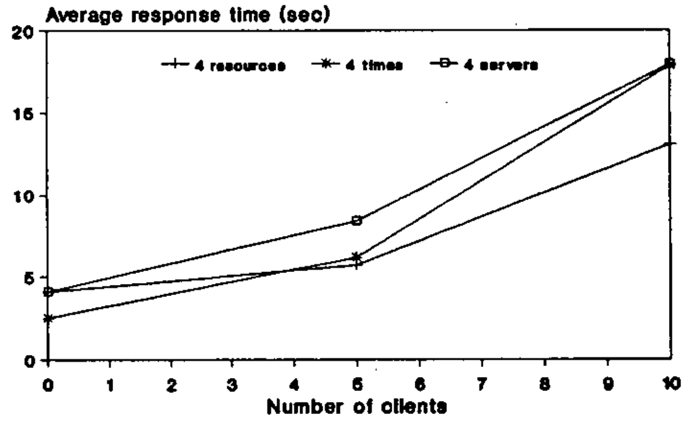


Figure 6.14.12 : 1856Kbytes

The average response time of the 4 resources case vs. the average response time of the 4 times better case vs. the average response time of the 4 file servers case in the distributed file system which consists of the Sun SPARCstation 10 workstations.

We also see that except in figure 6.14.7, that is, when the 8Kbytes workload is used, in each figure, there exist two intersecting points as in the previous comparison.

Figure 6.14.13 and figure 6.14.18 compare the average response when the system has 8 CPUs, 8 disks, 8 disk interface units and 8 network interface units at the same time in the file server, when the system has 8 times better file server and when the system has 8 file servers. The average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are shown respectively as the number of clients increases gradually in the figures.

In each case, there exists at least one crossing point even though when the 8Kbytes workload is used the crossing point is not shown in the given scale. In each case except when the 8Kbytes workload is used and when the 316kbtytes(B) workload is used, there exist two crossing points. As in the two previous comparisons, it is also observed that the crossing point occurs at fewer clients as the average transaction size increases. It is notable that there exist three crossing points when the 316Kbytes workload is used.

From the 3 comparisons, we find the following as common facts. First, the average response time of the distributed file system which has the better file server is more sensitive to the number of clients than the average response time of the distributed file system whose file server has multiple resources.

Second, when the 8Kbytes workload is used, there exist almost constant gaps between the average response times of the multiple file servers cases and those of the better file server cases, even though the number of clients increases. For

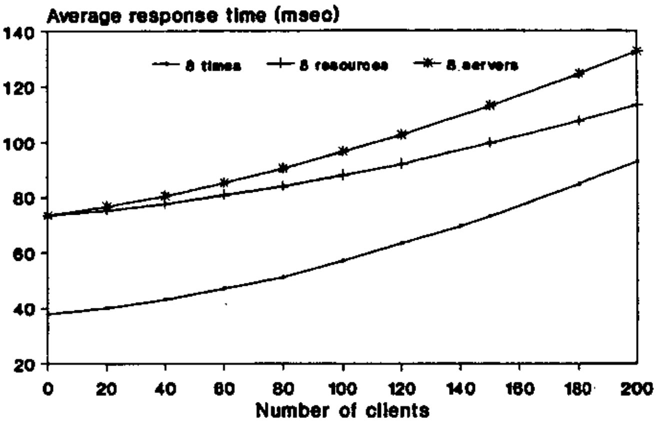


Figure 6.14.13 : 8Kbytes

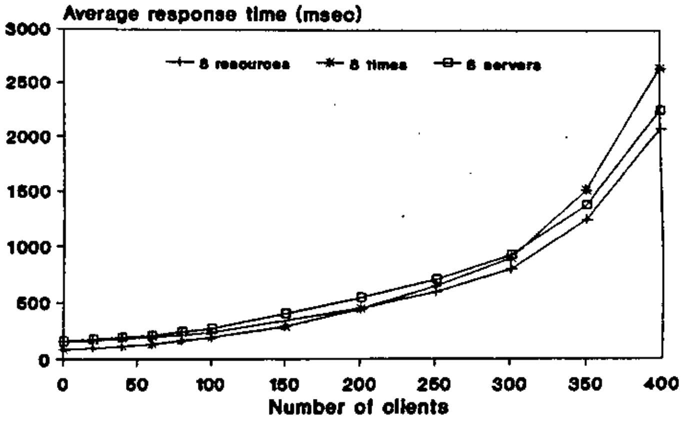


Figure 6.14.14 : 47Kbytes

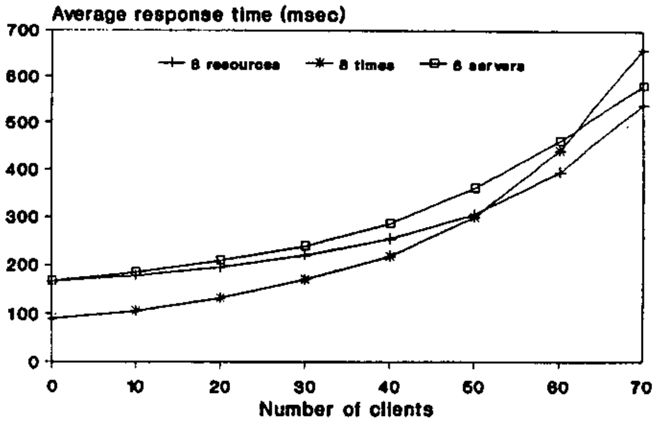


Figure 6.14.15 : 50.7Kbytes

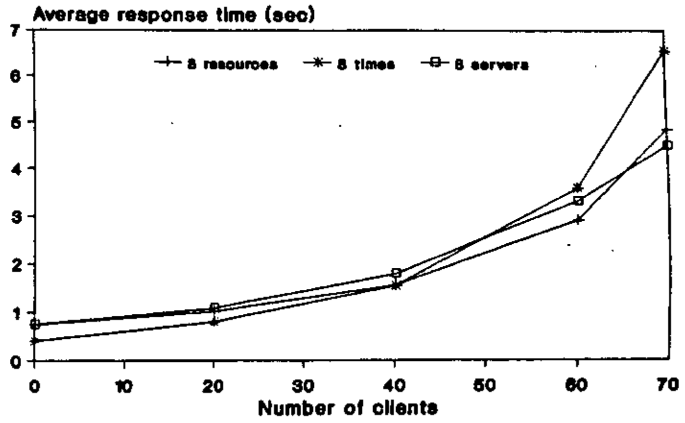


Figure 6.14.16 : 316Kbytes(B)

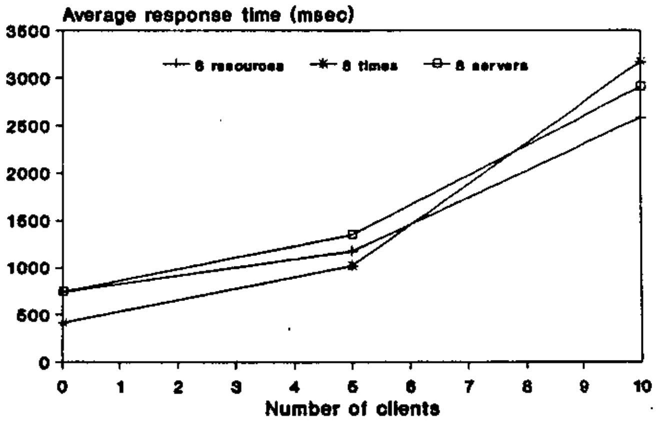


Figure 6.14.17 : 316Kbytes

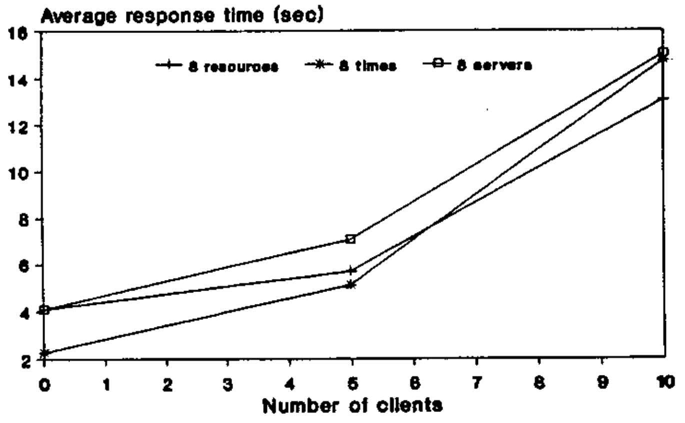


Figure 6.14.18 : 1856Kbytes

The average response time of the 8 resources case vs. the average response time of the 8 times better case vs. the average response time of the 8 file servers case in the distributed file system which consists of the Sun SPARCstation 10 workstations.

example, about 20msec gap in figure 6.14.1, about 40msec gap in figure 6.14.7 and 45msec gap in figure 6.14.13.

Third, the average response time of the distributed file system which has the better file server is more sensitive to the average transaction size than the average response time of the distributed file system whose file server has multiple resources.

Fourth, the better file server case always shows the best average response time, the multiple resources case and the multiple file servers case show the next best average response time, when there is no contention in the file server.

Fifth, as the contention grows beyond the first crossing point, the average response time of the better file server case develops faster than that of any other cases and becomes worse than that of the multiple resources case while the multiple file servers case still shows the worst average response time.

Sixth, as the contention grows beyond the second crossing point if it exists, the better file server case shows the worst average response time and the multiple file servers case shows the second worst average response time. As the contention grows beyond the third crossing point if it exists, the multiple file servers case shows the best average response and the better file server case shows the worst average response time.

Seventh, it is observed that the crossing point occurs at more clients as the degree of improvement or the number of multiple resources or the number of file servers increases regardless of the average transaction size used.

Eighth, no notable effect on the average response time due to workload fluctuation is found in the figures. Ninth, generally, the six workloads show a similar pattern in the average response times.

6.15 Multiple Resources in the Shared Memory System vs. Better Shared Memory System

This section compares the file access performance of a shared memory system when the system has multiple resources and when the system has better resources. The file access performance when multiple resources are used was already investigated in section 6.11 and the file access performance when a better resource is used was already investigated in section 6.12. The modification of the shared memory system in this section is based on the Sun SPARCstation 10 workstation.

Figure 6.15.1 to figure 6.15.6 compare the average response time when the system has 2 CPUs, 2 disks and 2 disk interface units at the same time and when the system is improved to be 2 times better. The two times better system means that the values of all parameters are improved to be two times better. The average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are shown respectively as the number of local users increases gradually in the figures.

Figure 6.15.7 to figure 6.15.12 compare the average response time when the system has 4 CPUs, 4 disks and 4 disk interface units at the same time and when the system is improved to be 4 times better. The average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are shown respectively as the number of local users increases gradually in the figures.

Figure 6.15.13 to figure 6.15.18 compare the average response time when the system has 8 CPUs, 8 disks and 8 disk interface units at the same time and when the system is improved to be 8 times better. The average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are shown respectively as the number of local users increases gradually in the figures.

The average response time in the figures shows the following pattern in general. First, in each figure of all 6 workloads, there exists one crossing point even though the crossing point is not shown in the given scale in some figures. The crossing point occurs since the average response time of the better system case grows faster than that of the multiple resources case.

Second, the better system case always shows better average response time, when there is no contention for the system resources in the shared memory system. As the contention grows the average response time of the better system case develops faster than that of the multiple resources case and beyond the first crossing point the better system case shows worse average response time than the multiple resources case.

Third, the average response time of the better system case is more sensitive to the average transaction size than the average response time of the multiple resources case and the crossing point occurs at fewer local users as the average transaction size increases.

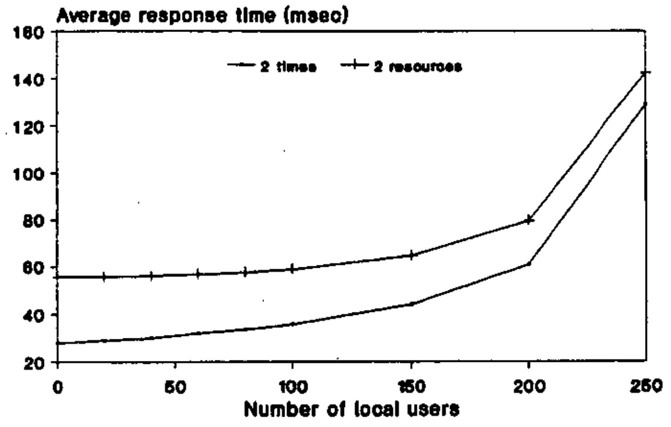


Figure 6.15.1 : 8Kbytes

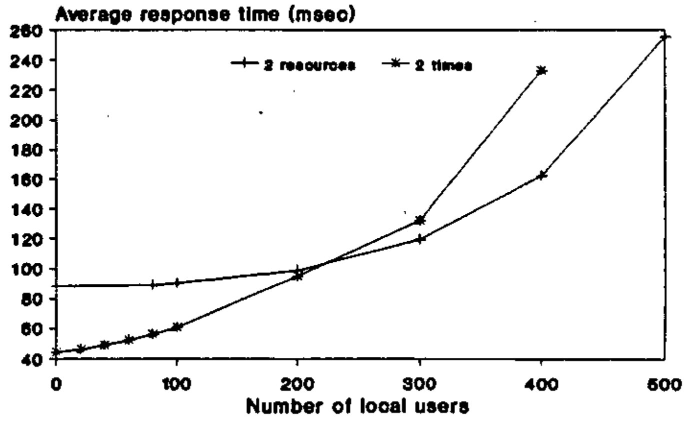


Figure 6.15.2 : 47Kbytes

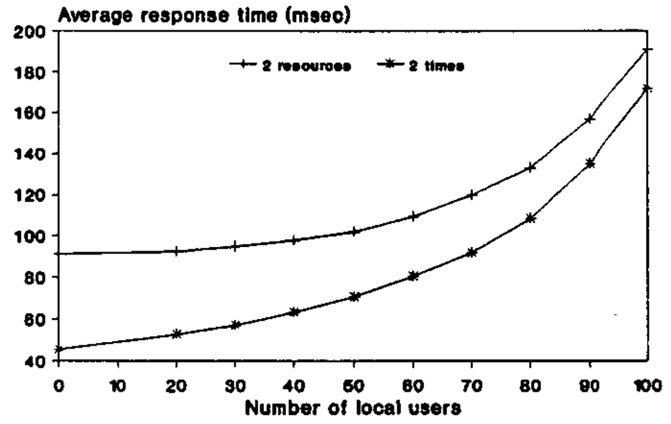


Figure 6.15.3 : 50.7Kbytes

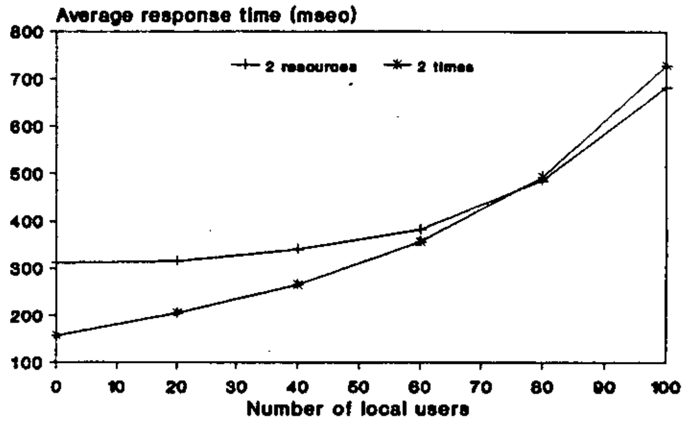


Figure 6.15.4 : 316Kbytes(B)

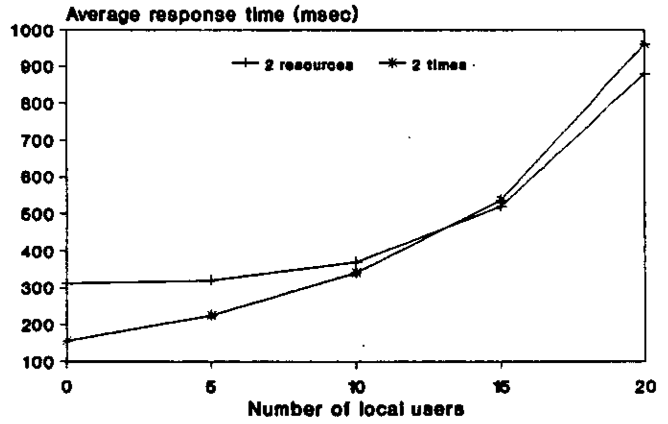


Figure 6.15.5 : 316Kbytes

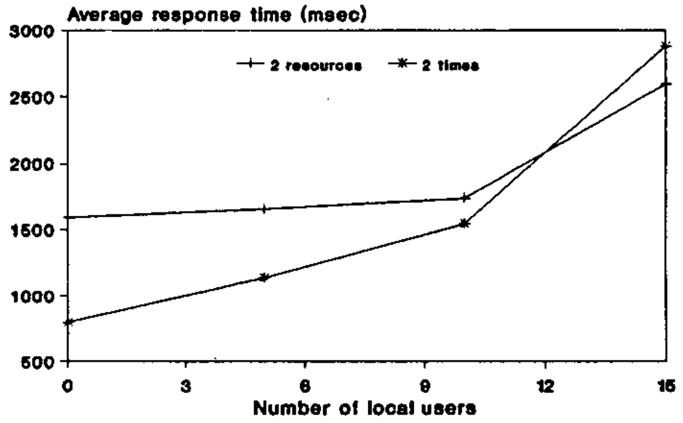


Figure 6.15.6 : 1856Kbytes

The average response time of the two times better case vs. the average response time of the two resources case in the shared memory system based on the Sun SPARCstation 10 workstation.

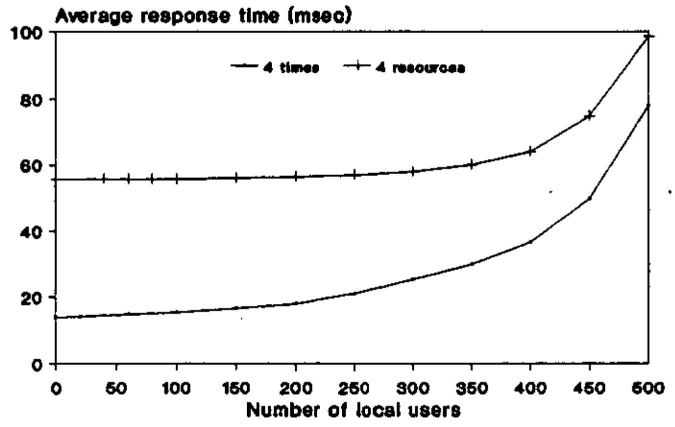


Figure 6.15.7 : 8Kbytes

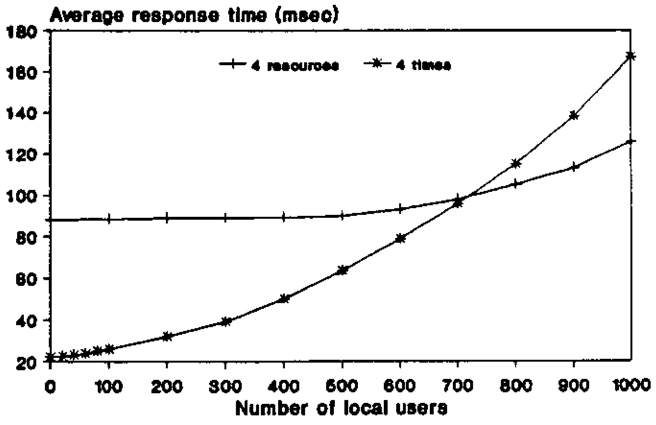


Figure 6.15.8 : 47Kbytes

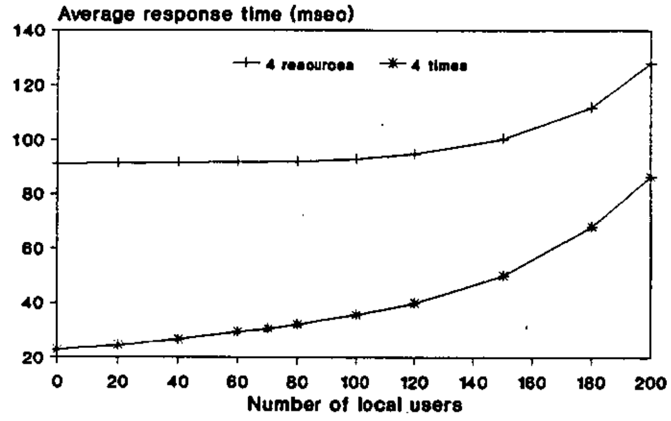


Figure 6.15.9 : 50.7Kbytes

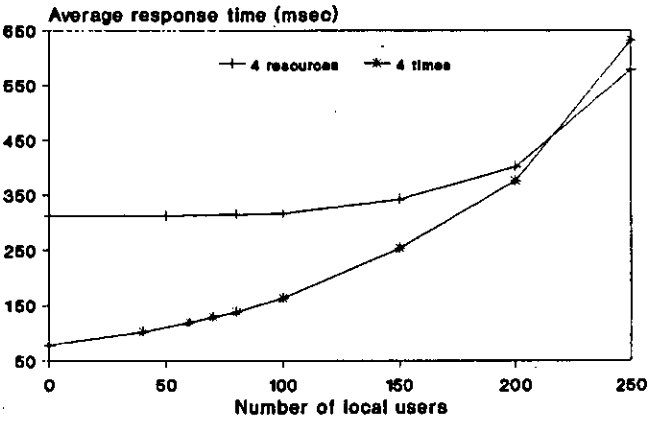


Figure 6.15.10 : 316Kbytes(B)

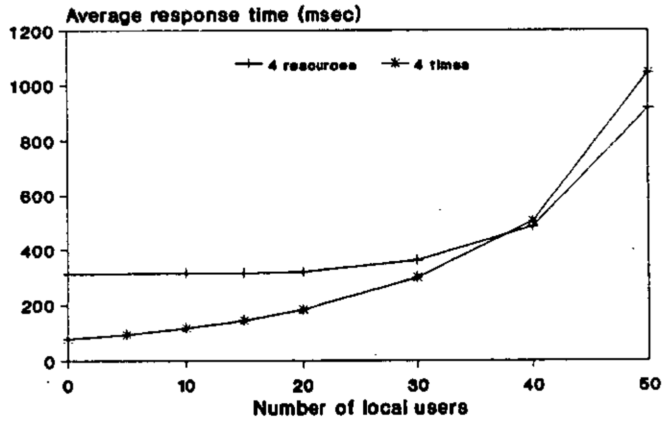


Figure 6.15.11 : 316Kbytes

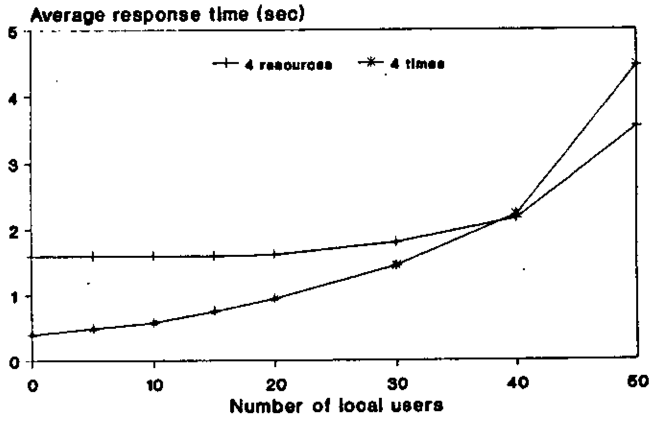


Figure 6.15.12 : 1856Kbytes

The average response time of the 4 times better case vs. the average response time of the 4 resources case in the shared memory system based on the Sun SPARCstation 10 workstation.

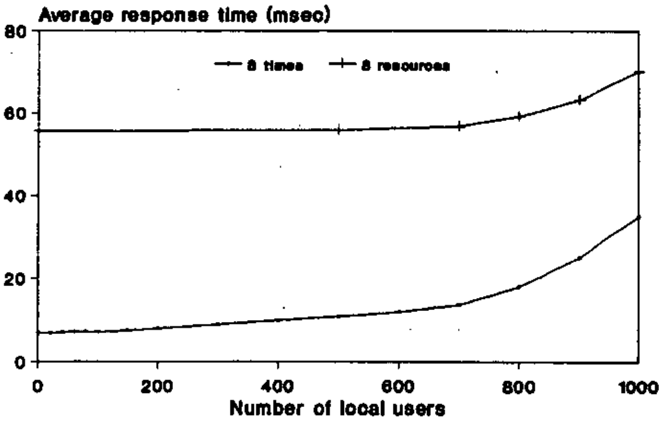


Figure 6.15.13 : 8Kbytes

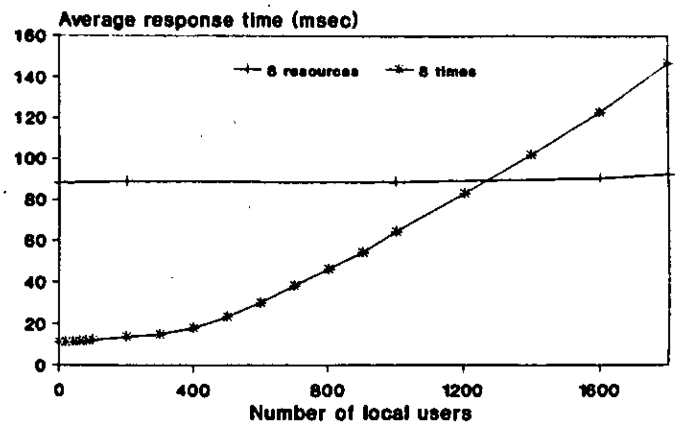


Figure 6.15.14 : 47Kbytes

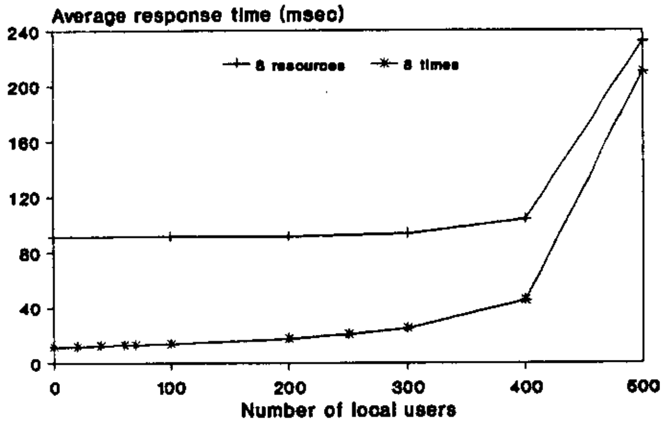


Figure 6.15.15 : 50.7Kbytes

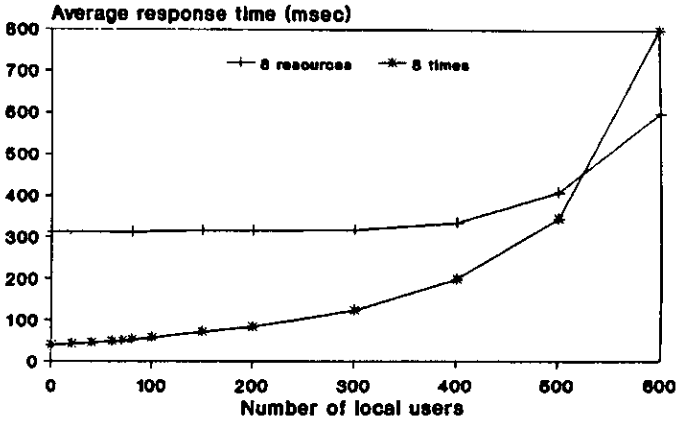


Figure 6.15.16 : 316Kbytes(B)

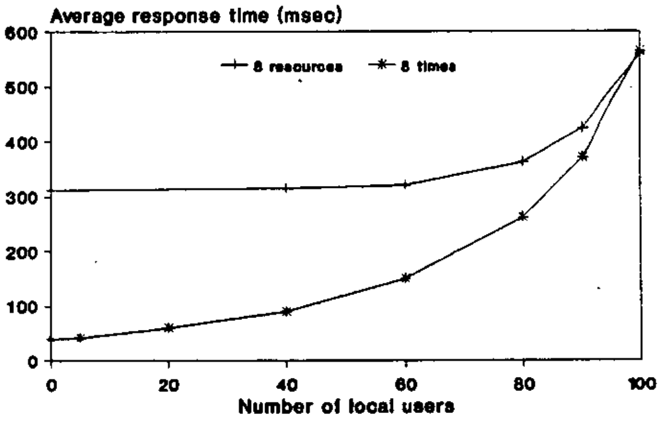


Figure 6.15.17 : 316Kbytes

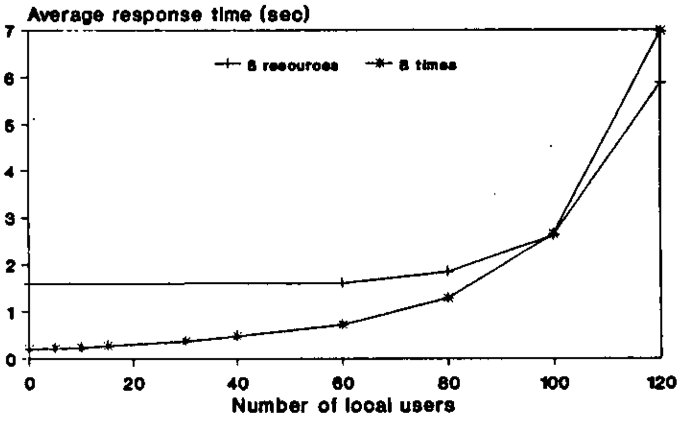


Figure 6.15.18 : 1856Kbytes

The average response time of the 8 times better case vs. the average response time of the 8 resources case in the shared memory system based on the Sun SPARCstation 10 workstation.

Fourth, the average response time of the system becomes less sensitive to the number of local users as the degree of improvement or the number of multiple resources increases regardless of the transaction size used and it is observed that the crossing point occurs at more clients as the degree of improvement or the number of multiple resources increases regardless of the transaction size used.

Sixth, no notable effect due to workload fluctuation on the average response time is found in the figures. Seventh, generally the six workloads show similar patterns for the average response times.

6.16 Concurrency

This section considers the effect of concurrency on the file access performance. Possible concurrency can happen in the following two cases. First, concurrency can happen between the CPU and the network interface unit during network communication(send/receive) operations in the clients and in the file server. Second, concurrency can happen between the CPU and the disk interface unit during disk I/O operations in the file server of the distributed file system and in the shared memory system. The degree of concurrency has an effect on the file access performance in both system paradigms. The following sections investigate the effect on the file access performance of concurrency in the two cases.

6.16.1 Concurrency during Disk I/O Operations

This section investigates the effect on file access performance of concurrency during disk I/O operations comparatively in both system paradigms.

Let's recall what was already explained about disk I/O operations in section 3.2.6. In the virtual server models, the disk interface unit and the CPU cooperate to do the preprocessing work such as disk I/O path set-up, etc., before starting the physical disk I/O operations. They also cooperate to do postprocessing work such as moving data from the buffers of the disk interface unit into the buffers of the memory, etc., after finishing the physical disk I/O operations. For cooperation for disk I/O operations, the disk interface unit and either the disk or the CPU are seized and released at the same time. If any of the two required resources is unavailable then the other must wait until the unavailable one becomes free and both of them can be seized at the same time.

If the disk interface unit is enhanced to do disk I/O operations for itself without the cooperation of the CPU, for the released time the CPU can better spend its power for other operations and the disk interface unit itself will be assigned with more opportunities when it is asked to serve since the two system resources have to be seized and released no longer at the same time and therefore, even though any of the two required resources is unavailable, the other does not have to wait until the unavailable one becomes free. This means the degree of concurrency is enhanced.

In the worst system in terms of the concurrency, the CPU has to cooperate with the disk even for low-level disk I/O operations. But the disk interface unit of the baseline Sun SPARCstation 10 workstation already provides some concurrency and the CPU does not have to do it there.

If the disk interface unit of the baseline Sun SPARCstation 10 workstation is replaced with a disk interface unit which has better mechanisms to improve the concurrency between them, what will be the effect on the file access performance? The enhancement is quantified as the relative percentage of the degree of concurrency to the degree of concurrency in the Sun SPARCstation 10 workstation in this study. For example, an improvement in concurrency of 20% means that 20% of the current CPU service time for disk I/O operations is reduced and during the period, the CPU is freed but on the other hand, the service time of the disk interface unit increases by that amount and the disk interface unit is that much more utilized or becomes that much busier. The disk interface unit is already the most heavily utilized system resource in the shared memory system and one of the heavily utilized system resources in the distributed file system. Therefore, asking more service of the disk interface unit will obviously damage the average response time in both system paradigms. Reducing the CPU service time demand by this amount will not improve the average response time much since the CPU is under-utilized and it is the most idle system resource all the time in both system paradigms. However, the CPU and the disk interface unit will provide better opportunities to be acquired when they are asked to serve since now they do not have to cooperate with each other for the disk I/O operations during the saved time period.

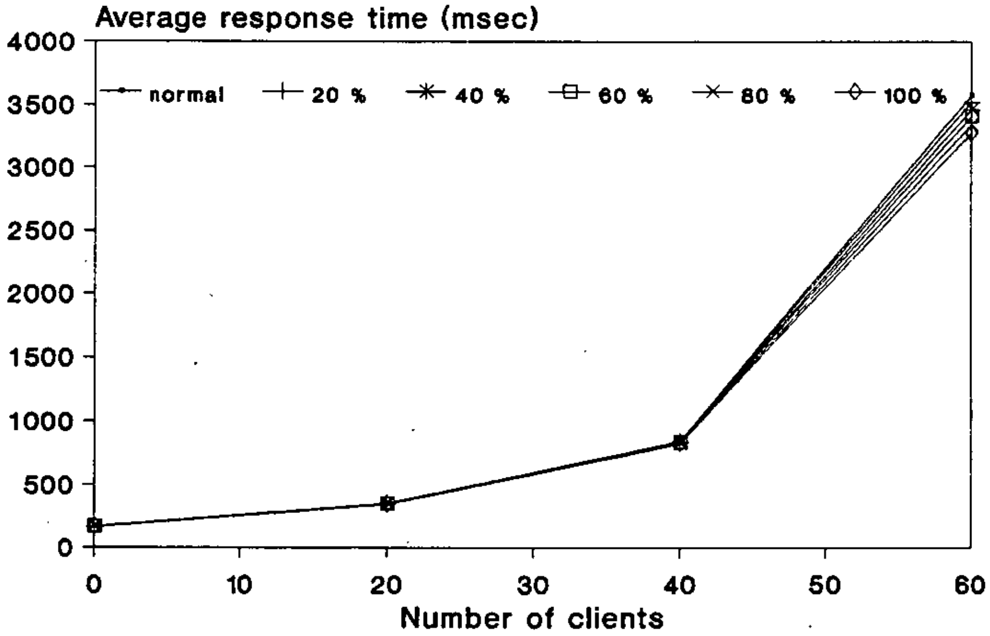


Figure 6.16.1.1 : The effect of the improved concurrency during disk I/O operations on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

Figure 6.16.1.1 shows the average response time of the 50.7Kbytes workload in the distributed file system which consists of the Sun SPARCstation 10 workstations as the number of clients increases gradually. The degree of concurrency in the disk interface unit of the file server is improved to be 20%, 40%, 60%, 80% and 100% better respectively. At 100% improvement, the CPU and the disk interface unit are absolutely independent of each other during the disk I/O operations. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

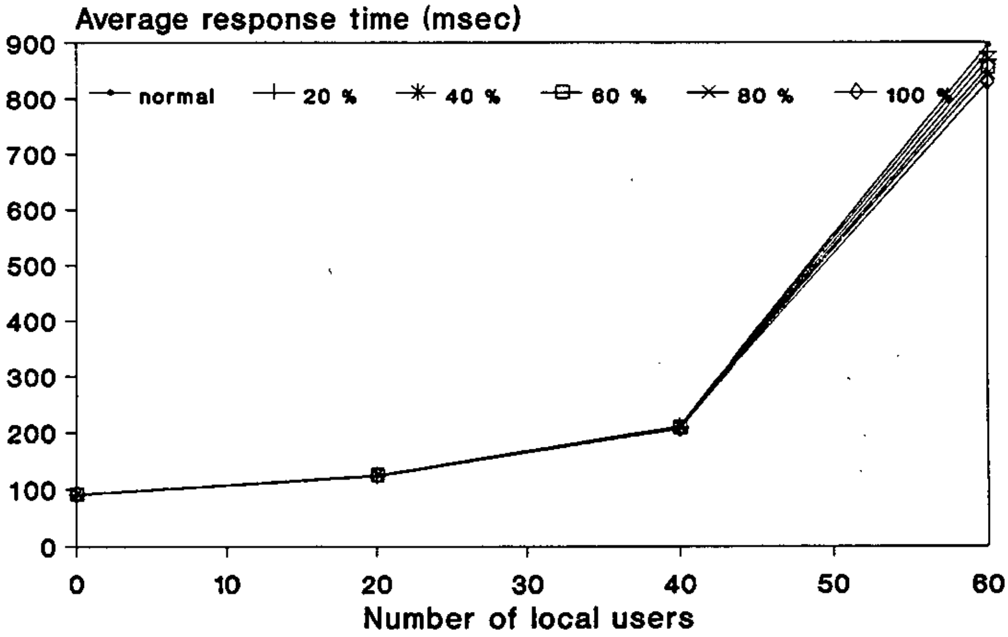


Figure 6.16.1.2 : The effect of the improved concurrency during disk I/O operations on the average response time in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

Figure 6.16.1.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The degree of concurrency in the disk interface unit is improved to be 20%, 40%, 60%, 80% and 100% better respectively. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation. See appendix C for the figures of other cases.

Contrary to our intuition, the file access performance of each case shows slight improvement, that is, the average response time decreases slightly. This means the effect of freeing the CPU and the disk interface unit for the times gained due to the improved concurrency and the effect of reducing the CPU service time demand by the time gained is larger than the effect of putting the burden of the time gained on the already busy disk interface unit. The pattern is similar in the figures for both system paradigms and for the six workloads.

6.16.2 Concurrency during Communication Operations

This section investigates the effect of concurrency during the network communication operations on the file access performance.

Let's recall what was already explained about network communication operations in section 3.2.6. In the virtual server model, before data transmission, both the network interface unit and the CPU of the client cooperate to do the preprocessing work for data sending, for example, moving data from the memory buffers to the buffers of the network interface unit at the sending site. After transmission activity, the network interface unit and the CPU of the file server cooperate to do postprocessing work for data receiving, for example, moving the received data in the buffers of the network interface unit into the memory buffers. For cooperation during network communication operations, the network interface unit and either the network or the CPU are seized and released at the same time. If any of the two required resources is unavailable then the other should wait until the unavailable one becomes free and both of them can be seized at the same time.

If the network interface unit is enhanced to do the network communication operations for itself without the cooperation of the CPU, during the released time the CPU can better spend its power for other operations and the network interface unit itself will be assigned with more opportunities when it is asked to serve since the two system resources have to be seized and released no longer at the same time and therefore even though any of the two required resources is unavailable the other does not have to wait any longer until the unavailable one becomes free. This means the degree of concurrency is enhanced.

In the worst system in terms of concurrency, the CPU has to cooperate with the network even for low-level data transmission operations through the network. But the network interface unit of the baseline Sun SPARCstation 10 workstation already provides some concurrency and the CPU does not have to do it there.

If the network interface unit of the baseline Sun SPARCstation 10 workstation is replaced with a network interface unit which has a better mechanism to improve the concurrency between them, what will be the effect on the file access performance?

To measure the effect on the file access performance, the improvement is quantified by the relative percentage of the degree of concurrency to the degree of concurrency of the Sun SPARCstation 10 workstation. For example, the improvement of the concurrency by 20% means that 20% of the current CPU service time for network communication is reduced and during the period the CPU is freed. However service time of the network interface unit increases by that amount and the network interface unit is that much more utilized and becomes that much busier. The network interface unit is already one of the most heavily utilized system resources in the distributed file system. Therefore, asking the network interface unit to do more service will obviously damage the average response time of the distributed file system. Reducing the CPU service time demand by the relevant amount will not contribute much to the improvement of the average response time since the CPU is under-utilized and it is the most idle system resource all the time in the distributed file system. However, the CPU and the network interface unit will provide more opportunities to be acquired when they are asked to serve since now they do not have to cooperate with each other for network communication operations during the saved time period.

Figure 6.16.2.1 shows the average response time of the 50.7Kbytes workload in the distributed file system which consists of the Sun SPARCstation 10 workstations as the number of clients increases gradually. The degree of concurrency is improved to be 20%, 40%, 60%, 80% and 100% better. At 100% improvement, the CPU and the network interface unit are absolutely independent from each other during the network communication operations. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix C for the figures of other cases.

Contrary to what was found about the effect on the file access performance when the concurrency during the disk I/O operations is improved, the file access performance shows slight deterioration, that is, the average response time increases slightly. This means the effect of freeing the CPU and the network interface unit during the time gained due to the improved concurrency and the effect of reducing the CPU service time demand by the time gained is smaller than the effect of putting the burden of the time gained on the already busy network interface unit. The patterns are similar in the figures for the six workloads.

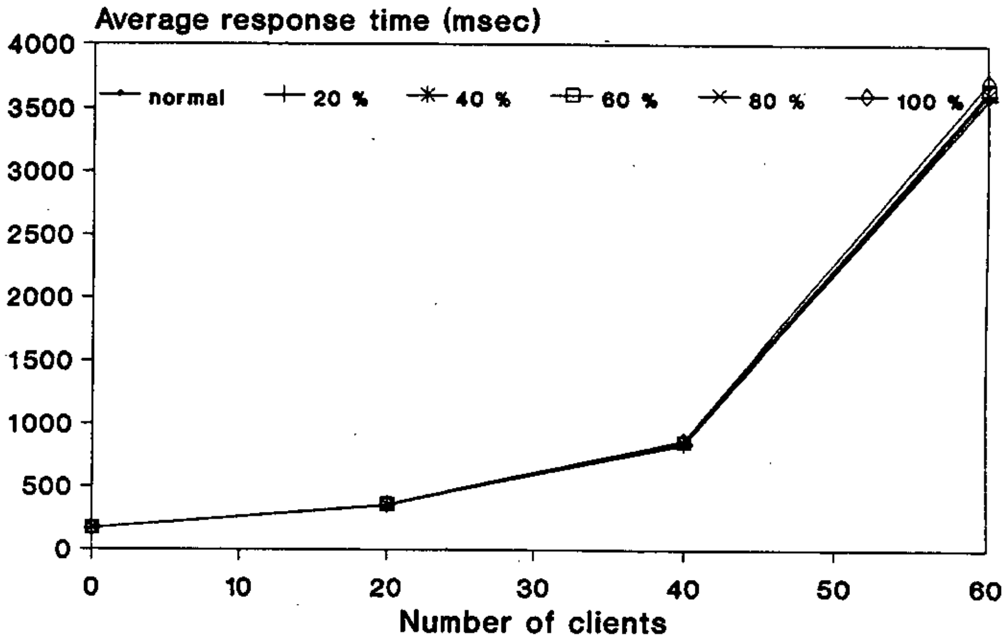


Figure 6.16.2.1 : The effect of the improved concurrency during communication operations on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

6.17 Everything Better

So far I have investigated the effect on the file access performance when we improve the power of the system resources or add more resources or enhance the processing mechanism separately one by one or group by group. This section investigates the file access performance of two different system paradigms when all parameter values of table 3.2.7.C or table 3.4.2.A which this study has investigated so far are reduced at the same time by enhancing the processing mechanisms or improving the powers of the system resources.

As this study already investigated, using a two times better system resource than a system resource used in a baseline system does not necessarily mean that the related parameter values are reduced to half of those of the baseline system. This is simply proved by observing the parameter values in table 3.2.7.C and table 3.4.2.A. The system power ratio among the three systems, that is, the ratio of the MIPS value of the Sun SPARCstation 10 workstation to the MIPS value of the Sun SPARCstation 470 workstation to the MIPS value of the Sun 3/60 workstation is 33.87 : 7.34 : 1 but no ratio among the parameter values in the table 3.2.7.C or table 3.4.2.A reaches the inverse of the MIPS ratio. The closest one is the ratio of the parameter value of the result processing i/o time(proportional portion) which is 1 : 4.6 : 23.7.

This section deals with the homogeneous distributed file systems. The baseline performance model of figure 3.2.6.B is used for the distributed file systems and the baseline performance model of figure 3.4.1.B is used for the shared memory systems.

Figure 6.17.1 shows the average response time of the 50.7Kbytes workload in the distributed file system which consists of the Sun SPARCstation 10 workstations when all parameter values are improved to be 2, 4, 8, 10, 20, 30, 100 and 1000 times better. For the distributed file system where all parameter values are improved to be four times better, a network of 50Mbps speed is used, therefore the network speed is five times faster, not four time faster. For the distributed file system where all parameter values are improved to be 8 times better and 16 times better, a network of 100Mbps speed is used, therefore the network speed is 10 times faster, not 8 times faster or 16 times faster. See appendix C for the figures of other cases.

In figure 6.12.2 the effect on the file access performance was already investigated when all parameter values of the Sun SPARCstation 10 workstation are improved to be 2, 4, 8, 10, 16, 20, 30, 100 and 1000 times better when the 50.7Kbytes workload are used.

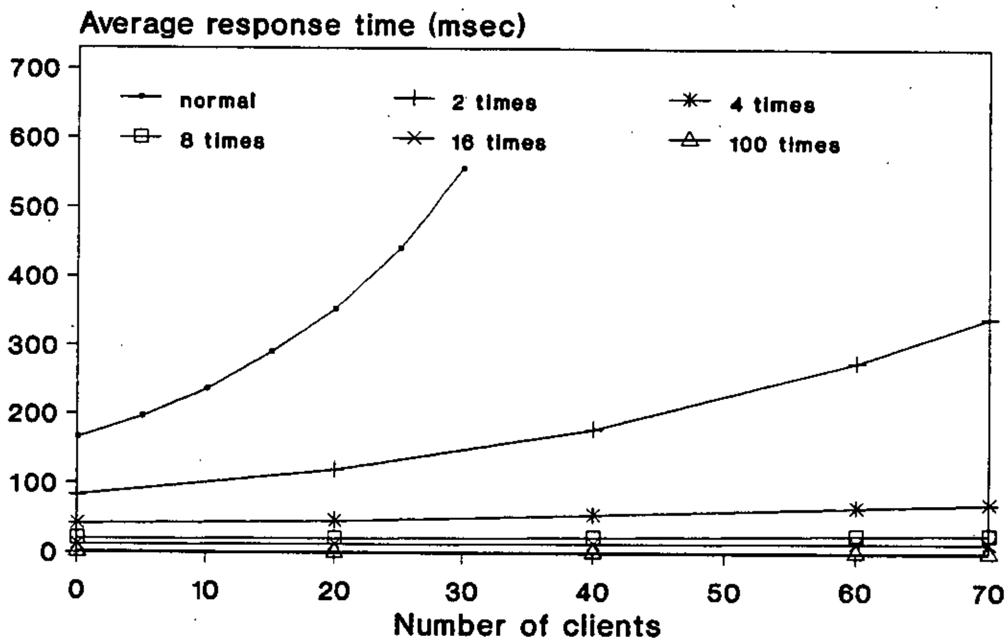


Figure 6.17.1 : The effect on the average response time of improving the power of all resources in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

It is observed that the ratio of the average response time in the baseline distributed file system to the average response time in the distributed file system where all parameters are improved to be X(2,4,8,...) times better including the network speed is equal to or larger than the degree of improvement, that is, X(2,4,8,..) up to a reasonable number of clients. This is also true in the shared memory system.

One method to measure the file access performance of a system is to find out when the average response time reaches to a given level as the contention increases. Let's look at the figures for the distributed file system. The average response time of the 8Kbyte workload is within 300msec up to 100 clients in the baseline system, and it is so up to more than 1000 clients in the system which are eight times better than the baseline system in all parameter values. The average response time of the 47Kbytes workload is within 660msec up to 100 clients in the baseline system, it is so up to 300 clients in the two times better system in all parameter values, and it is so up to far more than 1000 clients in the 8 times better system in all parameter values. When we use the 50.7Kbytes workload, the baseline distributed file system shows an average response time of 550msec at near 30 clients and the system where all parameters are improved to be two times better shows an average response time of 500msec at near 80 clients and the system where all parameters are improved to be 4 times better, shows an average response time of 500msec at 170 clients. The average response time of the 316Kbytes(B) workload is always larger than 1 second and already 3.5 seconds at near 30 clients in the baseline system but it is around 500msec at near 450 clients in a 16 times better system in all parameter values. The average response time of the 316Kbytes workload is 516msec at near 80 clients in a 16times better system in all parameter values and only 44msec at near 500 clients in a 100times better system in all parameter values. The average response time of the 1856Kbytes workload is more than 4 seconds even when there is no contention for the system resources and near 10 seconds at already 3 clients in the baseline system but it is within 200msec up to 400 clients in a 100 times better system in all parameter values.

Let's find out how much the file access performance is improved by looking at when the average response time reaches a given level as the number of local users increases in figure 6.15.7 to figure 6.15.12 for the shared memory system, where we improve the power by reducing all parameter values at the same time. The average response time of the 8Kbytes workload is within 130msec up to 100 local users in the baseline system, and it is so up to more than 500 local users in the 4 times better system in all parameter values. The average response time of the 47Kbytes workload is within 160msec up to near 100 local users in the baseline system, it is so up to more than 300 local users in the two times better system in all parameter values, and it is so up to 1000 local users in the 4 times better system in all parameter values. The average response time of the 50.7Kbytes workload is 160msec at near 30 local users in the baseline system and it is so up to more than 400 local users in the 8 times better system in all parameter values. The average response time of the 316Kbytes(B) workload is 540msec at near 20 local users in the baseline system but it is around 500msec at 500 local users in the 8 times better system in all parameter values. The average response time of the 316Kbytes workload is 670msec at 5 local users in the baseline system but it is only 61msec at near 100 local users in the 16 times better system in all parameter values. The average response time of the 1856Kbytes workload is more than 1.5 seconds when there is no contention for the system resources and already 7.3 seconds at 10 local users in the baseline system but it is 44msec at 200 local users in the 100 times better system in all parameter values.

Now we find out how much we have to improve the power of the baseline distributed file system which consists of the Sun SPARCstation 10 workstations in order that the average response time of the workload whose average transaction size is very large, for example, the 1856Kbytes workload, becomes similar to that of the 8Kbytes workload.. 1856Kbytes is 232 time as large as 8Kbytes. Therefore do we have to improve the system power by 232 times? From the figures, we find that if the baseline distributed file system is improved to be 100 times better in all parameter values, then the average response time of the 1856Kbytes workload becomes much better than that of the 8Kbytes workload. In the system, the average response time of the 1856Kbytes workload is 41msec when there is no contention for the system resources and 177msec at 400 clients and even 1000 clients do not saturate the system while in the baseline system the average response time of the 8Kbytes workload is 74msec when there is no contention for the system resources and 288msec at 100 clients and 150 clients saturate the system.

Let's also find out how much we have to improve the power of the Sun SPARCstation 10 workstation in order that the average response time of the 1856Kbytes workload becomes similar to that of the 8Kbytes workload as we did in the distributed file system. From the figures, we find that if the baseline shared memory system is improved to be 100 times better in all parameter values the average response time of the 1856Kbytes workload becomes 16msec when there is no contention for the system resources and 44msec at even 500 local users while in the baseline system the average response time of the 8Kbytes workload is 56msec when there is no contention for the system resources and 122msec at 100 local users. If the baseline shared memory system is improved to be 16 times better in all parameter values, the average response time of the 1856Kbytes workload is 100msec when there is no contention for the system resource and 416msec at 120 local users.

How much do we have to improve the power of the baseline distributed file system which consists of the Sun SPARCstation 10 workstations in order that the average response time becomes similar to that in the Sun SPARCstation 10 workstation? From the figures, we find that if the baseline distributed file system which consists of the Sun SPARCstation 10 workstations is improved to be 2 times better in all parameter values, the average response time of the 8Kbytes workload in the distributed file system is much better than that of the Sun SPARCstation 10 workstation all the time as the transaction arrival rate increases. This is also true when we use the 47Kbytes workload or the 50.7Kbytes workload. When we use the 316Kbytes(B) workload or the 316Kbytes workload or the 1856Kbytes workload, the average response time in the improved distributed file system is similar to that of the Sun SPARCstation 10 workstation. As the workload size grows, the gap between the average response time in the improved distributed file system and that in the Sun SPARCstation 10 workstation decreases gradually.

6.18 Summary

The six different workloads produce similar patterns of average response time in both system paradigms in each case.

The maximum improvement in the average response time by adding CPUs or improving the CPU power, that is, by getting rid of the queueing delay caused by the contention in the CPU, is small in percentage terms for the average response time of the baseline system in the two system paradigms. Both in the distributed file systems and in the shared memory systems, 2 CPUs or a two times better CPU get rid of most of the queueing delay caused by the contention in the CPU.

The average response time of the system which has a K(2,4,8,,,,) times better CPU is better than that of an equivalent system which has K(2,4,8,...) CPUs both in the distributed file system and in the shared memory system. And as the contention for the system resources of the file server in the distributed file system grows, the difference between the average response time of the better CPU case and that of the equivalent multiple CPUs case becomes larger in general. This was also observed in the shared memory system.

The average response time significantly improves in the system which has 2 disks and 2 disk interface units. Putting more than 4 disks and 4 disk interface units in the file server of the baseline distributed file system is not efficient in terms of the performance/cost.

When the CPU service time for disk I/O is improved, the overall improvement of the average response time in the distributed file system and in the shared memory system is not significant, as we expect.

The average response time in the system where the disk I/O time is improved to be two times faster is more sensitive to the number of clients than that in the system which has 2 disks and 2 disk interface units in both system paradigms.

The system which has a faster disk and the system which has multiple disks and multiple disk interface units becomes less sensitive to the number of clients or the number of local users as the disk I/O speed and the number of disks and disk interface units increase. System which has a faster disk is more sensitive to the average transaction size than a system which has multiple disks and multiple disk interface units.

The overall improvement in the average response time in the distributed file system is significant when multiple network interface units are used in the file server and multiple networks are used in the baseline distributed file system.

Most of the contention for the network disappears with a 100Mbps network. The overall improvement of the average response time in the distributed file system is significant when the performance of the network interface unit is improved. The contention for the network interface unit almost disappears when the parameter values are improved to be 16 times better. The overall improvement of the average response time in the distributed file system is small when the communication mechanism is enhanced.

The average response time of the distributed file system which has multiple networks and multiple network interface units in the file server is less sensitive to the number of clients than the average response time of the distributed file system which has the faster network and the better network interface unit in the file server.

The average response time of the distributed file system which has multiple networks and multiple network interface units in the file server is less sensitive to the average transaction size than the average response time of the distributed file system which has the faster network and the better network interface unit in the file server.

As the average transaction size of the workload increases, the effect of the overhead of the file system mechanism on the average response time decreases and becomes trivial. As the number of clients or the number of local users increases, the effect of the parameter of the file processing mechanism decreases further and becomes trivial. These facts hold in the case of the effect of RPC overhead on the average response time and the effect of command interpretation overhead on the average response time. The screen display overhead is proportional to the size of the transaction and therefore the effect of the overhead on the average response time overwhelms the other effects as the average size of the transaction increases.

The distributed file system which has 2 resources improves the average response time most efficiently so the best performance/cost can be obtained and the contention for the system resources almost disappears in the distributed file system which has 4 resources.

We observe that the distributed file system where all parameter values except the parameter of the network speed are improved to be 2 times better shows the best performance/cost.

The confidence of the normalized MIPS values of the three baseline shared memory systems in the file access performance is low or unacceptable regardless of the workload used. In the distributed file systems which consists of the Sun SPARCstation 10 workstations and the distributed file system which consists of the Sun SPARCstation 470 workstations, the confidence of the normalized MIPS values is high or acceptable when the 47Kbytes workload or the 1856Kbytes workload is used and low or unacceptable when any one of the other four workloads is used. The normalized MIPS value of the distributed file system which consists of the Sun 3/60 workstations is observed to have low or zero confidence in file access performance. Generally, the confidence of the normalized MIPS values in file access performance is observed to be better in the distributed file systems than in the shared memory systems.

When 2 file servers are used the distributed file system shows the best performance/cost and it is efficient in terms of performance/cost to use up to 4 file servers in the 10Mbps LAN environment. A 100Mbps network is desirable for the distributed file system which has multiple file servers when either 50.7Kbytes workload or 316Kbytes(B) workload is used. A 1Gbps network is desirable for the environment which has multiple file servers when either 316Kbytes workload or 1856Kbytes workload is used.

The average response time of the distributed file system which has the better file server is most sensitive to the number of clients among the three cases : the better file server case, the multiple file servers case and the multiple resources case. The average response time of the distributed file system which has the better file server is more sensitive to the average transaction size than the average response time of the distributed file system whose file server has multiple resources.

In shared memory systems, the average response time of the better system case grows faster than that of the multiple resources case and the average response time of the better system case is more sensitive to the average transaction size than the average response time of the multiple resources case.

The file access performance of each case shows slight improvement when the degree of concurrency in the disk interface unit is improved. When the concurrency during the communication operations is improved, the average response time increases slightly.

Chapter 7

File Access Performance Evaluation of Caching in the Two System Paradigms

This chapter investigates the file access performance of caching comparatively in the two system paradigms using the virtual server models.

In the following sections in this chapter, the following conditions hold unless otherwise specified. Write file access is performed unless read file access is explicitly specified to be performed. The workload pattern of the Poisson distribution for input arrival and the log-normal distribution for input transaction size is used unless the workload pattern used is specified. The Sun SPARCstation 10 workstation is used as the base system for the shared memory system and the distributed file system which consists of the Sun SPARCstation 10 workstations is used as the base distributed file system unless the base system is explicitly specified.

Many operating systems and distributed file systems have used caches to improve file access performance. Before an actual request for data occurs, the data can be prefetched into the cache by prediction so that the request is serviced directly by the cached data if it is requested later. The requested data can also be written into the cache and later the data written to the designated disk. Successive accesses to the same data in the cache are carried out without accessing the disk where the actual data reside.

If the cached data are used just one time, then no system power is saved since the caching expense is paid sometime somewhere after all. For example, in read-ahead caching and write-back caching, the response time of the request will be better than the response time without caching but the expense which is saved by using the cached data should be paid before the cached data are used or after the cached data are used. So by using the cached data, the system shows faster response time but actually all operations for the file access occur after all and no operation is saved at all. In this caching, the data traffic amount is the same, that is, the system load is same as that without caching.

However, when the same cached content is reused, there exists no hidden overhead due to the cache hit except the cache consistency maintenance overhead and the cache access overhead. So the hidden expense is saved.

If the cached data are used just one time, that is, if caching overhead is required before or after the cached data are used, then that cache hit is not the concern in this section. This study deals with the cache hits which do not require any pre-operations or post-operations at all, that is, if the same cached data are accessed more than one time, then the first access is not the concern but from the second access to the last access among all accesses are the concern of this chapter.

This caching has two distinct advantages. First, delays are reduced by caching since the requested data are already in the cache. This is also true even though the cached data are accessed just one time. Second, the contention for the disk I/O related devices such as the CPU, the disk interface unit, the disk, etc., is reduced so that processes attempting to access the same I/O related devices will have a better chance to access them with less waiting time. This is not true if the cached data are accessed just one time but true only if they are accessed more than one time.

There are overheads for the system to operate a caching mechanism such as the cache consistency mechanism overhead, the caching policy overhead, etc.. However the overheads are usually small compared with the benefits gained by caching, as we can see in the Sprite distributed file system[BAKER etal 91]. Measurement studies of some time-sharing systems also show that caching gives substantial benefits and the large size of caches in large physical memories give more benefits[BAKER etal 91],[LEFFLER etal 84],[OUSTERHOUT etal 85].

In designing the distributed file system or the file system of the shared memory system, we have to decide several things for the caching. First, shall we have to use the caching mechanism? Second, if we use the caching mechanism, where shall we have the cache : only in the file server or both in the file server and in the clients in case of designing the distributed file systems? Third, if we do caching in the clients as well, where shall we put the cached files : in the main memories of the clients or the local disks of the clients in case of designing the distributed file systems? Fourth, shall we do additional file caching in the disk interface unit as well as in the main memory of the file server of the distributed file system or in the main memory of the shared memory system? That is, is it worthwhile to do caching in the disk interface unit of the file server which already does caching in the main memory? File caching is usually performed in the memory. Additional file caching in the disk interface unit is now wide spread and will continue.

Performance evaluation of caching mechanisms is one of the benchmarks which we should rely on when we have to decide the above matters. This chapter studies the effect of the caching mechanisms on file access performance. This chapter investigates the effect at given cache hit rates but does not discuss how the cache hit rates can be achieved in each mechanism. This chapter does not discuss the details of the caching mechanism such as the cache replacement algorithm, the cache size, the block size, the cache consistency maintenance mechanism, etc..

Baker et al.[BAKER etal 91] show the measured data for the file caches in the Sprite distributed system, discuss issues of file caching such as file cache sizes, the effect of caching on file traffic, cache consistency mechanisms, etc. and show simulation results of a cache consistency mechanism which is similar to the cache consistency mechanism in some Sun NFS implementations. Ousterhout et al.[OUSTERHOUT etal 85] show the simulation results of file caching in local UNIX systems and discuss the issues of file caching such as file cache size, block size and write policy, which this study does not deal with. Lilja[LILJA 93] surveys cache coherence mechanisms in shared memory systems, discusses design issues, and studies the performance effect of the issues using trace driven simulations, which this study does not deal with. Karedla[KAREDLA 94] discusses caching strategies and studies the performance effect of cache replacement algorithms by simulation which this study does not deal with. Smith[SMITH 82],[SMITH 85] discusses various cache memories and caching mechanisms in general and in detail.

Below, what are investigated in the following sections is described. With which caching mechanism, does the system show the best file access performance? How much does the file access performance improve with a given caching mechanism? What operations are saved with the given caching mechanism? At what cache hit rate, does the average response time become acceptable even when the workloads of large average transaction size such as the 316Kbytes workload and the 1856Kbytes workload are used? What is the pattern of the average response time as the contention grows given a caching mechanism? When we use the caching mechanism, is there any difference in the pattern of the average response time between the distributed file system and the shared memory system? Does the pattern of the average response time vary as the average transaction size varies when we fix the cache hit rate to a given value? What is the pattern of the average response time when the cache hit rate varies? These are investigated in the following sections.

The performance effects of the four standalone caching mechanisms such as caching in the memory of the file server, caching in the disk interface unit of the file server, caching in the memory of the client and caching in the disk of the client of the distributed file system are investigated respectively in section 7.1, section 7.2, section 7.3 and section 7.4. The effects on the file access performance of the two standalone caching mechanisms such as caching in the memory of the shared memory system and caching in the disk interface unit of the shared memory system are also investigated respectively in section 7.1 and section 7.2. Section 7.5 compares the effects on the file access performance of four caching mechanisms in the distributed file system and of two caching mechanisms in the shared memory system.

The effects on file access performance of the combinations among the four standalone caching mechanisms in the distributed file system and of the combination of the two standalone mechanisms in the shared memory system are investigated in the following 5 sections. They are the combination of caching in the memory of the client and caching in the memory of the file server in the distributed file system in section 7.6, the combination of caching in the disk of the client and caching in the memory of the file server in the distributed file system in section 7.7, the combination of caching in the memory of the client, caching in the memory of the file server and caching in the disk interface unit of the file server in the distributed file system in section 7.8, the combination of caching in the disk of the client, caching in the memory of the file server and caching in the disk interface unit of the file server in the distributed file system in section 7.9 and the combination of caching in the memory and caching in the disk interface unit in the shared memory system in section 7.10. Finally, the effects on file access performance of the 4 standalone caching mechanisms and the 5 combined caching mechanisms are compared in section 7.11.

In all following sections, it is assumed that the cache consistency maintenance overhead is zero, which is the theoretical limit. Additional operations to read the cached data from the cache are required. I measured the memory access time to be 0.1msec per 1500bytes data in the three systems for the case of memory cache.

In the following sections, unless this study explicitly specifies the configuration of the system used, for the simulations this study uses the distributed file system which consists of the Sun SPARCstation 10 workstations and the Sun SPARCstation 10 workstation for the shared memory system. In the following sections, unless this study explicitly specifies the performance model used and the performance parameter table used, the performance model of figure 3.2.6.F and the baseline performance parameter values in table 3.2.7.C are used for the distributed file system and the performance model of figure 3.4.1.C and the baseline performance parameter values in table 3.4.2.A are used for the shared memory system.

7.1 Standalone Caching in the Memory of the File Server

This section comparatively investigates the effect of caching in the memory of the file server of the distributed file systems and caching in the memory of the shared memory systems on file access performance. If the requested data are in the cache, then all disk I/O operations are bypassed as shown in figure 3.2.6.F and figure 3.4.1.C. Therefore the CPU service time for disk I/O, the service time of the disk interface unit for the disk I/O and disk I/O time are saved and the utilization of the CPU, the disk interface unit and the disk are reduced.

Figure 7.1.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. Figure 7.1.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for these, all others are kept the same as the baseline Sun SPARCstation 10 workstation. See appendix D for the figures of other cases.

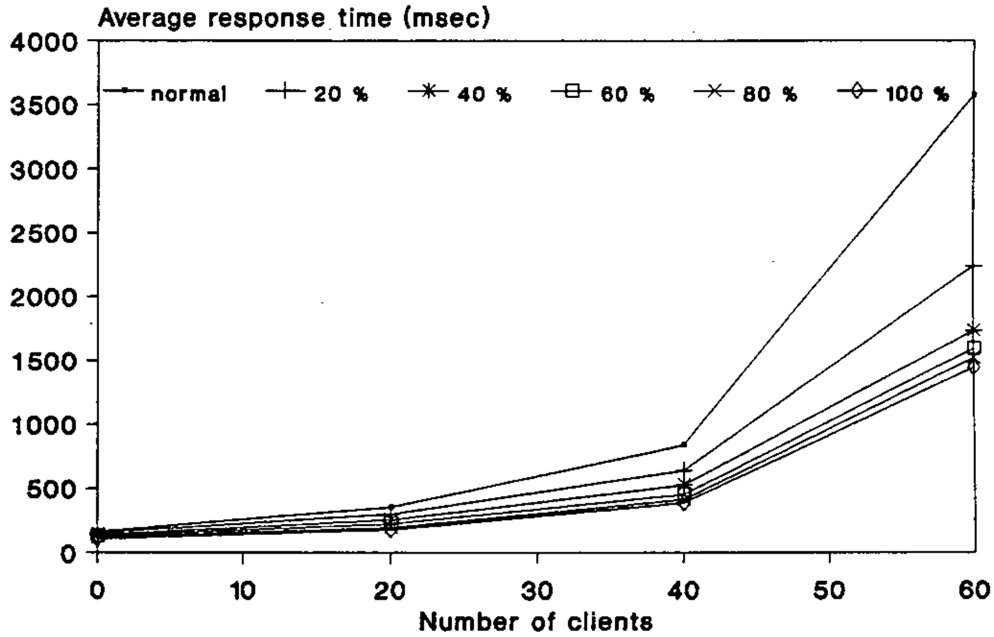


Figure 7.1.1 : The effect on the average response time of caching in the memory of the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

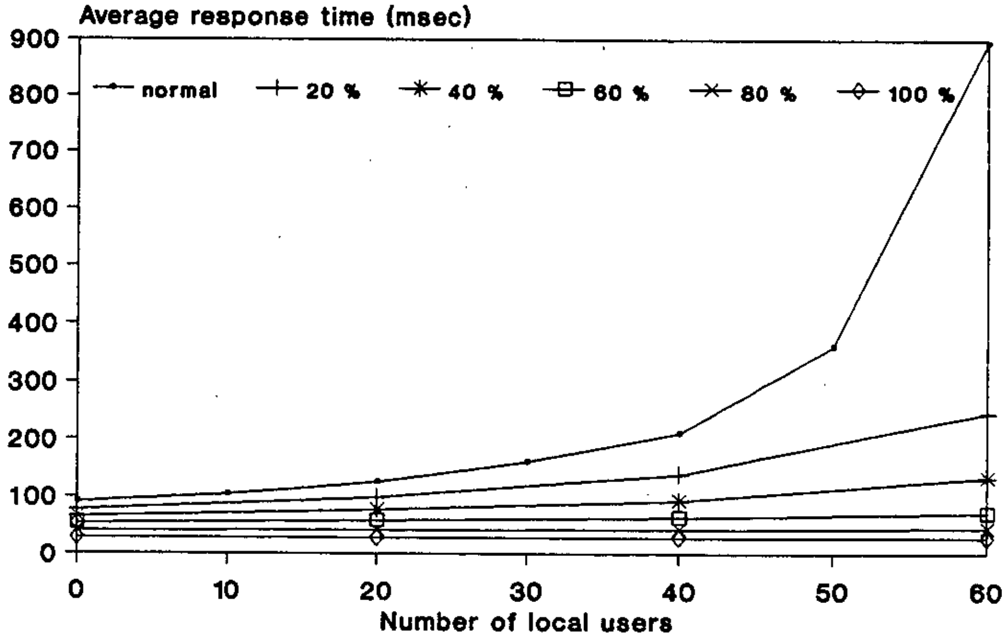


Figure 7.1.2 : The effect on the average response time of caching in the memory of the file server in the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

In the distributed file system, it is observed that at 20% hit rate, the improvement rate of the average response time per cache hit rate is the largest, then gradually it reduces. In the shared memory system, a regular improvement in the average response time is observed as the cache hit rate increases unlike in the figures of the distributed file system. In the distributed file system, the saturation point does not significantly increase but increases a little up to the saturation point of the network interface unit as the cache hit rate increases. In the shared memory system, the saturation point increases significantly and almost linearly as the cache hit rate increases.

In the distributed file system the queueing delay due to the contention for the system resources related to network communication service remains unchanged even though the overall improvement in the average response time is significant, but in the shared memory system the queueing delay gradually disappears as the cache hit rate increases. Because of this, the patterns of the average response times are different in the two system paradigms. In the distributed file system, even at 100 % cache hit rate, the average response time of the 316Kbytes workload and 1856Kbytes workload are still far above 1 second all the time but in the shared memory system, the average response time of the 1856Kbytes workload are below 1 second up to more than 15 clients at 80% cache hit rate. The average response time of the 8Kbytes workload in the distributed file system when the 40% cache hit occurs shows a similar trend to the average response time of the 8Kbytes workload in the shared memory system when no caching occurs.

All six workloads show similar trends in the average response times. No notable change is observed in the pattern of the average response time as the workload size increases and no notable difference is observed between the patterns of the average response times when steady workloads are used and those when bursty workloads are used.

7.2 Standalone Caching in the Disk Interface Unit

This section comparatively investigates the effect on file access performance when we use caching in the disk interface unit of the file server of the distributed file system and caching in the disk interface unit of the shared memory system. If the requested data are in the cache, then the disk I/O operations are bypassed as shown in figure 3.2.6.F and figure 3.4.1.C. Therefore the service time of the disk interface unit for the disk I/O and the disk I/O time are saved and the utilization of the disk interface unit and that of the disk are reduced.

Figure 7.2.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. Figure 7.2.2 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for these, all others are kept the same as the baseline Sun SPARCstation 10 workstation. See appendix D for the figures of other cases.

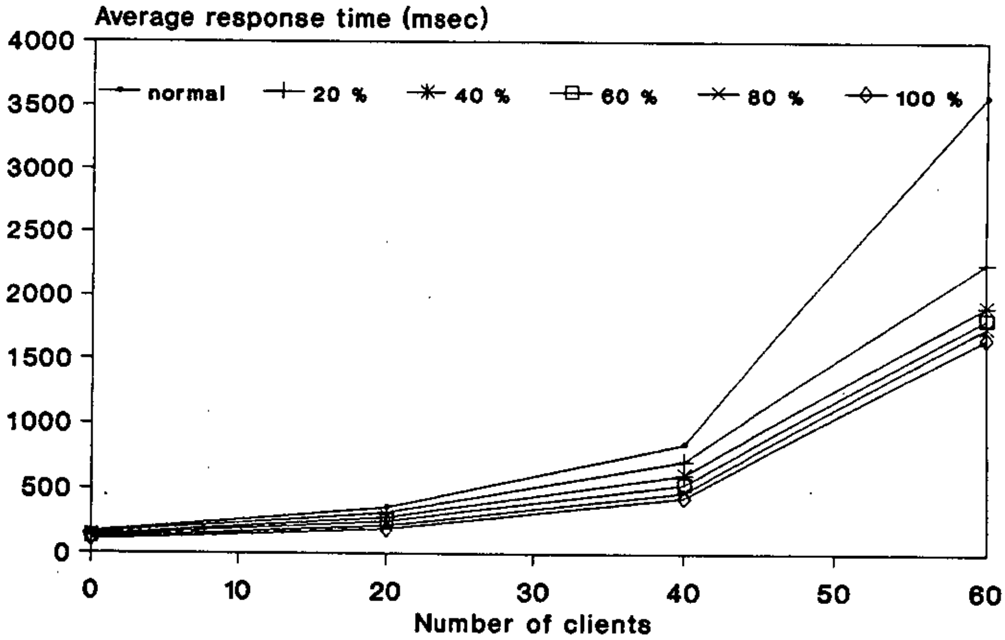


Figure 7.2.1 : The effect on the average response time of caching in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

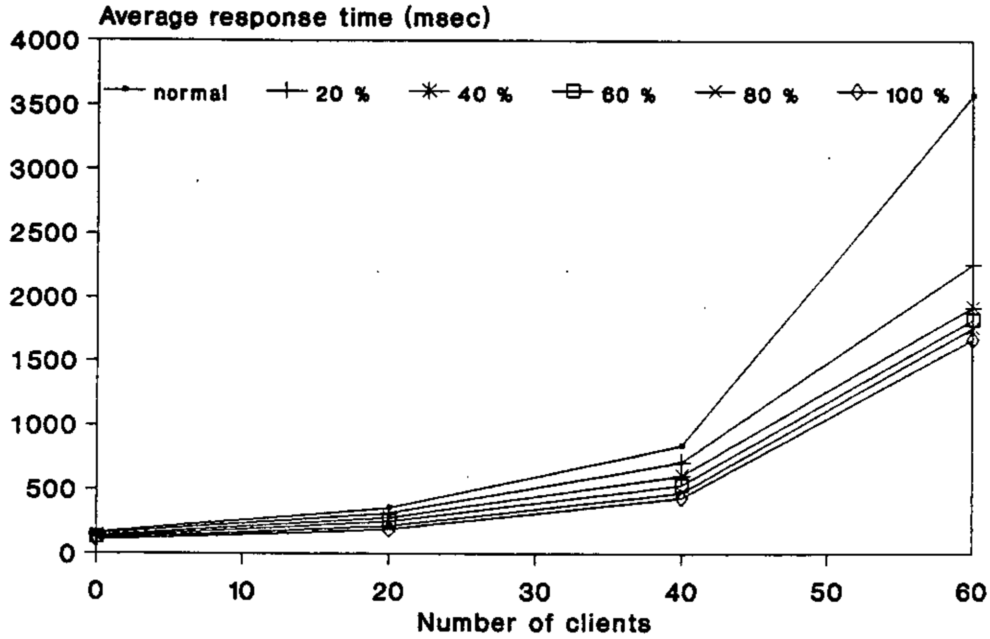


Figure 7.2.2 : The effect on the average response time of caching in the disk interface unit of the file server in the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

The average response time in the standalone caching in the disk interface unit of the file server shows the same pattern as the average response time in the standalone caching in the memory of the file server, even though the former is always larger than the latter. In the distributed file system, even at 100% cache hit rate, the average response time of the 316Kbytes workload and that of the 1856Kbytes workload are still far above 1 second since the network communication overhead remains unchanged but in the shared memory system, the average response time of the 1856Kbytes workload is below 1 second up to more than 15 clients at 80% cache hit rate. When the 60% cache hit occurs, the average response time of the 8Kbytes workload in the distributed file system already shows a similar trend to the average response time of the 8Kbytes workload in the shared memory system when no caching occurs. All six workloads show similar trends in the average response times. No notable change is observed in the pattern of the average response time as the workload size increases and no notable difference is observed between the patterns of the average response times when the steady workloads are used and those when the bursty workloads are used.

7.3 Standalone Caching in the Memory of the Client

This section investigates the effect on file access performance when we use caching in the memory of the client of the distributed file system. If the requested data are in the cache, then all operations in the file server and the network communication operations are bypassed as shown in figure 3.2.6.F. Therefore, the utilization of the file server and that of the network are reduced. In this case, the required operations are similar to those when the cache hit occurs in the memory of the shared memory system : in fact, this is better since there is no contention for the system resources in the clients but there is contention for the system resources in the shared memory system.

Figure 7.3.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

In the distributed file system, regular improvement in the average response time per cache hit rate is observed as the cache hit rate increases since all queueing delays gradually disappear as the cache hit rate increases. The saturation point increases significantly as the cache hit rate increases. It is notable that the average response time of the 1856Kbytes workload is below 1 second up to more than 20 clients at 80% cache hit rate.

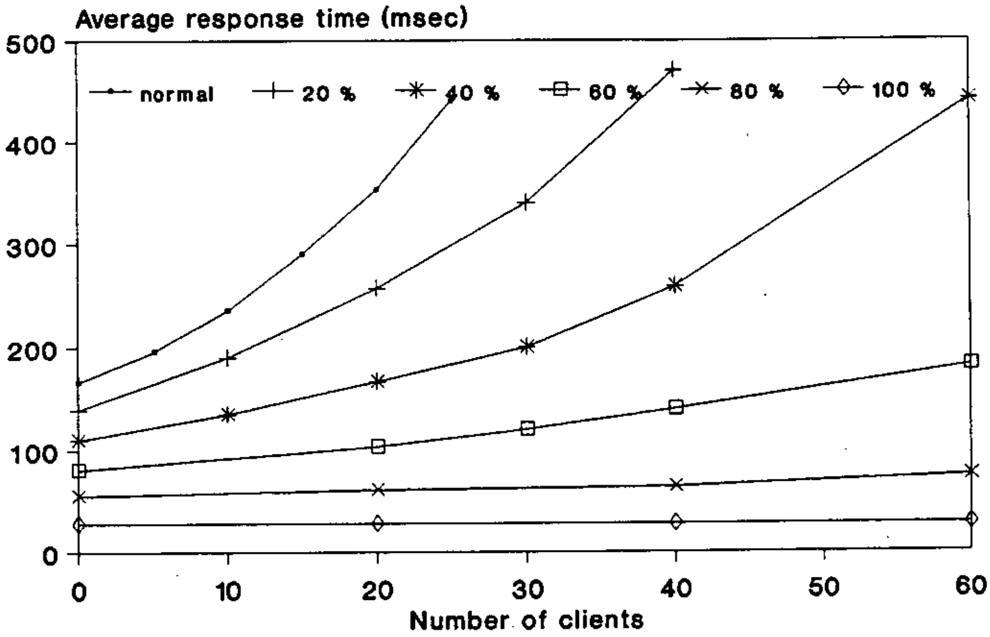


Figure 7.3.1 : The effect on the average response time when we use caching in the disk interface unit of the Sun SPARCstation 10 workstation : the 50.7Kbytes workload.

At 20% cache hit rate, the average response time of the 8Kbytes workload, the 47Kbytes workload and the 50.7Kbytes workload already show better trends than those in the baseline shared memory system where no caching occurs. At 60% cache hit rate, the average response time of the 316Kbytes(B) workload, the

316Kbytes workload and the 1856Kbytes workload already show better trends than those in the baseline shared memory system where no caching occurs.

In all cases except the case for the 100% cache hit, the average response time is slightly higher than that in caching in the memory of the shared memory system since the requests which are missed in the cache must perform all the required operations and the operations are more expensive in the distributed file system than in the shared memory system.

However, at 100% cache hit, regardless of the kind of workload, the average response time is slightly lower than that in caching in the memory of the shared memory systems since there is no contention for the related system resources during operations such as command interpretation, file searching, etc., in the clients of the distributed file system but there are contentions for the related system resources during operations in the shared memory system.

All six workloads show similar trends in the average response times. No notable change is observed in the pattern of the average response time as the average transaction size increases and no notable difference is observed between the patterns of the average response times when the steady workloads are used and those when the bursty workloads are used.

7.4 Standalone Caching in the Disk of the Client

This section investigates the effect on file access performance when we use caching in the disk of the client of the distributed file system.

If the requested data are in the cache, then all operations in the file server and the network communication operations are bypassed as shown in figure 3.2.6.F.

The utilization of the file server and the utilization of the network are reduced. However, additional operations to read the cached data from the disk cache of the client are required. The required operations for the requests are similar to those in the baseline shared memory system where there is no caching except that there is no queueing delay due to the contention for the disk I/O subsystem in these cases since they are performed in the client. Therefore, it is expected that the average response time at 100% cache hit rate should be better than the average response time in the baseline shared memory system where no caching occurs.

Figure 7.4.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

In the distributed file system, regular improvement in the average response time per cache hit rate is observed as the cache hit rate increases since all queueing delays gradually disappear as the cache hit rate increases. The saturation point increases significantly as the cache hit rate increases. The average response time of the 316Kbytes workload is below 1 second up to more than 20 clients at 80% cache hit rate. At 40% cache hit rate, the average response times of the 8Kbytes workload, the 47Kbytes workload and the 50.7Kbytes workload are better than those in the baseline shared memory system where no caching occurs respectively. At 60% cache hit rate, the average response times of the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload are better than those in the baseline shared memory system where no caching occurs. It is observed that the average response time at 100% cache hit rate is constant regardless of the number of clients. This is because there exists no contention in the clients.

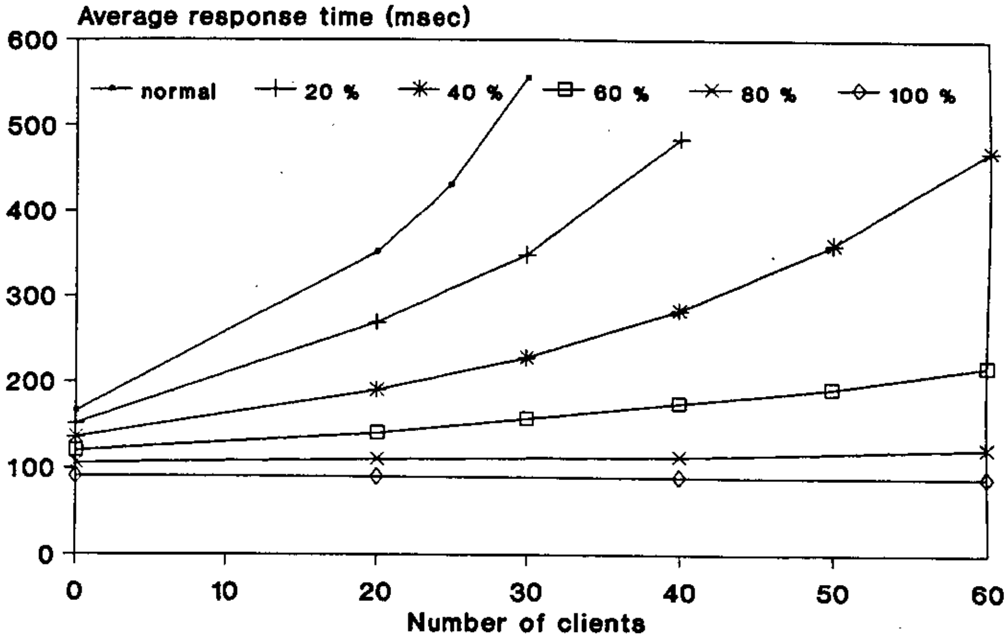


Figure 7.4.1 : The effect on the average response time when we use caching in the disk of the client in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

All six workloads show similar patterns for the average response times. No notable change is observed in the pattern of the average response time as the average transaction size increases and no notable difference is observed between the patterns of the average response times of the steady workloads and those of the bursty workloads.

7.5 Comparison of the Standalone Caching Mechanisms

This section compares the effects on file access performance when we use the four standalone caching mechanisms which were investigated in the previous four sections, that is, standalone caching in the memory of the file server, standalone caching in the disk interface unit of the file server, standalone caching in the memory of the client and standalone caching in the disk of the client in the distributed file system. This section also compares the performances of the previously investigated two caching mechanisms, that is, standalone caching in the memory and standalone caching in the disk interface unit of the shared memory system.

Figure 7.5.1 to figure 7.5.5 compare the average response times of the 50.7Kbytes workload when the cache hit rate is 20%, 40%, 60%, 80% and 100% respectively both in the distributed file system and in the shared memory system. Similar patterns are found in the other cases and the figures of the other cases are not included in this section.

Among the four standalone caching mechanisms in the distributed file system, the best performance, that is, the lowest average response time and the lowest utilizations of the systems resources such as the CPU, the disk I/O subsystem and the network interface unit of the file server are found in the cases when the caching is done in the memory of the client. The next best performance is found in the cases when the caching is done in the disk of the client. The third best performance is found in the cases when the caching is done in the memory of the file server and the worst performance is found in the cases when the caching is done in the disk interface unit of the file server.

As expected, caching in the memory of the shared memory system shows better performance, that is, a lower average response time than for caching in the disk interface unit of the system.

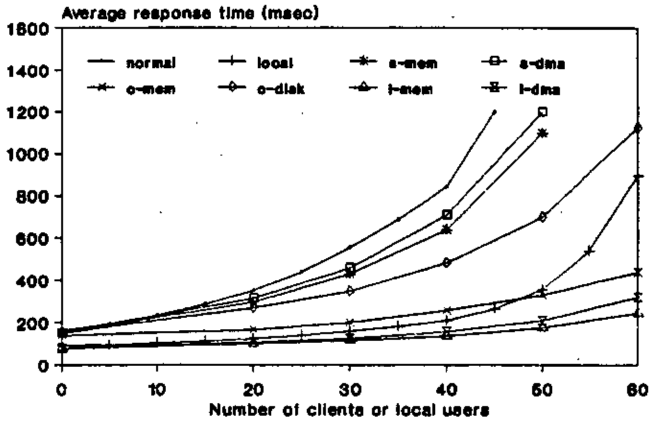


Figure 7.5.1 : 20% cache hit rate

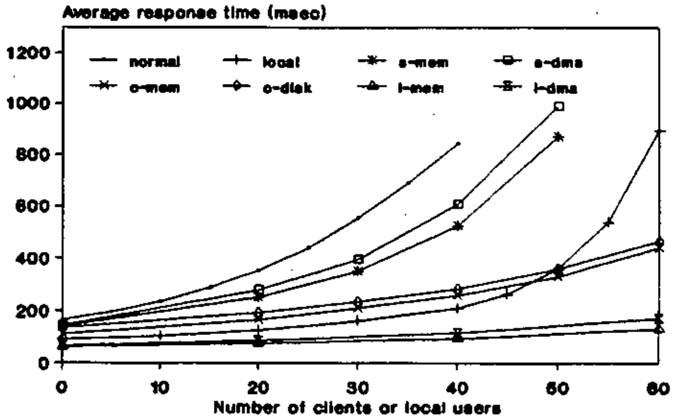


Figure 7.5.2 : 40% cache hit rate

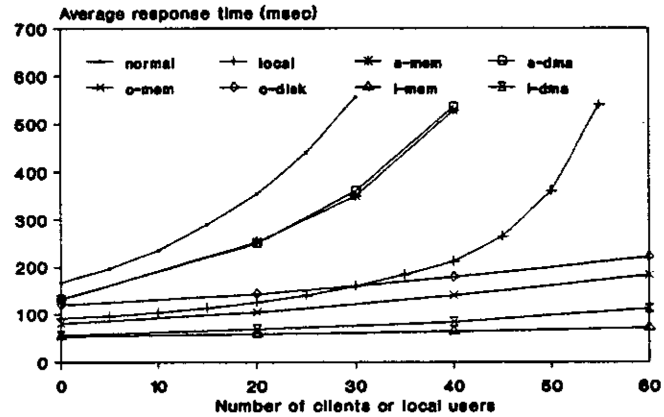


Figure 7.5.3 : 60% cache hit rate

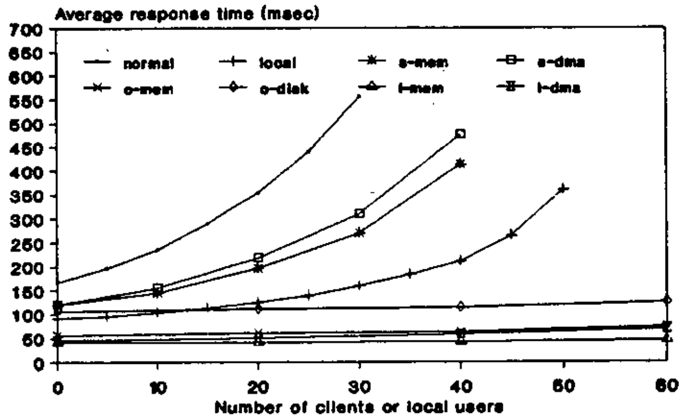


Figure 7.5.4 : 80% cache hit rate

Comparison of the average response times of the four standalone caching mechanisms when the 50.7Kbytes workload is used in the Sun SPARCstation 10 workstation and in the distributed file system which consists of the Sun SPARCstation 10 workstations respectively. Abbreviation : Normal means the average response time in the distributed file system without caching, local means the average response time in the shared memory system without caching, s-mem means the average response time in the caching in the memory of the file server of the distributed file system, l-mem means the average response time in the caching in the memory of the shared memory system, s-dma means the average response time in the caching in the disk interface unit of the file server of the distributed file system, l-dma means the average response time in the caching in the disk interface unit of the shared memory system, c-mem means the average response time in the caching in the memory of the client of the distributed file system and c-disk means the average response time in the caching in the disk of the client of the distributed file system.

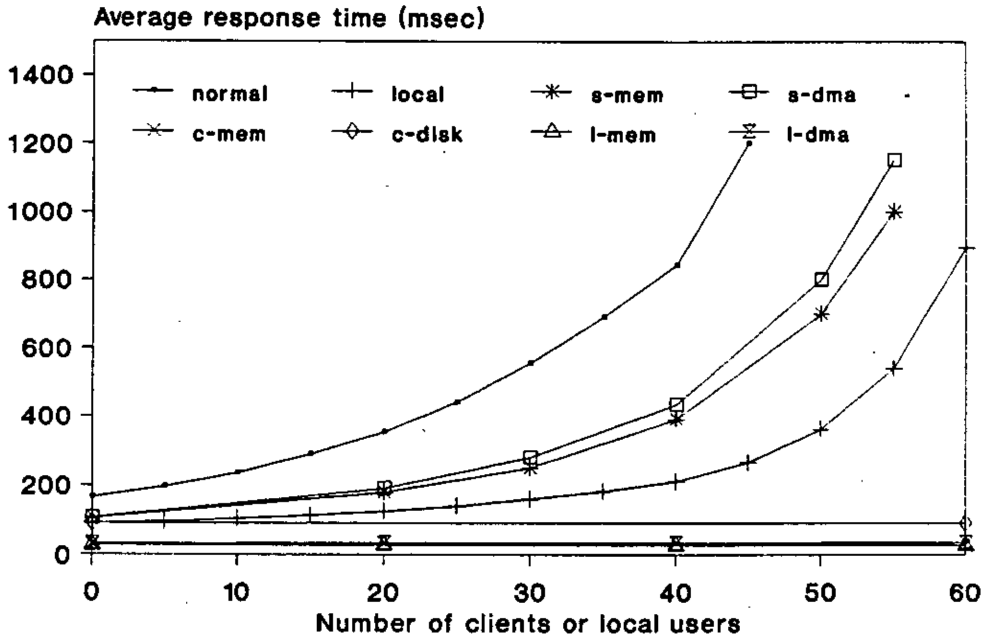


Figure 7.5.5 : Comparison of the average response times of the standalone caching mechanisms at 100% cache hit when the 50.7Kbytes workload is used in the Sun SPARCstation 10 workstation and in the distributed file system which consists of the Sun SPARCstation 10 workstations respectively.

The utilizations of the network communication facilities such as the network and the network interface unit in the file server and in the client when the caching is done in the memory of the client are same as those when the caching is done in the disk of the client in the distributed file system. Therefore, in the two caching mechanisms, the saturation points are same. It increases almost linearly as the cache hit rate increases regardless of the kind of the used workload. But when caching occurs in the memory of the file server or in the disk interface unit of the file server in the distributed file system, the saturation point increases a little up to the saturation point of the network control unit as the cache hit rate increases regardless of the kind of workload used, because the utilization of the network control unit remains unchanged.

In the shared memory system, the caching in the memory shows slightly better average response time than caching in the disk interface unit since caching in the memory saves the CPU service time for the disk I/O operations further as well as it bypasses the operations which caching in the disk interface unit also bypasses.

Figure 7.5.6 compares the average response times of the 50.7Kbytes workload when the caching is done in the memory of the client of the distributed file system and the average response times of the 50.7Kbytes workload when the caching is done in the memory of the shared memory system.

When caching is done in the memory of the client of the distributed file system, except in the cases for the 100% cache hit, the average response time is still slightly higher than that when the caching is done in the memory of the shared memory system since the requests which are missed in cache cause the full operations and they are more expensive in the distributed file system than in the shared memory system.

However, at 100% cache hit, regardless of the kind of the workload, the average response time in caching in the memory of the client of the distributed file system is slightly lower than that in caching in the memory of the shared memory system since there is no contention for the system resources during the operations such as command interpretation, file searching, etc., in the client of the distributed file system but there is contention for the system resources during the operations in the shared memory system.

Generally the six workloads show similar file access performance patterns. It is found that the workload fluctuation does not cause any noticable effect on file access performance.

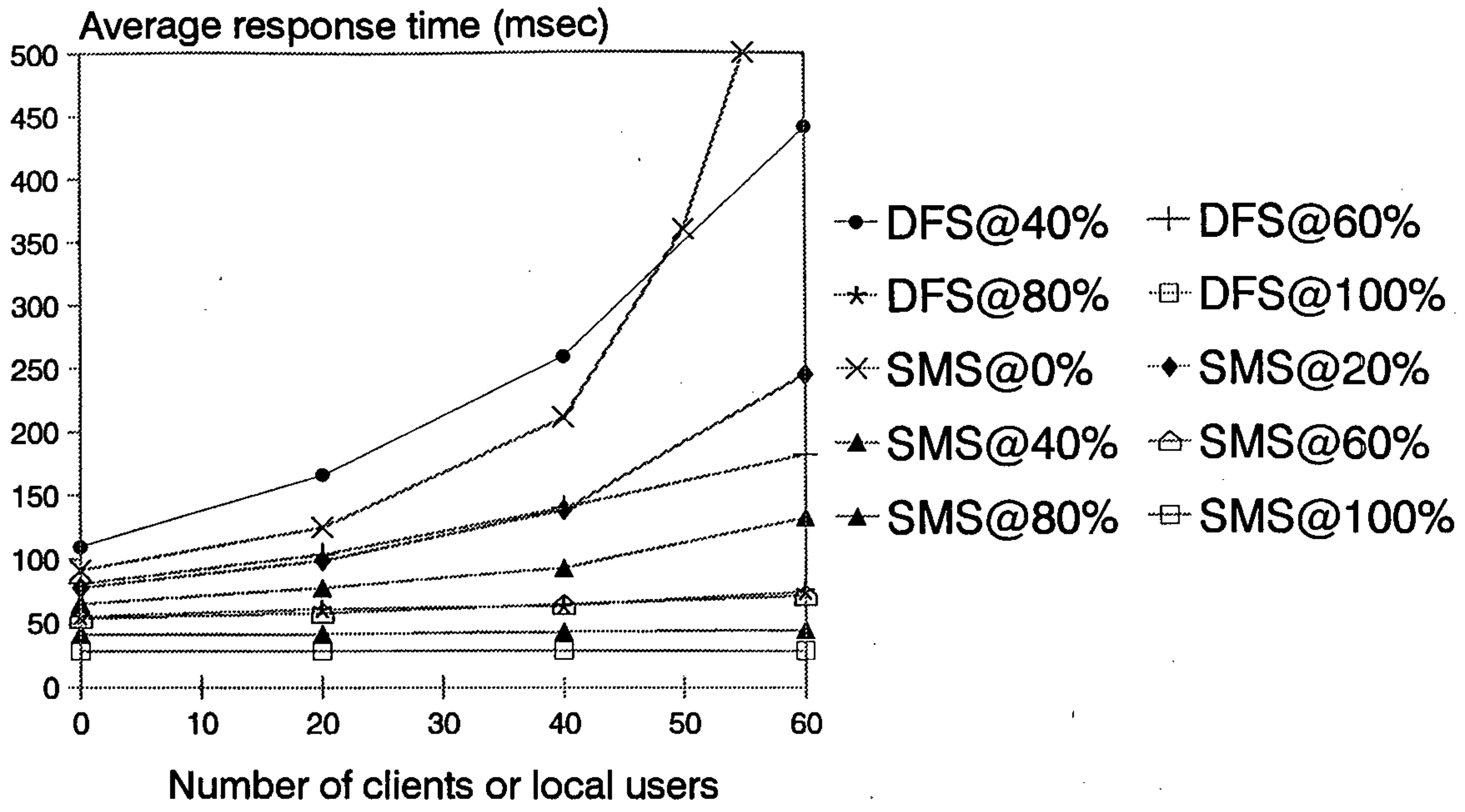


Figure 7.5.6 : The average response times of the 50.7Kbytes workload when caching is done in the memory of the client of the distributed file system which consists of the Sun SPARCstation 10 workstations vs. the average response times of the 50.7Kbytes workload when caching is done in the memory of the Sun SPARCstation 10 workstation. Abbreviation : SMS@20% stands for 20% cache hit in the shared memory system and DFS@20% stands for 20% cache hit in the distributed file system.

So far this study has investigated the effects on file access performance when we use the standalone caching mechanisms but the following sections investigate the effects on file access performance when we use the combinations of the standalone caching mechanisms.

7.6 Combination of Caching in the Memory of the Client and Caching in the Memory of the File Server

This section investigates the effect on file access performance when we use the combination of caching in the memory of the client and caching in the memory of the file server at the same time in the distributed file system.

In this combination, the requests from the client are screened first by the cache in the memory of the client and second by the cache in the memory of the file server. If the requested data are in the memory of client, then the data are fetched for the response and the remaining operations are bypassed. Therefore the network communication cost and all costs in the file server are saved as explained in section 7.3. The utilization of the CPU, the disk interface unit, the disk and the network interface unit of the file server and the network are reduced.

If the requested data are not in the memory of the client but in the memory of the file server, then the cost of the disk I/O operations is saved as explained in section 7.1 and the utilization of the CPU, the disk interface unit and the disk of the file server are reduced.

Figure 7.6.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for these, all others are kept same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

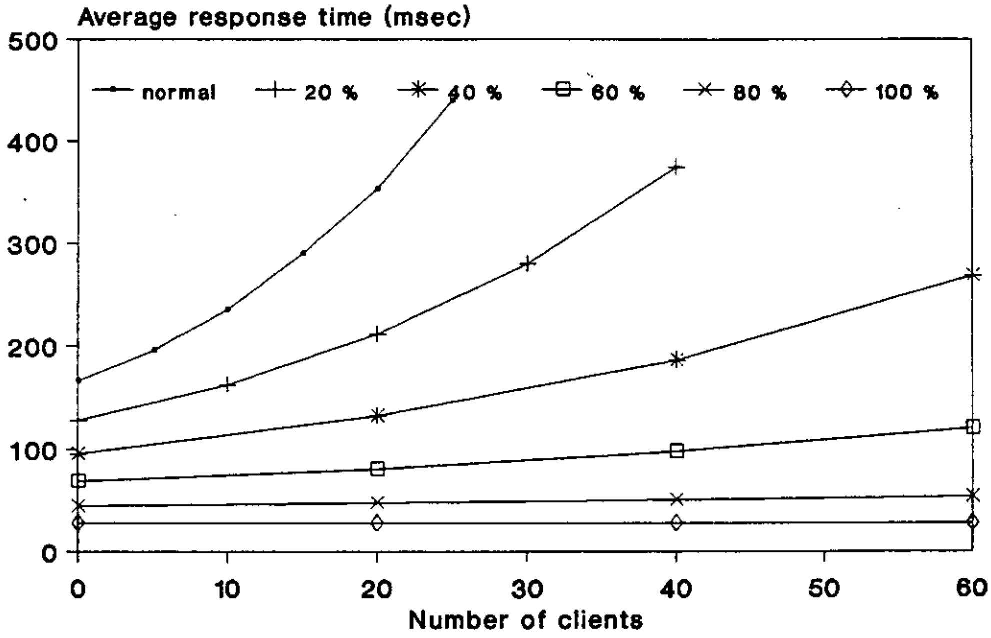


Figure 7.6.1 : The effect on the average response time when we use caching both in the memory of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the both caches improves at the same time : the 50.7Kbytes workload.

Figure 7.6.2 shows the average response time of the 50.7Kbytes workload in the distributed file system when the hit rate of the cache in the memory of the client is improved to be 20%, 40%, 60%, 80% and 100% while the hit rate of the cache in the memory of the file server is fixed to be 60%. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

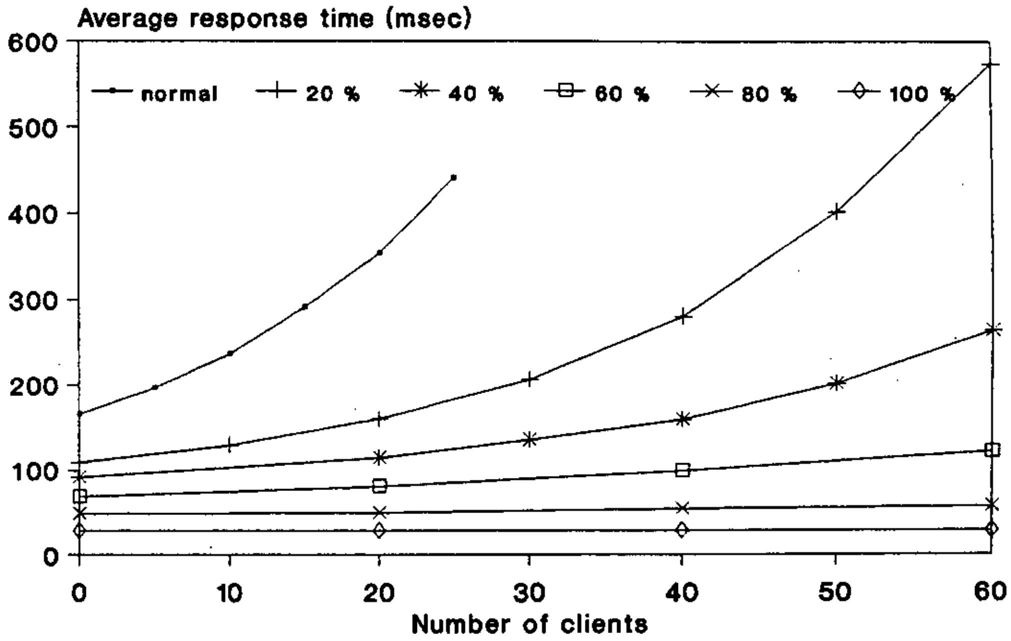


Figure 7.6.2 : The effect on the average response time when we use caching both in the memory of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the cache in the memory of the client improves while the hit rate of the cache in the memory of the file server is fixed to be 60% : the 50.7Kbytes workload.

In the distributed file system, regular improvement in the average response time is observed as the cache hit rate increases since all queueing delays gradually disappear at the same rate as the cache hit rate increases. The saturation point increases significantly as the cache hit rate increases. Generally the performance pattern when the hit rate is varied in both caches at the same time is similar to that when the hit rate is varied in one of the two caches while the hit rate in the other cache is fixed all the time.

At 100% cache hit rate, the average response time is the same as the average response time of the standalone caching in the memory of the client. At 100% cache hit rate, the average response time is constant as the number of clients increases since there is no contention for the system resources.

The combined caching shows better average response time than the standalone caching in the memory of the client. All six workloads show similar patterns of the average response times. No notable change is observed in the pattern of the average response time as the average transaction size increases and no notable difference is observed between the patterns of the average response times of the steady workloads and those of the bursty workloads.

7.7 Combination of Caching in the Disk of the Client and Caching in the Memory of the File Server

This section investigates the file access performance when caching is done in the disk of the client and in the memory of the file server at the same time in the distributed file system.

In this combination, the requests from the client are screened first by the cache in the disk of the client and second by the cache in the memory of the file server. If the requested data are in the disk of the client, then the data are fetched for the response and the remaining operations are bypassed. Therefore the network communication cost and all costs in the file server are saved as explained in section 7.3. The utilization of the CPU, the disk interface unit, the disk and the network interface unit of the file server and the network are reduced. However, the cost of the disk I/O operations is paid in the client where there is no contention for the system resources. If the requested data are not in the disk of the client but in the memory of the file server, then only the cost of the disk I/O operations is saved as explained in section 7.1. The utilization of the CPU, the disk interface unit and the disk of the file server are reduced.

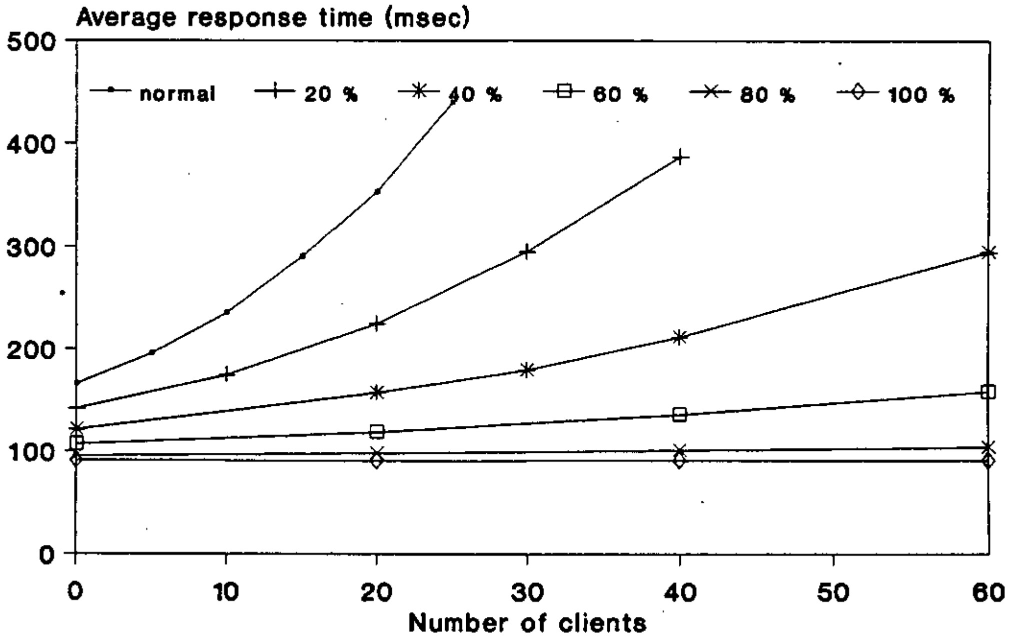


Figure 7.7.1 : The effect on the average response time when we use caching both in the disk of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the both caches improves at the same time : the 50.7Kbytes workload.

Figure 7.7.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

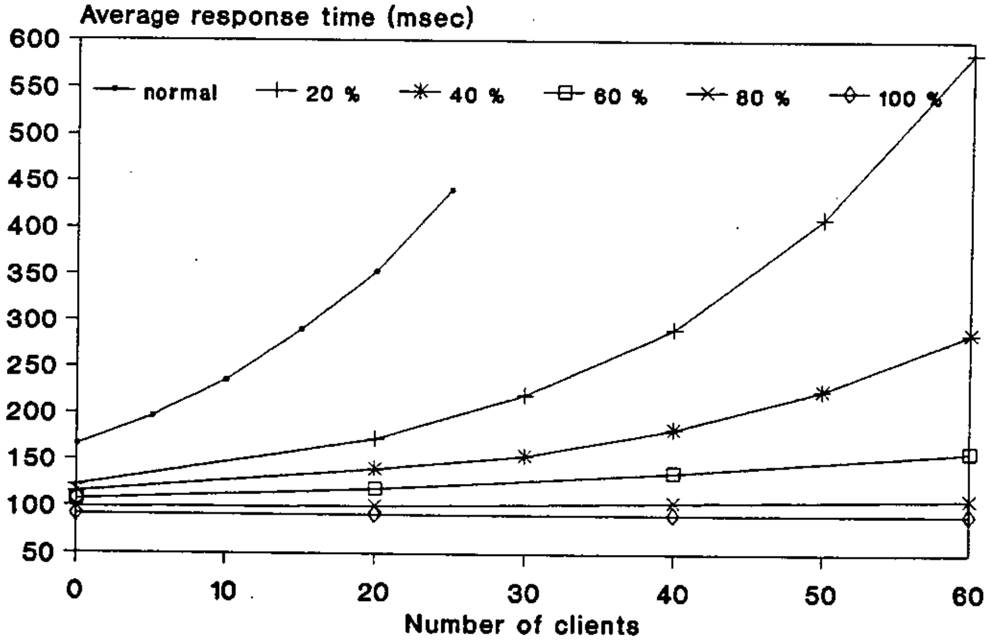


Figure 7.7.2 : The effect on the average response time when we use caching both in the disk of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the cache in the memory of the client improves while the the hit rate of the cache in the memory of the file server is fixed to be 60% : the 50.7Kbytes workload.

Figure 7.7.2 shows the average response time of the 50.7Kbytes workload in the distributed file system when the hit rate of the cache in the disk of the client is improved to be 20%, 40%, 60%, 80% and 100% while the hit rate of the cache in the memory of the file server is fixed to be 60% all the time. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

In the distributed file system, regular improvement in the average response time is observed as the cache hit rate increases since all queueing delays gradually disappear at the same rate as the cache hit rate increases. The saturation point increases significantly as the cache hit rate increases. Generally the performance pattern when the hit rate is varied in both caches at the same time is similar to the performance pattern when the hit rate is varied in one of the two caches while the hit rate in the other cache is fixed all the time.

At 100% cache hit rate, the average response time is same as the average response time of the standalone caching in the disk of the clients. At 100% cache hit rate, the average response time is constant as the number of clients increases since there is no contention for the system resources.

The combined caching shows better average response time than the standalone caching in the disk of the client. All six workloads show similar patterns of the average response times. No notable change is observed in the pattern of the average response time as the workload size increases and no notable difference is observed between the patterns of the average response times of the steady workloads and those of the bursty workloads.

At 100% cache hit rate, the average response time is the same as the average response time of the standalone caching in the memory of the client. At 100% cache hit rate, the average response time is constant as the number of clients increases since there is no contention for the system resources in the client.

The combined caching shows better average response time than the combination of caching in the memory of the client and caching in the memory of the file server whose file access performance was investigated in section 7.5.

All six workloads show similar patterns of average response times. No notable change is observed in the pattern of the average response time as the workload size increases and no notable difference is observed between the patterns of the average response times of steady workloads and those of bursty workloads.

7.9 Combination of Caching in the Disk of the Client, Caching in the Memory of the File Server and Caching in the Disk Interface Unit of the File Server

This section investigates the effect on file access performance when we use the combination of caching in the disk of the client, caching in the memory of the file server and caching in the disk interface unit of the file server in the distributed file system.

In this combination, the requests from the client are screened first by the cache in the disk of the client, second by the cache in the memory of the file server and third and last, by the cache in the disk interface unit of the file server. If the requested data are in the disk of the client, then the data are fetched for the response and the remaining operations are bypassed. Therefore the network communication cost and all costs in the file server are saved. The utilization of the CPU, the disk interface unit, the disk and the network interface unit of the file server and the network are reduced. However, the cost of the disk I/O operations accessing the disk cache is paid in the client where there is no contention for the system resources as explained in section 7.3. If the requested data are not found in the disk of the client but found in the memory of the file server, then all disk I/O operations in the file server are saved as explained in section 7.1. The utilization of the CPU, the disk interface unit and the disk of the file server are reduced. If the requested data are not found in the cache in the disk of the client and not in the cache in the memory of the file server but found in the cache in the disk interface unit of the file server, then the cost of the operations for I/O in the disk interface unit and the disk is saved as explained in section 7.2. The utilization of the CPU, the disk interface unit and the disk of the file server are reduced.

Figure 7.9.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in the three caches at the same time. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

In the distributed file system, regular improvement in the average response time is observed as the cache hit rate increases since all queueing delays gradually disappear at the same rate as the cache hit rate increases. The saturation point increases significantly as the cache hit rate increases.

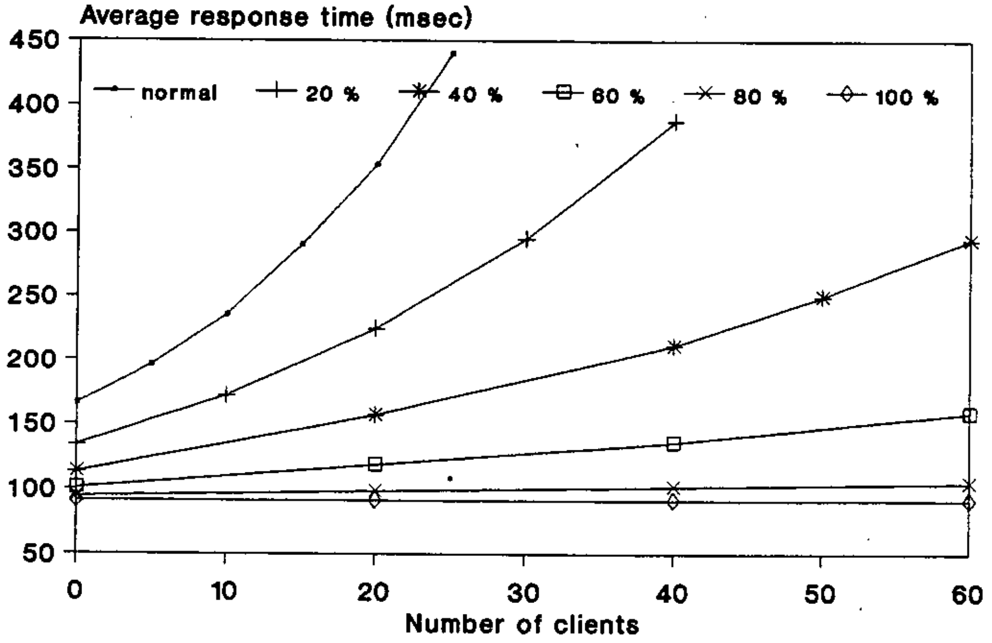


Figure 7.9.1 : The effect on the average response time when we use caching in the disk of the client, in the memory of the file server and in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations : the 50.7Kbytes workload.

At 100% cache hit rate, the average response time is the same as the average response time of the standalone caching in the disk of the client. At 100% cache hit rate, the average response time is constant as the number of clients increases since there is no contention for system resources.

This combined caching shows better average response time than the combined caching in the disk of the client and the memory of the file server whose file access performance was investigated in section 7.7.

All six workloads show similar patterns of the average response time. No notable change is observed in the pattern of the average response time as the workload size increases and no notable difference is observed between the patterns of the average response times of the steady workloads and those of the bursty workloads.

7.10 Combination of Caching in the Memory and Caching in the Disk Interface Unit in a System

This section investigates the effect on file access performance when we use the combination of caching in the memory of the file server and in the disk interface unit of the file server in the distributed file system. Comparatively in the shared memory system, the effect on file access performance is also investigated when the caching is done in the memory and in the disk interface unit at the same time.

In this combination, the requests from the client are screened first by the cache in the memory of the file server and second by the cache in the disk interface unit of the file sever in the distributed file system. If the requested data are found in the memory, then all disk I/O operations are saved as explained in section 6.21.1. The utilization of the CPU, the disk interface unit and the disk are reduced. If the requested data are not found in the memory but found in the disk interface unit, then the operations for the disk I/O in the disk interface unit and the disk are saved as explained in section 7.2. The utilization of the disk interface unit and the disk are reduced.

Figure 7.10.1 shows the average response time of the 50.7Kbytes workload in the distributed file system as the number of clients increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

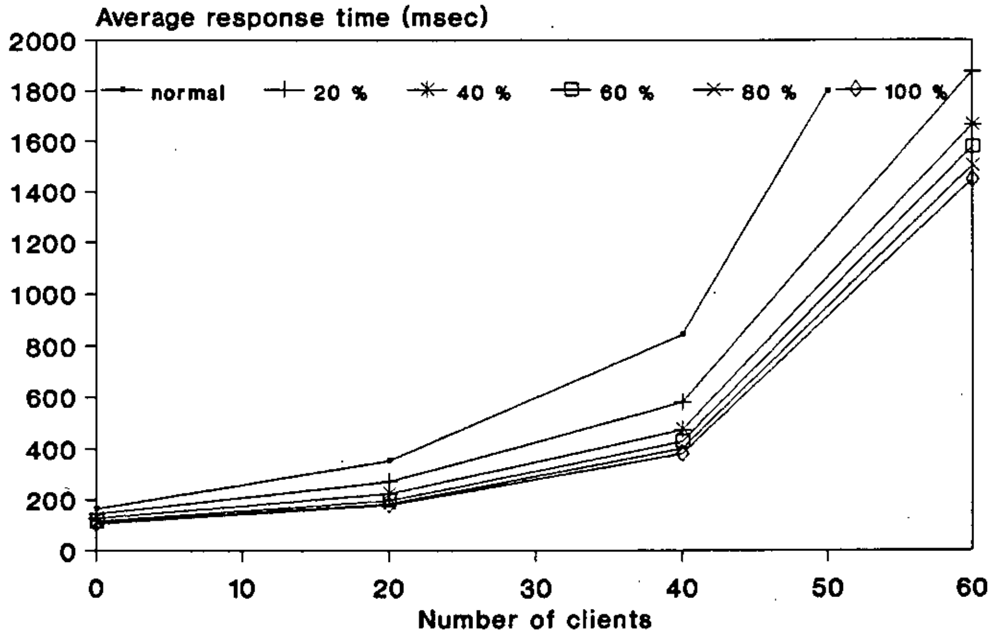


Figure 7.10.1 : The effect on the average response time when we use caching both in the memory of the file server and in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the both caches improves at the same time : the 50.7Kbytes workload.

Figure 7.10.2 shows the average response time of the 50.7Kbytes workload in the distributed file system when the hit rate of the cache in the disk interface unit of the file server is improved to be 20%, 40%, 60%, 80% and 100% while the hit rate of the cache in the memory of the file server is fixed to be 60% all the time. Except for these, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. See appendix D for the figures of other cases.

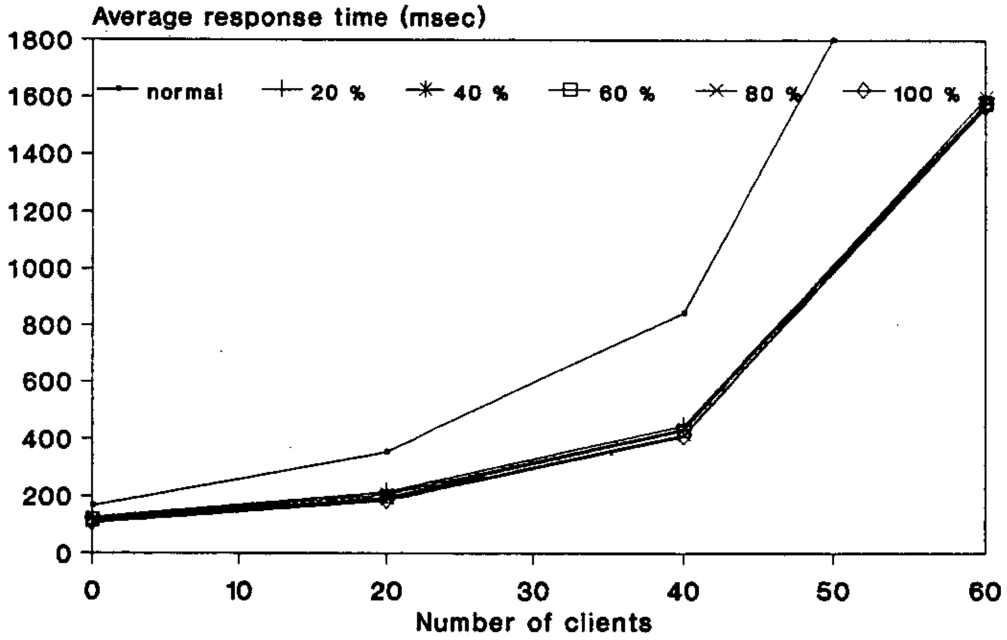


Figure 7.10.2 : The effect on the average response time when we use caching both in the memory of the file server and in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the cache in the memory of the client improves while the the hit rate of the cache in the memory of the file server is fixed to be 60% : the 50.7Kbytes workload.

Figure 7.10.3 shows the average response time of the 50.7Kbytes workload in the shared memory system as the number of local users increases gradually. The cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for these, all others are kept the same as the baseline Sun SPARCstation 10 workstation. See appendix D for the figures of other cases.

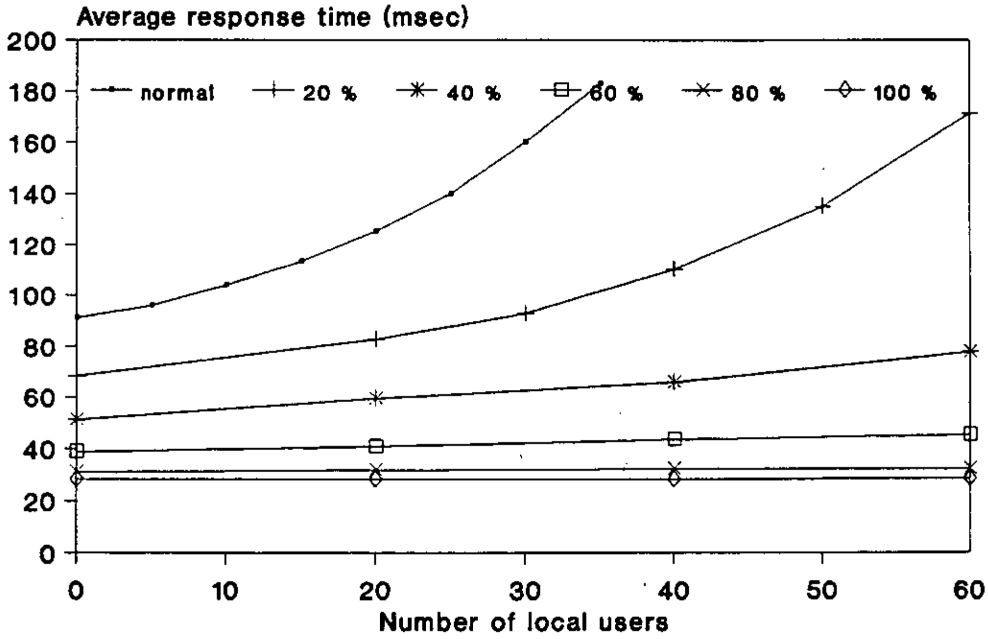


Figure 7.10.3 : The effect on the average response time when we use caching both in the memory and in the disk interface unit of the Sun SPARCstation 10 workstation when the hit rate of the both caches improves at the same time : the 50.7Kbytes workload.

In the distributed file system, it is observed that the 20% hit rate case shows the best improvement rate of the average response time per cache hit rate, then gradually the improvement rate reduces.

In the shared memory system, almost linear improvement of the average response time is observed as the cache hit rate increases since all queueing delays gradually disappear at the same rate as the cache hit rate increases.

The queueing delay caused by the contention for system resources during network communication remains unchanged in the distributed file system but all queueing delays gradually disappear in the shared memory system as the cache hit rate increases. The saturation point of the distributed file system increases a little up to the saturation point of the network interface unit but the saturation point for the shared memory system increases significantly as the cache hit rate increases.

At 100% cache hit rate, the average response time is the same as the average response time of the standalone caching in the memory in both system paradigms. The combined caching shows better average response time than the standalone caching in the memory. All six workloads show similar patterns of the average response times. No notable change is observed in the pattern of the average response time as the workload size increases and no notable difference is observed between the patterns of the average response times of the steady workloads and those of the bursty workloads.

7.11 Comparison of All Caching Mechanisms

This section compares the effects on the file access performance in the distributed file system when we use the 4 standalone caching mechanisms and 5 combined caching mechanisms which were investigated in the previous 9 sections. They are the following.

- Standalone caching in the memory of the file server.
- Standalone caching in the disk interface unit of the file server.

- Standalone caching in the memory of the client.
- Standalone caching in the disk of the client.
- The combination of caching in the memory of the client and caching in the memory of the file sever.
- The combination of caching in the disk of the client and caching in the memory of the file sever.
- The combination of caching in the memory of the client, caching in the memory of the file sever and caching in the disk interface unit of the file server.
- The combination of caching in the disk of the client, caching in the memory of the file sever and in the disk interface unit of the file server.
- The combination of caching in the memory of the file server and caching in the disk interface unit of the file sever.

This section also compares the effects on the file access performance in the shared memory system when we use the two standalone caching mechanisms and one combined caching mechanism which were investigated in the previous 3 sections. They are the following.

- Standalone caching in the memory.
- Standalone caching in the disk interface unit.
- The combination of caching in the memory and in the disk interface unit.

So far this study has used absolute cache hit rates at each cache all the time. In order to compare all caching mechanisms including the combined caching mechanisms, it is useful to know relative cache hit rates at each cache. For example, when caching is done in the memory of the client, in the memory of the file server and in the disk interface unit of the file server at the same time, the 60% cache hit rate at each of the three caches, which is called a 60% absolute cache hit rate, means a 60% cache hit rate in the first cache in the memory of the client, a 24% cache hit rate, which is called a 24% relative cache hit rate in the second cache in the memory of the file server, since the second cache hit occurs among the portions which are missed in the first cache, and a 9.6% relative cache hit rate in the third cache in the disk interface unit of the file server. Therefore, when the absolute hit rate is 60% in each of the three caches, the total relative hit rate is 60% at the first cache, 84% at the second cache and 93.6% at the third cache. The table 7.11.1 shows the relative cache hit rate, the total relative cache hit rate and the total relative cache miss rate at the absolute cache hit rate of 20%, 40%, 60%, 80% and 100% respectively.

Absolute cache hit rate (%)	Relative cache hit rate (%)			Total relative cache hit rate (%)			Total relative cache miss rate (%)		
	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
20	20	16	12.8	20	36	48.8	80	64	51.2
40	40	24	14.4	40	64	78.4	60	36	21.6
60	60	24	9.6	60	84	93.6	40	16	6.4
80	80	16	3.2	80	96	99.2	20	4	0.8
100	100	0	0	100	100	100	0	0	0

Table 7.11.1 : Absolute cache hit rate, relative cache hit rate, total relative cache hit rate and total relative cache miss rate.

Figure 7.11.1 and figure 7.11.2 compare the average response times of the 9 caching mechanisms at the cache hit rate of 40% and 60% when the 50.7Kbytes workload is used in the distributed file system and the average response times of the 3 caching mechanisms at the cache hit rate of 40% and 60% when the 50.7Kbytes workload is used in the shared memory system. In the figures, the following abbreviations are used.

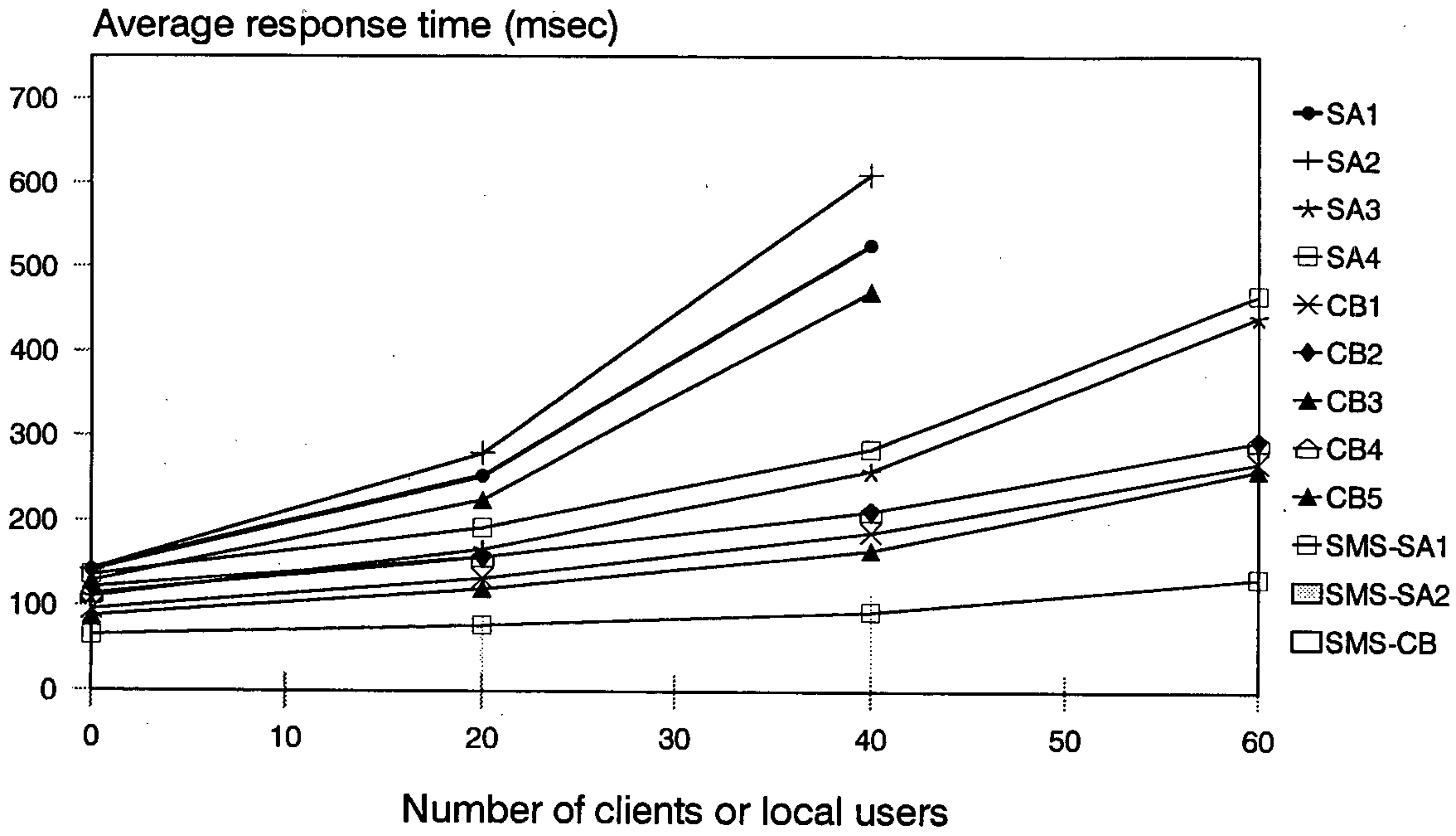


Figure 7.11.1 : The average response times of the 9 caching mechanisms at 40% hit in each cache when the 50.7Kbytes workload is used in the distributed file system which consists of the Sun SPARCstation 10 workstations and the average response times of the 3 caching mechanisms at 40% hit in each cache when the 50.7Kbytes workload is used in the Sun SPARCstation 10 workstation.

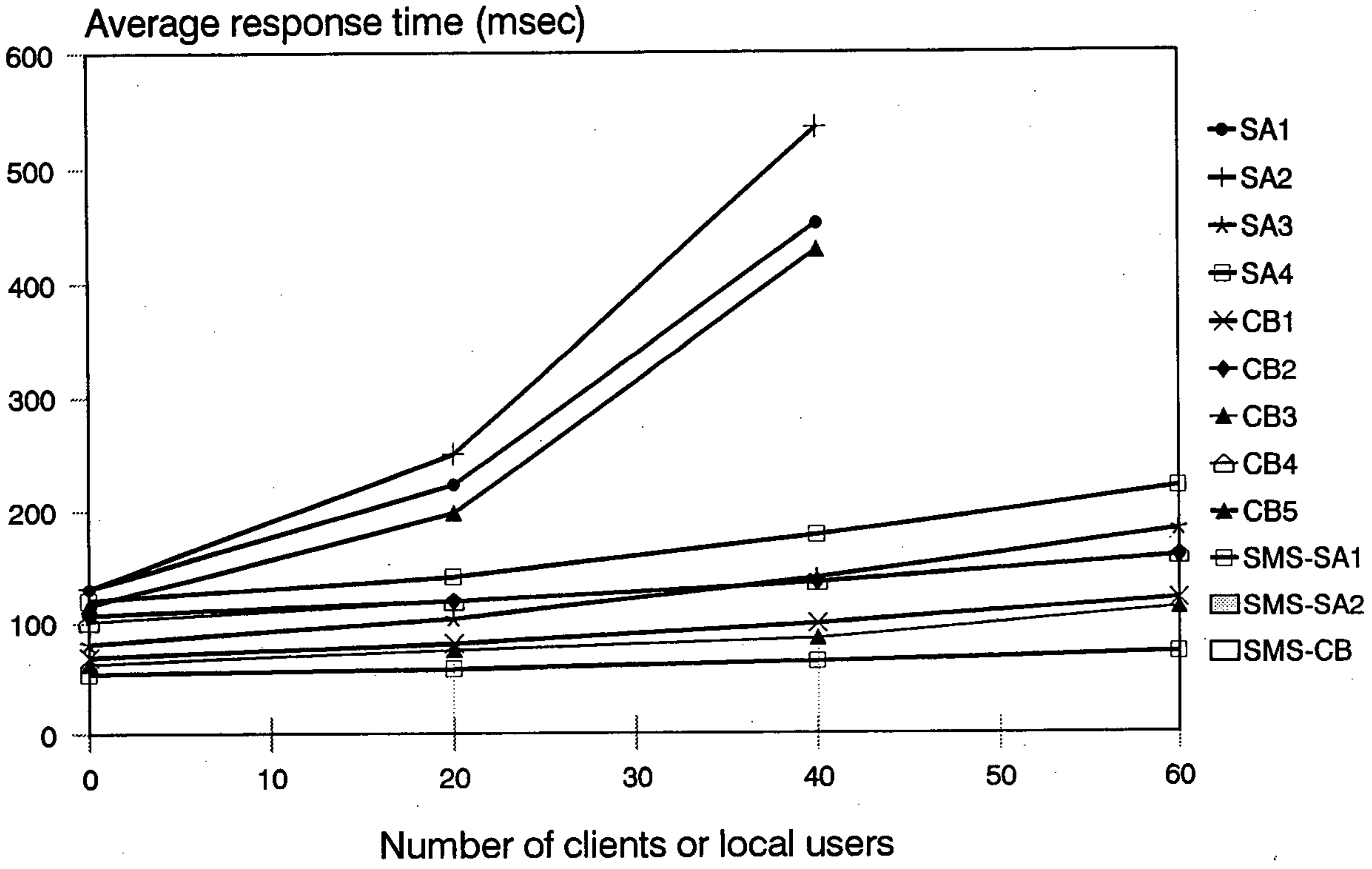


Figure 7.11.2 : The average response times of the 9 caching mechanisms at 60% hit in each cache when the 50.7Kbytes workload is used in the distributed file system which consists of the Sun SPARCstation 10 workstations and the average response times of the 3 caching mechanisms at 60% hit in each cache when the 50.7Kbytes workload is used in the Sun SPARCstation 10 workstation.

In the distributed file system,

- SA1 : Standalone caching in the memory of the file server.
- SA2 : Standalone caching in the disk interface unit of the file server.
- SA3 : Standalone caching in the memory of the client.
- SA4 : Standalone caching in the disk of the client.
- CB1 : The combination of caching in the memory of the client and caching in the memory of the file sever.
- CB2 : The combination of caching in the disk of the client and caching in the memory of the file sever.
- CB3 : The combination of caching in the memory of the client, caching in the memory of the file sever and caching in the disk interface unit of the file server.
- CB4 : The combination of caching in the disk of the client, caching in the memory of the file sever and in the disk interface unit of the file server.
- CB5 : The combination of caching in the memory of the file server and caching in the disk interface unit of the file sever.

In the shared memory system,

- SMS-SA1 : Standalone caching in the memory.
- SMS-SA2 : Standalone caching in the disk interface unit.
- SMS-CB1 : The combination of caching in the memory and in the disk interface unit.

At 100 % cache hit rate in the distributed file system, the average response time is the same in the three caching mechanisms, that is, the standalone caching in the memory of the client, the combination of caching in the memory of the client and caching in the memory of the file server and the combination of caching in the memory of the client, caching in the memory of the file server and caching in the disk interface unit of the file server and the average response time is the same in the three caching mechanisms, that is, the standalone caching in the disk of the client, the combination of caching in the disk of the client and caching in the memory of the file server, and the combination of caching in the disk of the client, caching in the memory of the file server and caching in the disk interface unit of the file server.

Also at 100% cache hit rate, the average response time is constant for the six caching mechanisms above as the number of clients increases since there is no contention for the system resources.

At 100 % cache hit rate in the shared memory system, the average response time is the same in the two caching mechanisms, that is, the standalone caching in the memory and the combination of caching in the memory and caching in the disk interface unit.

Among the 9 caching mechanisms in the distributed file system, the best performance, that is, the lowest average response time and the lowest utilization is found when caching is done in the memory of the client, in the memory of the file sever and in the disk interface unit of the file server at the same time. The worst performance is found when standalone caching is done in the disk interface unit of the file server. The following shows the descending order from the best to the worst in terms of the file access performance in the distributed file system.

1) The combination of caching in the memory of the client, in the memory of the file sever and in the disk interface unit of the file server.
2) The combination of caching in the memory of the client and in the memory of the file sever.
3) or 4) or 5) Standalone caching in the memory of the client.
4) or 3) The combination of caching in the disk of the client, in the memory of the file sever and in the disk interface unit of the file server.

5) or 4) The combination of caching in the disk of the client and in the memory of the file sever.

6) Standalone caching in the disk of the client.

7) The combination of caching in the memory of the file server and in the disk interface unit of the file sever.

8) Standalone caching in the memory of the file server.

9) Standalone caching in the disk interface unit of the file server.

In the shared memory system, the combination of caching in the memory and in the disk interface unit shows the best file access performance, the standalone caching in the memory shows the second best file access performance and the standalone caching in the disk interface unit shows the worst file access performance. In the shared memory system, caching in the memory shows slightly better average response time than caching in the disk interface unit since caching in the memory saves the CPU service time for the disk I/O operations as well as it bypasses the operations which caching in the disk interface unit also bypasses.

The utilization of the network is lowest when caching is done in the client and highest when caching is done in the file server. The utilizations of the network communication facilities such as the network and the network interface unit in the file server and the client when caching is done in the memory of the client are the same at given cache hit rates as those when caching is done in the disk of the client in the distributed file system. Therefore, in the two caching mechanisms, the saturation points are same. The saturation points of the two caching mechanisms increase almost linearly as the cache hit rate increases regardless of the kind of the workload used. But when caching is done in the memory of the file server or in the disk interface unit of the file server in the distributed file system, the saturation point increases a little up to the saturation point of the network interface unit as the cache hit rate increases regardless of the kind of the workload used, since the utilization of the network interface unit remains unchanged.

At 100% cache hit, regardless of the kind of the workload used, the average response time of caching in the memory of the client in the distributed file system is slightly lower than the average response time of caching in the memory of the shared memory system since the operations such as command interpretation, file searching, etc., are performed in the client and it is assumed that there is no contention for the system resources in the client of the distributed file system but in the shared memory system there is contention for the system resources and the operations performed there compete with other operations.

Generally the six workloads show similar file access performance patterns. It is found that the workload fluctuation does not cause any noticable effect on file access performance.

7.12 Summary

This study dealt with the cache hits which did not require any pre-operations or post-operations at all. It was assumed that the cache consistency maintenance overhead was zero, which was the theoretical limit. This study has used absolute cache hit rates at each cache all the time.

All six workloads show similar trends in the average response times. In each case, no notable change is observed in the pattern of the average response time as the workload size increases and no notable difference is observed between the patterns of the average response times when steady workloads are used and those when bursty workloads are used.

The average response time of the 8Kbytes workload in the distributed file system with a 40% cache hit shows a similar trend to the average response time of the

8Kbytes workload in the shared memory system when no caching occurs.

The saturation point increases significantly as the cache hit rate increases in the memory of the client of the distributed file system. At 20% cache hit rate, the average response time of the 8Kbytes workload, the 47Kbytes workload and the 50.7Kbytes workload already show better trends than those in the baseline shared memory system where no caching occurs. At 60% cache hit rate, the average response time of the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload already show better trends than those in the baseline shared memory system where no caching occurs.

Among the four standalone caching mechanisms in the distributed file system, the best performance, that is, the lowest average response time and the lowest utilizations of the systems resources such as the CPU, the disk I/O subsystem and the network interface unit of the file server are found in the cases when the caching is done in the memory of the client. The next best performance is found in the cases when the caching is done in the disk of the client. The third best performance is found in the cases when the caching is done in the memory of the file server and the worst performance is found in the cases when the caching is done in the disk interface unit of the file server.

Among the 9 caching mechanisms in the distributed file system, the best performance, that is, the lowest average response time and the lowest utilization is found when caching is done in the memory of the client, in the memory of the file sever and in the disk interface unit of the file server at the same time. The worst performance is found when standalone caching is done in the disk interface unit of the file server. The utilization of the network is lowest when caching is done in the client and highest when caching is done in the file server.

In the shared memory system, the combination of caching in the memory and in the disk interface unit shows the best file access performance, the standalone caching in the memory shows the second best file access performance and the standalone caching in the disk interface unit shows the worst file access performance.

Chapter 8

Remarks

8.1 Conclusions

At the beginning of this dissertation, I presented the research problems and the research objectives. From chapter 2 to chapter 7, this study proceeded to seek the solutions of the research problems and to achieve the research objectives. Below, I summarize the solutions of the research problems.

1) How to accurately and efficiently model the two computer system paradigms using the queueing network theory?

Chapter 3 presents the virtual server models. It is easy to construct the performance models using the virtual server concept. The virtual server models are flexible and easily modified to accommodate the changes in the target systems and yet the models which were used are found to predict the file access performance of the real systems very precisely.

2) What performance parameters will this study use for the performance models and how to obtain the parameter values?

Chapter 3 presents the special parameterization methodology. Chapter 3 and chapter 4 describe the measurement methodology to obtain the parameter values. No special performance measurement tool except the available standard UNIX facilities was used to measure the file access performance to obtain the parameter values. Nonetheless I got very accurate parameter values using the parameterization methodology. This enables me and others to reproduce easily what has been studied in this thesis in other UNIX environments or to apply them to other UNIX environments.

3) How to obtain the accurate, realistic and representative artificial workloads for the performance models from the real measured workloads in the two system paradigms?

This study proposed the workload characterization methodology which consists of six steps. As the baseline data, this study used file I/O statistics measured in the three VAX 11/780 systems with BSD 4.2 UNIX and the file I/O statistics measured in Sprite distributed system of the Computer Science Department of University of California, Berkeley. The six realistic and representative artificial workloads were obtained after the representativeness of them was carefully investigated in another very large scale distributed system.

4) How to solve the performance models?

Simulation was used as the main methodology and the analytic approach was used as an auxiliary method to solve the performance models in this research. Using SLAM-II simulation packages, the virtual server models were easily implemented as simulation programs. It was observed that the simulation predicted the file access performance of the target systems very precisely. This study used most of the typical performance indices such as response time, queue length, waiting time, utilization, etc. during the simulations.

5) How to verify the simulation programs?

This study compared the simulation results with the analytic solutions case by case after obtaining the parameter values and confirmed the two were exactly same.

6) How to measure the real performance and validate the performance models?

This study performed standalone measurement experiments and real world measurement experiments in the environments of the two system paradigms to validate the performance models and the simulation results. It is more difficult than simply measuring the performance in the real environments of the two system paradigms since deliberately designed scenarios should be carefully executed and we have to capture the real performance accurately in time. As in the measurement to obtain the performance parameter values, no special performance measurement tool except the standard UNIX facilities was used to measure the file access performance to validate the parameter models.

Below I summarize what this study has found while achieving the research objectives, recalling the objectives presented in chapter 1.

The first objective is to comparatively evaluate the file access performances of the two system paradigms using currently available systems.

The distributed file systems and the shared memory systems showed similar patterns of file access performance in general. The average response time of the distributed file system was always larger than the average response time of the equivalent shared memory system as expected. When the communication overhead was reduced to be infinitesimal by using faster computer communication, better hardware and better mechanisms, the average response time of the distributed file system became very close to that of the equivalent shared memory system as expected.

The second objective is to explore the file system design issues.

When this study compared the file access performance of the better CPU cases with the file access performance of the equivalent multiple CPU cases, the average response times of the systems which had the K(2,4,8,,,,) times better CPU were better than those of equivalent systems which had K(2,4,8,...) CPUs both in the distributed file system and in the shared memory system. And as the contention for the system resources of the file server in the distributed file system grew, the difference between the average response times of the better CPU cases and those of the equivalent multiple CPU cases became larger. This was also observed in the shared memory system.

When this study compared the file access performances of the faster disk I/O subsystem cases of section 6.5.1. with the file access performances of the equivalent multiple disk I/O subsystems cases of section 6.4, the average response times of the faster disk I/O subsystem cases were more sensitive to the number of clients and the number of local users than the average response times of the equivalent multiple disk I/O subsystems cases up to a certain number of clients and up to a certain number of local users in both system paradigms. When there was no contention for the system resources, the former was always smaller than the latter.

When this study compared the file access performance of the distributed file system which used the faster network and the better network interface unit in the file server with the file access performance of the distributed file system which used the equivalent number of multiple networks and the equivalent number of multiple network interface units, the average response time of the former was more sensitive to the number of clients than the average response time of the latter up to a certain number of clients. When there was no contention for the system resources, the average response time of the former was always smaller than the average response time of the latter.

This study compared the file access performance of the distributed file system when the system had multiple resources in the file server, when the system used a better file server and when the system used multiple file servers. The average response time of the distributed file system which had the better file server was most sensitive to the number of clients and to the average transaction size among the three cases. The better file server cases always showed the best average response time, the multiple resource cases show the next best average response time and the multiple file server cases showed the worst average response time, when there was no contention in the file server. The three cases became less sensitive to the number of clients as the degree of improvement and the number of multiple resources and the number of file servers increased regardless of the workload size. No notable performance effect due to the workload fluctuation was found. Generally, the six workloads showed similar patterns for the average response time.

This study investigated the effect on the file access performance when the file system mechanism was enhanced in the distributed file systems and in the shared memory systems comparatively. The average response time improved very little and the effect on the average response time decreased as the average transaction size of the workload increased and became trivial due to amortization in the two system paradigms.

This study investigated the effect on the file access performance when the RPC mechanism was improved in the distributed file systems. The average response time improved very little and the effect of the RPC parameter on the file access performance decreased as the number of clients increased and became trivial.

This study comparatively investigated the effect on the file access performance when the command interpretation mechanism was enhanced respectively in the distributed file system and in the shared memory system. The effect on the average response time decreased due to amortization. The effect on the average response time was a little larger in the shared memory system than in the distributed file system. The relative effect on the average response time became smaller when the workload of the larger average transaction size was supplied or more clients used the system even though the effect was significant when the 8Kbytes workload was used and there was very low contention in the system.

In chapter 7, this study comparatively investigated the effect on the file access performance when we used the 4 standalone caching mechanisms and the 5 combined caching mechanisms in the distributed file system and the 2 standalone caching mechanisms and the 1 combined caching mechanism in the shared memory system. It was observed that caching improved the file access performance significantly in most cases.

The third objective is to evaluate the effect of the changes in computing practice on the file access performance.
This study investigated the effect on the file access performance when the CPU of the baseline distributed file system and the CPU of the baseline shared memory system were replaced with better CPUs up to the theoretical limit and found that the overall improvement of the average response time of the distributed file system and that of the shared memory system were not significant since the contention in the CPU was low.

This study investigated the effect on the file access performance when only the disk I/O time was improved, when only the CPU time for the disk I/O was improved and when the two parameter values were improved up to the theoretical limit at the same time separately and comparatively in the two system paradigms. The overall improvement of the average response time in the distributed file system and in the shared memory system was significant.

This study investigated the effect on the file access performance when the network transmission speed was improved in section 6.8.1, when the performance of the network interface unit was improved in section 6.8.2, when the communication mechanism was enhanced in section 6.8.3 and when the three factors investigated in section 6.8.1, section 6.8.2 and section 6.8.3 were improved up to the theoretical limit at the same time respectively. In all cases, the overall improvement of the average response time in the distributed file system was significant since the communication facility was one of the major bottleneck points. With the infinitely faster network, the file access performance of the distributed file system was close to that of the shared memory system as expected.

This study investigated the effect on the file access performance effect when better systems were used in the distributed file system. The distributed file system where the all parameter values except the parameter of the network speed were improved to be 2 times better showed the best performance/cost in the 10Mbps local area network.

It was observed that the ratio of the average response time in the distributed file system, of which all parameters were improved to be X(2,4,8,...) time better including the network speed, to the average response time in the baseline distributed file system was equal to or larger than the degree of improvement, that is, X(2,4,8,..) up to a reasonable number of clients. This was also observed in the shared memory system.

The baseline distributed file system which consists of the Sun SPARCstation 10 workstations supports up to around 140 clients when the 8Kbytes workload is used, around 60 clients when the 50.7Kbytes workload is used and around 15 clients when the 316Kbytes workload is used. Therefore, the 316Kbytes workload or larger workloads seem to be too large to be accommodated in the system. Only when the disk I/O speed and the communication speed are improved at the same time, does the maximum number of supportable clients increase significantly. If the baseline system is improved to be 100 times better in all parameter values, then the average response time of the 1856Kbytes workload is 41msec when there is no contention for the system resources and 177msec at 400 clients and even 1000 clients do not saturate the system while in the baseline system the average response time of the 8Kbytes workload is 74msec when there is no contention for the system resources and 288msec at 100 clients and 150 clients saturate the system. The 316Kbytes workload and larger workloads are too big to be accommodated also in the baseline shared memory system.

When this study investigated the effect on the file access performance of concurrency during the disk I/O operation comparatively in the two system paradigms, it was observed that the file access performance showed slight improvement, that is, the average response time decreased slightly. When this study investigated the effect on the file access performance of concurrency during the network communication operation, it was observed that the file access performance showed slight deterioration, that is, the average response time increased slightly.

The fourth objective is to quantitatively evaluate the effect of the workload characteristics on the file access performance.

It was observed that the read operation was less sensitive to the contention for the system resources and in real environment, caching occurred more frequently in reading than in writing.

The best average response time was always found in the workload pattern with constant inter-arrival time distribution and constant transaction size distribution. The worst average response time was found in the workload pattern with log-normal inter-arrival time distribution and log-normal transaction size distribution most time. The workload pattern with Poisson arrival distribution and log-normal transaction size distribution showed the second or third worst average response time most times. It was observed that when steady workloads were used, the workload pattern with Poisson arrival distribution and constant transaction size distribution always showed worse average response time than the workload pattern with log-normal inter-arrival time distribution and constant transaction size distribution but when bursty workloads were used, the reverse was true.

At 100% remote file access, in other words, 0% local processing in a job, we see the average response time in the distributed file systems as it is. The average response time in the distributed file system becomes closer to the average response time of the equivalent local system as the percentage of local processing increases in a job. Therefore, the slowness of the remote file access is hidden to the users when the total response time is observed by the users.

It cannot be emphasized too much that I have to be careful in attempting to generalize the research results. Nonetheless, I believe that many research results obtained in this research are not only the properties of the particular systems but also have generality.

Below, I highlight the major contributions made in this dissertation.

1) This study developed the queueing network performance models for the two system paradigms. They are accurate, flexible in accommodating the changes in the target systems easily and can be simulated with reasonable effort.

2) This study presents the virtual server concept which enables us to easily construct precise and yet flexible performance models based on the queueing network theory.

3) This study presents an accurate and yet easy parameterization methodology which does not require any special performance measurement tool but uses only the standard UNIX facilities.

4) This study proposes a workload characterization methodology which consists of six steps.

5) Six realistic and representative file access workloads were obtained from the real measured data of the two system paradigms.

6) This study presents the standalone performance measurement methodology and the real world performance measurement methodology for the validation and uses the two methodologies to measure the real file access performance and validate the simulation results.

7) The file access performance of the two system paradigms was comparatively and quantitatively investigated and the various design topics were quantitatively discussed.

8) This study evaluates the file access performances of the various design alternatives in the two system paradigms comparatively so that the system designers can find the optimal solutions for their needs.

9) This study evaluates the effect on the file access performance of the major changes in computing practices such as computer communications speed growth, computing power growth, transaction size growth, etc. in the two system paradigms comparatively so that the system designers can interpret the changes quantitatively from the viewpoint of file access performance.

10) This study quantitatively finds out the theoretical limit of the file access performance from the various improvements in the two system paradigms comparatively so that the system designers can have better understanding of the two system paradigms.

8.2 Further Work

The research in this thesis mainly focuses on the homogeneous distributed file systems which consist of the same type and the same power of systems for the file server and the clients, though heterogeneous distributed file systems are also evaluated. This study finds that the confidence of the simulation values becomes worse in the heterogeneous distributed file system when the current parameter values are used. I think the reason is because some parameter values in the sending systems are assumed to be the same as those in the receiving systems. I think the confidence of the simulation results of the heterogeneous distributed file systems can be improved to the level of that of the homogeneous distributed file systems by getting rid of this assumption.

The workloads used in this thesis do not include voice data and image data explicitly. Jones and Hopper[JONES etal 93] describe the methodology used in the Pandora project to handle audio and video streams in a local area network based distributed environment. Audio and video data should be delivered in time. Real time synchronization is essential in order to maintain the integrity of the data being presented. Coulson and Blair[COULSON etal 94] address the real-time synchronization requirements of multi-media data in distributed environments. Anderson and Osawa[ANDERSON etal 92] present a file system called as the CMFS(Continuous Media File System) which supports real-time storage and retrieval of digital audio data and video data on disk. Further work is required in this area.

This study investigated the file access performances in the shared memory systems when only local users used the systems. It is also common that these systems are accessed via local area network using "telnet" or any other remote access facility. It will be interesting to compare the file access performance of the networked access case with the that of the case done in this thesis. I think the networked access case can be investigated with the expansion of the performance models developed in this research and the performance parameters obtained in this research.

As explained in chapter 7, this dissertation only deals with the cache hit which does not require any pre-operation or post-operation at all. If the same cached content is reused then there exists no overhead before and after cache hit except the cache consistency maintenance overhead and the cache access overhead. In other words, if the same cached data are accessed more than one time, then the first access is not dealt with in this thesis but all accesses from the second access to the last access are dealt with in this thesis. If the cached data are used just one time, then no system power is saved since the caching expense is paid sometime somewhere after all. In this cache hit, the data traffic amount and the system load are the same. Further work is required in order to represent this kind of cache hit using the performance models.

References

[ABATE etal 68] J.ABATE, H.DUBNER & S.WEINBERG, "Queueing analysis of the IBM 2314 disk storage facility," Journal of ACM, Vol.15, No.4, October 1968, pp.577-589.

[ABEYSUNDARA etal 91] B.W.ABEYSUNDARA & A.E.KAMAL, "High speed local area networks and their performance : A survey," ACM Computing Survey, Vol.23, No.2, June 1991, pp.222-264.

[ANANDA etal 93] A.L.ANANDA, B.H.TAY & E.K.KOH, "A survey of asynchronous remote procedure calls," Dept. of Information Systems and Computer Science, National University of Singapore, Singapore, 1993.

[ANDERSON 84] G.E.ANDERSON, "The Coordinated Use of Five Performance Evaluation Methodologies," Communication of ACM, Vol.27, No.2, Jan. 1984, pp.119-125.

[ANDERSON etal 92] D.P.ANDERSON & Y.OSAWA, "A file system for continuous media," ACM Transactions on Computer Systems, Vol.1, No.4, November 1992, pp.311-337.

[ARTIS 94] H.P.ARTIS, "DASD subsystems : evaluating the performance envelope," CMG transactions, Winter 1994, pp.3-12.

[AS 94] Harmen R.V. AS, "Media access techniques : the evolution toward terabit/s LANs and MANs," Computer Networks and ISDN Systems, 26, 1994, pp.603-656.

[AT&T 94] AT&T, "UNIX System V Release 4.2 Multiprocessor," System

Administration, Vol.2, May 3, 1994.

[BACON etal 87] J.M.BACON & K.G.HAMILTON, "Distributed Computing with RPC : The Cambridge approach," Technical Report No.117, Computing Laboratory, University of Cambridge, October 1987.

[BAKER etal 91] M.BAKER, J.HARTMAN, M.KUPFER and K.SHIRRIFF, "Measurements of a distributed file system," Presented in Proceedings of the 13th ACM Symposium on Operating System Principles, October 1991, Published in Operating Systems Review, Vol.25, No.5, pp.198-212.

[BARD 80] Y.BARD, "A model of shared DASD and multipathing," CACM, October 1980, Vol.23, No.10.

[BARNETT 86] C.C.BARNETT, "Simulation in Pascal with Micro PASSIM," Proceedings of 1986 Winter Simulation Conference, 1986, pp.151-155.

[BAYLOR etal 94] S.J.BAYLOR, C.BENVENISTE and Y.HSU, "Performance evaluation of a massively parallel i/o subsystem," Proceedings of the IPPS '94, April 1994, Mexico, pp.5-10.

[BELL 89] G.BELL, "The future of high performance computers in science and engineering," CACM, Vol.32, No.9, September 1989, pp.1091-1101.

[BELL 93] Gordon BELL, "Ultracomputers : a teraflop before its time," CACM, Vol.35, No.8, August 1993.

[BENETT 89] Geoff BENNETT, "Souped up token ring," Computer Systems Europe, Feb. 1989, pp.55-56

[BESTER etal 84] J.BESTER et al., "A dual priority MVA model for a large distributed system : LOCUS," Proceedings of Performance '84, 1984.

[BHUYAN etal 89] L.N.BHUYAN, D.GHOSAL, and Q.YANG, "Approximate analysis of single and multiple ring networks," IEEE transactions on computers, Vol.38, No.7, July 1989, pp.1027-1040.

[BRANDWAJN 81] A.BRANDWAJN, "Models of DASD subsystems : Basic model of reconnection," Performance Evaluation 1, 1981, pp.263-281.

[BRANDWAJN 83] A.BRANDWAJN, "Models of DASD subsystems with multiple access paths : a throughput-driven approach," IEEE transactions on computers, Vol.C-32, No.5, May 1983, pp.451-463.

[BROWNBRIDGE 82] D.R.BROWNBRIDGE, L.F.MARSHALL and B.RANDELL, "The Newcastle Connection or UNIXes of the World Unite!," Software-Practice and Experience, Vol.12, 1982, pp.1147-1162.

[BRYANT 80] R.M.BRYANT, "SIMPAS : A Simulation Language Based on Pascal," Proceedings of 1980 Winter Simulation Conference, 1980.

[BURR 86] W.E.BURR, "The FDDI Optical Data Link," IEEE Computer, Vol.25, May 1986, pp.18-23.

[BUX 89] W.BUX, "Token ring local area networks and their performance," Proceedings of IEEE, 77(2), Feb. 1989, pp.238-256.

[CALHOUN etal 87] J.CALHOUN & E.KORTRIGHT, "VSIM : A Graphics Based Model Engineering Tool," Proceedings of 6th Annual Modeling and Simulation Conference, San Diego, CA, U.S.A., January 1987.

[CALZAROSSA & FERRARI 86] M.CALZAROSSA and D.FERRARI, "A sensitivity study of the clustering approach to workload modeling," Performance Evaluation 6, 1986, pp.25-33, Elsevier Publishers B.V.

[CHANDRAS 90] R.G.CHANDRAS, "Distributed Message Passing Operating

Systems," ACM Operating System Review, Vol.24, No.1, Jan. 1990, pp.7-17.

[CHEN etal93] P.M.CHEN and A.PATTERSON, "Storage performance - metrics and benchmarks," Proceedings of the IEEE, Vol. 81, No.8, August 1993, pp.1151-1165.

[CHEN etal 94] P.M.CHEN, E.K.LEE, G.A.GIBSON, R.H.KATZ & D.A.PATERSON, "RAID : High performance, reliable secondary storage," ACM Computing Surveys, Vol.26, No.2, June 1994, pp.145-185.

[CHERITON etal 83] D.R.CHERITON & W.ZWAENEPOEL, "The distributed V Kernel and Its Performance for Diskless Workstations," Proceedings of the 9th Symposium on Operating System Principles, ACM, New York, 1983, pp.128-140.

[CHERITON 84] D.R.CHERITON, "The V Kernel : A Software Base for Distributed System IEEE Software," April 1984, pp.19-42.

[CHERITON 88] D.R.CHERITON, "The V Distributed System," Communications of the ACM, Vol.31, No.3, March 1988.

[CHRISTENSEN 79] G.S.CHRISTENSEN, "Links Between Computer-room Networks," Telecommunications, Vol.13, NO.2, February 1979, pp.47-50.

[CLARK 83] D.W.CLARK, "Cache performance in the VAX-11/780," ACM Transactions on Computer Systems, Vol.1, 1983, pp.24-37.

[CLARK etal 89] D.D.CLARK et al., "An analysis of TCP processing overhead," IEEE Communication, Vol.27, No.6, June 1989, pp.23-29.

[COLEMAN etal 93] S.S.COLEMAN & R.W.WATSON, "The emerging paradigm shift in storage system architectures," Proceedings of IEEE, Vol.81, No.4, April, 1993, pp.607-620.

[COOPER etal 90] E.COOPER et al., "Protocol Implementation on the Nectar Communication Processor," Proceedings of SIGCOMM 90 Symposium, Communication Architectures and Protocols, ACM Press, New York, 1990, pp.135-143.

[CORMIER etal 83] R.L.CORMIER, R.J.DUGAN & R.R.GUYETTE, "System/370 extended architecture : the channel subsystem," IBM Journal of Research and Development, 27, 3, May 1983, pp.206-218.

[COULORIS etal 88] G.F.COULOURIS & J.DOLLIMORE, "Distributed Systems : Concepts and Design," Addison-Wesley, 1988.

[COULSON etal 94] G.COULSON & G.S.BLAIR, "Meeting the real-time synchronization requirements of multimedia in open distributed processing," Distributed Systems Engineering, 1, 1994, pp.135-144.

[CRAFT 85] D.H.CRAFT, "Resource Management in a Distributed Computing System," Ph.D. Thesis, Computer Laboratory, University of Cambridge, March 1985.

[DAHL etal 66] O.J.DAHL & K.NYGAARD, "SIMULA - An ALGOL Based Simulation Language," Communication of ACM, Vol.9, 1966, pp.671-678.

[DAIGLE etal 90] J.N.DAIGLE, R.B.KUEHL and J.D.LANGFORD, "Queueing analysis of an optical disk jukebox based office system," IEEE Transactions on computers, Vol.39, No.6, June 1990.

[DATAPRO] DATAPRO Reports on International UNIX systems, 1995.

[DAVIDS etal 94] P.DAVIDS, T.MEUSER & O.SPANIOL, "FDDI : Status and perspectives," Computer Networks and ISDN Systems, 26, 1994, pp.657-677.

[FEITELSON etal 95] D.G.FEITELSON & P.F.CORBETT, "Parallel I/O subsystems in massively parallel supercomputers," IEEE Parallel & Distributed Technology, Fall

1995, pp.33-47.

[FERRARI etal 83] D.FERRARI et al., "Modeling file system organizations in a local area network environment," Report No. UCB/CSD, 83/142, Progress report No. 83.7, EECS, University of California, Berkeley 94720, October 1983.

[FLYNN 72] Michael J. FLYNN, "Some computer organizations and their effectiveness," IEEE Transactions on Computers, CT21, 1972, pp.948-960.

[GANGER etal 94] G.R.GANGER, B.L.WORTHINGTON, R.Y.HOU and Y.N.PATT, "Disk Arrays : high performance, high reliability storage subsystems," IEEE Computer, March 1994, pp.30-36.

[GARRISON 87] W.J.GARRISON, "NETWORK II.5 Tutorial," Proceedings of '87 Winter Simulation Conference, 1987, pp.247-257.

[GARZIA 90] Mario R. GARZIA, "Discrete Event Simulation Methodologies and formalisms," Simulation Digest, Vol.21, No.1, Summer 1990.

[GEIST etal 82] R.M.GEIST & K.S.TRIVEDI, "Optimal design of multilevel storage hierarchies," IEEE Transactions on computers, Vol.C-31, No.3, March 1982, pp.249-260.

[GOTLIEB & MacEWEN 73] C.C.GOTLIEB & G.H.MacEWEN, "Performance of moveable-head disk storage devices," Journal of ACM, 20, 4, Oct. 1973, pp.604-623.

[GOLDBERG etal 83] A.GOLDBERG, G.POPEK, and S.S.LEVENBERG, "A validated distributed system performance model," PERFORMANCE '83, A.K.Agrawala and S.K.Tripathi, Eds., Amsterdam, The Netherlands, North-Holland, 1983, pp.251-268.

[GORDON etal 86] R.F.GORDON, E.A.MACNAIR & P.D.WELCH, "Examples of Using the Research Queueing Package Modeling Environment (RESQME)," Proceedings of '86 Winter Simulation Conference, 1986, pp.494-503.

[GOYA etal 84] A.GOYAL and T.AGERWALA, "Performance analysis of future shared storage systems," IBM Journal of Research and Development, Vol.28, No.1, January 1984, pp.95-108.

[GUSELLA 90] R.GUSELLA, "A measurement study of diskless workstation traffic on an Ethernet," IEEE transactions on communications, Vol.38, No.9, September 1990, pp.1557-1568.

[HEIDELBERGER etal 81] P.HEIDELBERGER and P.A.W.LEWIS, "Regression adjusted estimates for regenerative simulations with graphics," CACM, Vol.24, 1981, pp.260-273.

[HEIDELBERGER etal 83B] P.HEIDELBERGER and P.D.WELCH, "Simulation run length control in the presence of an initial transient .....," Operational Research, Vol.31, 1983, pp.1109-1144.

[HEIDELBERGER etal 84] P.HEIDELBERGER and A.A.LAVENBERG, "Computer performance evaluation methodology," IEEE Transaction on Computer, C-33, 1984, pp.1195-1120.

[HEIDEMANN etal 93] R.HEIDEMANN, B.WEDDING, and G.VEITH, "10-GB/S Transmission and Beyond," Proceedings of the IEEE, Vol.81, No.11, November 1993, pp.1558-1567.

[HOUTEKAMER 85] G.E.HOUTEKAMER, "The local disk controller," ACM SIGMETRICS performance evaluation review, special issue, Vol.13, No.2, 1985. or Proceedings of the 1985 ACM SIGMETRICS conference on measurement and modeling of computer systems, August 1985.

[HOWARD etal 88] J.H.HOWARD et al., "Scale and performance in a distributed file system," ACM transactions on computer systems, Vol.6, No.1, Feb. 1988.

[HOWE etal 87] C.D.HOWE & B.MOXON, "How to program parallel computers," IEEE Spectrum, Vol.24, No.9, September 1987, pp.36-41.

[IGLEHART 76] D.L.IGLEHART, "Simulating stable stochastic systems : VI. Quantile estimation," Journal of ACM, Vol.23, 1976, pp.347-360.

[IGLEHART 78] D.L.IGLEHART, "The regenerative method for simulation analysis : Current Trends in Programming Methodology," Vol.III: Software Modeling, K.M.Chandy and R.T.Yeh Eds., Englewood Cliffs, NJ, Prentice-Hall, 1978, pp.52-71.

[JAIN 90] R.JAIN, "Performance analysis of FDDI token ring networks : effect of parameters and guidelines for setting TTRT," SIGCOMM '90 Symposium : Communication Architectures and protocols, September 24-27, 1990, Computer Communication Review, Vol.20, No.4, September 1990.

[JOHNSON 88] E.E.JOHNSON, "Completing an MIMD Multiprocessor Taxonomy," ACM Computer Architecture News, Vol.16, No.3, June 1988, pp.44-47.

[JONES etal 93] A. JONES & A.HOPPER, "Handling audio and video streams in a distributed environment," ACM Proceedings of SIGOPS '93, NC, U.S.A., Dec. 1993.

[JOSHI 86] S.P.JOSHI, "High Performance Networks - A Focus on the Fiber Distributed Data Interface (FDDI) Standard," IEEE Micro, Vol.6, June 1986, pp.8-14.

[KAREDLA etal. 94] R.KAREDLA, J.S.LOVE & B.G.WHERRY, "Caching strategies to improve disk system performance," IEEE Computer, March 1994, pp.38-46.

[KARP 89] A.KARP, "Programming for parallelism," IEEE Computer, Vol.20, No.5, May 1987, pp.43-57.

[KIM 86] M.Y.KIM, "Synchronized disk interleaving," IEEE Transactions on computers, Vol.C-35, No.11, November 1986, pp.978-988.

[KIVIAT etal 73] P.J.KIVIAT, R.VILLANUEVA & H.M.MARKOWITZ, "SIMSCRIPT II.5 Programming Language," CACI, Los Angeles, CA, U.S.A., 1973.

[KLEIJNEN 74] J.P.C. KLEIJNEN, "Statistical Techniques in Simulation : Part I," New York, Marcel Dekker, 1974.

[KLEIJNEN 75] J.P.C. KLEIJNEN, "Statistical Techniques in Simulation, Part II," New York, Marcel Dekker, 1975.

[KRONENBERG etal 86] Nancy P. KRONENBERG, Henry M. LEVY, and William D. STRECKER, "VAXclusters: A Closely-Coupled Distributed System," ACM Transactions on Computer Systems, Vol.4, No.2, May 1986, pp.130-146.

[KUROSE etal 86] J.F.KUROSE & K.J.GORDON, "A Graphics-Oriented Modeler's Workstation Environment for the RESearch Queueing Package (RESQ)," Proceedings of '86 Fall Joint Computer Conference of ACM/IEEE, Dallas, TX, U.S.A., November 1986, pp.719-728.

[LANG etal 90] Lawrence J. LANG and James WATSON, "Connecting Remote FDDI Installations with Single-Mode Fiber, Dedicated Lines, or SMDS," ACM Computer Communication Review, Vol.20, No.3, July 1990, pp.72-82.

[LAVENBERG etal 77] S.S.LAVENBERG and C.H.SAUER, "Sequential stopping rules for the regenerative method of simulation," IBM Journal of Research and Development, Vol.21, 1977, pp.545-558.

[LAVENBERG 83] S.S.LAVENBERG Ed., "Computer Performance Modeling Handbook," Academic Press, New York, 1983.

[LAW etal 82] A.M.LAW and W.D.KELTON, "Simulation Modeling and Analysis," New York: McGraw-Hill, 1982.

[LAW etal 83] S.S.LAM and Y.L.LIEN, "A tree convolution algorithm for the solution of queueing networks," CACM, 26, 1983, pp.203-215.

[LAZOWSKA etal 84]] E.D.LAZOWSKA, J.ZAHORJAN, G.S.GRAHAM and K.C. SEVCIK, "Quantitative System Performance--Computer System Analysis Using Queueing Network Models," Englewood Cliffs, NJ, Prentice-Hall, 1984.

[LAZOWSKA etal 86] E.D.LAZOWSA, J.ZAHORJAN, D.R.CHERITON & W.ZWAENEPOEL, "File access performance of diskless workstations," ACM Trans. on Computer systems, Vol.4, No.3, August, 1986.

[LECUYER etal 87] P.L'ECUYER & N.GIROUX, "A Process Oriented Simulation Package Based on Modula-2," Proceedings of '87 Winter Simulation Conference, 1987, pp.165-174.

[LEE etal 93] Young Woo LEE, Alex S. WIGHT and Dan H. LEE, "Performance modeling and simulation of the 1993 Daejeon International EXPO network," Proceedings of UK Simulation Society '93 Conference, Keswick, U.K., September 1993.

[LEE etal 94] Young Woo LEE, Alex S. WIGHT and Yeong Wook CHO, "Workload characterization of Cray supercomputer systems running UNICOS for the optimal design of NQS configuration in a site," Proceedings of the 33th International Cray User Group Conference, San Diego, CA, U.S.A., March 14-18, 1994.

[LEE etal 95] Young Woo LEE, Alex S. WIGHT, Sung W. CHOI and Dan H. LEE, "Simulation of Compound Local Area Networks for a Large Scale Client Server Type Multi-media Computer System," International Pritsker User Group Conference, June 7-9, 1995, Indianapolis, U.S.A..

[LEFFLER etal 84] S.LEFFLER & M.KARELS & M.K.McKUSICK, "Measuring and improving the performance of 4.2 BSD," In Proceedings of the USENIX 1984 Summer Conference, Salt Lake City, Utah, June 1984, USENIX Association, Berkeley, CA, pp.237-252.

[LELANN 81] G.LELANN, "Motivation, objectives and characterization of distributed Systems In 'Distributed Systems : Architecture and Implementation'," edited by B.W.LAMPSON, M.PAUL & H.J.SIEGERT, Springer-Verlag, 1981.

[LEVY etal 90] E.LEVY & A.SILBERSCHATZ, "Distributed file systems : Concepts and Examples," ACM Computing Surveys, Vol.22, No.4, December 1990, pp.321-374.

[LILJA 93] David J. LILJA, "Cache coherence in large-scale shared memory multiprocessors : issues and comparisons," ACM Computing Surveys, Vol.25, No.3, September 1993, pp.303-338.

[MACDOUGALL 87] M.H.MACDOUGALL, "Simulating Computer Systems : Techniques and Tools," The MIT Press, 1987.

[MAJOR 81] J.B.MAJOR, "Processor, I/O path, and DASD configuration capacity," IBM System Journal, Vol.20, No.1, 1981.

[MALLOY etal 86] B.MALLOY & M.L.SOFFA, "Simcal : The Merger of SIMULA and Pascal," Proceedings of '86 Winter Simulation Conference, 1986, pp.397-402.

[MARATHE etal 81] M.MARATHE and S.KUMAR, "Analytic models for an Ethernet-like LAN link," ACM SIGMETRICS Conference Proceedings, September 1981.

[MECHANIC 66] H.MECHANIC and W.McKAY, "Confidence intervals for averages of dependent data in simulations II," IBM Corp., Yorktown Heights. NY. Tech. Rep. ASDD17-202, 1966.

[MELDE etal 88] J.E.MELDE & P.G.GAGE, "Ada Simulation Technology - Methods and Metrics," Simulation, Vol.51, No.2, August 1988, pp.57-69.

[MERLE etal 78] D.MERLE, D.POTIER and M.VERAN, "A tool for computer system performance analysis," D. Ferrari, ed., Performance of Computer Installations, North-Holland, Amsterdam, 1978, pp.195-213.

[METCALFE etal 76] Robert M. METCALFE and David R. BOGGS, "Ethernet : Distributed Packet Switching for Local Computer Networks," Communications of the ACM, Vol.19, No.7, July 1976, pp.395-404.

[MORRIS etal 86] J.H.MORRIS et al., "Andrew : A Distributed Personal Computing Environment," Communications of the ACM, Vol.29, No.3, March 1986.

[MUELLER 84] B.MUELLER, "NUMAS : A Tool for the Numerical Modeling of Computer Systems," Proceedings of International Conference on Modeling Techniques and Tools for Performance Analysis, Paris, France, May 1984.

[MULLENDER 89] S.J.MULLENDER, "Amoeba - High Performance Distributed Computing," EUUG Spring '89, Brussels, April 3-7, 1989, pp.17-26.

[MULLENDER etal 90] S.J.MULLENDER, G.V.ROSSUM, A.S.TANENBAUM, R.V.RENESSE & H.V.STAVEREN, "Amoeba : A Distributed Operating System for the 1990s," IEEE Computer, May 1990.

[NEEDHAM etal 82] R.M.NEEDHAM & A.J.HERBERT, "The Cambridge Distributed Computing Systems," Addison-Wesley, Reading, MA 1982.

[OKEEFE 86B] R.M.O'KEEFE, "Simulation and Expert Systems - A taxonomy and Some Examples Simulation," Vol.46, No.1, January 1986, pp.10-16.

[OKEEFE etal 86] R.M.O'KEEFE & R.M.DAVIES, "Discrete Visual Simulation with Pascal_SIM," Proceedings of '86 Winter Simulation Conference, 1986, pp.517-521.

[OUSTERHOUT etal 85] J.K.OUSTERHOUT et al., "A trace driven analysis of the UNIX 4.2 BSD system," Proceedings of the 10th ACM Symposium on Operating System Principles, Washington, Dec 1-4, 1985, ACM, New York, pp.15-24.

[PADEGS 83] PADEGS, "System/370 Extended Architecture : design considerations," IBM Journal of Research and Development, Vol.27, No.3, May 1983, pp.198-205.

[PAWS 83] "PAWS/A User Guide," Information Research Associates, Austin, TX, 1983.

[PEGDEN 86] C.D.PEGDEN, "Introduction to SIMAN," Proceedings of '86 Winter Simulation Conference, 1986, pp.95-103.

[PERROS etal 85] H.G.PERROS & D.MIRCHANDANI, "An analytic model of a file server for bulk file transfer," ACM SIGMETRICS Performance Evaluation Review, Vol.1.3, No.3&4, November 1985, pp.14-22.

[PERRY etal 89] T.S.PERRY & G.ZORPETTE, "Supercomputer experts predict expansive growth," IEEE Spectrum, Vol.26, No.2, February 1989, pp.26-33.

[POOLEY 86] R.J.POOLEY, "An introduction to programming in SIMULA," Blackwell Scientific, Oxford, 1986.

[POPEK etal 85] G.J.POPEK & B.J.WALKER, "The LOCUS Distributed System Architecture," The MIT Press, 1985.

[PRITSKER 74] A.A.B.PRITSKER, "The GASP IV Simulation Language," John Wiley and Sons, New York, 1974.

[PRITSKER 84] A.A.B.PRITSKER, "Introduction to simulation and SLAM II," Second edition by Alan B. Pritsker, John Wiley and Sons, 1984.

[RAMAKRISHNAN etal 82] K.G.RAMAKRISHNAN and D.MITRA, "An overview of PANACEA, A software package for analyzing Markovian queueing networks," Bell Systems Technical Journal, Vol.61, 1982, pp.2849-2872.

[RAMAKRISHNAN etal 86] K.K.RAMAKRISHNAN and J.S.EMER, "A model of file server performance for a heterogeneous distributed system," SIGCOMM '86 Symposium : Communications architectures and protocols, August, 1986, pp.338-347.

[RAMAKRISHNAN etal 89] K.K.RAMAKRISHNAN & J.S.EMER, "Performance analysis of mass storage service alternatives for distributed systems," IEEE Transactions on Software Engineering, Vol.15, No.2, Feb. 1989.

[RHUEMMLER etal 93] C.RHUEMMLER and J.WILKES, "UNIX disk access patterns," Proceedings of 1993 Winter USENIX, January 25-29, 1993, San Diego, U.S.A..

[RHUEMMLER etal 94] C.RHUEMMLER and J.WILKES, "An introduction to disk drive modeling," IEEE Computer, March 1994, pp.17-28.

[ROSARIO etal 94] J.M.ROSARIO and A.N.CHOUDHARY, "High-performance I/O for massively parallel computers : problems and prospects," IEEE Computer, March 1994, pp.59-68.

[ROSS 86] F.E.ROSS, "FDDI - A Tutorial," IEEE Communication, Vol.24, May 1986, pp.10-15.

[ROSS etal 90] Floyd E. ROSS, James R. HAMSTRA & Robert L. FINK, "FDDI - A LAN Among MANs," ACM Computer Communication Review, Vol.20, No.3, July 1990, pp.16-31.

[SANDBERG etal 85] R.SANDBERG et al., "Design and Implementation of the SUN Network File System," Proceedings of Summer Usenix Conference, Portland, 1985, pp.119-130.

[SATYANARAYANAN 90A] M.SATYANARAYANAN, "A Survey of Distributed File Systems," In "Annual Review of Computer Science" edited by J.F.TRAUB et al. Annual Reviews Inc., Palo Alto, CA, U.S.A., 1990, pp.73-104.

[SATYANARAYANAN 90B] M.SATYANARAYANAN et al., "Coda : A Highly Available File System for a Distributed Workstation Environment," IEEE transactions on Computers, Vol.39, No.4, April 1990, pp.447-459.

[SATYANARAYANAN 90C] M.SATYANARAYANAN, "Scalable, Secure, and Highly Available Distributed File Access," IEEE Computer, May 1990.

[SAUER etal 83] C.H.SAUER & E.A.MAcNAIR, "Simulation of Computer Communication Systems," Englewood Cliffs, NJ, Prentice-Hall, 1983.

[SAUER etal 87] C.H.SAUER et al., "RT PC Distributed Services Overview," ACM Operating Systems Review, Vol.21, No.3, July 1987, pp.18-29.

[SCHRIBER 74] J.J.SCHRIBER, "Simulation Using GPSS," John wiley and Sons, New York, 1974.

[SCHRUBEN 81] L.W.SCHRUBEN, "Control of initialization bias in multivariate simulation response," Communication of ACM, Vol.24, 1981, pp.246-252.

[SCHRUBEN 82] L.W.SCHRUBEN, "Detecting initialization bias in simulation output," Operational Techniques, Vol.30, 1982, pp.569-590.

[SEILA 88] A.F.SEILA, "SIMTOOLS : A Software Tool Kit for Discrete Event Simulation in Pascal," Simulation, Vol.50, No.3, March 1988, pp.93-99.

[SHANTIKUMAR etal 83] J.G.SHANTIKUMAR and R.G.SARGENT, "A unifying view of hybrid simulation/analytic models and modeling," Operational Research, Vol.31, 1983, pp.1030-1052.

[SHERMAN etal 72] S.W.SHERMAN, F.BASKETT and J.C.BROWNE, "Trace-driven modeling and analysis of CPU scheduling in a multiprogramming system," Communication of ACM, Vol.15, 1972, pp.1063-1069.

[SHOCH etal 80] J.F.SHOCH & J.A.HUPP, "Measured performance of an Ethernet local network," CACM, Vol.23, No.12, Dec. 1980, pp.711-721.

[SHROEDER etal 90] M.SCHROEDER and M.BURROWS, "Performance of Firefly RPC," ACM Trans. on Computer Systems, Vol.8, No.1, Feb. 1990, pp.1-17.

[SINCLAIR etal 86] J.B.SINCLAIR & S.MADALA, "A Graphical Interface for Specification of Extended Queueing Network Models," Proceedings of '86 Fall Joint Computer Conference of ACM/IEEE, Dallas, TX, U.S.A. November 1986, pp.709-718.

[SKINNER etal 69] C.E.SKINNER & J.R.ASHER, "Effects of storage connection on system performance," IBM System Journal, No.4, 1969, pp.319-333.

[SMITH 81] Alan Jay SMITH, "Long term file migration : development and evaluation of algorithms," CACM, Vol.24, No.8, August 1981, pp.521-532.

[SMITH 82] A.J.SMITH, "Cache memories," ACM Computing surveys, Vol.14, No.3, September 1982, pp.473-530.

[SMITH 85] A.J.SMITH, "Disk-cache miss-ratio analysis and design considerations," ACM Transactions on Computer Systems, Vol.3, No.3, August 1985, pp.161-203.

[STALLINGS 84] William STALLINGS, "Local Area Networks," ACM Computing Surveys, Vol. 16, No.1, March 1984, pp.3-41.

[STEENKISTE 94] Peter A. STEENKISTE, "A systematic approach to host interface design for high speed networks," IEEE Computer, March 1994, pp.47-57.

[STEWART 79] H.M.STEWART, "Performance analysis of complex communications systems," IBM System Journal, Vol.18, 1979, pp.356-373.

[SVOBODOVA 84] Liba SVOBODOVA, "File Servers for Network-Based Distributed Systems," ACM Computing Surveys, Vol.16, No.4, Dec. 1984, pp.353-398.

[TANENBAUM etal 85] A.S.TANENBAUM & R.V.RENESSE, "Distributed Operating Systems," ACM Computing Surveys, Vol.17, December 1985, pp.419-470.

[TANENBAUM etal 88] A.S.TANENBAUM, R.V.RENESSE & H.V.STAVERN, "Performance of The World's Fastest Distributed Operating System," ACM Operating Systems Review, Vol.22, No.4, October 1988, pp.25-34.

[TANENBAUM etal 89] A.S.TANENBAUM, R.V.RENESSE & H.V.STAVERN, "The Performance of The Amoeba Distributed Operating System," Software Practice and Experience, Vol.19, No.3, March 1989.

[TAY etal 90] B.H.TAY & A.L.ANANDA, "A Survey of Remote Procedure Calls," ACM Operating System Review, Vol.24, No.3, July 1990, pp.68-79.

[TEOREY etal 72] TEOREY and PINKERTON, "A comparative analysis of disk scheduling policies," CACM, Vol.15, No.3, March 1972, pp.177-184.

[UYENSO etal 80] D.UYENSO & W.VAESSEN, "PASSIM : A Discrete Event Simulation Package for Pascal," Simulation, Vol.35, 1980, pp.183-190.

[WATERS 75] S.J.WATERS, "Estimating magnetic disk seeks," Computing Journal, 18, 1975, pp.12-19.

[WILHELM 77] N.WILHELM, "A general model for the performance of disk systems," Journal of ACM, January 1977, Vol.24, No.1, pp.14-31.

[WOOD etal 93] C.WOOD and P.HODGES, "DASD trends : cost, performance, and form factor," Proceedings of the IEEE, Vol.81, No.4, April 1993, pp.573-585.

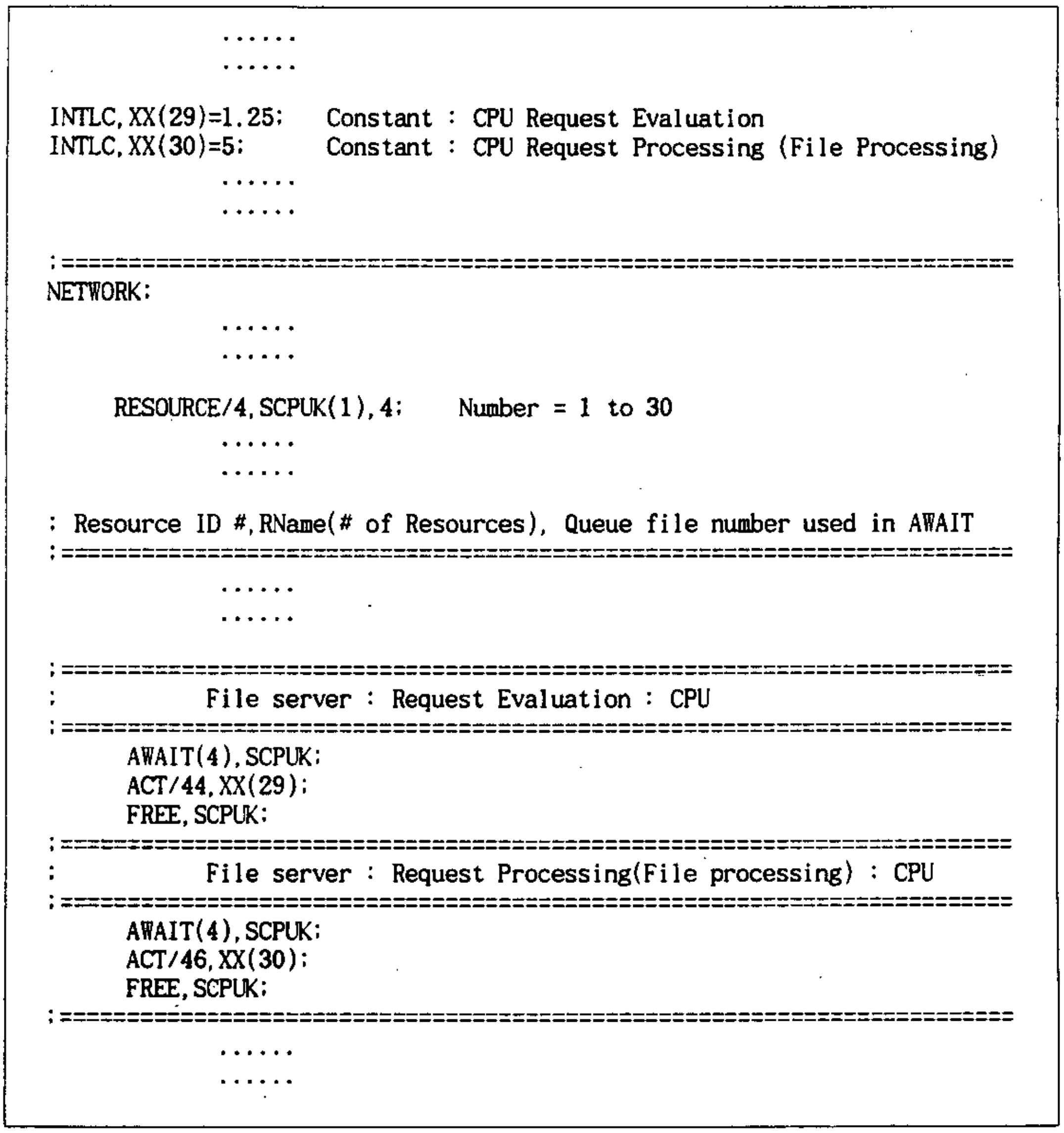
Appendix A

The Implementation of the Virtual Server Concept

Let us look at the two virtual CPU servers - the request evaluation virtual CPU server and the request processing(file processing) virtual CPU server - of the file server in figure 3.2.6 as a sample case for explanation. Figure A.1 shows part of a SLAM-II program which implements those two virtual servers.

In figure A.1, a real CPU server is represented as a resource. The identification number of this resource is "4" : "RESOURCE/4". "SCPUK(1), 4" means it has one resource named as SCPUK and a queue with identification number "4" is assigned for the resource.

If the request evaluation virtual CPU server is called for service, then that virtual server calls the SCPUK resource for acquisition. That virtual CPU server should compete with other virtual servers, for example, the request processing virtual CPU server for acquisition of the SCPUK resource. After using the SCPUK resource during the activity period, that virtual server releases(frees) the SCPUK resource. In figure A.1, the queue "4" is assigned to the resource SCPUK with FCFS(First Come First Served) queueing discipline. However, the two virtual queues for the two virtual CPU servers can be represented in many ways according to the mechanism of the real system. For example, it can be represented as multiple queues with various queueing disciplines such as Round Robin, etc..



```
                 ......
                 ......

INTLC,XX(29)=1.25;   Constant : CPU Request Evaluation
INTLC,XX(30)=5;      Constant : CPU Request Processing (File Processing)
                 ......
                 ......

;=================================================================
NETWORK:
                 ......
                 ......

     RESOURCE/4,SCPUK(1),4;     Number = 1 to 30
                 ......
                 ......

; Resource ID #,RName(# of Resources), Queue file number used in AWAIT
;=================================================================
                 ......
                 ......

;=================================================================
;           File server : Request Evaluation : CPU
;=================================================================
      AWAIT(4),SCPUK;
      ACT/44,XX(29);
      FREE,SCPUK;
;=================================================================
;           File server : Request Processing(File processing) : CPU
;=================================================================
      AWAIT(4),SCPUK;
      ACT/46,XX(30);
      FREE,SCPUK;
;=================================================================
                 ......
                 ......
```

Figure A.1 : A SLAM-II program.

Appendix B

The Effect of the Paradigms

B.1 The Effect of Workload

B.1.1 Read and Write

Figure B.1.1.1 to figure B.1.1.6 show the average response time of the read and the average response time of the write in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure B.1.1.7 to figure B.1.1.12 show the average response time of the read and the average response time of the write in the distributed file system which consists of the Sun SPARCstation 470 workstations.

Figure B.1.1.13 to figure B.1.1.18 show the average response time of the read and the average response time of the write in the distributed file system which consists of the Sun 3/60 workstations.

B.1.2 Workload Pattern

Figure B.1.2.1 shows the average response times of the six workload patterns when the 8kbytes workload is used as the number of clients increases in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure B.1.2.2 to figure B.1.2.6 show the average response time of the 47kbytes workload, the 50.7kbytes workload, the 316kbyte(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure B.1.2.7 to figure B.1.2.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbyte(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system of the Sun SPARCstation 10 workstation.

Figure B.1.2.13 to figure B.1.2.18 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbyte(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system which consists of the Sun SPARCstation 470 workstations.

Figure B.1.2.19 to figure B.1.2.24 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbyte(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system of the Sun SPARCstation 470 workstation.

Figure B.1.2.25 to figure B.1.2.30 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbyte(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system which consists of the Sun 3/60 workstations.

Figure B.1.2.31 to figure B.1.2.36 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbyte(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the Sun 3/60 workstation.

The effect of the workload pattern on the average response time is analyzed as follow. First, see the figures obtained when this study used the 8kbytes workload in the distributed file systems. The workload pattern with the log-normal inter-arrival time distribution and the log-normal transaction size distribution shows the worst average response time and the workload pattern with the Poisson arrival distribution and the log-normal transaction size distribution shows the next worst average response time. In the distributed file system which consists of the Sun SPARCstation 10 workstations, the workload pattern with the constant inter-arrival time distribution and the log-normal transaction size distribution shows the third worst average response time and the workload pattern with the log-normal inter-arrival time distribution and the constant transaction size distribution shows the fourth worst average response time. In the distributed file system which consists of the Sun SPARCstation 470 workstations and in the distributed file system which consists of the Sun 3/60 workstations the order of the third worst and the fourth worst is reversed. The second best average response time is observed in the workload pattern with the Poisson arrival distribution and the constant transaction size distribution. The workload pattern with the constant inter-arrival time distribution and the constant transaction size distribution always shows best average response time in both system paradigms. I find an interesting fact that the workload pattern with the constant inter-arrival time distribution and the constant transaction size distribution shows constant average response time as the number of clients or the number of local terminals increases up to near the saturation point. It is notable that even when the inter-arrival time is smaller than the average response time, the average response time seldom increases. This is commonly observed in the six workloads, in all three systems and in the two system paradigms. For example, when the 8kbyte workload is used, the average response time is 73.33msec all the time and when the 316kbytes workload is used, the average response time is 740.7msec all the time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Second, see the figures obtained when the 8kbytes workload was used in the shared memory systems. In the three systems such as the Sun SPARCstation 10 workstation, the Sun SPARCstation 470 workstation and the Sun 3/60 workstation, the order of the average response time is the same whatever workload is used. The order from the worst average response time to the best average response time is the following.

1) The workload pattern which has the log-normal inter-arrival time distribution and the log-normal transaction size distribution.
2) The workload pattern which has the log-normal inter-arrival time distribution and the constant transaction size distribution.
3) The workload pattern which has the Poisson arrival distribution and the log-normal transaction size distribution.
4) The workload pattern which has the Poisson arrival distribution and the constant transaction size distribution.
5) The workload pattern which has the constant inter-arrival time distribution and the log-normal transaction size distribution.
6) The workload pattern which has the constant inter-arrival time distribution and the constant transaction size distribution.

When the 8kbytes workload is used in the two different paradigms, it is commonly observed that the workload pattern with the log-normal inter-arrival time distribution and the log-normal transaction size distribution shows the worst average response time and the workload pattern with the constant inter-arrival time distribution and the constant transaction size distribution shows the best average response time. The workload pattern with the Poisson arrival distribution and the log-normal transaction size distribution, which is taken as the baseline workload pattern in this study, shows the second worst average response time in the distributed file systems and the third worst average response time in the shared memory systems.

Third, see the figures obtained when the 47kbytes workload was used in the distributed file systems. In the three distributed file systems, the order of average response time is the same except for the order of the worst average response time and the second worst average response time. The order from the worst average response time to the best average response time is the following.

1) The workload pattern which has the log-normal inter-arrival time distribution and the log-normal transaction size distribution, or, the workload pattern which has the Poisson arrival distribution and the log-normal transaction size distribution.
3) The workload pattern which has the constant inter-arrival time distribution and the log-normal transaction size distribution.
4) The workload pattern which has the Poisson arrival distribution and the constant transaction size distribution.
5) The workload pattern which has the log-normal inter-arrival time distribution and the constant transaction size distribution.
6) The workload pattern which has the constant inter-arrival time distribution and the constant transaction size distribution.

Fourth, see the figures obtained when the 47kbytes workload was used in the shared memory systems. In the three systems, the order of the average response time is the same. The order from the worst average response time to the best average response time is also the same as the order in the distributed file systems. The difference between the worst average response time and the second worst average response time is very small. When the 8kbytes workload is used, the workload pattern with the Poisson arrival distribution and the constant transaction size distribution shows worse average response time than the workload pattern with the log-normal inter-arrival time distribution and the constant transaction size distribution but when the 47kbytes workload is used, the former shows better average response time than the latter. This is also observed in the workload pair of the 50kbytes workload and the 316kbytes(B) workload and the workload pair of the 316kbytes workload and the 1856kbytes workload. This means that when steady workloads are used, the workload pattern with the Poisson arrival distribution and the constant transaction size distribution shows worse average response time than the workload pattern with the log-normal inter-arrival time distribution and the constant transaction size distribution but when bursty workloads are used, the reverse is true.

Fifth, see the figures obtained when the 50.7kbytes workload was used in the distributed file systems. In the three systems the order of the average response time is same. The order from the worst average response time to the best average response time is the following.

1) The workload pattern which has the log-normal inter-arrival time distribution and the log-normal transaction size distribution.
2) The workload pattern which has the Poisson arrival distribution and the log-normal transaction size distribution.
3) The workload pattern which has the constant inter-arrival time distribution and the log-normal transaction size distribution.
4) The workload pattern which has the log-normal inter-arrival time distribution and the constant transaction size distribution.
5) The workload pattern which has the Poisson arrival distribution and the constant transaction size distribution.
6) The workload pattern which has the constant inter-arrival time distribution and the constant transaction size distribution.

Sixth, see the figures when the 50.7kbytes workload was used in the shared memory systems. In the three systems the workload pattern with the log-normal inter-arrival time distribution and the log-normal transaction size distribution shows the worst average response time and the workload pattern with the constant inter-arrival time distribution and the constant transaction size distribution shows the best average response time like the previous cases. The workload pattern with the Poisson arrival distribution and the log-normal transaction size distribution shows the second worst average response time in the Sun SPARCstation 10 workstation and the third worst average response time in the Sun SPARCstation 470 workstation and the Sun 3/60 workstation. The average response times of the six workload patterns in the Sun SPARCstation 470 workstations and the Sun 3/60 workstations are in the same order.

Seventh, see the figures obtained when the 316kbytes(B) workload, a bursty workload, was used in the distributed file systems. In the three systems the order of the average response time is same. The order from the worst average response time to the best average response time is the following.

1) The workload pattern which has the log-normal inter-arrival time distribution and the log-normal transaction size distribution.
2) The workload pattern which has the Poisson arrival distribution and the log-normal transaction size distribution.
3) The workload pattern which has the constant inter-arrival time distribution and the log-normal transaction size distribution.
4) The workload pattern which has the Poisson arrival distribution and the constant transaction size distribution.
5) The workload pattern which has the log-normal inter-arrival time distribution and the constant transaction size distribution.
6) The workload pattern which has the constant inter-arrival time distribution and the constant transaction size distribution.

The order is the same as the order when the 50.7kbytes workload is used in the distributed file systems except that the fourth and the fifth are reversed which is commonly observed when the bursty workloads are used.

Eighth, see the figures obtained when the 316kbytes(B) workload, a bursty workload, was used in the shared memory systems. In the three systems the average response time is in the same order. The order is also the same as the order which was observed in the distributed file systems.

Ninth, see the figures when the 316kbytes workload, a steady workload, was used in the distributed file systems. In the three systems the average response time is in the same order. The average response times is in the the same order as the order when the 316kbytes workload, a bursty workload, is used except that the fourth and the fifth are reversed which is commonly observed when steady workloads are used.

Tenth, see the figures obtained when the 316kbytes workload, a steady workload, was used in the shared memory systems. In the three systems the average response time is in the same order. The order of the average response times is the same as the order when the 316kbytes workload is used in the distributed file systems.

Eleventh, see the figures when the 1856kbytes workload, a bursty workload, was used in the distributed file systems. The best average response time is observed in the workload pattern with the constant inter-arrival time distribution and the constant transaction size distribution. The second best average response time is observed in the workload pattern with the log-normal inter-arrival time distribution and the constant transaction size distribution. The third best average response time is observed in the workload pattern with the Poisson arrival distribution and the constant transaction size distribution. This order is the same in the three systems.

Twelfth, see the figures when the 1856kbytes workload, a steady workload, was used in the shared memory systems. The order of the best three in terms of the average response time is the same in the three systems and also the same as the order in the distributed file systems.

B.2 The Two System Paradigms

Figure B.2.1 to figure B.2.6 shows the average response time as the number of clients or the number of local users increases when this study uses the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and 1856kbytes workload respectively in the environments which consist of the SUN SPARCstation 10 workstations.

Figure B.2.7 to figure B.2.12 shows the average response time as the number of clients or the number of local users increases when this study uses the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and 1856kbytes workload respectively in the environments which consist of the SUN SPARCstation 470 workstations.

Figure B.2.13 to figure B.2.18 show the average response time as the number of clients or the number of local users increases when this study uses the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and 1856kbytes workload respectively in the environments which consist of the SUN 3/60 workstations.

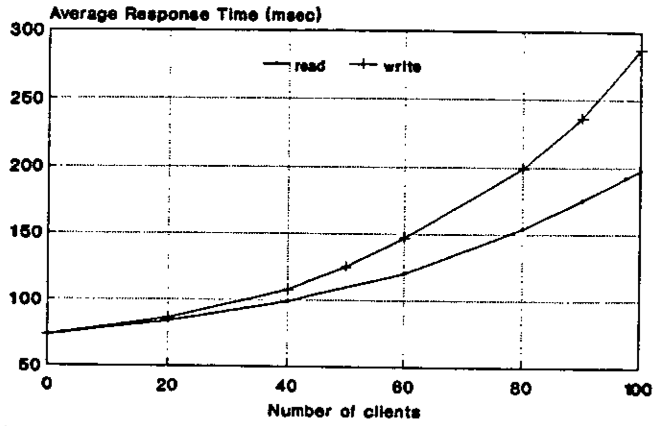


Figure B.1.1.1 : 8Kbytes

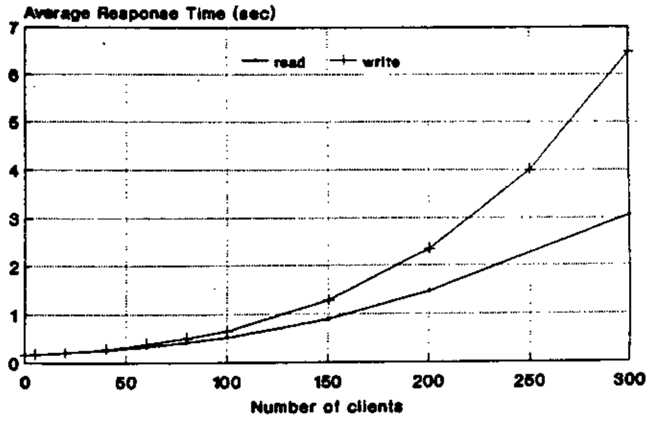


Figure B.1.1.2 : 47Kbytes

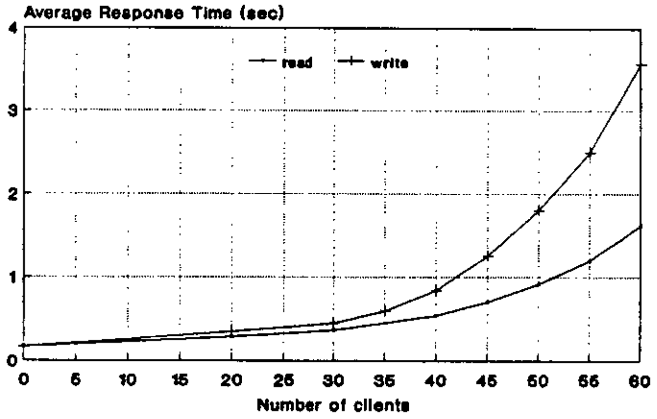


Figure B.1.1.3 : 50.7Kbytes

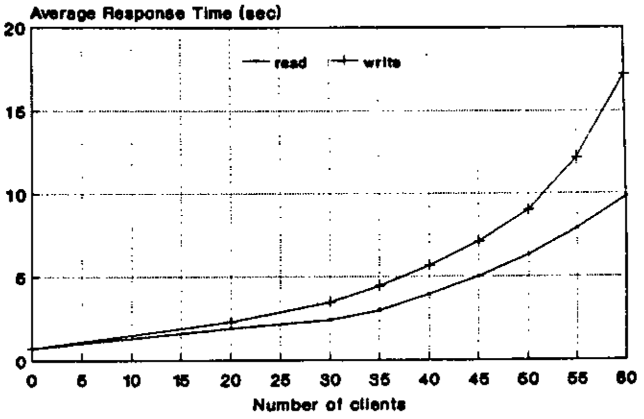


Figure B.1.1.4 : 316Kbytes(B)

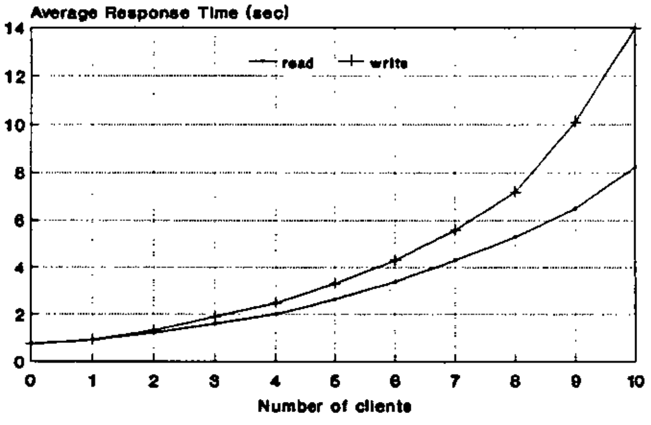


Figure B.1.1.5 : 316Kbytes

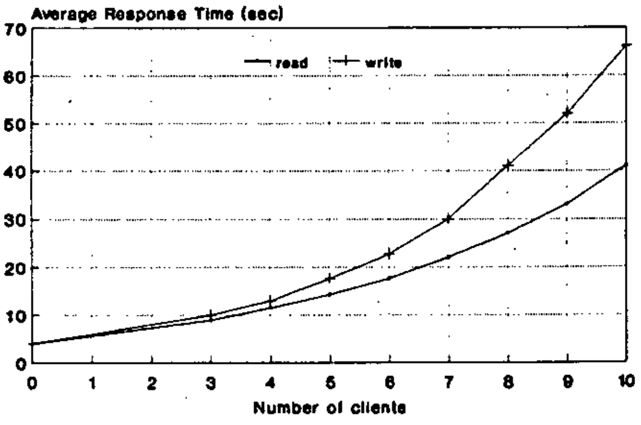


Figure B.1.1.6 : 1856Kbytes

The average response time of the read vs. the average response time of the write in the distributed file system which consists of the Sun SPARCstation 10 workstations.

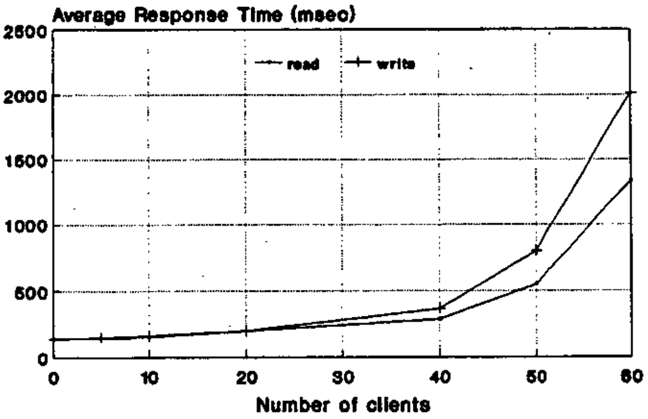


Figure B.1.1.7 : 8Kbytes

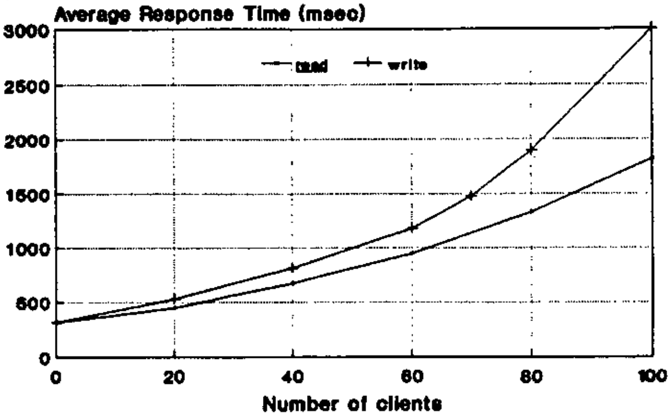


Figure B.1.1.8 : 47Kbytes

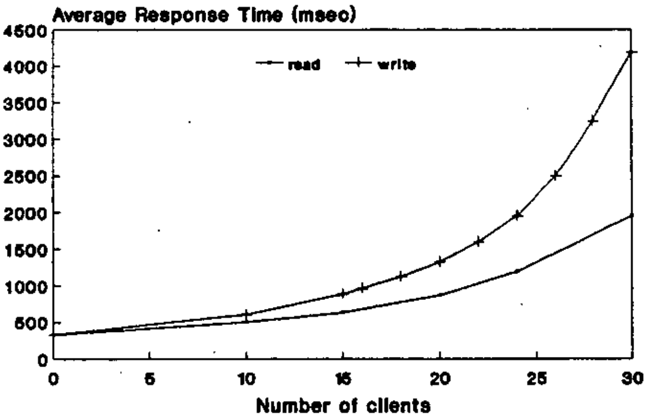


Figure B.1.1.9 : 50.7Kbytes

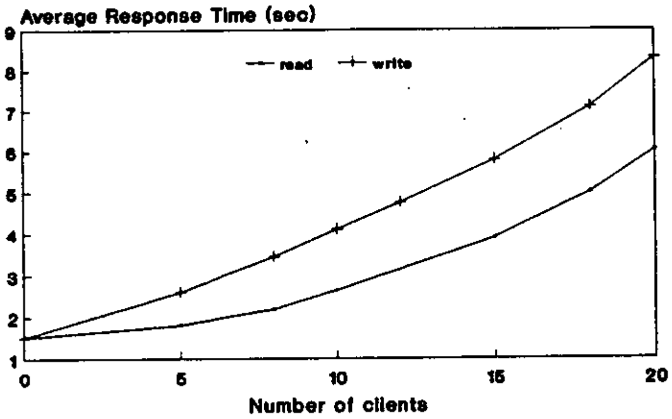


Figure B.1.1.10 : 316Kbytes(B)

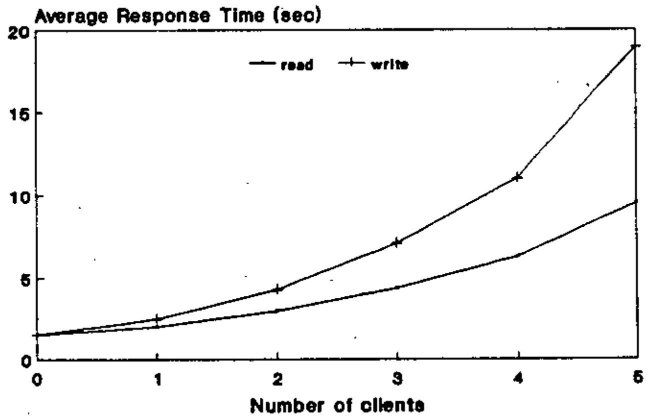


Figure B.1.1.11 : 316Kbytes

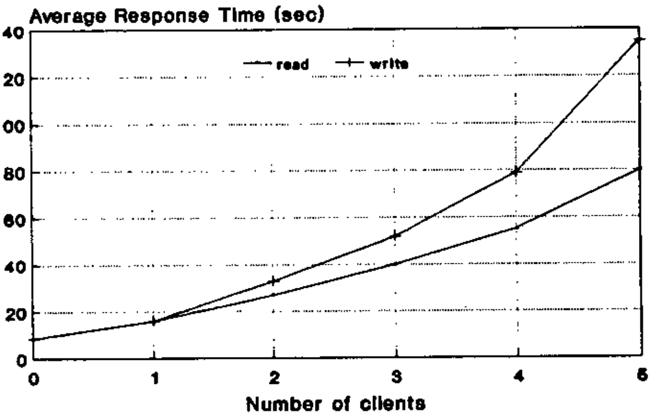


Figure B.1.1.12 : 1856Kbytes

The average response time of the read vs. the average response time of the write in the distributed file system which consists of the Sun SPARCstation 470 workstations.

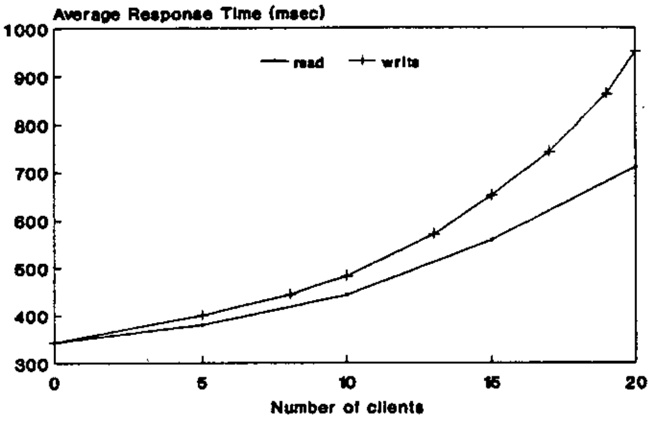


Figure B.1.1.13 : 8Kbytes

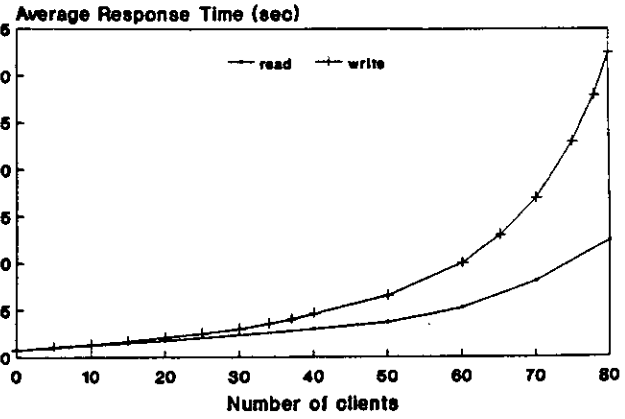


Figure B.1.1.14 : 47Kbytes

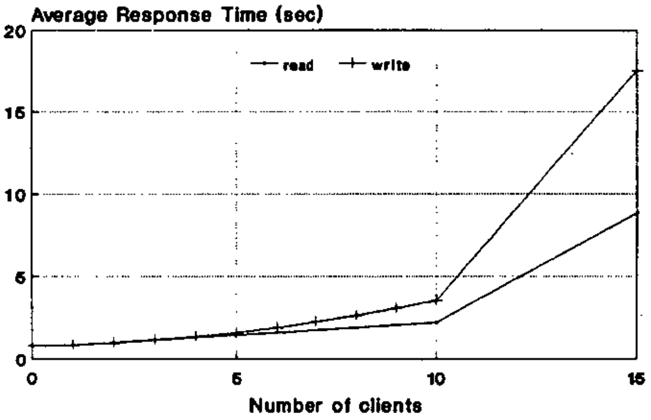


Figure B.1.1.15 : 50.7Kbytes

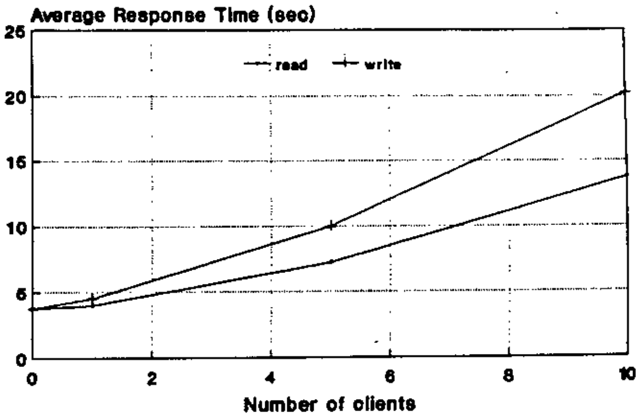


Figure B.1.1.16 : 316Kbytes(B)

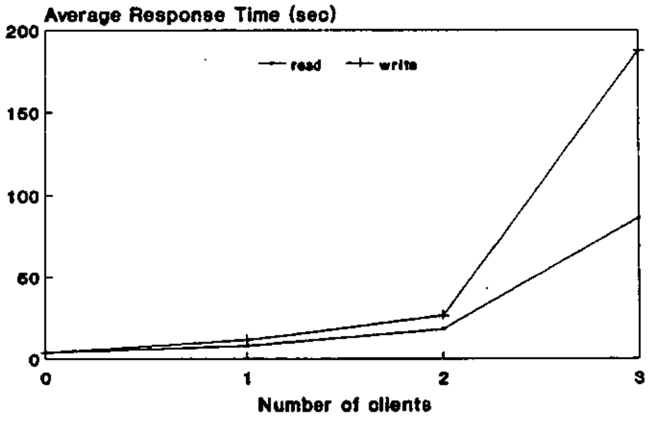


Figure B.1.1.17 : 316Kbytes

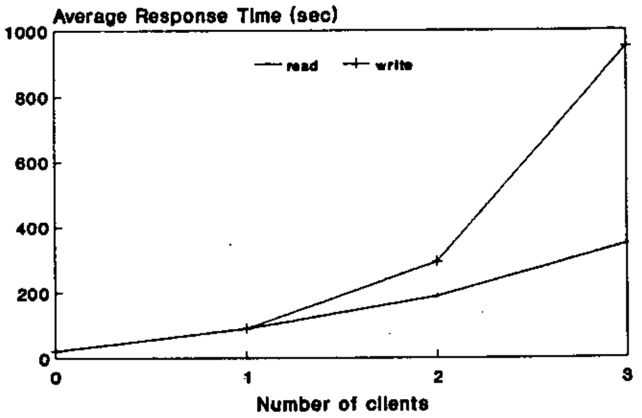


Figure B.1.1.18 : 1856Kbytes

The average response time of the read vs. the average response time of the write in the distributed file system which consists of the Sun 3/60 workstations.

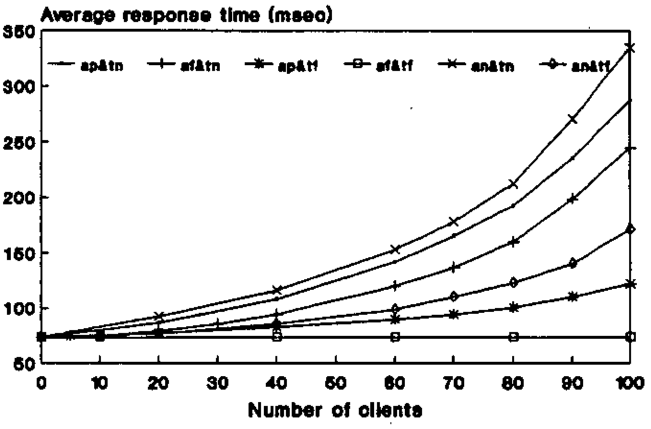


Figure B.1.2.1 : 8Kbytes

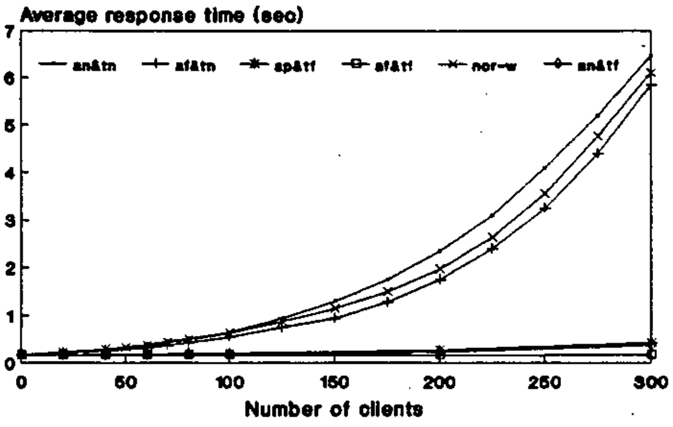


Figure B.1.2.2 : 47Kbytes

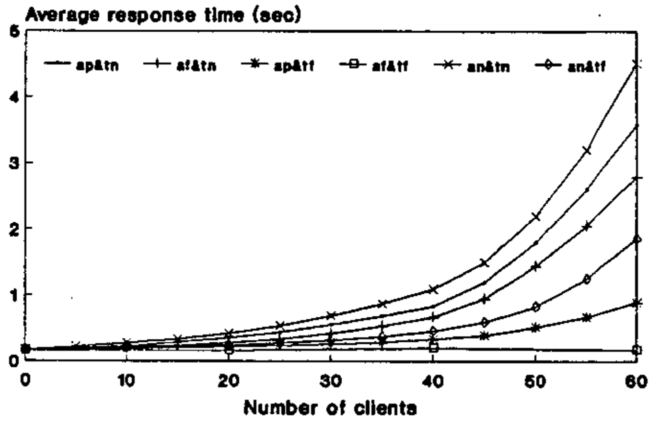


Figure B.1.2.3 : 50.7Kbytes

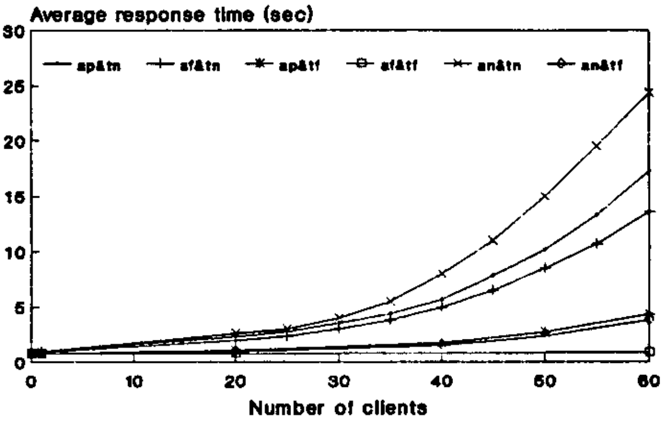


Figure B.1.2.4 : 316Kbytes(B)

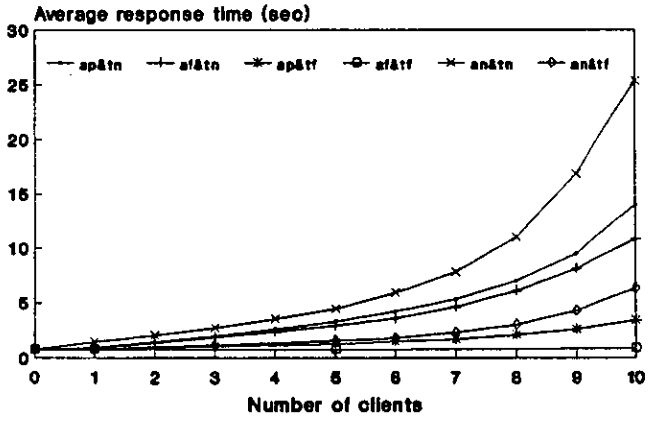


Figure B.1.2.5 : 316Kbytes

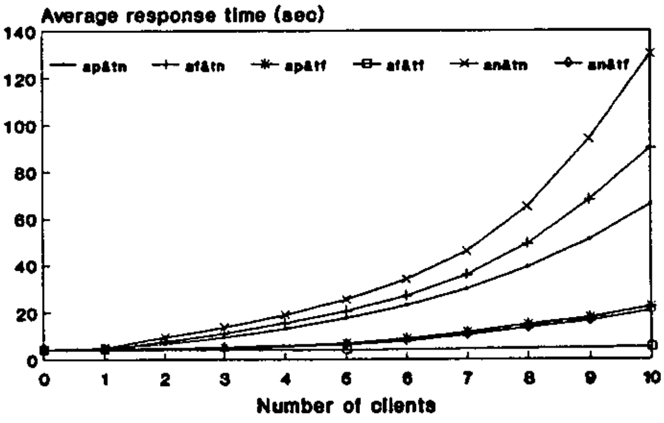


Figure B.1.2.6 : 1856Kbytes

The effect of the workload pattern on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

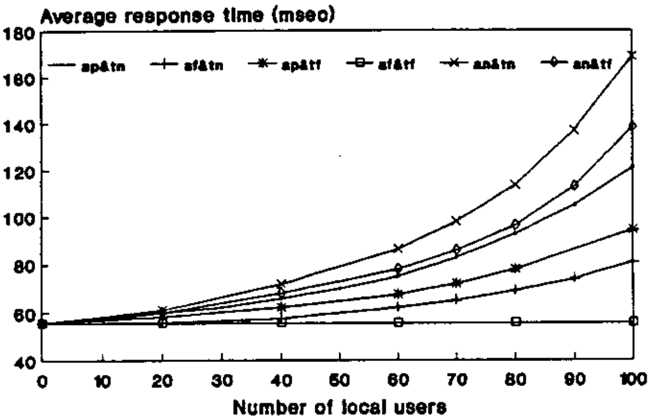


Figure B.1.2.7 : 8Kbytes

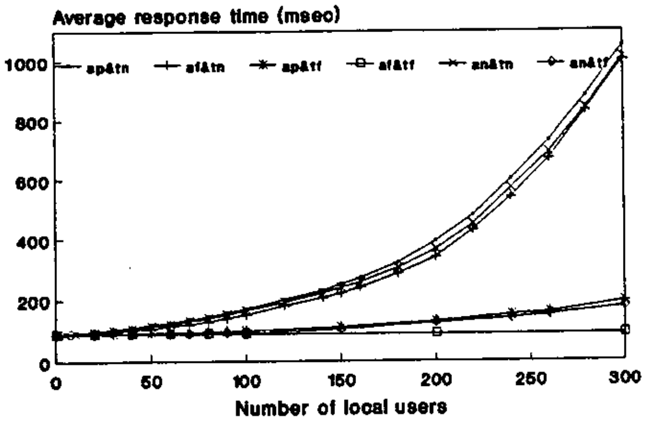


Figure B.1.2.8 : 47Kbytes

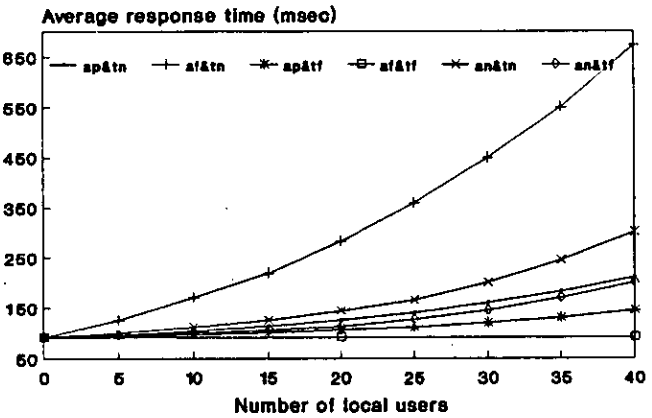


Figure B.1.2.9 : 50.7Kbytes

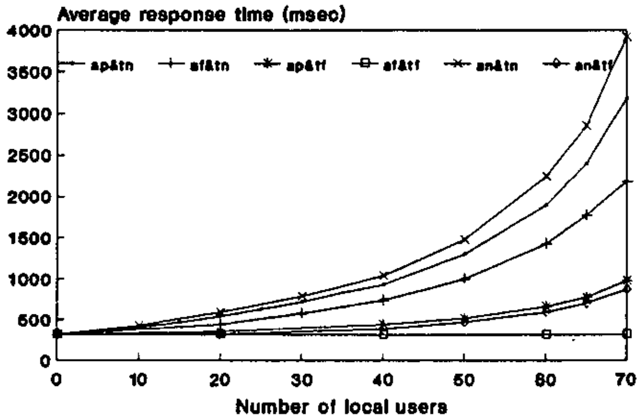


Figure B.1.2.10 : 316Kbytes(B)

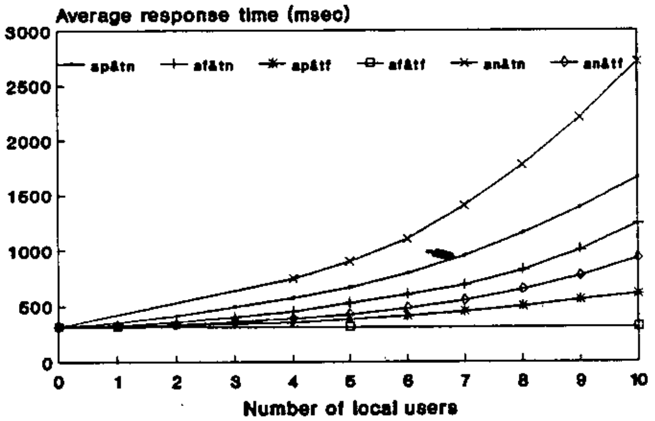


Figure B.1.2.11 : 316Kbytes

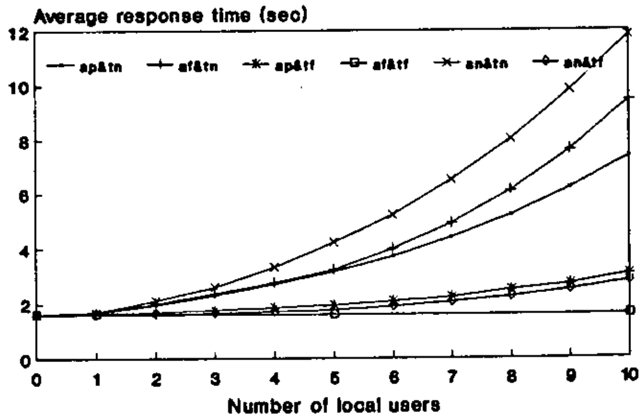


Figure B.1.2.12 : 1856Kbytes

The effect of the workload pattern on the average response time in the Sun SPARCstation 10 workstation.

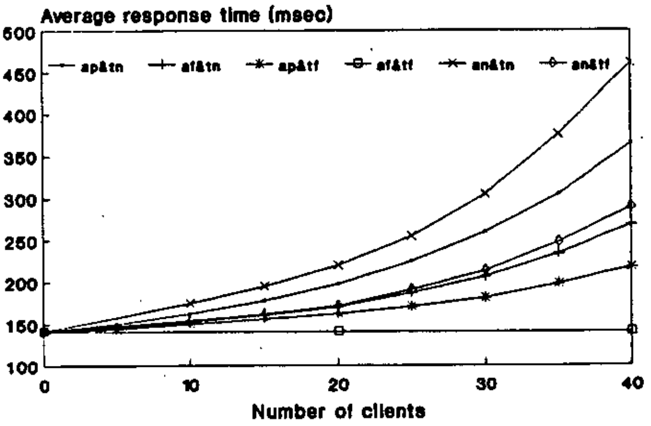


Figure B.1.2.13 : 8Kbytes

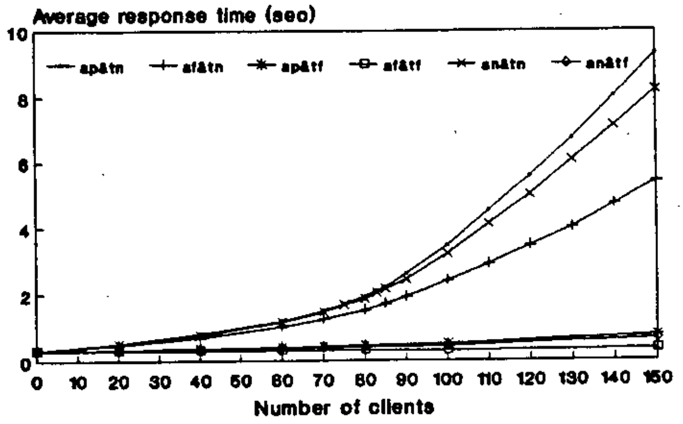


Figure B.1.2.14 : 47Kbytes

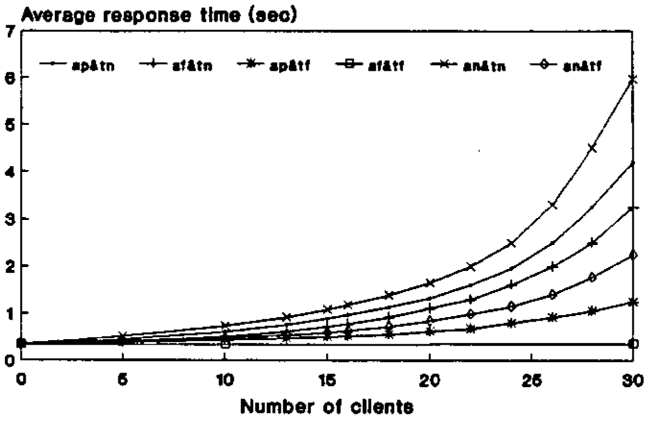


Figure B.1.2.15 : 50.7Kbytes

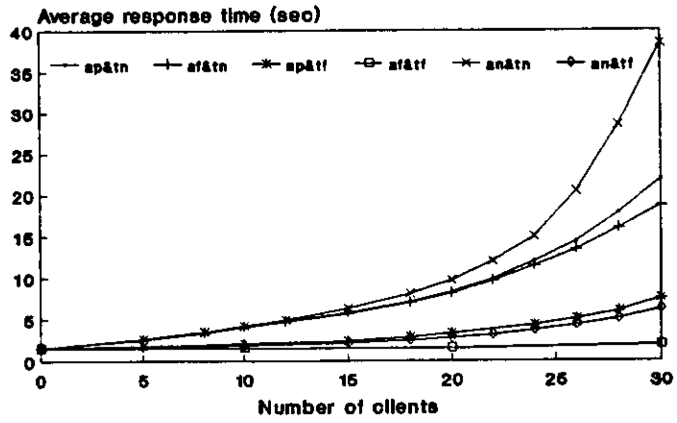


Figure B.1.2.16 : 316Kbytes(B)

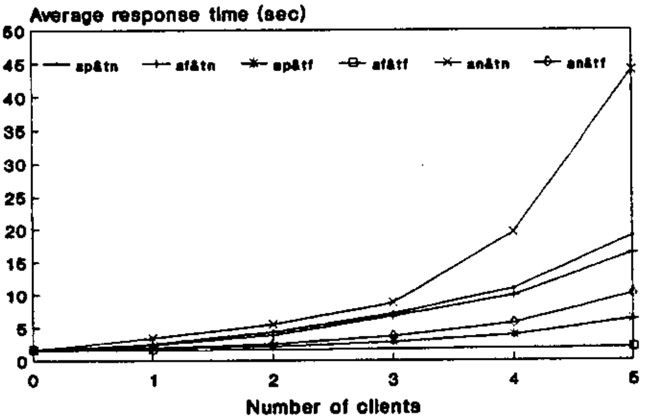


Figure B.1.2.17 : 316Kbytes

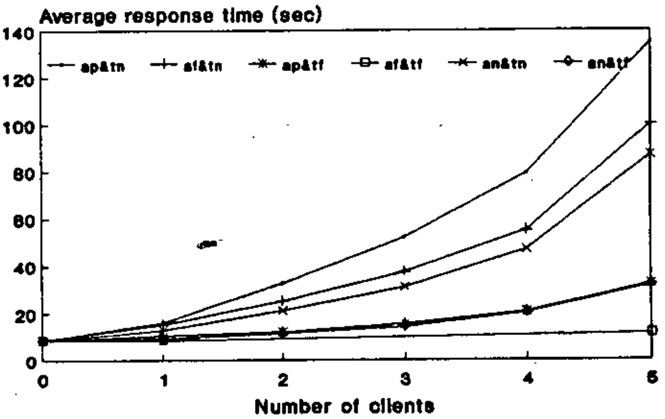


Figure B.1.2.18 : 1856Kbytes

The effect of the workload pattern on the average response time in the distributed file system which consists of the Sun SPARCstation 470 workstations.

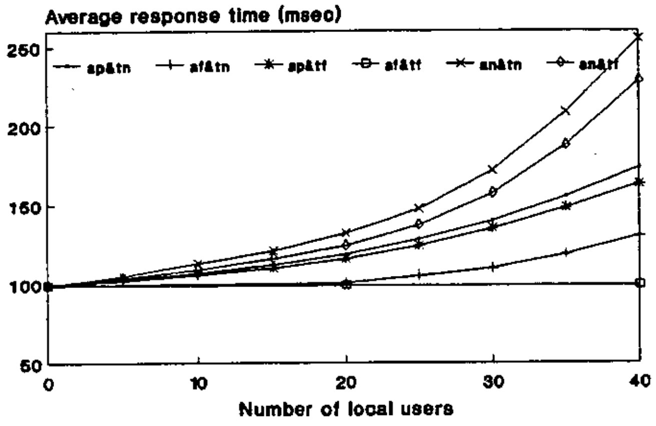


Figure B.1.2.19 : 8Kbytes

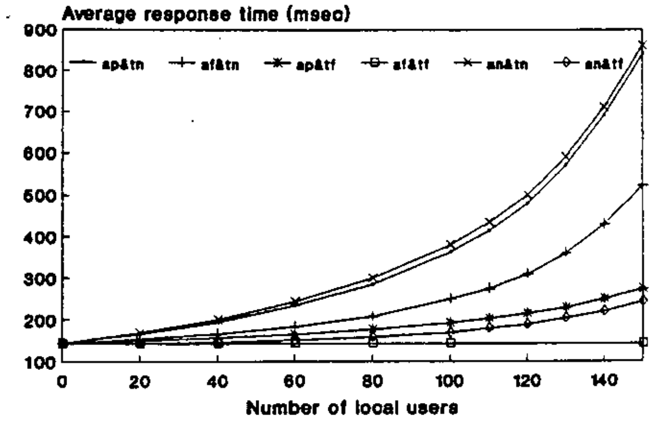


Figure B.1.2.20 : 47Kbytes

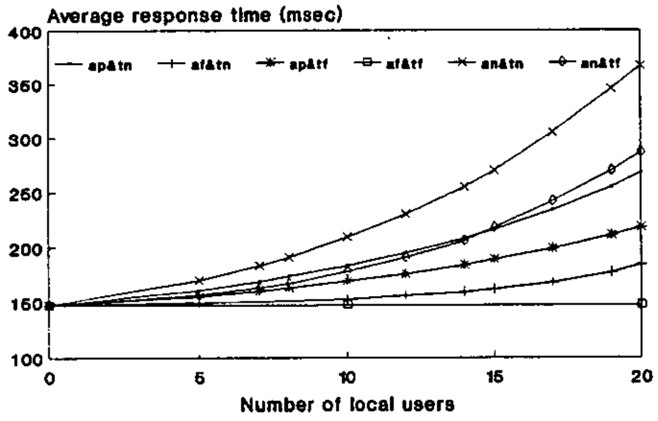


Figure B.1.2.21 : 50.7Kbytes

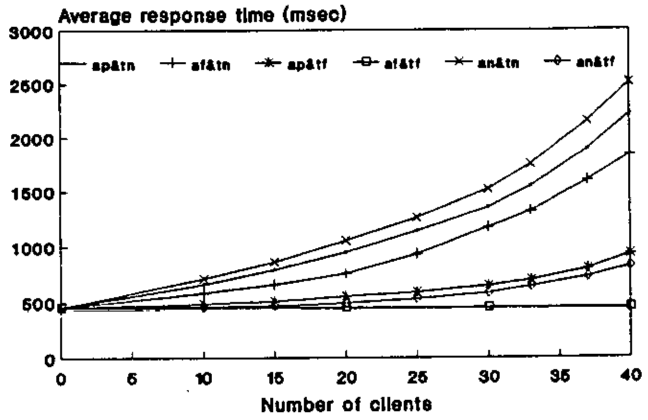


Figure B.1.2.22 : 316Kbytes(B)

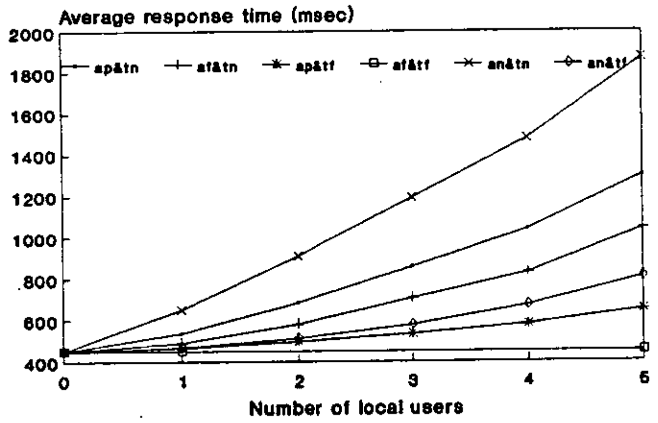


Figure B.1.2.23 : 316Kbytes

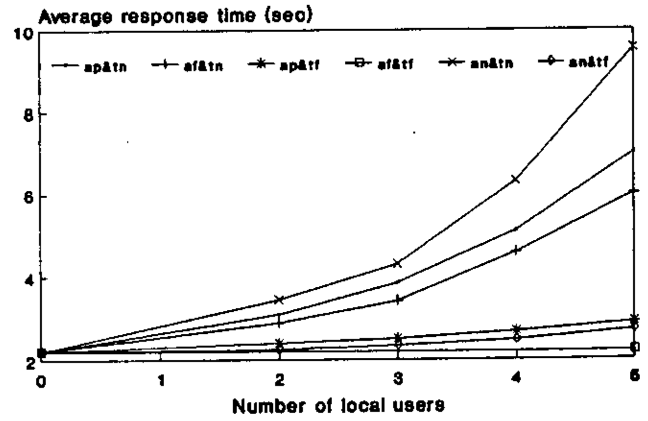


Figure B.1.2.24 : 1856Kbytes

The effect of the workload pattern on the average response time in the Sun SPARCstation 470 workstation.

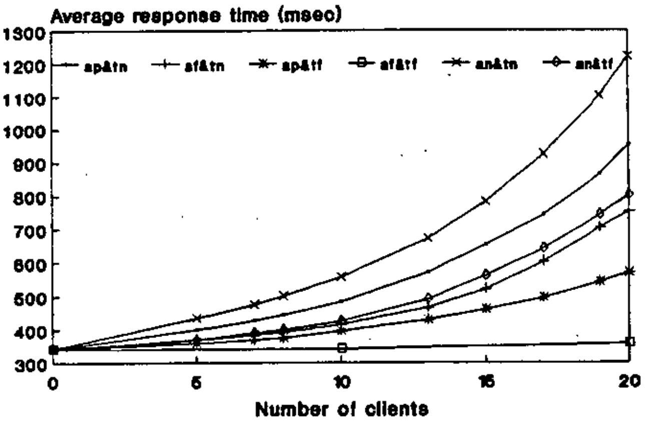


Figure B.1.2.25 : 8Kbytes

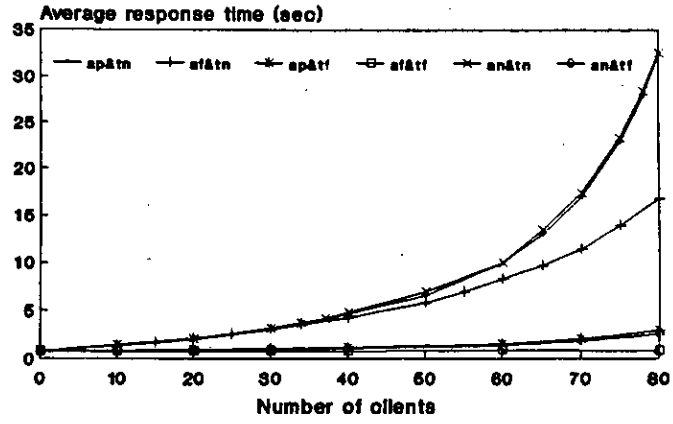


Figure B.1.2.26 : 47Kbytes

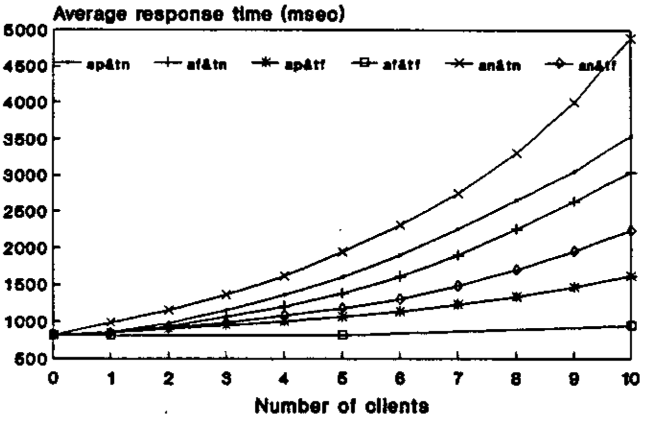


Figure B.1.2.27 : 50.7Kbytes

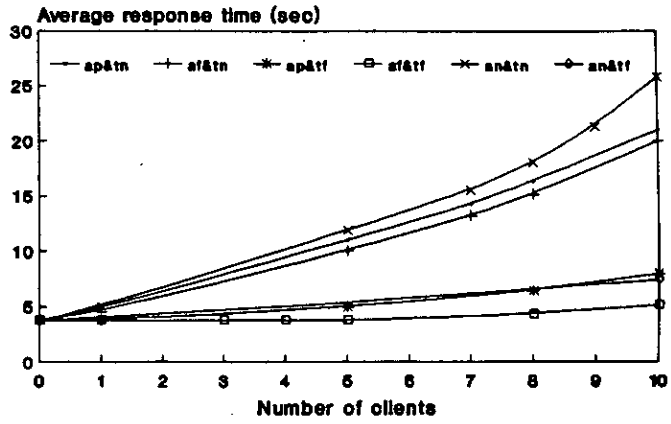


Figure B.1.2.28 : 316Kbytes(B)

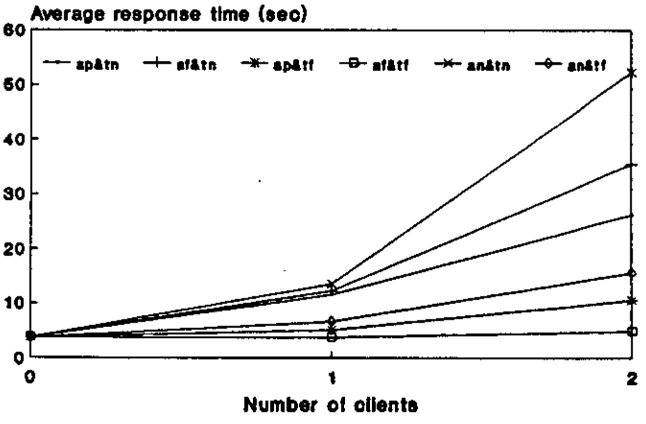


Figure B.1.2.29 : 316Kbytes

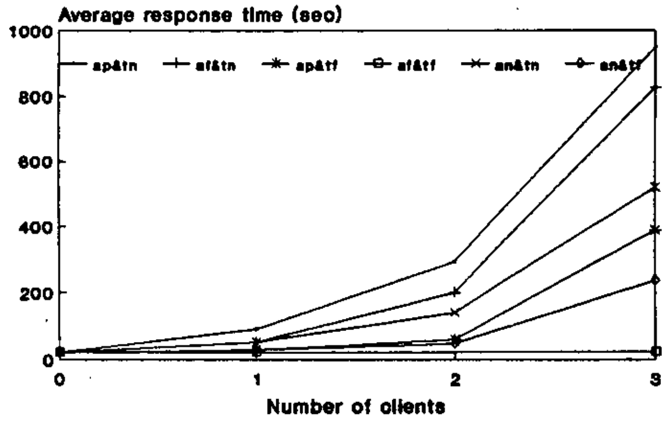


Figure B.1.2.30 : 1856Kbytes

The effect of the workload pattern on the average response time in the distributed file system which consists of the Sun 3/60 workstations.

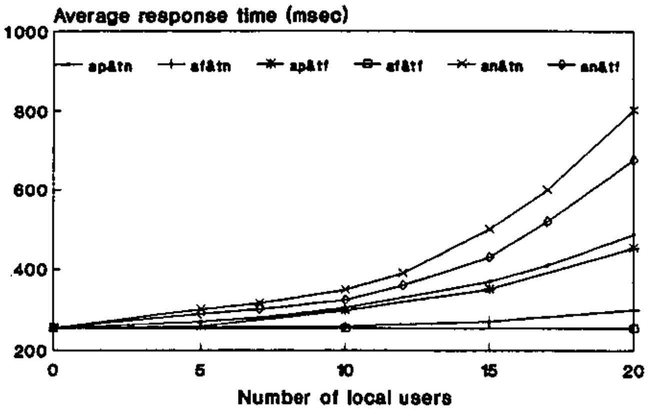


Figure B.1.2.31 : 8Kbytes

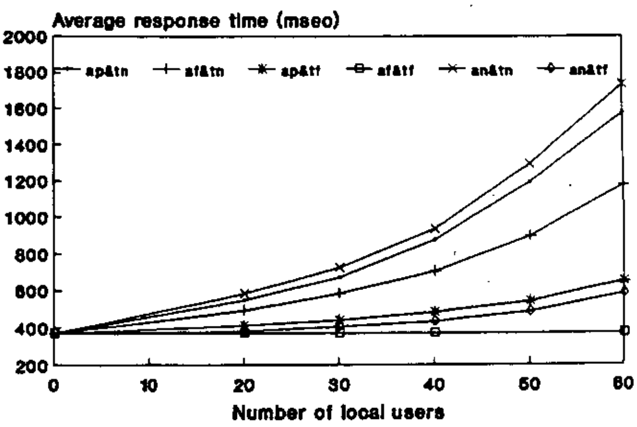


Figure B.1.2.32 : 47Kbytes

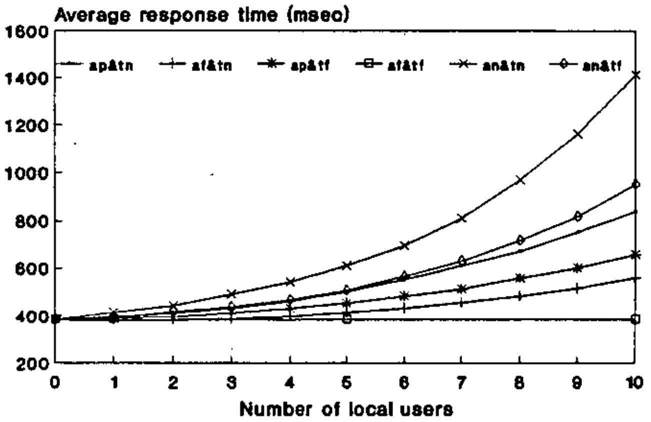


Figure B.1.2.33 : 50.7Kbytes

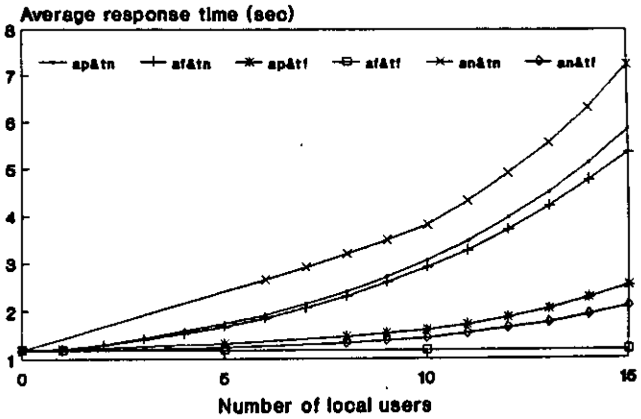


Figure B.1.2.34 : 316Kbytes(B)

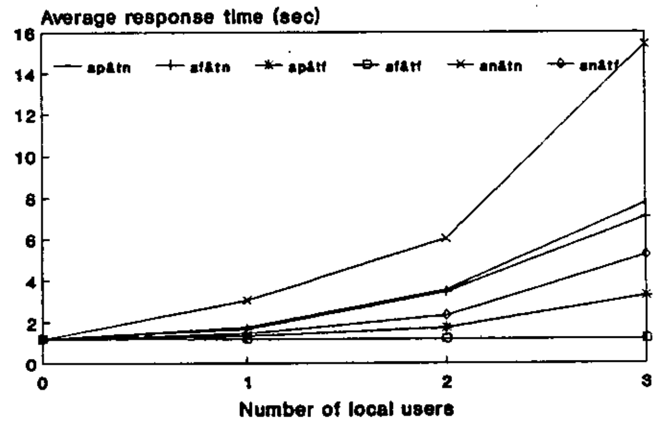


Figure B.1.2.35 : 316Kbytes

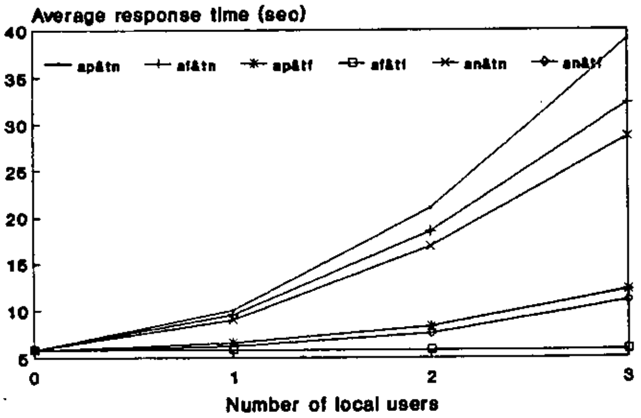


Figure B.1.2.36 : 1856Kbytes

The effect of the workload pattern on the average response time in the Sun 3/60 workstation.

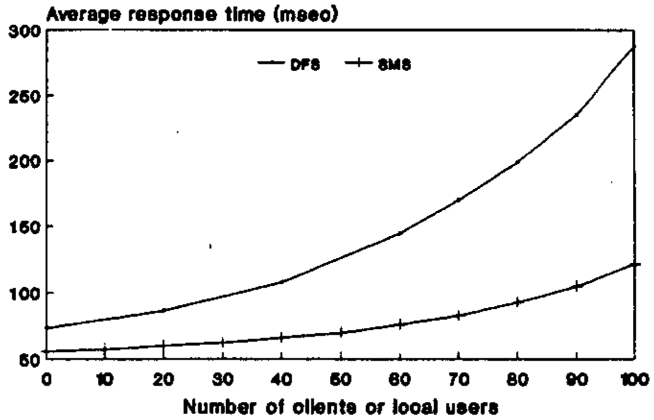


Figure B.2.1 : 8Kbytes

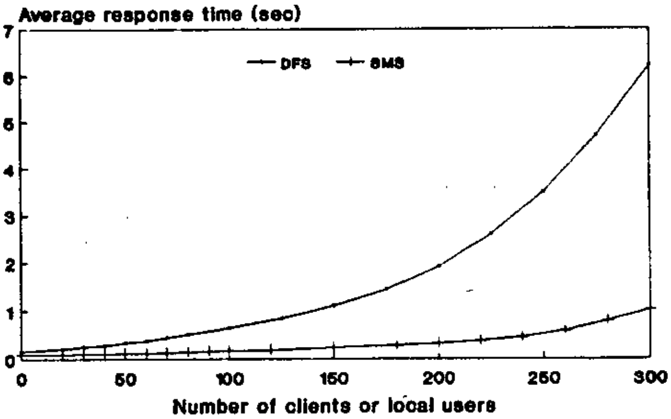


Figure B.2.2 : 47Kbytes

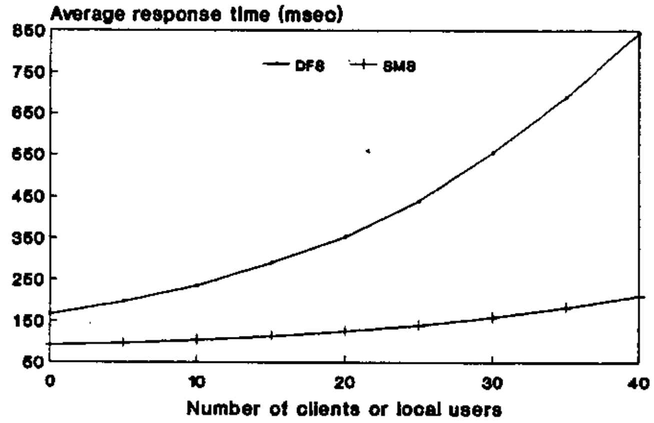


Figure B.2.3 : 50.7Kbytes

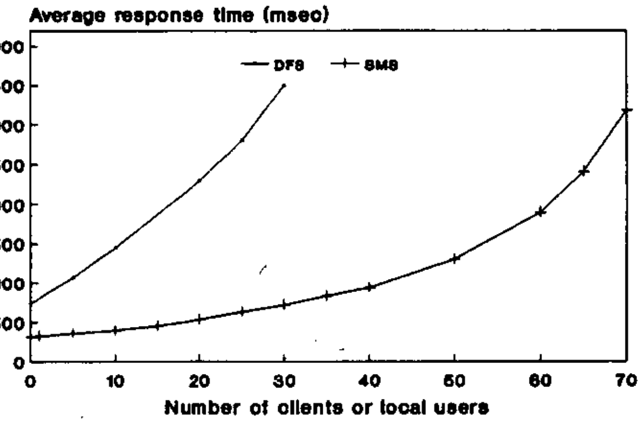


Figure B.2.4 : 316Kbytes(B)

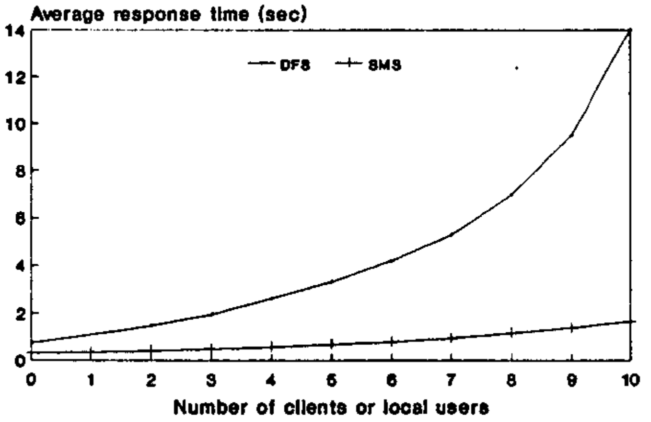


Figure B.2.5 : 316Kbytes

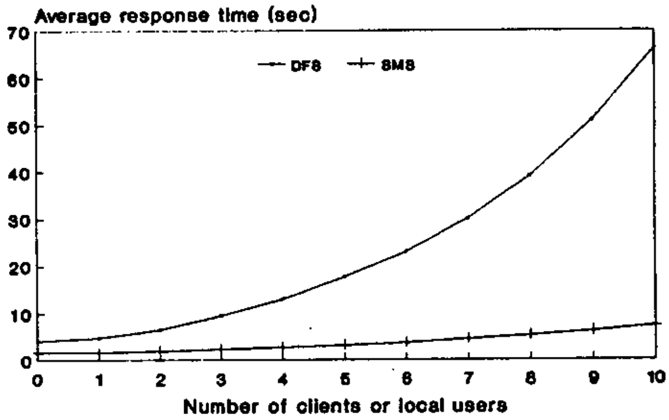


Figure B.2.6 : 1856Kbytes

The average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations vs. the average response time in the Sun SPARCstation 10 workstation.

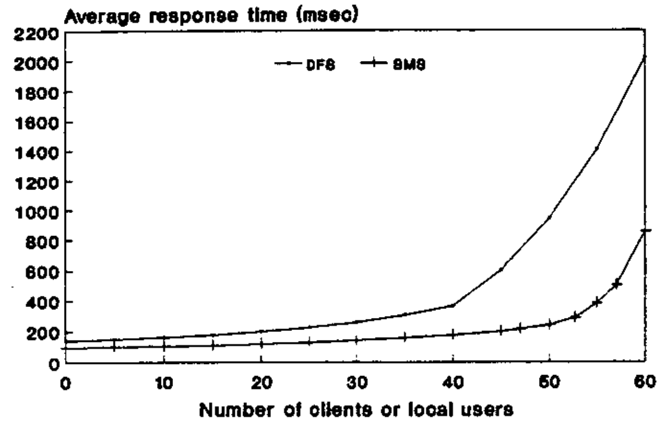


Figure B.2.7 : 8Kbytes

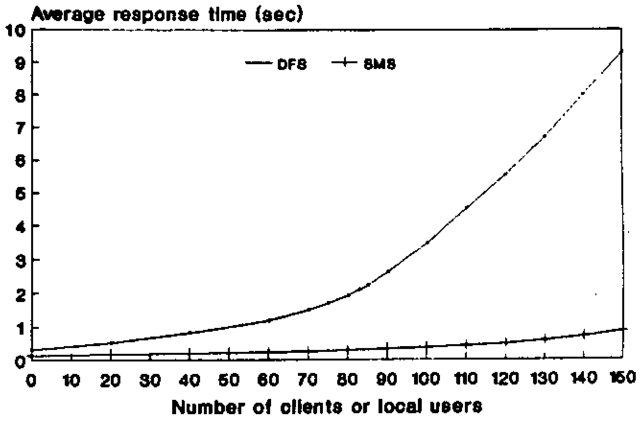


Figure B.2.8 : 47Kbytes

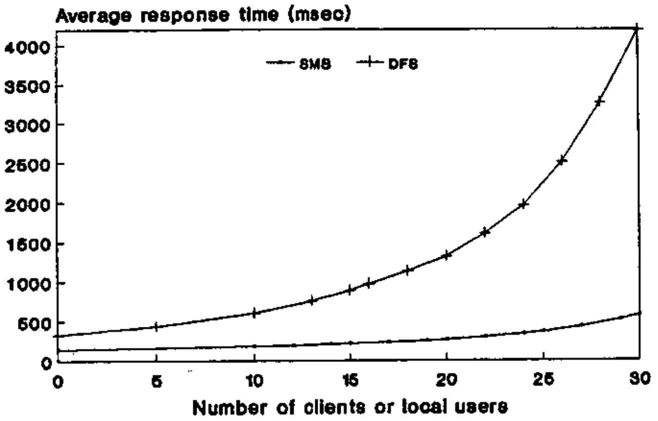


Figure B.2.9 : 50.7Kbytes

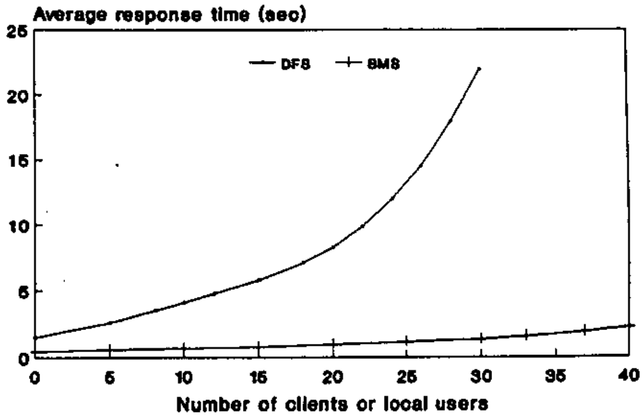


Figure B.2.10 : 316Kbytes(B)

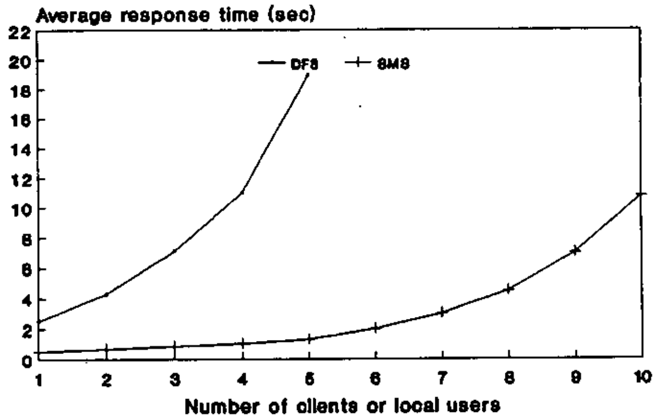


Figure B.2.11 : 316Kbytes

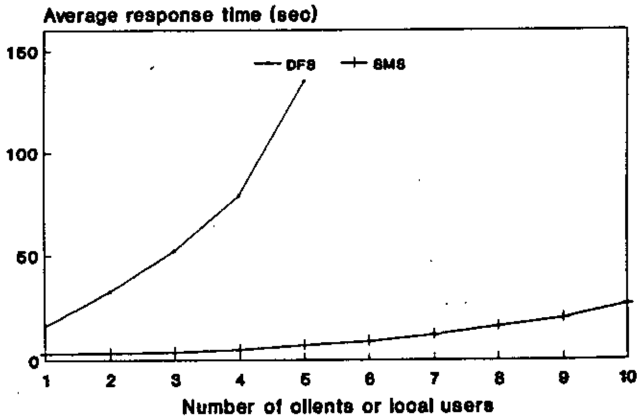


Figure B.2.12 : 1856Kbytes

The average response time in the distributed file system which consists of the Sun SPARCstation 470 workstations vs. the average response time in the Sun SPARCstation 470 workstation.

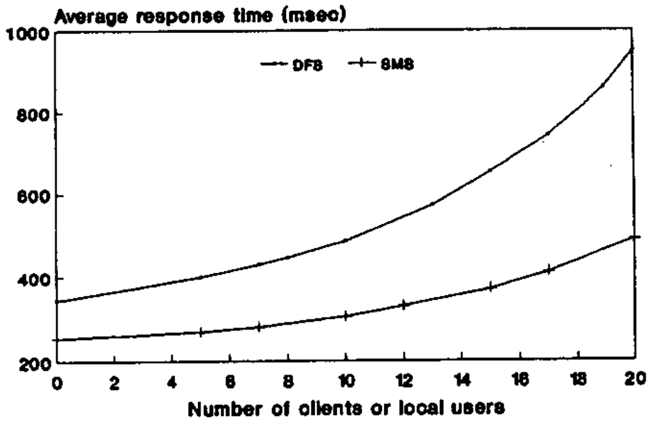


Figure B.2.13 : 8Kbytes

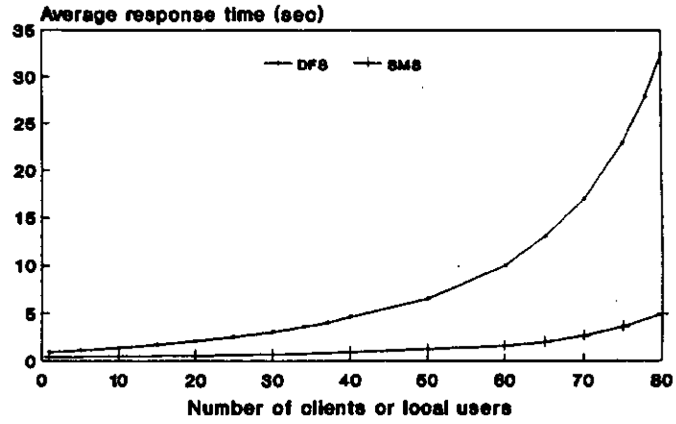


Figure B.2.14 : 47Kbytes

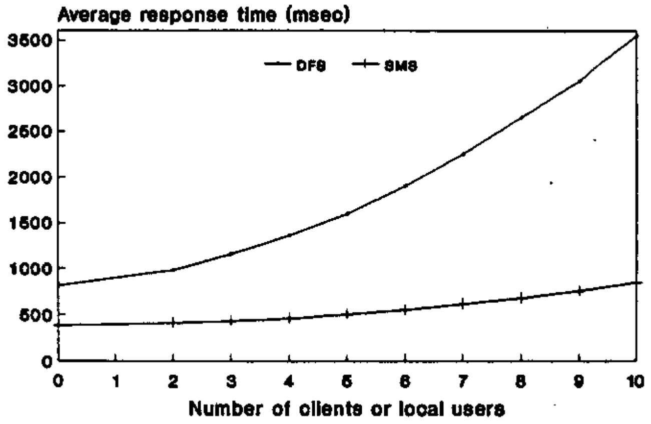


Figure B.2.15 : 50.7Kbytes

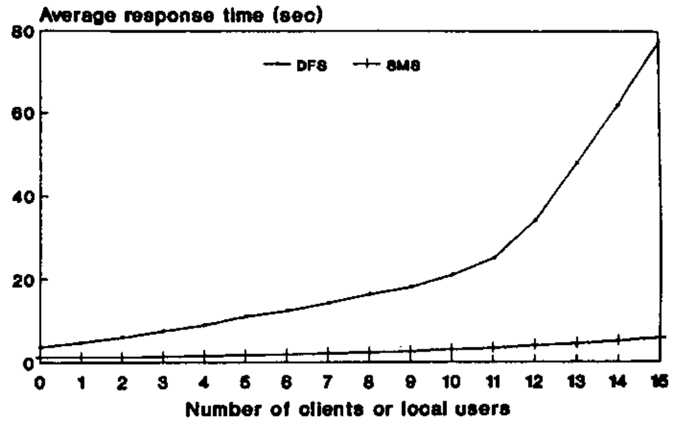


Figure B.2.16 : 316Kbytes(B)

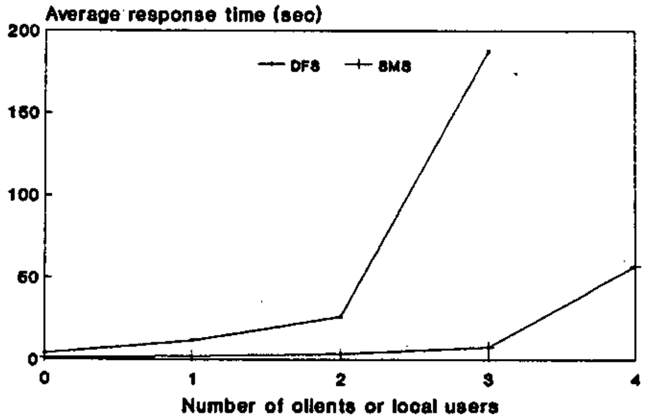


Figure B.2.17 : 316Kbytes

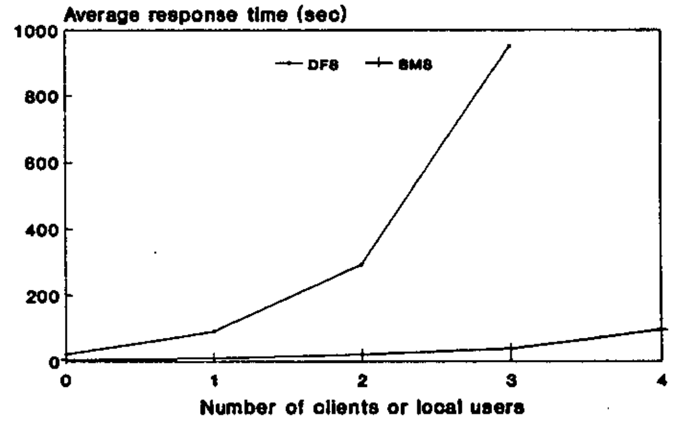


Figure B.2.18 : 1856Kbytes

The average response time in the distributed file system which consists of the Sun 3/60 workstations vs. the average response time in the Sun 3/60 workstation.

Appendix C

The Effect of the Design Alternatives

C.1 Multiple CPUs

Figure C.1.1 shows the average response time of the 8kbytes workload in the distributed file system which consists of Sun SPARCstation 10 workstations where CPUs are added as the number of clients increases gradually. Figure C.1.2 to figure C.1.6 show the average response time of the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively.

Figure C.1.7. to figure C.1.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively when the number of CPUs of the shared memory system is increased to be 2 CPUs, 4 CPUs, 8 CPUs, 10 CPUs, 16 CPUs, 20 CPUs, 24 CPUs, 26 CPUs and 30 CPUs. The base system to which the CPUs are added is the Sun SPARCstation 10 workstation in the figures, as in the distributed file system.

C.2 Better CPU

Figure C.2.1 to figure C.2.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In the simulations for each figure, the CPUs of the baseline distributed file system which consists of the Sun SPARCstation 10 workstations are replaced by the 2 times, 4 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better CPUs.

Figure C.2.7 to figure C.2.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory systems as the number of local users increases gradually. In each figure, the CPU of the baseline Sun SPARCstation 10 workstation is replaced by a 2 times, 4 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better CPU individually.

C.3 Multiple Disk I/O Subsystems

Figure C.3.1 to figure C.3.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the number of disks and the number of disk interface unit in the file server are increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity at the same time. Except the disk and the disk interface unit in the file server, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure C.3.7 to figure C.3.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the number of disks and the number of disk interface units are increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity at the same time. Except the disks and the disk interface units, all others are kept the same as the baseline Sun SPARCstation 10 workstation.

C.4 Better Disk I/O Subsystem

C.4.1 Reduced Disk I/O Time

Figure C.4.1.1 to figure C.4.1.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, both the constant portion and the proportional portion of the disk I/O time are improved to be 2, 4, 8, 10, 16, 20, 30, 100, 1000 and infinitely faster. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The baseline performance model of figure 3.2.6.B is used for the distributed file system.

Figure C.4.1.7 to figure C.4.1.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, both the constant portion and the proportional portion of the disk I/O time are improved to be 2, 4, 8, 10, 16, 20, 30, 100, 1000 and infinitely faster. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation. The baseline performance model of figure 3.4.1.B is used for the shared memory system.

C.4.2 Other improvements

Figure C.4.2.1 to figure C.4.2.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the CPU service time parameter value for disk I/O is improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The baseline performance model of figure 3.2.6.B is used for the distributed file system.

Figure C.4.2.7 to figure C.4.2.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the CPU service time parameter value for disk I/O is improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation. The baseline performance model of figure 3.4.1.B is used for the shared memory system.

C.4.3 All Improvements at the Same Time

Figure C.4.3.1 to figure C.4.3.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the values of all parameters for disk I/O are improved to be 2, 4, 8, 10, 16, 20, 30

100, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The baseline performance model of figure 3.2.6.B is used for the distributed file system.

Figure C.4.3.7 to figure C.4.3.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the values of all parameters for disk I/O are improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation. The baseline performance model of figure 3.4.2.B is used for the shared memory system.

C.5 Multiple Networks and Multiple Network Interface Units

Figure C.5.1 to figure C.5.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, both the number of networks and the number of network interface units in the file server are increased to be K(2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity). Except the number of networks and the number of network interface units in the file server, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

C.6 Faster Network Communication

C.6.1 Faster Network

Figure C.6.1.1 to figure C.6.1.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the baseline performance model of figure 4.2.6.B is used and only the network transmission speed is improved to be 2(20Mbps), 5(50Mbps), 10(100Mbps), 50(500Mbps), 100(1Gbps), 1000(10Gbps) times and infinitely faster. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The network retransmission delay may be adjusted when the transmission speed is changed.

C.6.2 Better Network Interface Unit

Figure C.6.2.1 to figure C.6.2.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the baseline performance model of figure 4.2.6.B is used and the I/O time for the request send operation and that for the response receive operation in the network interface units of the clients and that for the request receive operation and that for the response send operation in the network interface unit of the file server are improved to be 2 times, 4 times, 6 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

C.6.3 Enhanced Communication Mechanism

Figure C.6.3.1 to figure C.6.3.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the baseline performance model of figure 4.2.6.B is used and the CPU time for the request send operation and that for the response receive operation in the clients and that for the request receive operation and that for the response send operation in the file server are improved to be 2 times, 4 times, 6 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

C.6.4 All Improvements at the Same Time

Figure C.6.4.1 to figure C.6.4.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the baseline performance model of figure 4.2.6.B is used and the values of all parameters for the network communication are improved to be 2 times, 4 times, 6 times, 8 times, 10 times, 16 times, 20 times, 30 times, 100 times, 1000 times and infinitely better. Except them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations. The network speed is set to be 50Mbps not 40Mbps for the 4 times better case and 100Mbps for the 8 times better case and for the 16 times better case. In all other cases, the degree of improvement is kept same in all parameters.

C.7 Multiple Resources in the System

Figure C.7.1 to figure C.7.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system which consists of the Sun SPARCstation 10 workstations as the number of clients increases gradually. In each figure, the number of resources in the file server is increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity. Except for them, all others are kept the same as the baseline distributed file system.

Figure C.7.7 to figure C.7.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system of the Sun SPARCstation 10 workstation as the number of local users increases gradually. In each figure, the number of resources is increased to be 2, 4, 8, 10, 16, 20, 30 100, 1000 and infinity. Except for them, all others are kept the same as the baseline shared memory system.

C.8 Better System

Figure C.8.1 to figure C.8.6 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the values of all parameters except for the network transmission speed in table 4.2.7.C are improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure C.8.7 to figure C.8.12 show the average response time of the 8kbytes workload, the 47kbytes workload, the 50.7kbytes workload, the 316kbytes(B) workload, the 316kbytes workload and the 1856kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the values of all parameters are improved to be 2, 4, 8, 10, 16, 20, 30 100, 1000 times and infinitely better. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation.

C.9 Concurrency

C.9.1 Concurrency during Disk I/O Operations

Figure C.9.1.1 to figure C.9.1.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system which consists of the Sun SPARCstation 10 workstations as the number of clients increases gradually. In each figure, the degree of concurrency in the disk interface unit of the file server is improved to be 20%, 40%, 60%, 80% and 100% better respectively. At 100% improvement, the CPU and the disk interface unit are absolutely independent of each other during the disk I/O operations. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure C.9.1.7 to figure C.9.1.12 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the degree of concurrency in the disk interface unit is improved to be 20%, 40%, 60%, 80% and 100% better respectively. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation.

C.9.2 Concurrency during Communication Operations

Figure C.9.2.1 to figure C.9.2.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system which consists of the Sun SPARCstation 10 workstations as the number of clients increases gradually. In each figure, the degree of concurrency is improved to be 20%, 40%, 60%, 80% and 100% better. At 100% improvement, the CPU and the network interface unit are absolutely independent of each other during the network communication operations. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

C.10 Everything Better

Figure C.10.1 shows the average response time of the 8Kbyte workload in the distributed file system which consists of the Sun SPARCstation 10 workstations when all parameter values are improved to be 2, 4, 8, 10, 20, 30, 100 and 1000 times better. Figure C.10.2 to figure C.10.6 show the average response time of the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively. For the distributed file system where all parameter values are improved to be four times better, a network of 50Mbps speed is used, therefore the network speed is five times faster, not four times faster. For the distributed file system where all parameter values are improved to be 8 times better and 16 times better, a network of 100Mbps speed is used, therefore the network speed is 10 times faster, not 8 times faster or 16 times faster.

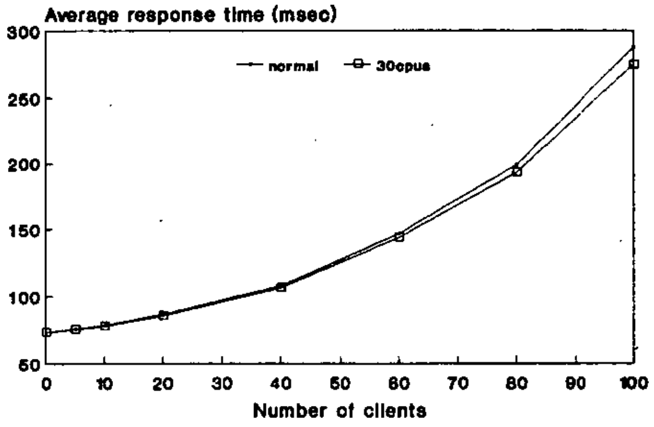


Figure C.1.1 : 8Kbytes

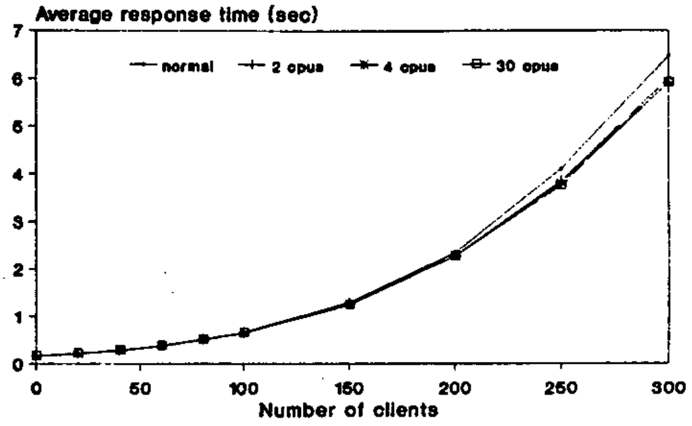


Figure C.1.2 : 47Kbytes

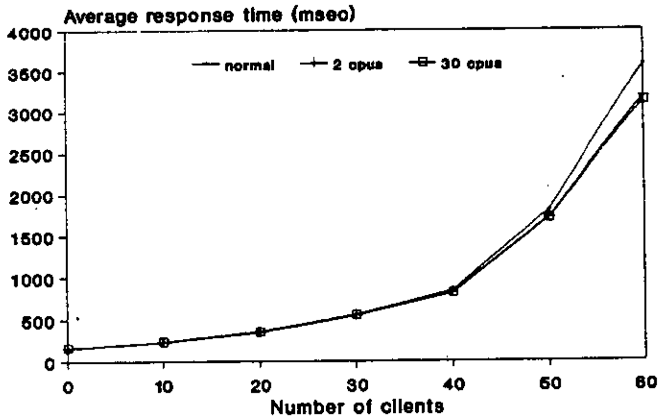


Figure C.1.3 : 50.7Kbytes

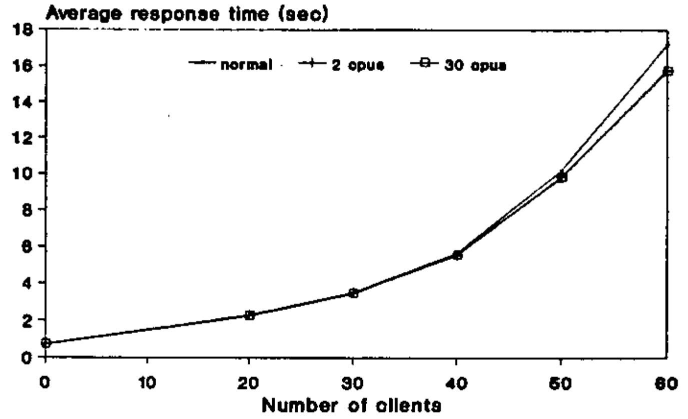


Figure C.1.4 : 316Kbytes(B)

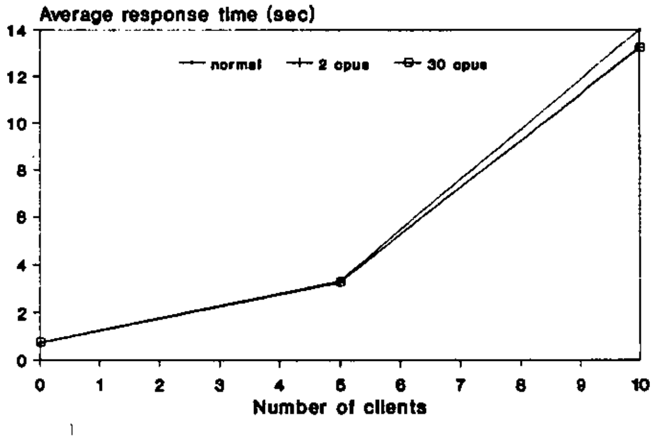
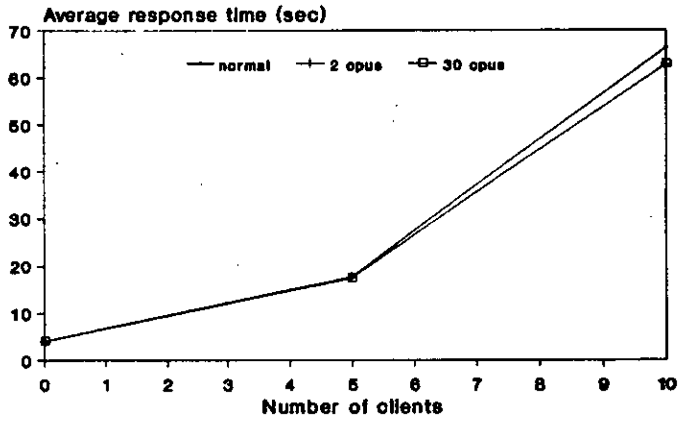


Figure C.1.5 : 316Kbytes



Cigure B.1.6 : 1856Kbytes

The effect of having multiple CPUs on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

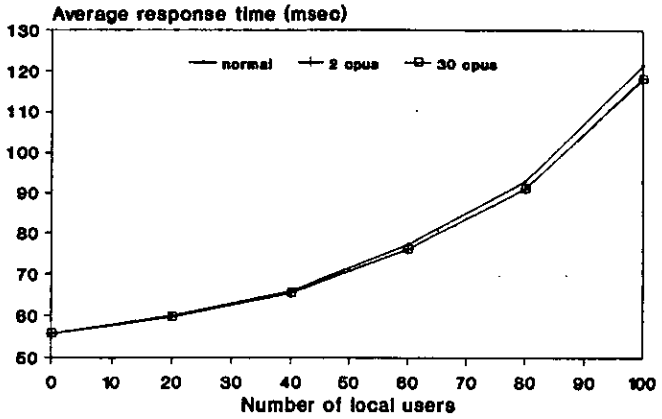


Figure C.1.7 : 8Kbytes

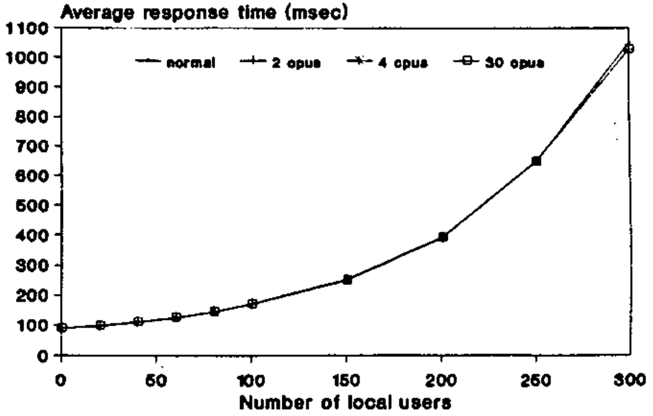


Figure C.1.8 : 47Kbytes

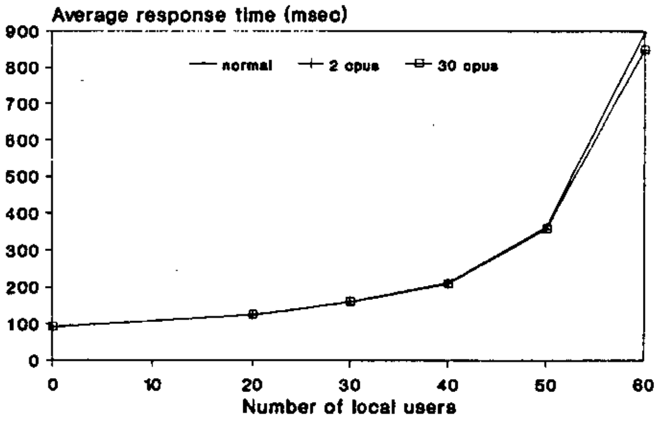


Figure C.1.9 : 50.7Kbytes

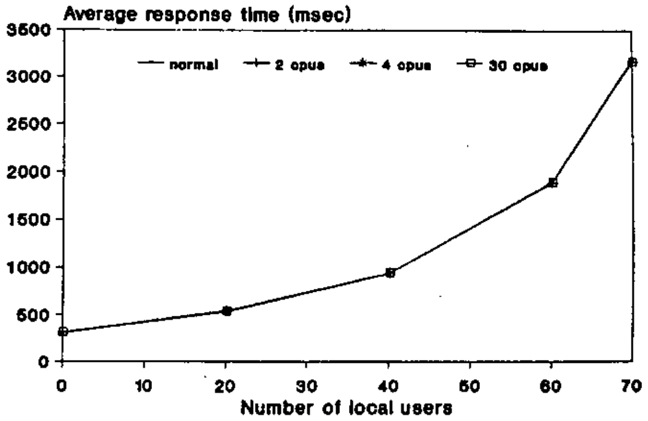


Figure C.1.10 : 316Kbytes(B)

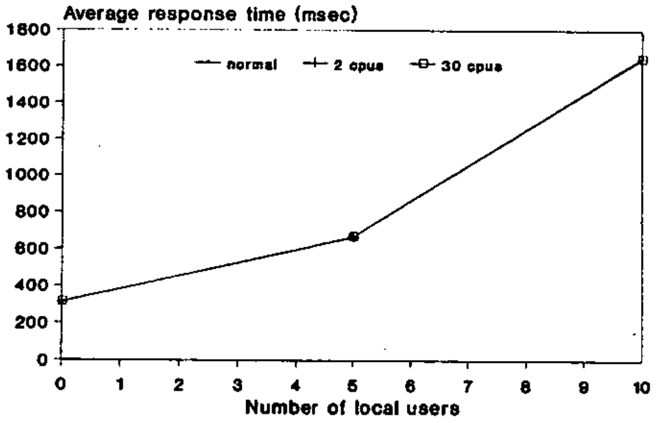


Figure C.1.11 : 316Kbytes

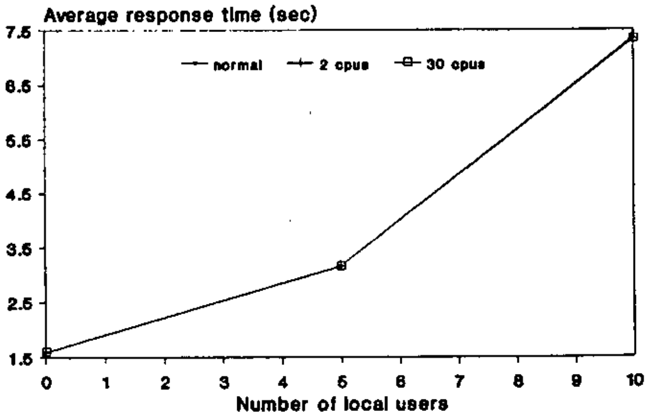


Figure C.1.12 : 1856Kbytes

The effect of having multiple CPUs on the average response time in the Sun SPARCstation 10 workstations.

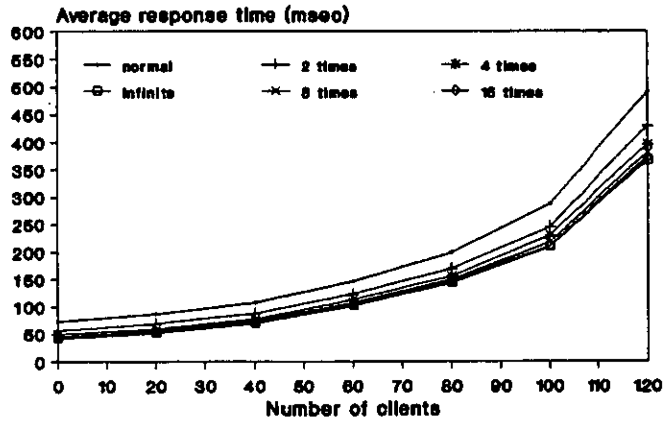


Figure C.2.1 : 8Kbytes

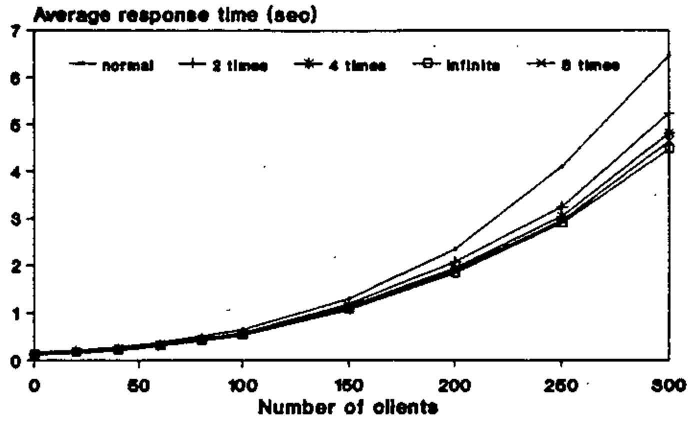


Figure C.2.2 : 47Kbytes

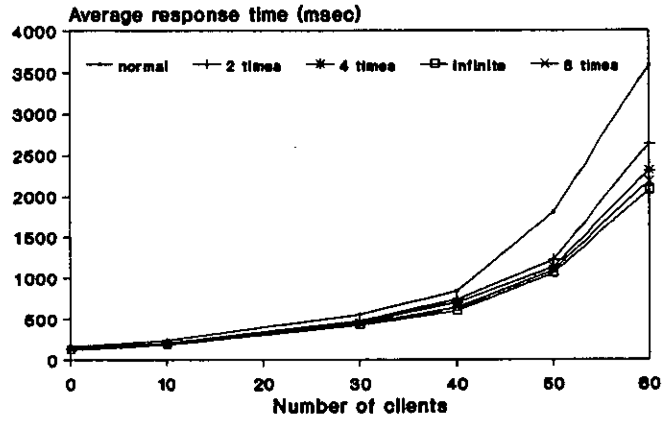


Figure C.2.3 : 50.7Kbytes

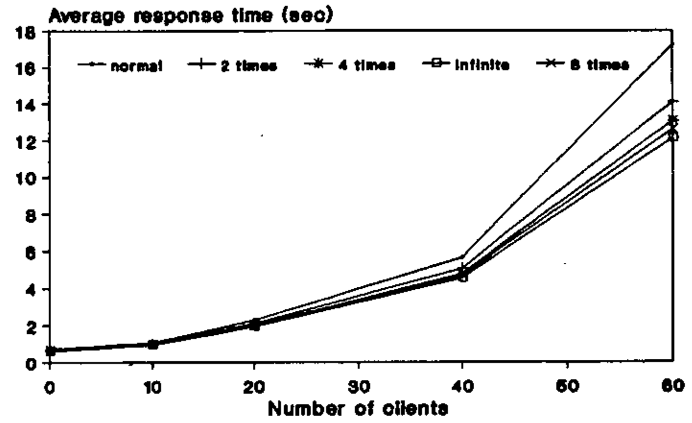


Figure C.2.4 : 316Kbytes(B)

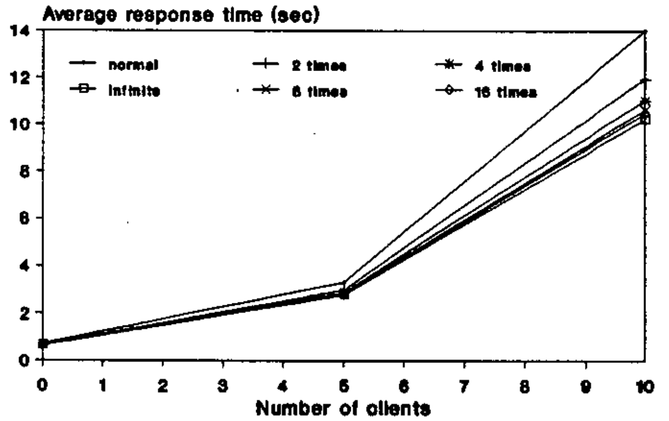


Figure C.2.5 : 316Kbytes

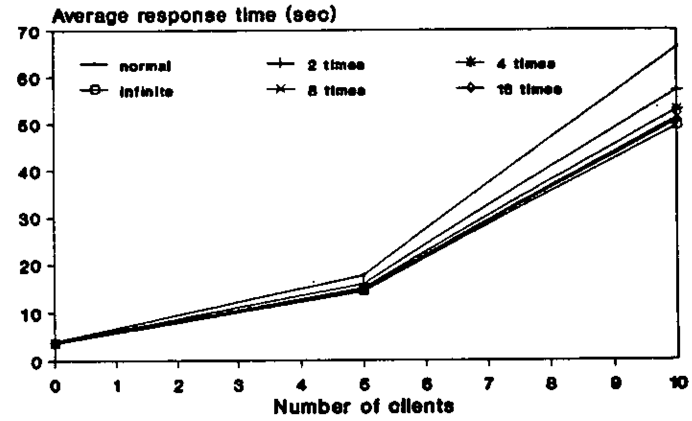


Figure C.2.6 : 1856Kbytes

The effect of the better CPU on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

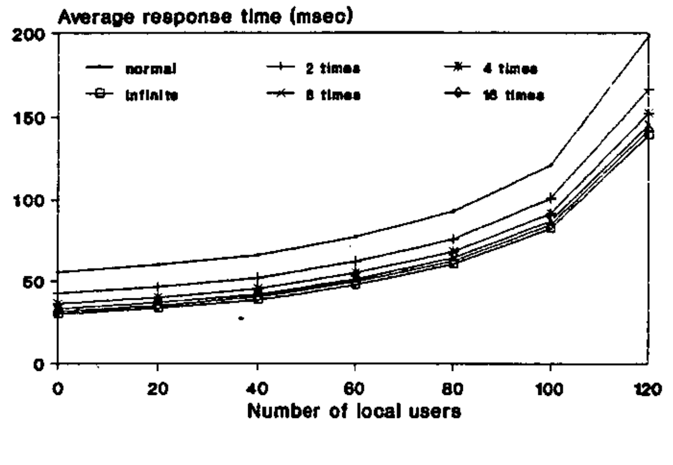


Figure C.2.7 : 8Kbytes

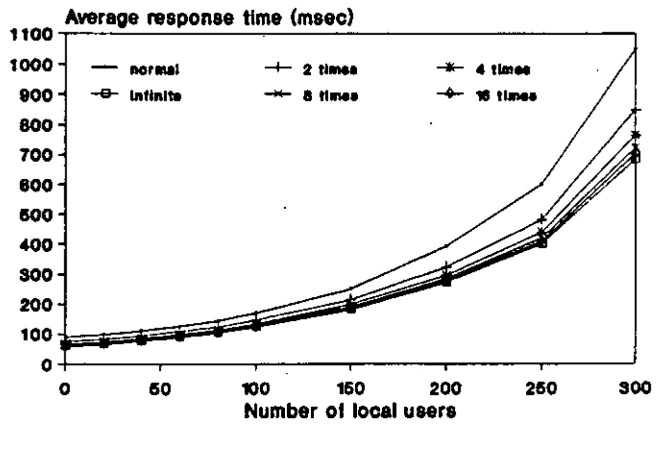


Figure C.2.8 : 47Kbytes

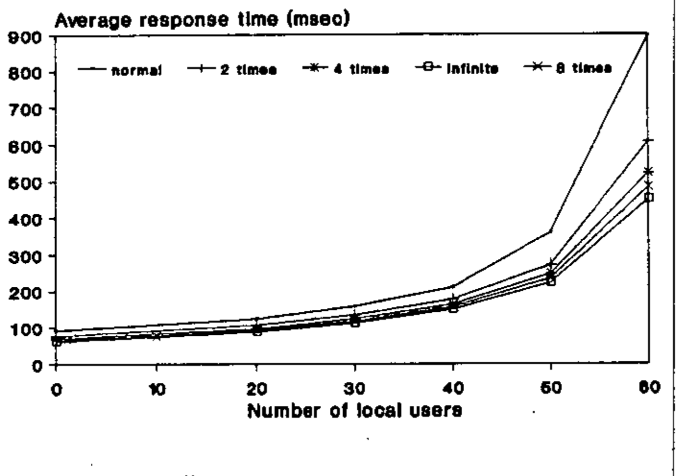


Figure C.2.9 : 50.7Kbytes

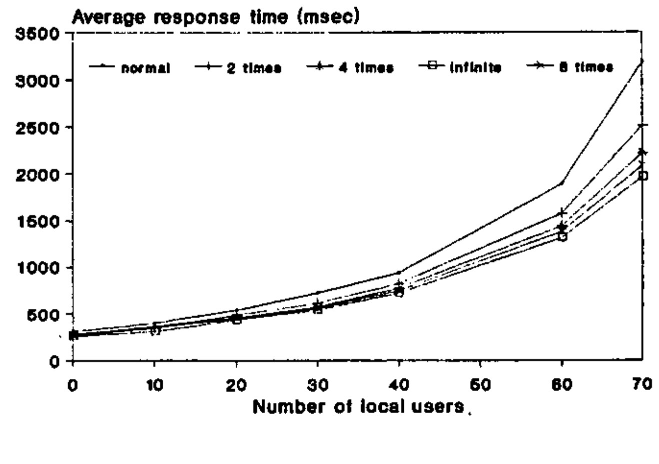


Figure C.2.10 : 316Kbytes(B)

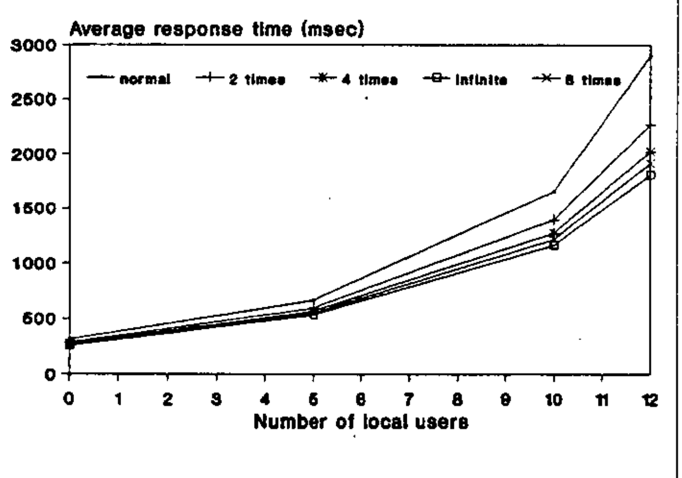


Figure C.2.11 : 316Kbytes

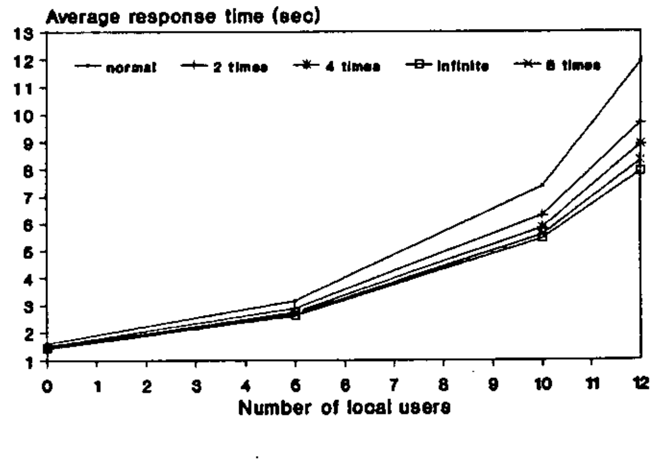


Figure C.2.12 : 1856Kbytes

The effect of the better CPU on the average response time in the Sun SPARCstation 10 workstation.

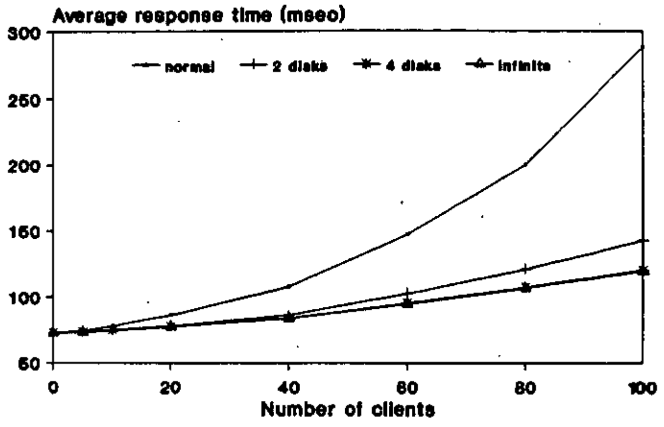


Figure C.3.1 : 8Kbytes

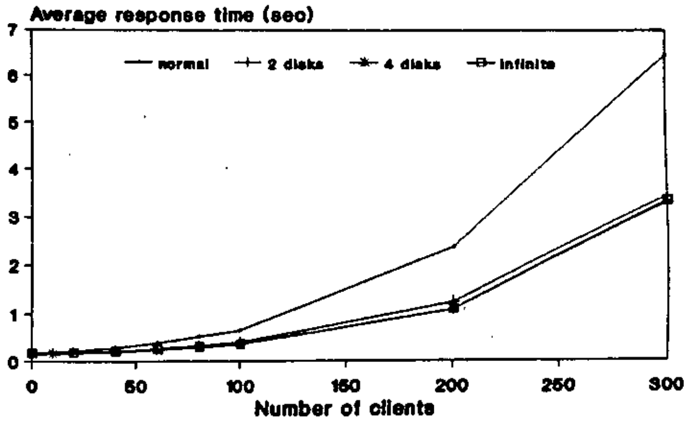


Figure C.3.2 : 47Kbytes

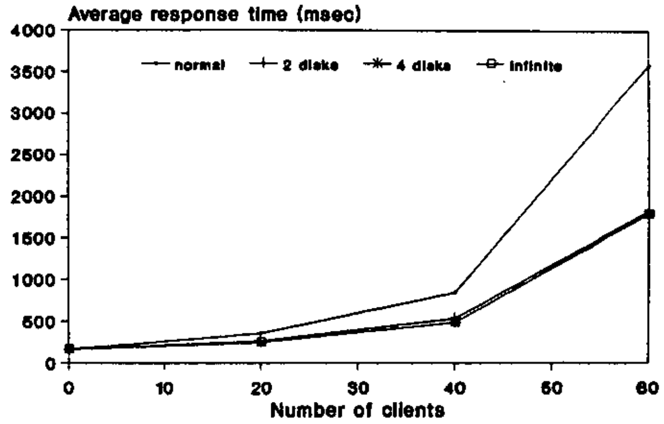


Figure C.3.3 : 50.7Kbytes

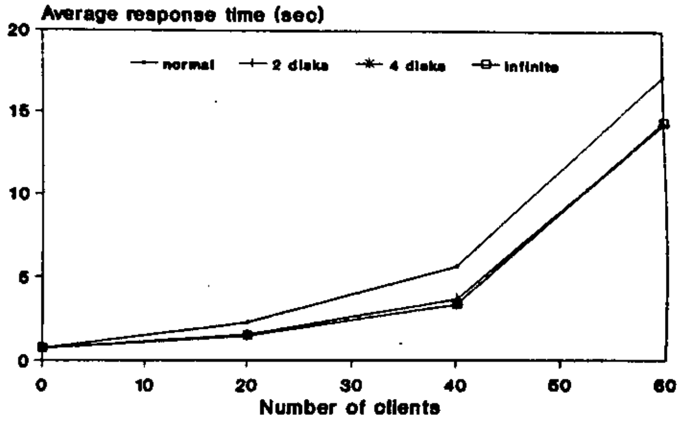


Figure C.3.4 : 316Kbytes(B)

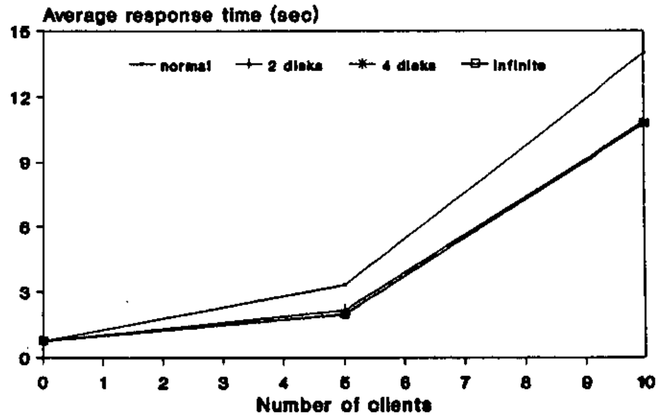


Figure C.3.5 : 316Kbytes

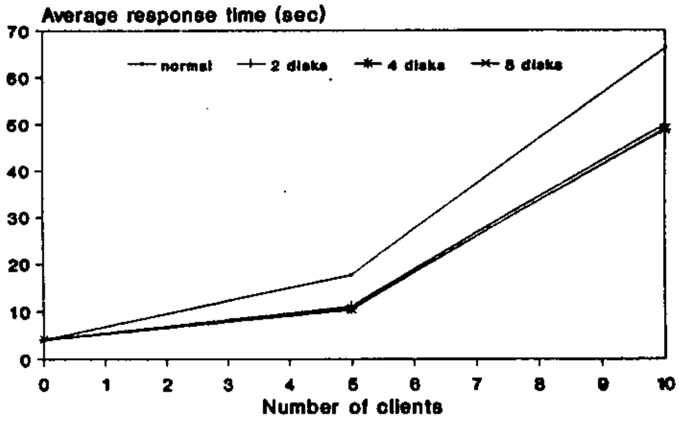


Figure C.3.6 : 1856Kbytes

The effect on the average response time of having multiple disk I/O subsystems in the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations.

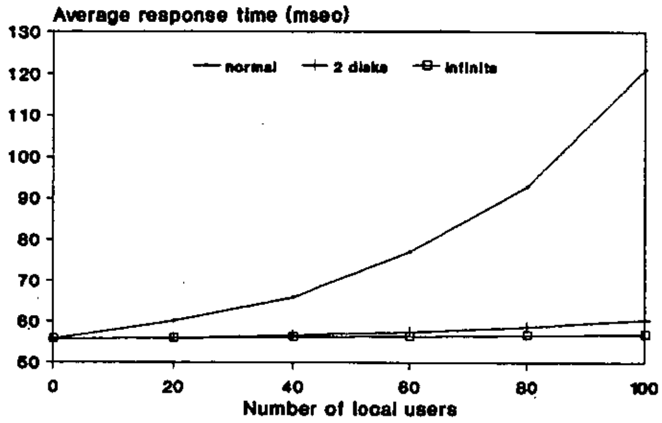


Figure C.3.7 : 8Kbytes

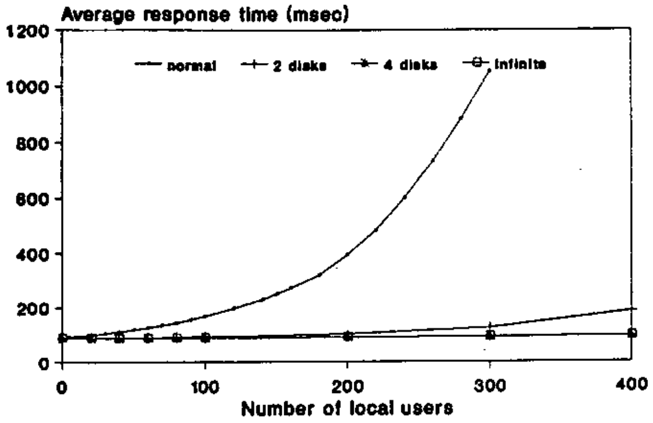


Figure C.3.8 : 47Kbytes

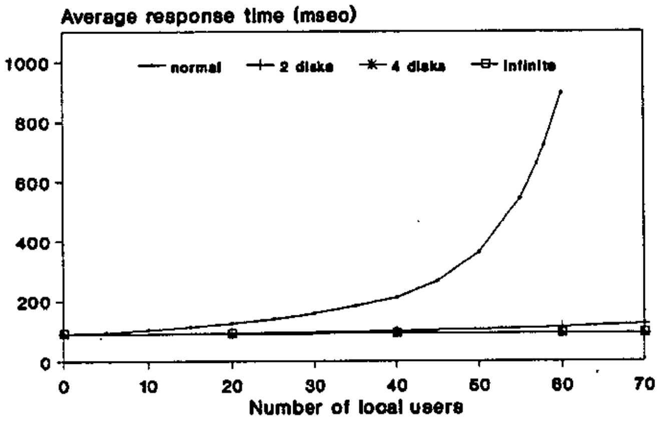


Figure C.3.9 : 50.7Kbytes

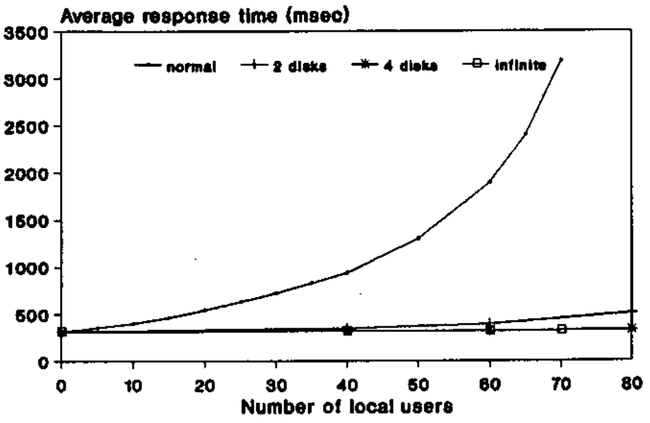


Figure C.3.10 : 316Kbytes(B)

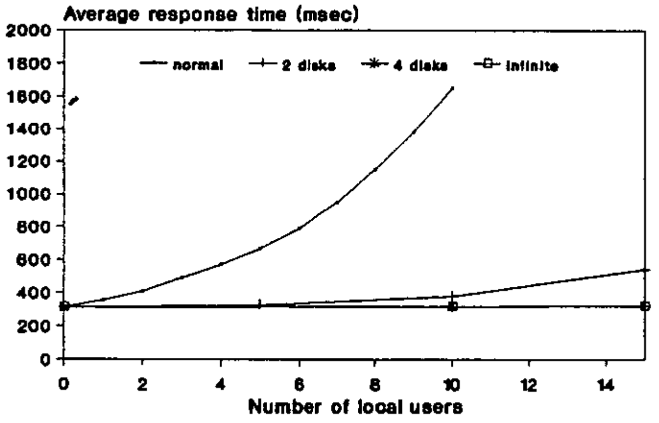


Figure C.3.11 : 316Kbytes

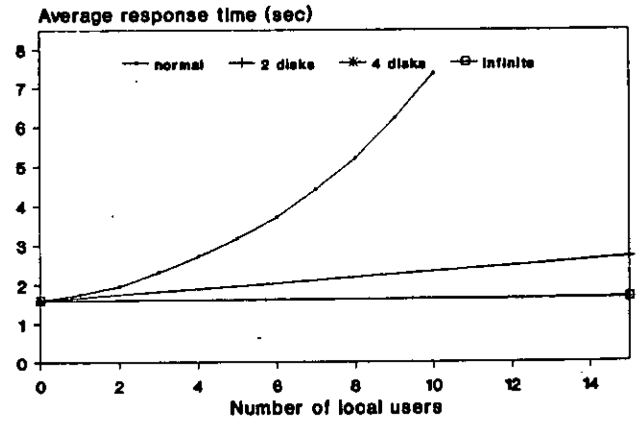


Figure C.3.12 : 1856Kbytes

The effect on the average response time of having multiple disk I/O subsystems in the Sun SPARCstation 10 workstation.

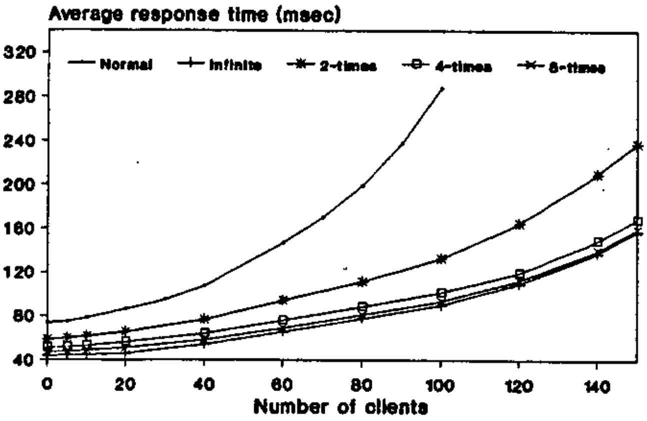


Figure C.4.1.1 : 8Kbytes

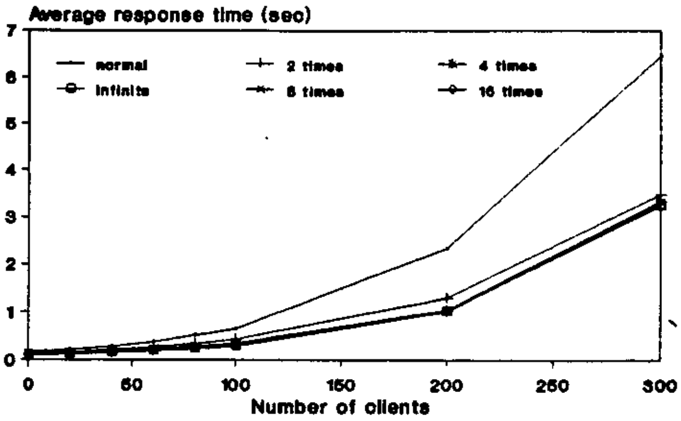


Figure C.4.1.2 : 47Kbytes

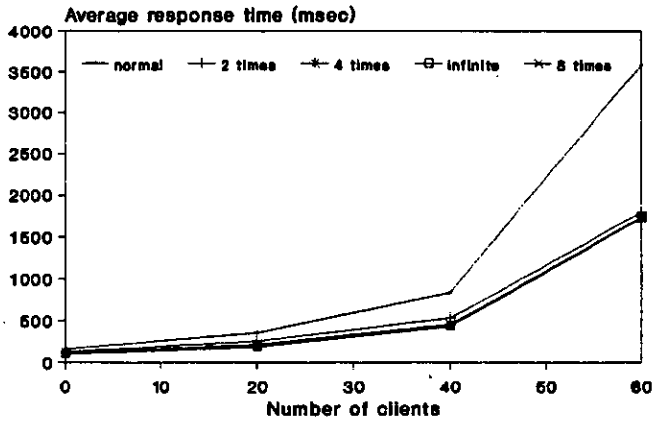


Figure C.4.1.3 : 50.7Kbytes

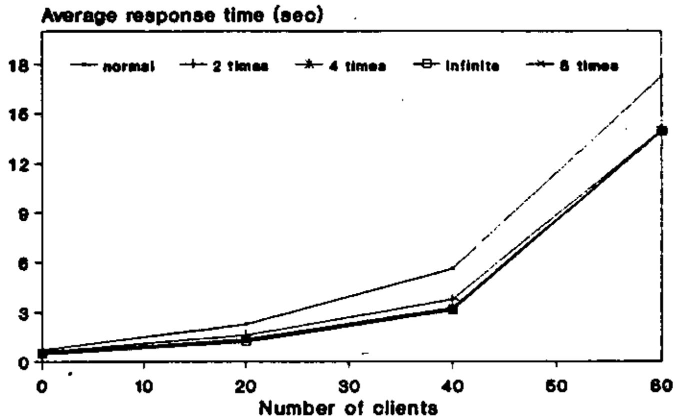


Figure C.4.1.4 : 316Kbytes(B)

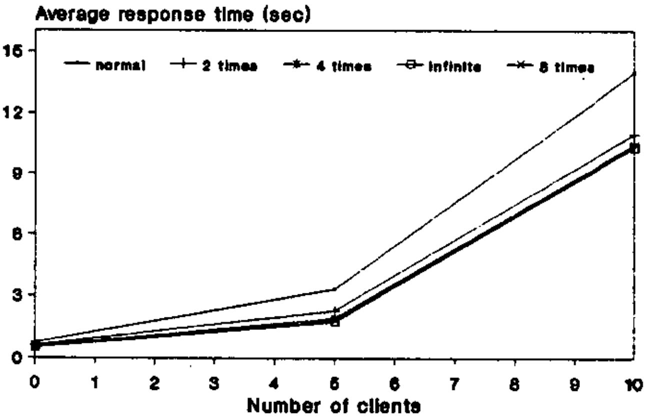


Figure C.4.1.5 : 316Kbytes

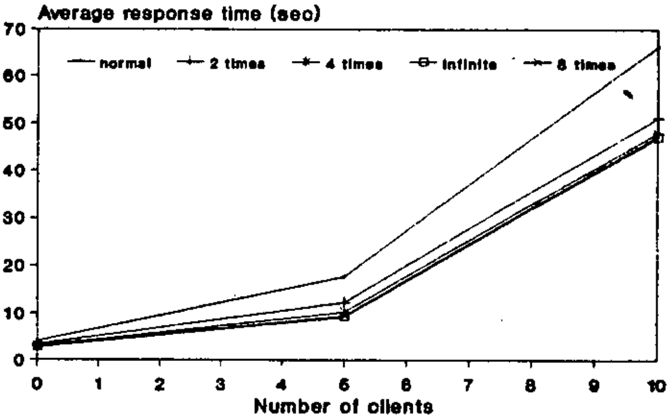


Figure C.4.1.6 : 1856Kbytes

The effect of having the better disk I/O subsystem on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

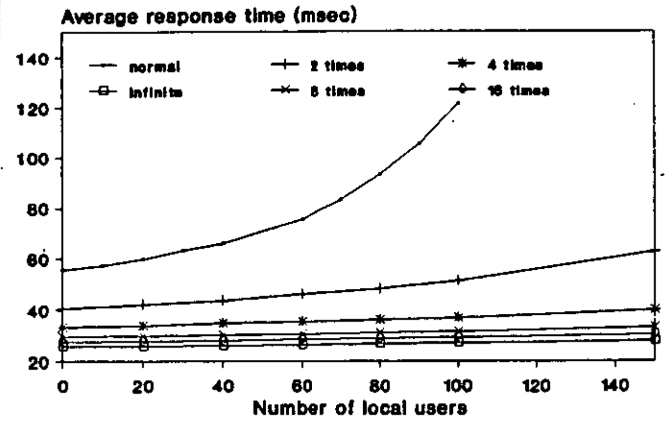


Figure C.4.1.7 : 8Kbytes

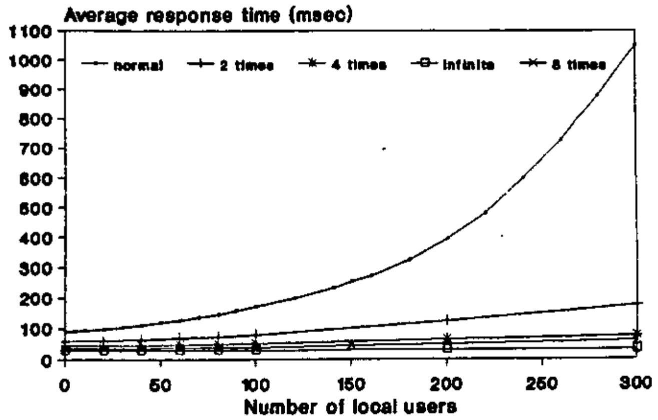


Figure C.4.1.8 : 47Kbytes

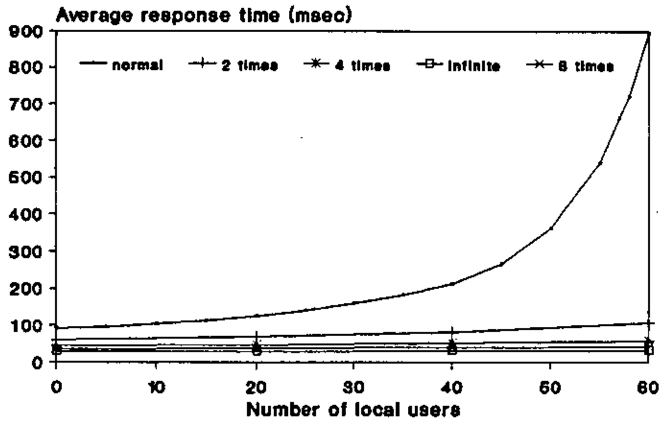


Figure C.4.1.9 : 50.7Kbytes

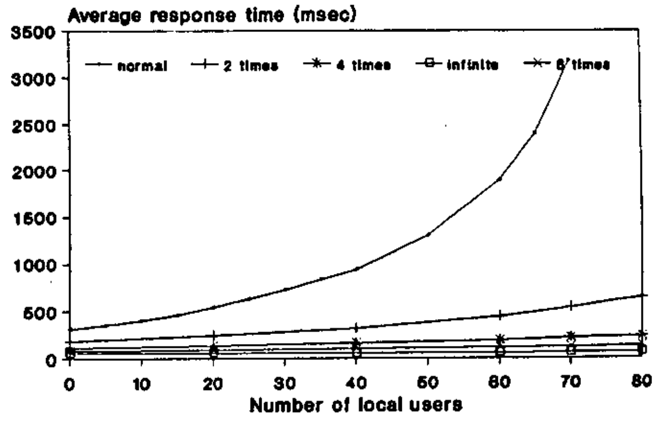


Figure C.4.1.10 : 316Kbytes(B)

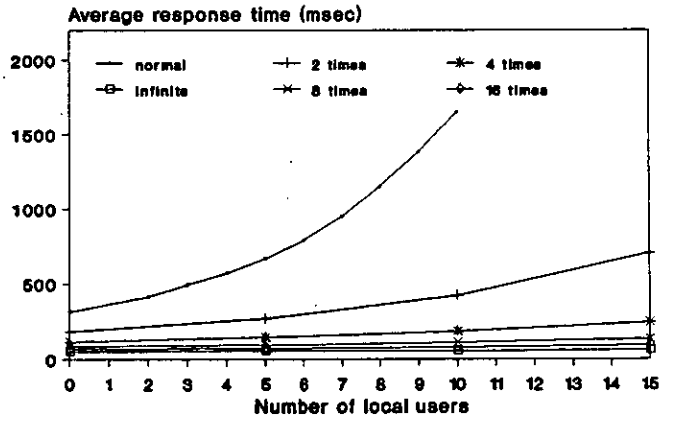


Figure C.4.1.11 : 316Kbytes

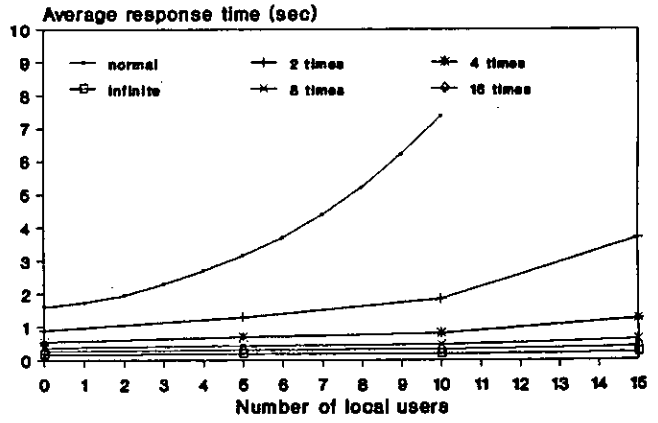


Figure C.4.1.12 : 1856Kbytes

The effect of having the better disk I/O subsystem on the average response time in the Sun SPARCstation 10 workstation.

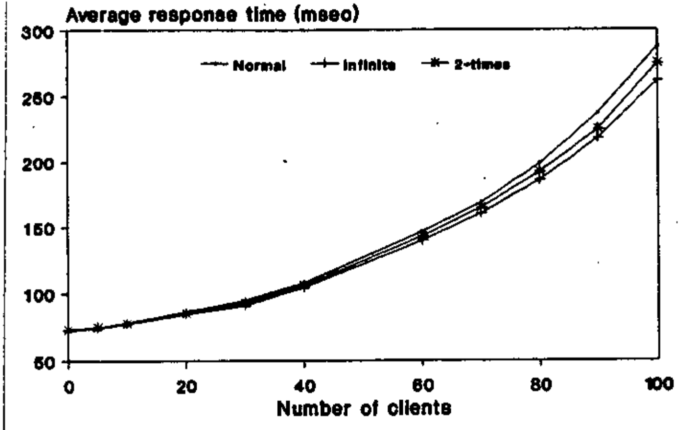


Figure C.4.2.1 : 8Kbytes

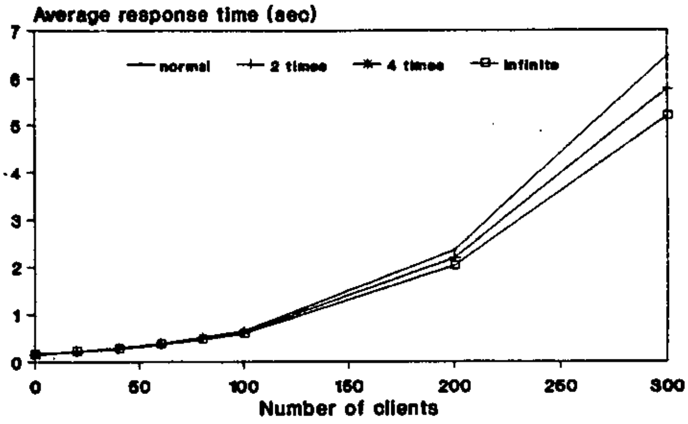


Figure C.4.2.2 : 47Kbytes

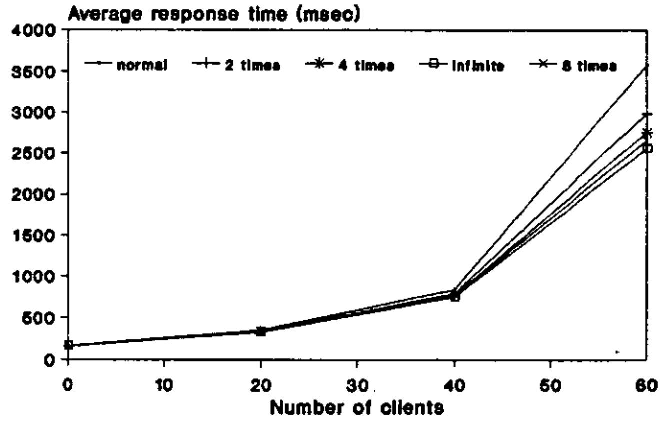


Figure C.4.2.3 : 50.7Kbytes

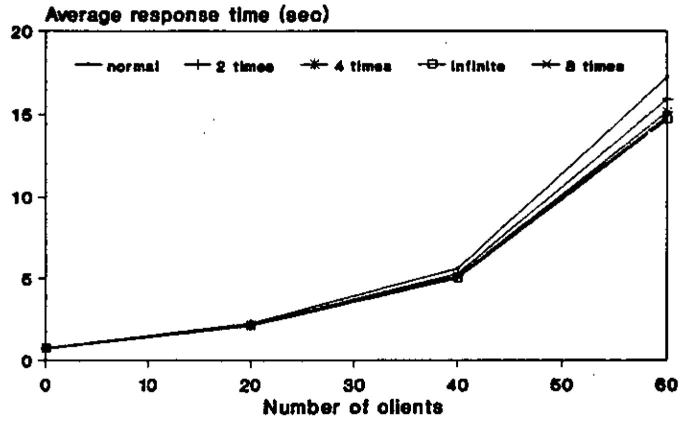


Figure C.4.2.4 : 316Kbytes(B)

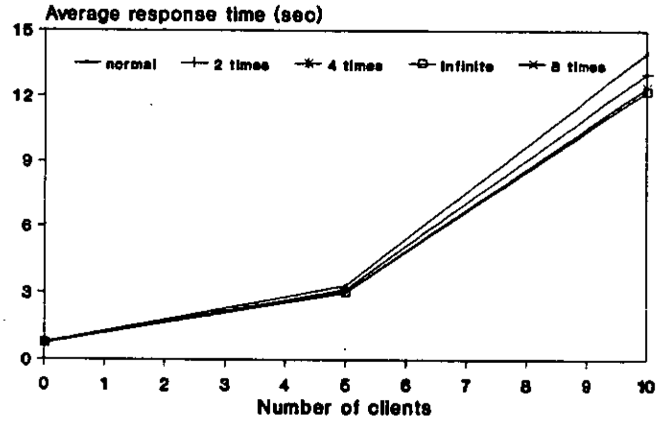


Figure C.4.2.5 : 316Kbytes

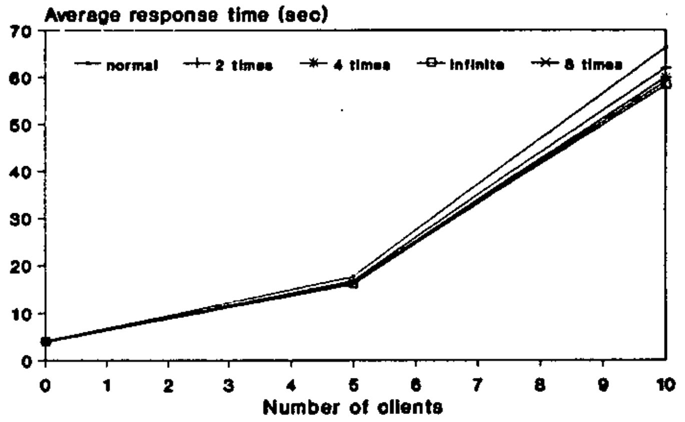


Figure C.4.2.6 : 1856Kbytes

The effect of the improved CPU service time for disk I/O on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

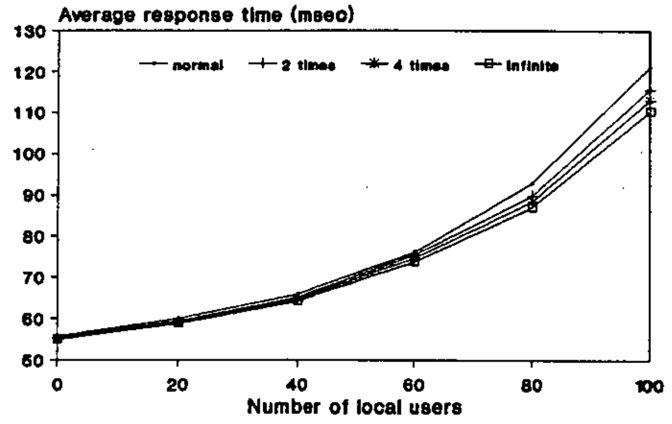


Figure C.4.2.7 : 8Kbytes

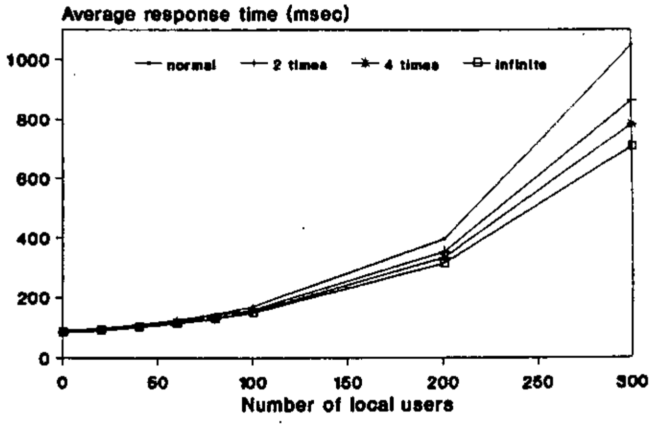


Figure C.4.2.8 : 47Kbytes

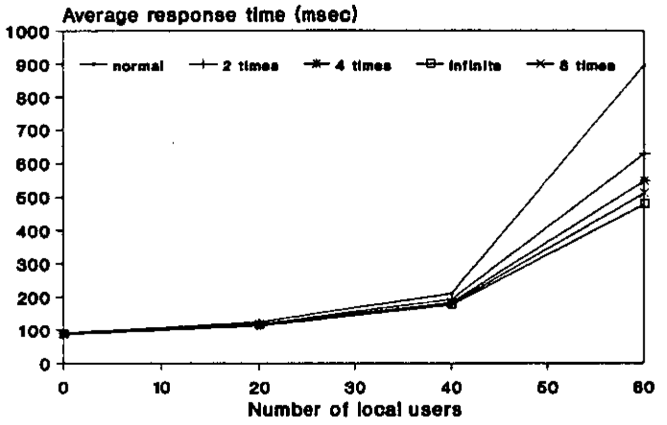


Figure C.4.2.9 : 50.7Kbytes

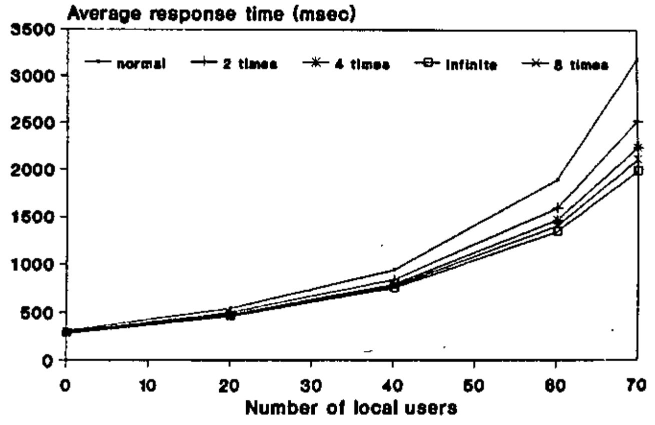


Figure C.4.2.10 : 316Kbytes(B)

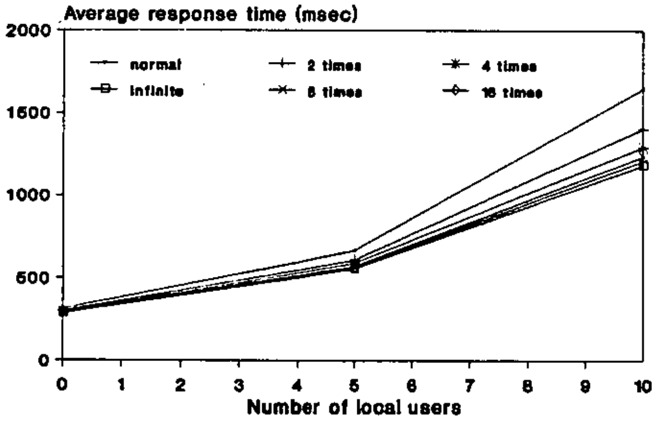


Figure C.4.2.11 : 316Kbytes

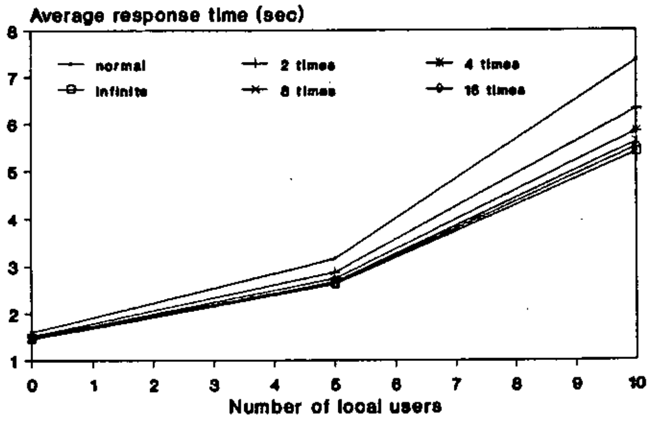


Figure C.4.2.12 : 1856Kbytes

The effect of the improved CPU service time for disk I/O on the average response time in the Sun SPARCstation 10 workstation.

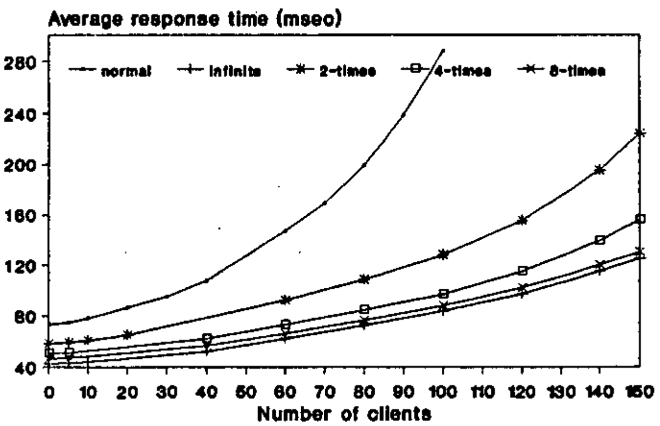


Figure C.4.3.1 : 8Kbytes

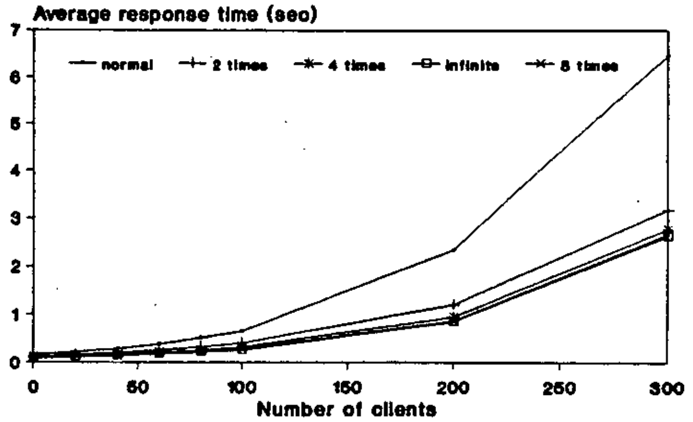


Figure C.4.3.2 : 47Kbytes

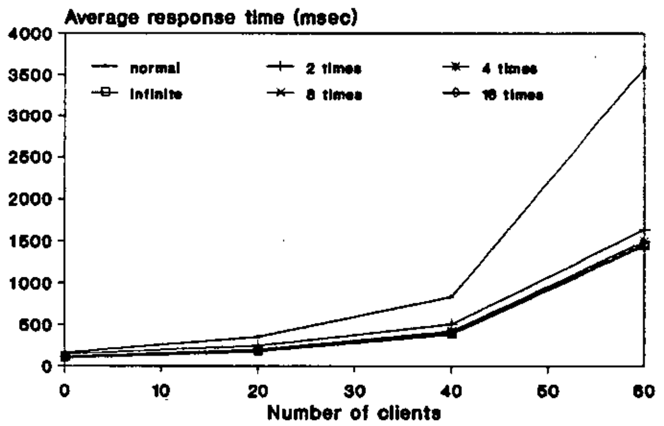


Figure C.4.3.3 : 50.7Kbytes

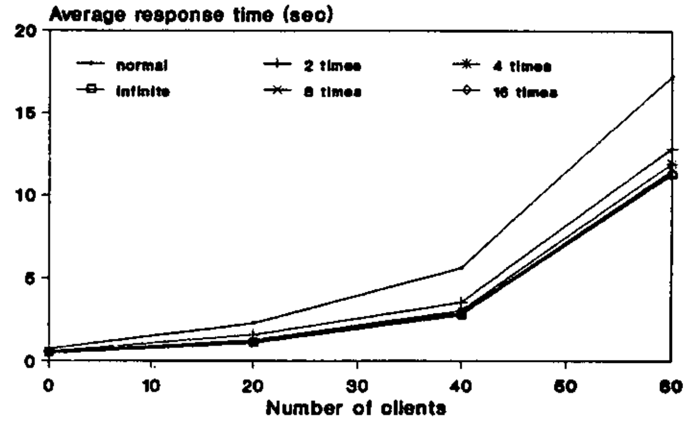


Figure C.4.3.4 : 316Kbytes(B)

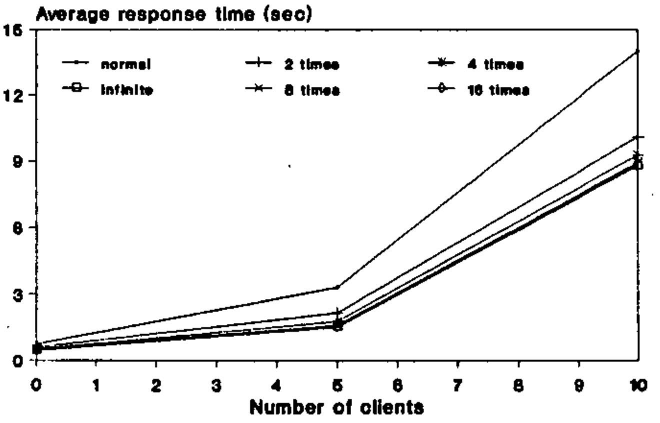


Figure C.4.3.5 : 316Kbytes

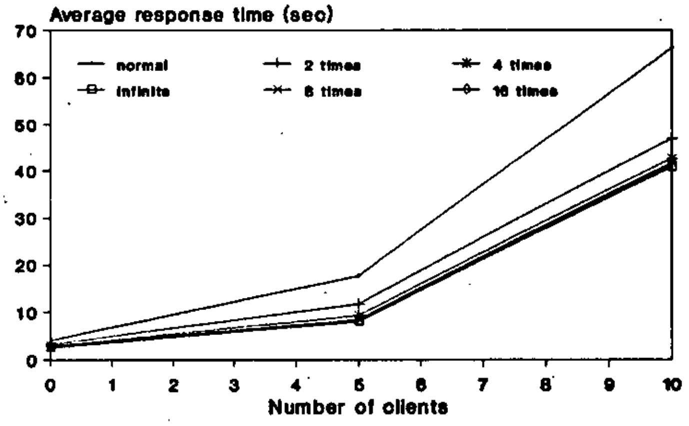


Figure C.4.3.6 : 1856Kbytes

The effect on the average response time when the values of all parameters for disk I/O are improved at the same time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

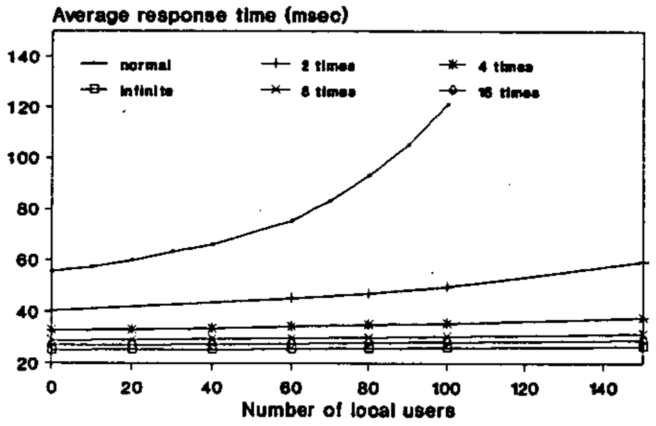


Figure C.4.3.7 : 8Kbytes

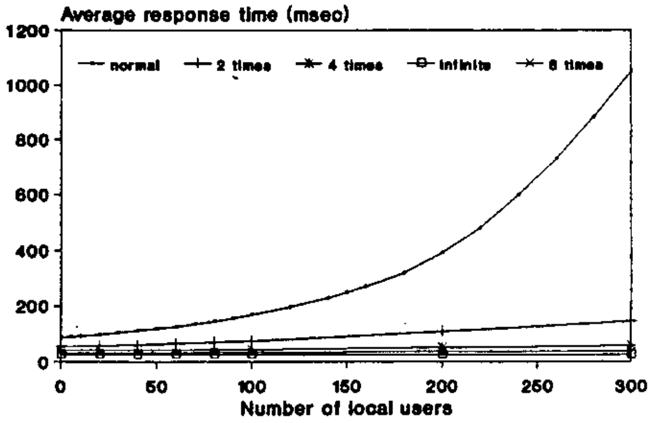


Figure C.4.3.8 : 47Kbytes

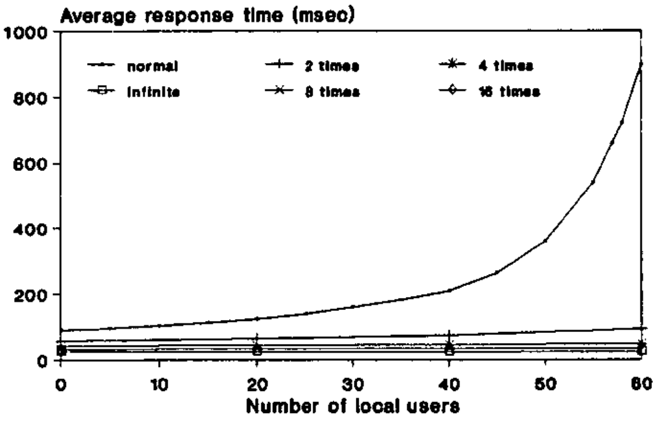


Figure C.4.3.9 : 50.7Kbytes

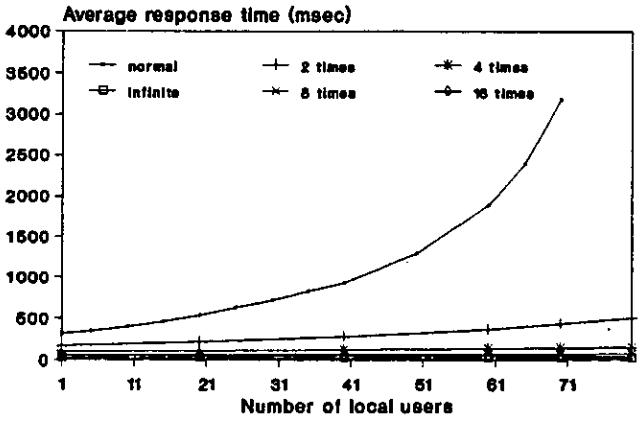


Figure C.4.3.10 : 316Kbytes(B)

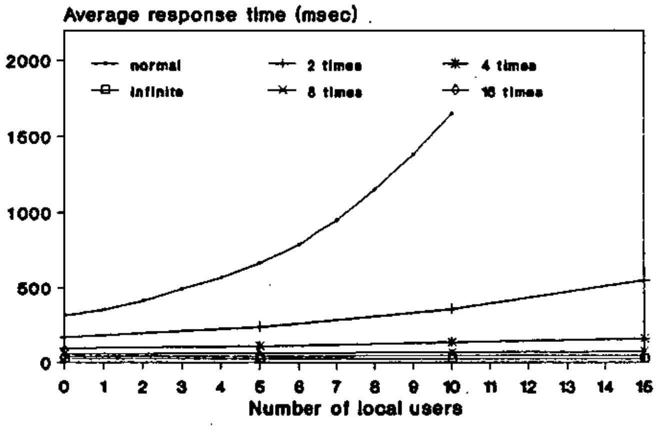


Figure C.4.3.11 : 316Kbytes

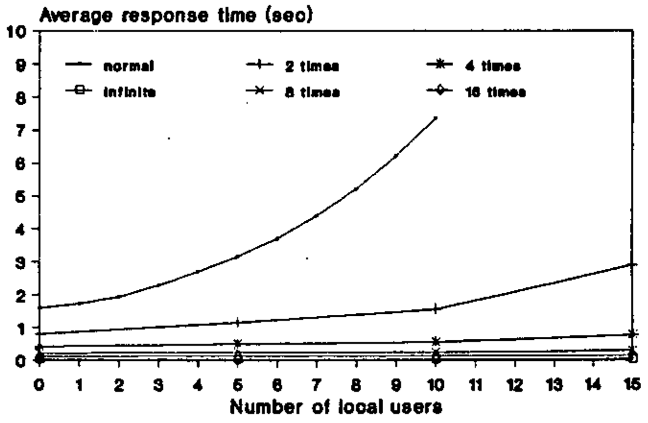


Figure C.4.3.12 : 1856Kbytes

The effect on the average response time when the values of all parameters for disk I/O are improved at the same time in the Sun SPARCstation 10 workstation.

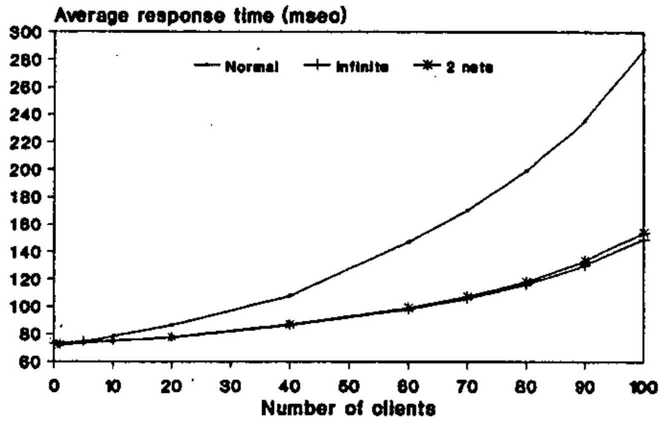


Figure C.5.1 : 8Kbytes

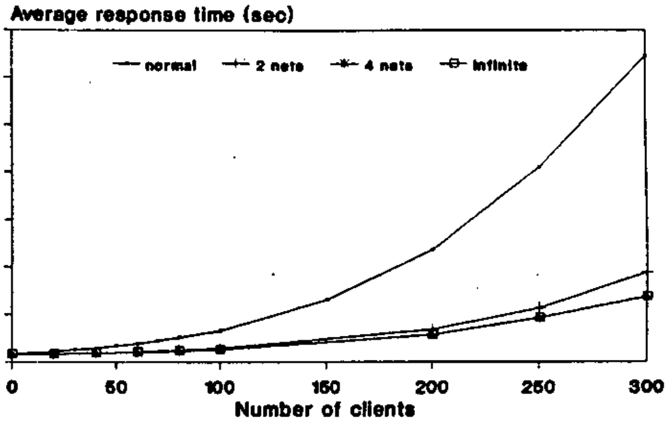


Figure C.5.2 : 47Kbytes

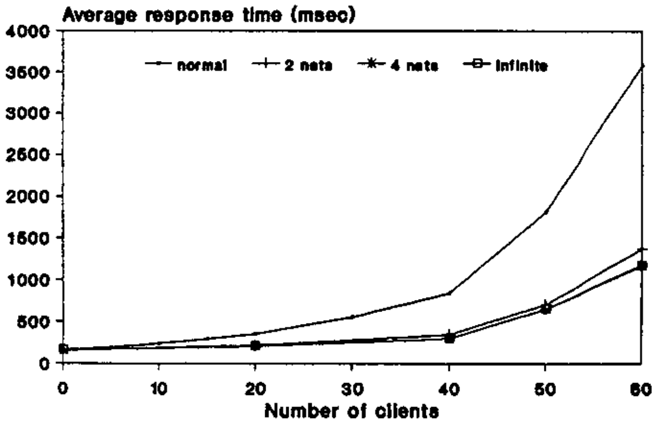


Figure C.5.3 : 50.7Kbytes

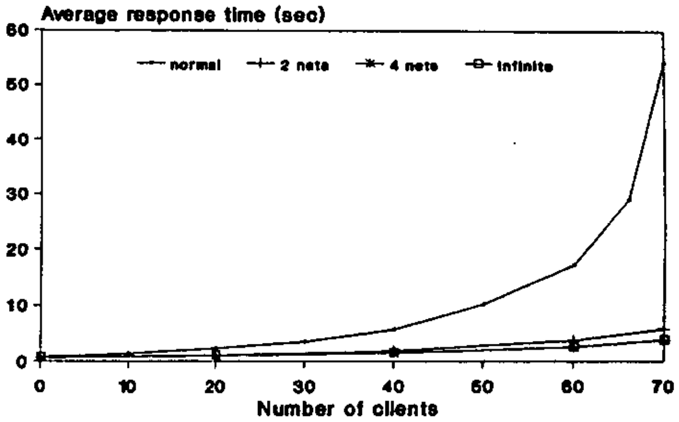


Figure C.5.4 : 316Kbytes(B)

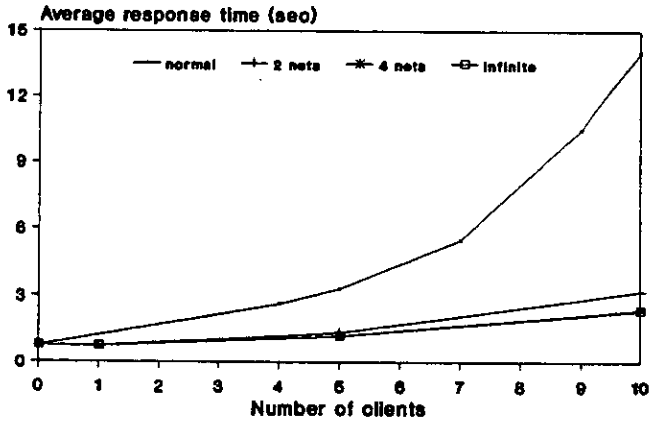


Figure C.5.5 : 316Kbytes

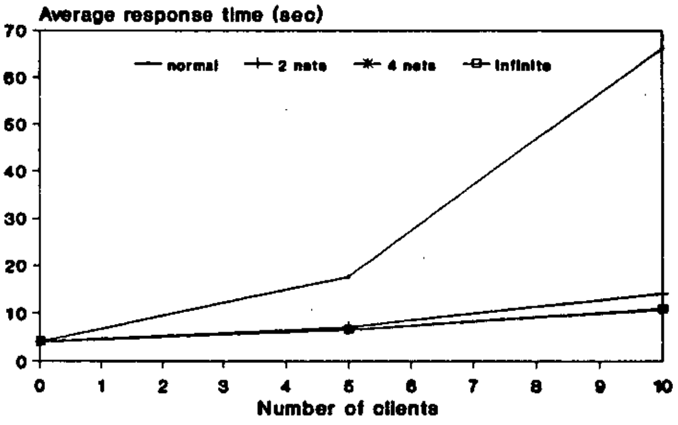


Figure C.5.6 : 1856Kbytes

The effect on the average response time of having multiple networks and multiple network interface units in the distributed file system which consists of the Sun SPARCstation 10 workstations.

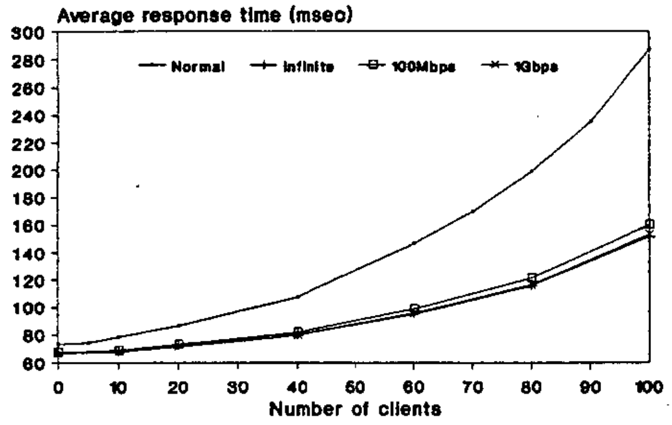


Figure C.6.1.1 : 8Kbytes

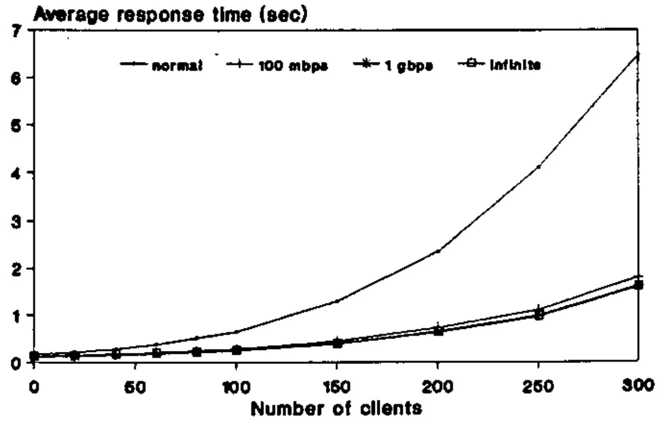


Figure C.6.1.2 : 47Kbytes

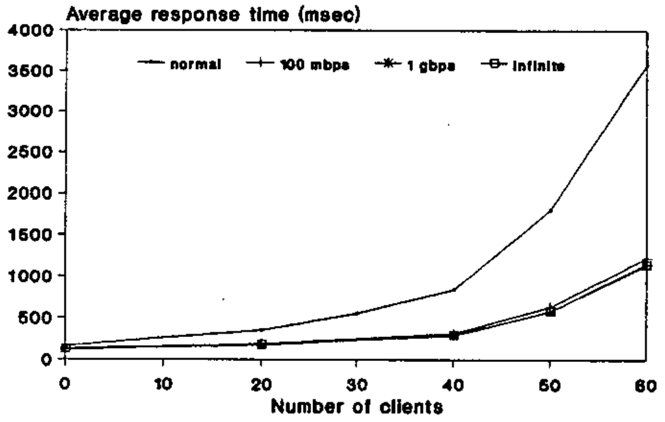


Figure C.6.1.3 : 50.7Kbytes

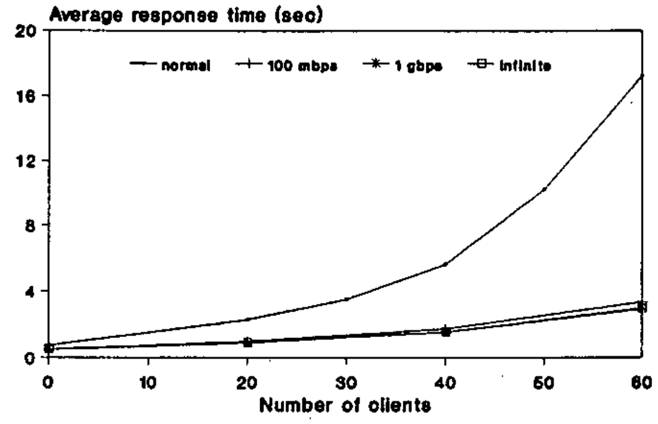


Figure C.6.1.4 : 316Kbytes(B)

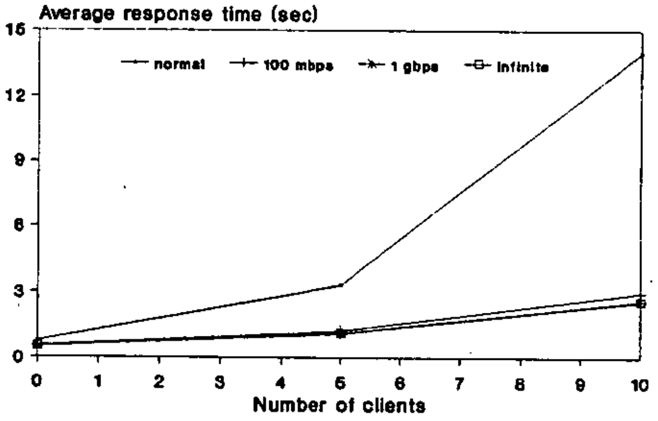


Figure C.6.1.5 : 316Kbytes

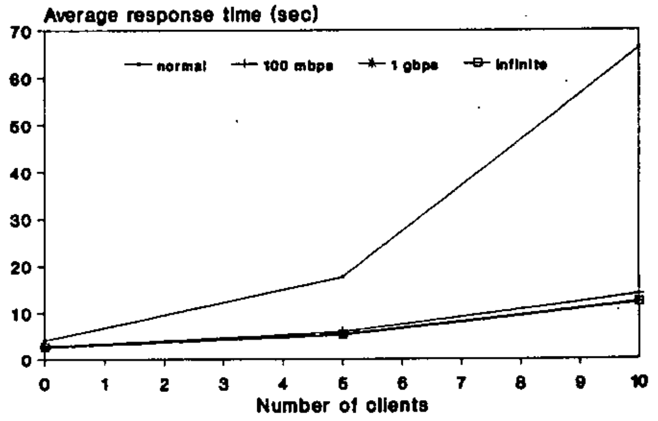


Figure C.6.1.6 : 1856Kbytes

The effect of having the faster network on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

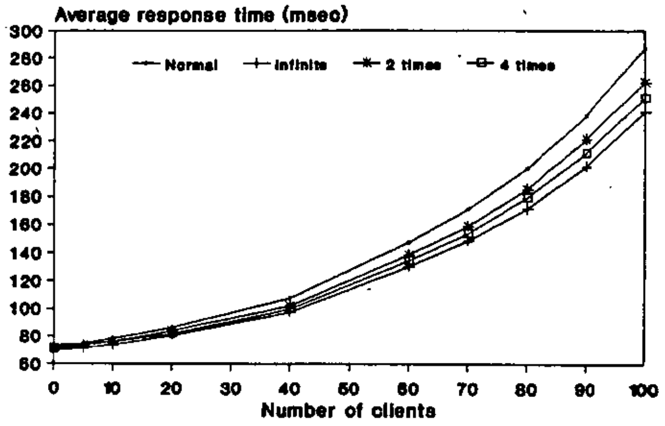


Figure C.6.2.1 : 8Kbytes

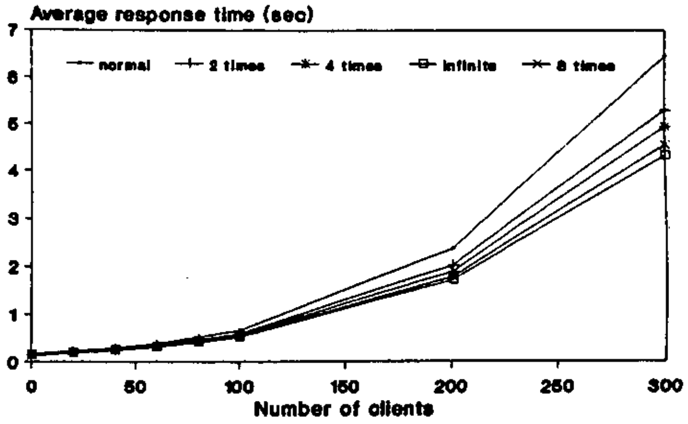


Figure C.6.2.2 : 47Kbytes

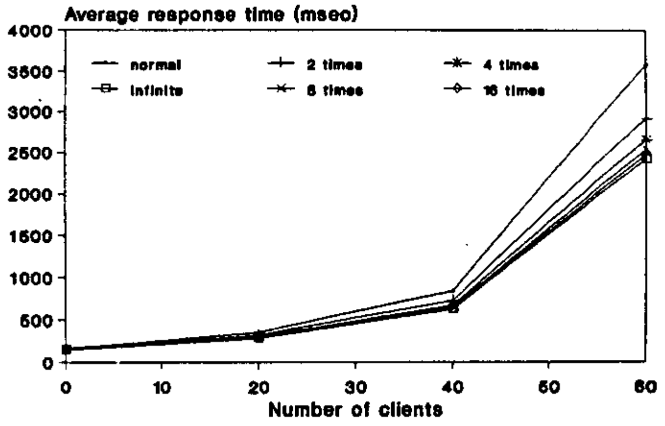


Figure C.6.2.3 : 50.7Kbytes

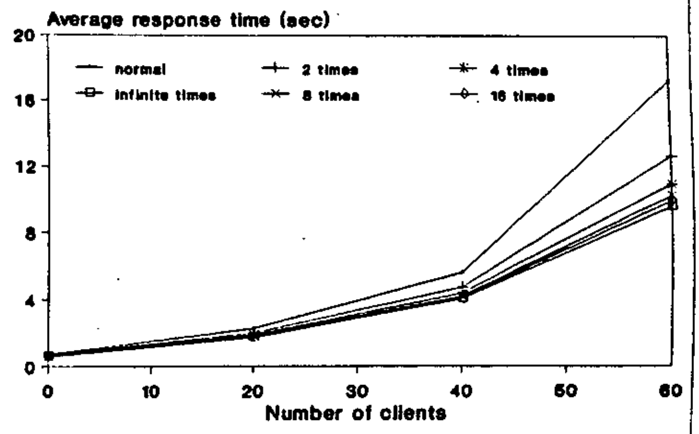


Figure C.6.2.4 : 316Kbytes(B)

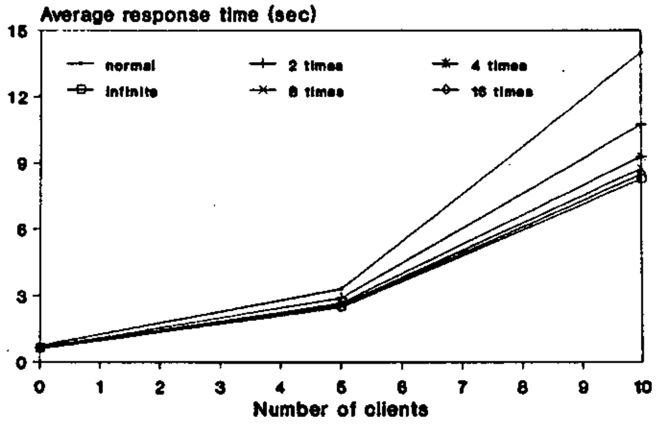


Figure C.6.2.5 : 316Kbytes

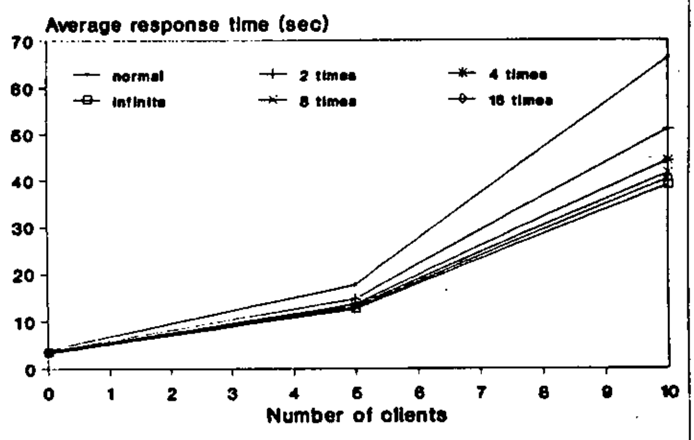


Figure C.6.2.6 : 1856Kbytes

The effect on the average response time of having the better network interface unit in the distributed file system which consists of the Sun SPARCstation 10 workstations.

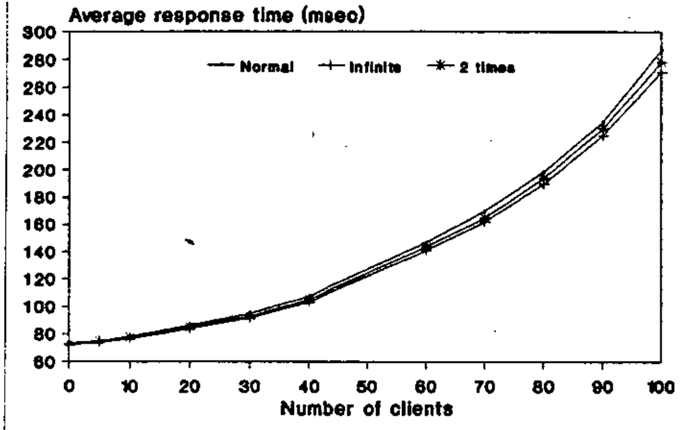


Figure C.6.3.1 : 8Kbytes

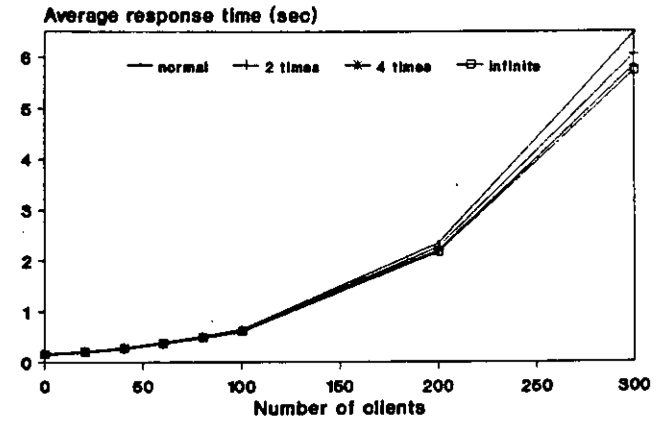


Figure C.6.3.2 : 47Kbytes

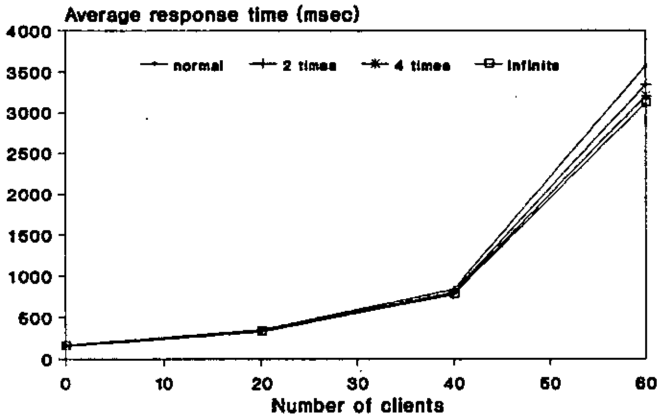


Figure C.6.3.3 : 50.7Kbytes

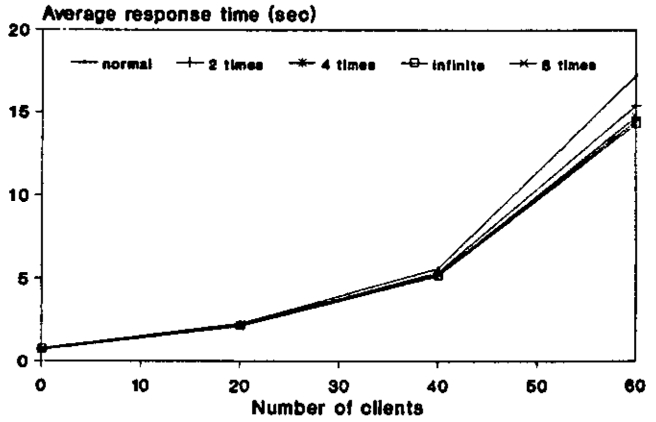


Figure C.6.3.4 : 316Kbytes(B)

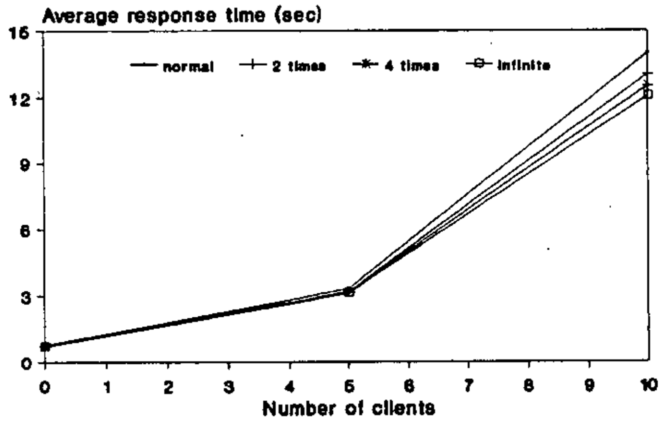


Figure C.6.3.5 : 316Kbytes

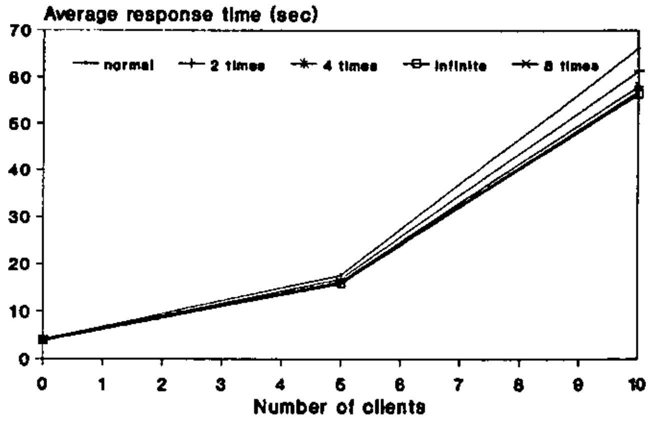


Figure C.6.3.6 : 1856Kbytes

The effect of having the better communication mechanism on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

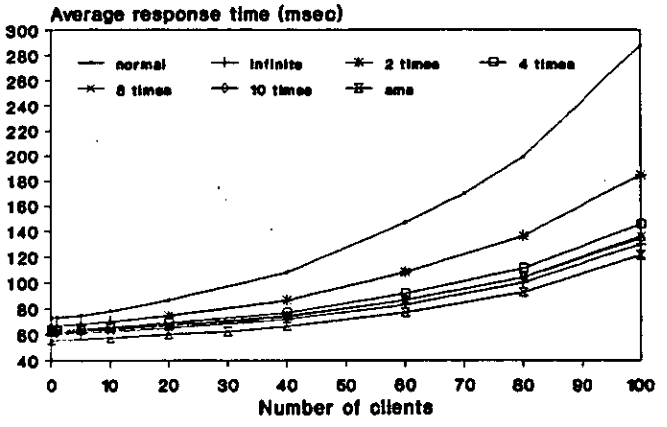


Figure C.6.4.1 : 8Kbytes

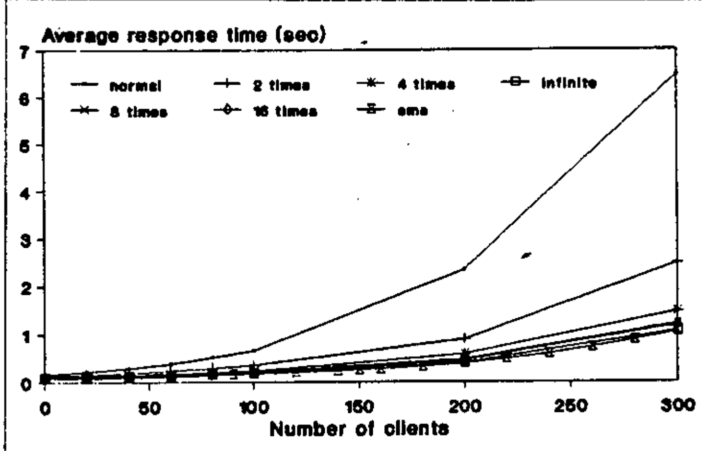


Figure C.6.4.2 : 47Kbytes

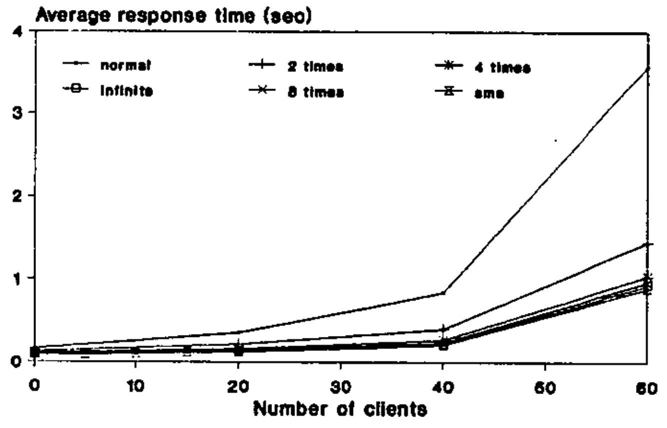


Figure C.6.4.3 : 50.7Kbytes

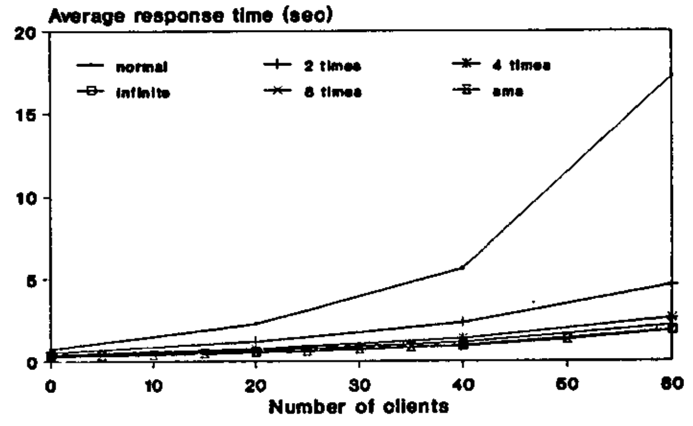


Figure C.6.4.4 : 316Kbytes(B)

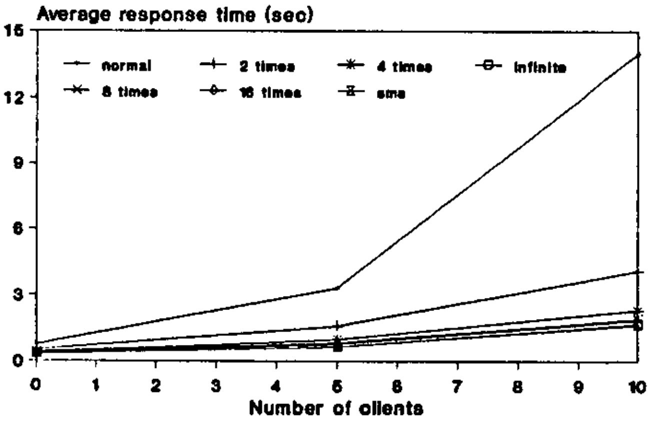


Figure C.6.4.5 : 316Kbytes

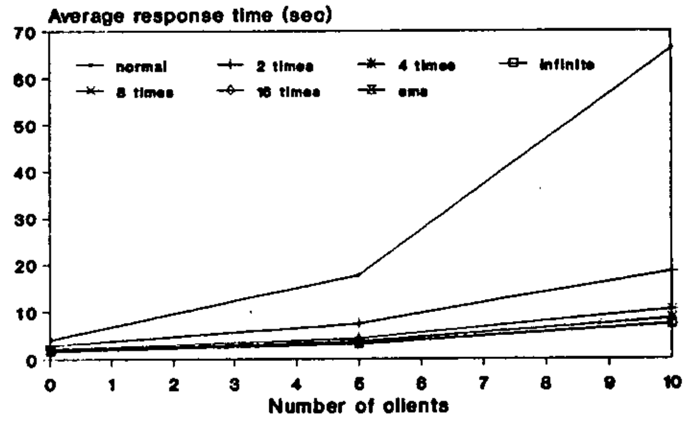


Figure C.6.4.6 : 1856Kbytes

The effect of the better communication on the average response time in the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations.

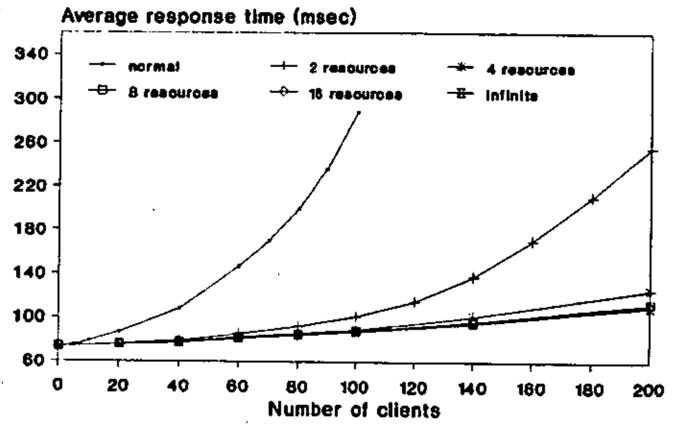


Figure C.7.1 : 8Kbytes

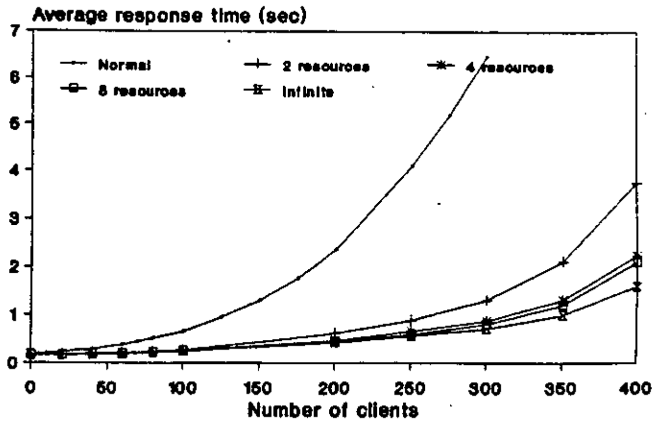


Figure C.7.2 : 47Kbytes

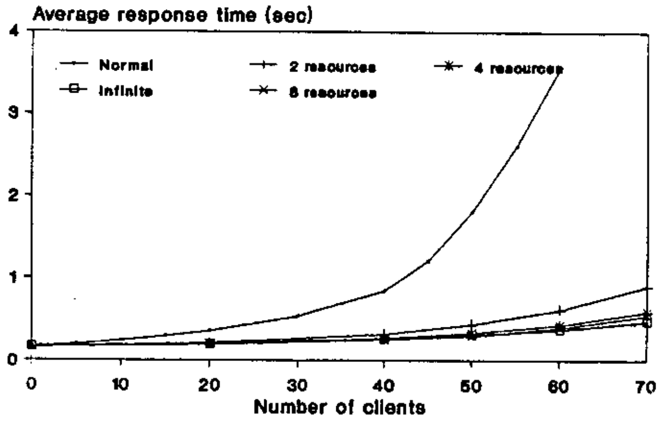


Figure C.7.3 : 50.7Kbytes

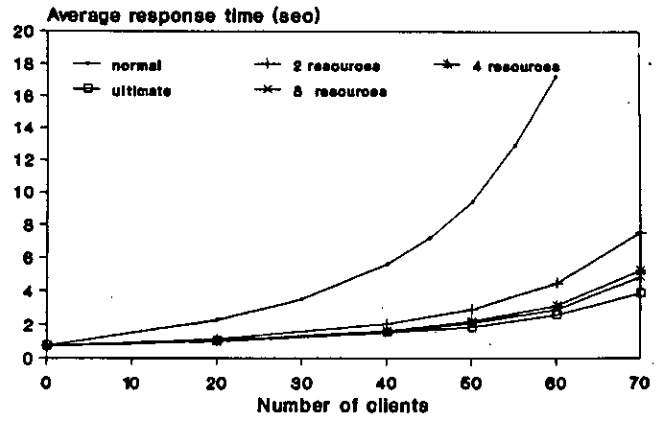


Figure C.7.4 : 316Kbytes(B)

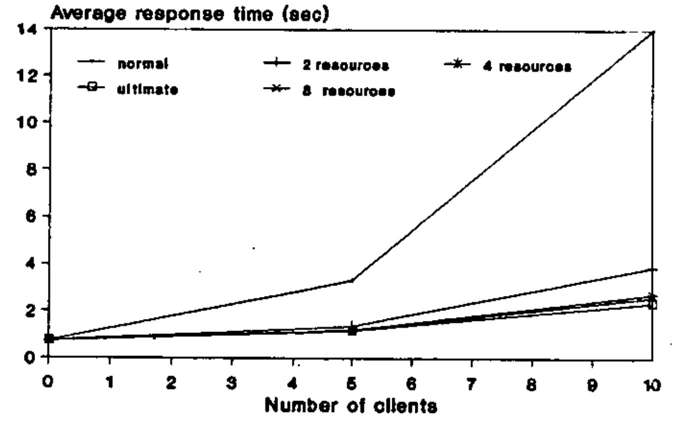


Figure C.7.5 : 316Kbytes

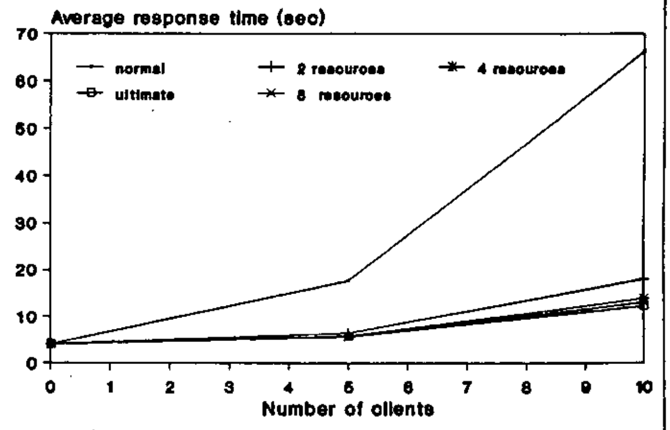


Figure C.7.6 : 1856Kbytes

The effect on the average response time of having multiple resources in the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations.

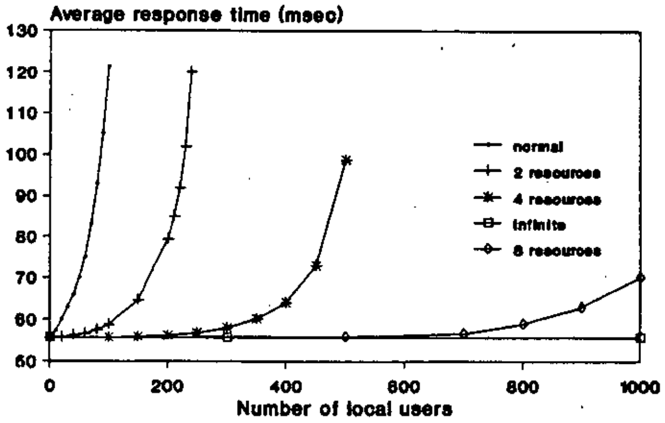


Figure C.7.7 : 8Kbytes

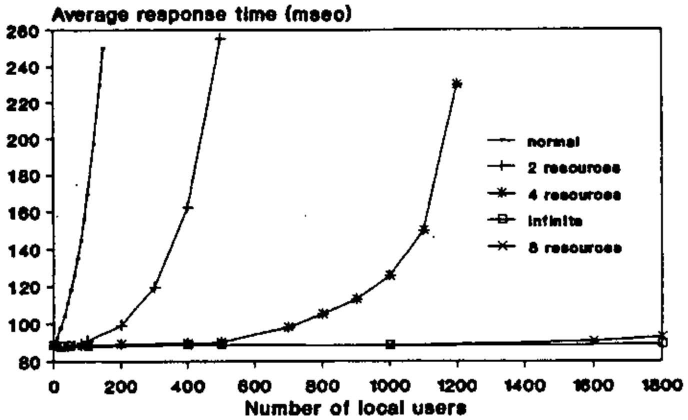


Figure C.7.8 : 47Kbytes

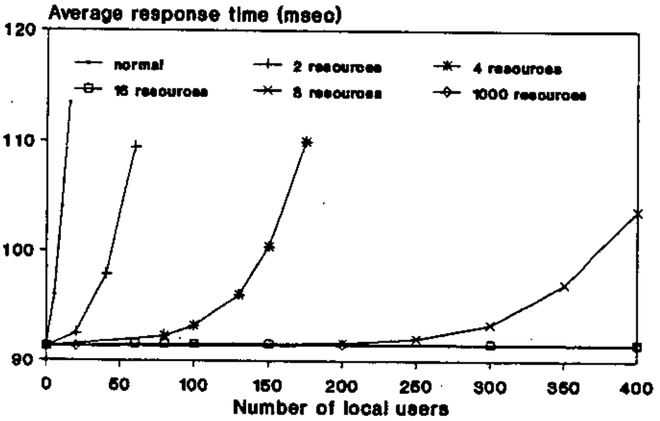


Figure C.7.9 : 50.7Kbytes

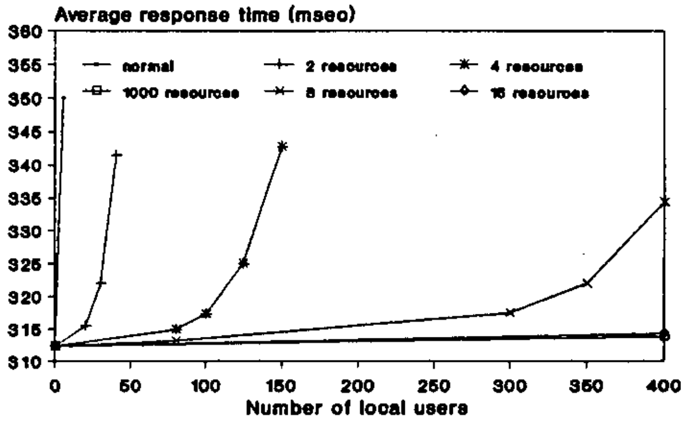


Figure C.7.10 : 316Kbytes(B)

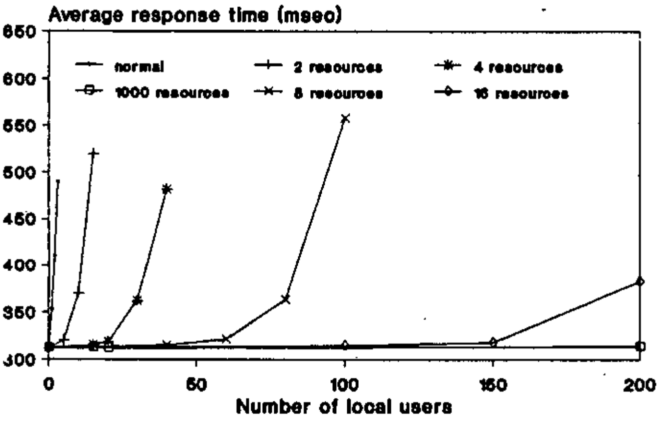


Figure C.7.11 : 316Kbytes

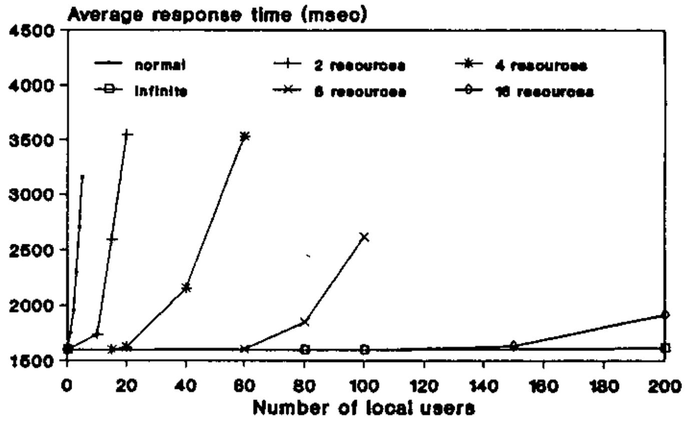


Figure C.7.12 : 1856Kbytes

The effect on the average response time of having multiple resources in the Sun SPARCstation 10 workstation.

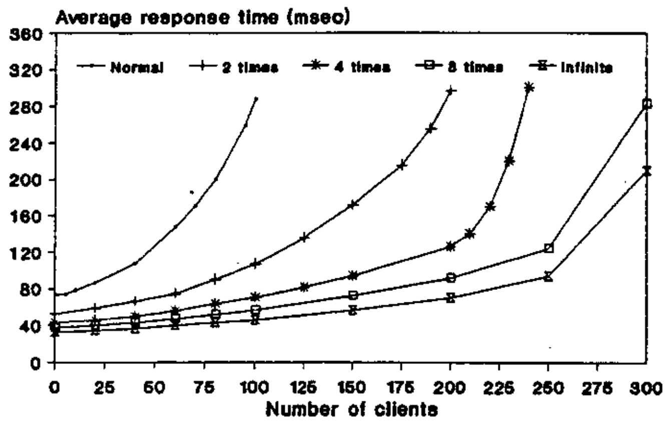


Figure C.8.1 : 8Kbytes

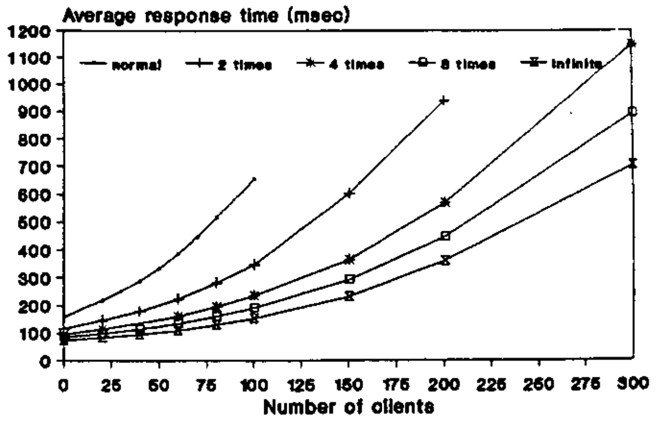


Figure C.8.2 : 47Kbytes

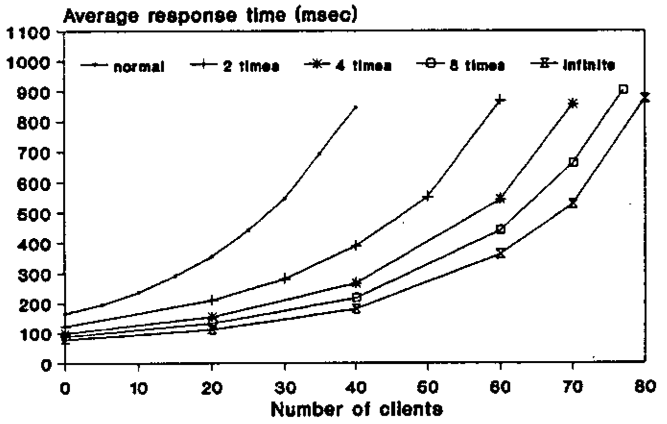


Figure C.8.3 : 50.7Kbytes

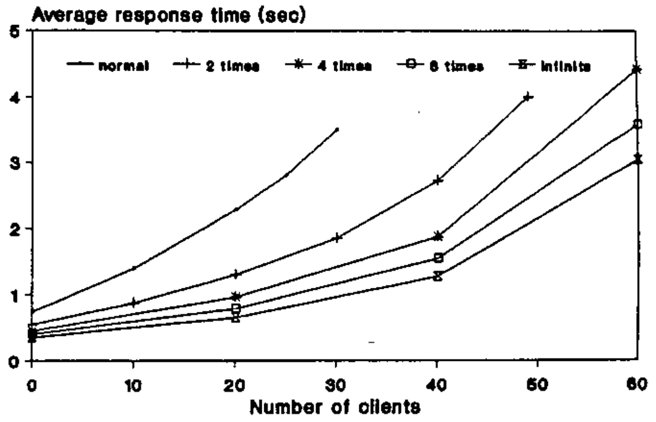


Figure C.8.4 : 316Kbytes(B)

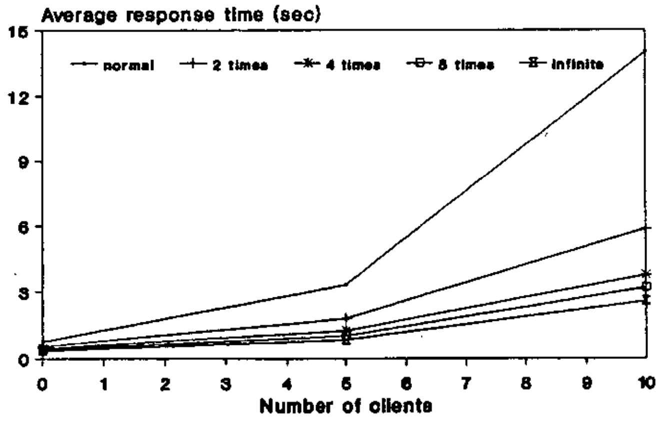


Figure C.8.5 : 316Kbytes

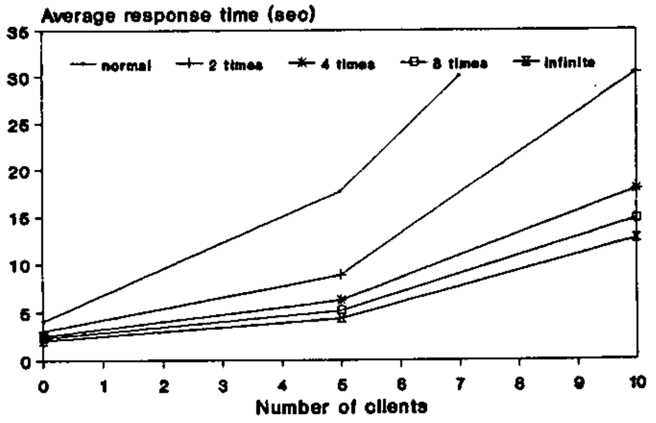


Figure C.8.6 : 1856Kbytes

The effect of the better system on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

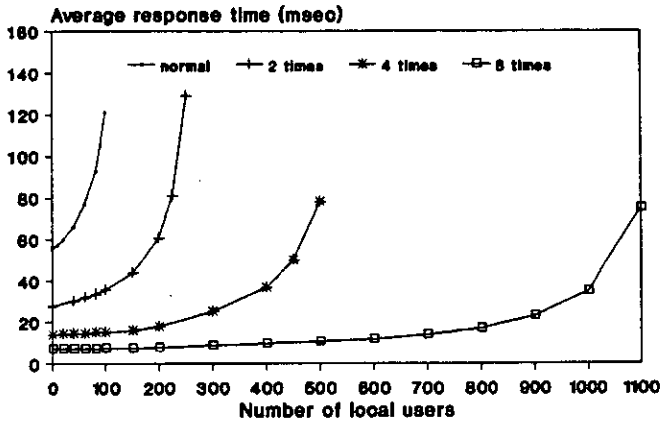


Figure C.8.7 : 8Kbytes

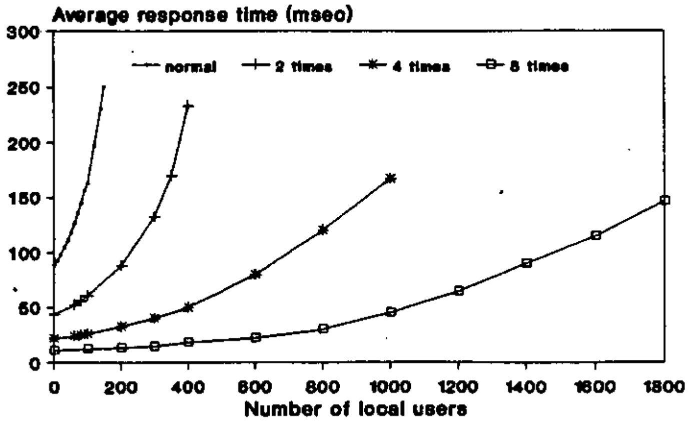


Figure C.8.8 : 47Kbytes

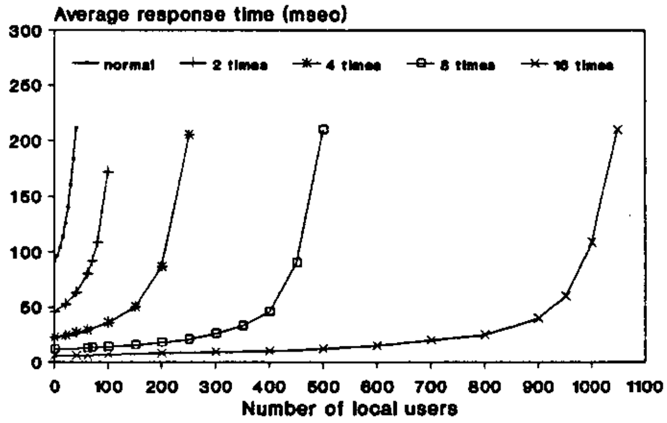


Figure C.8.9 : 50.7Kbytes

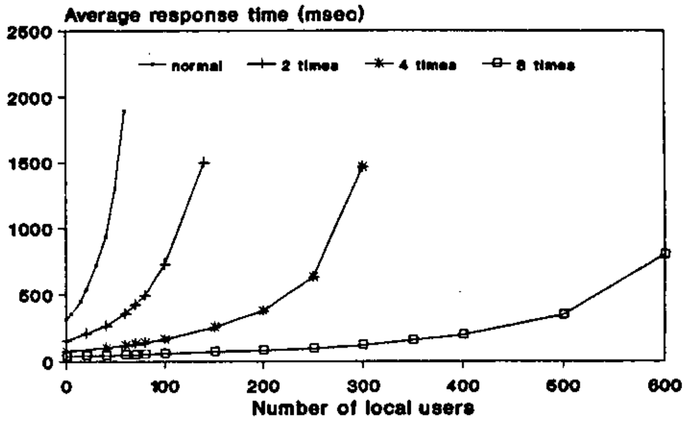


Figure C.8.10 : 316Kbytes(B)

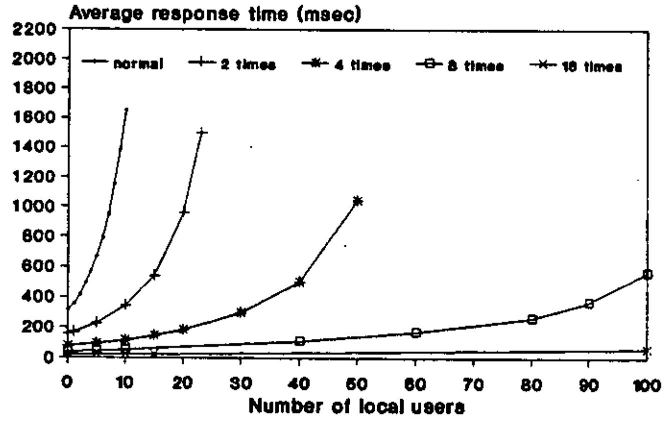


Figure C.8.11 : 316Kbytes

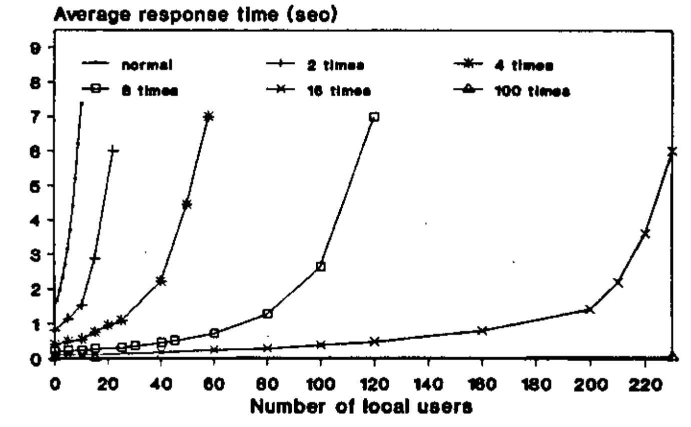


Figure C.8.12 : 1856Kbytes

The effect of the better system on the average response time in the Sun SPARCstation 10 workstation.

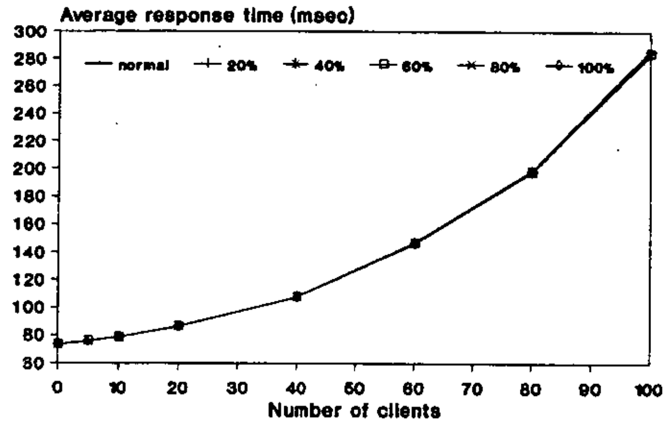


Figure C.9.1.1 : 8Kbytes

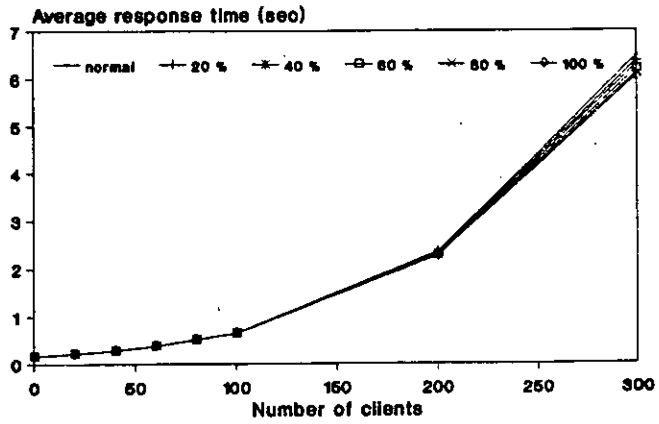


Figure C.9.1.2 : 47Kbytes

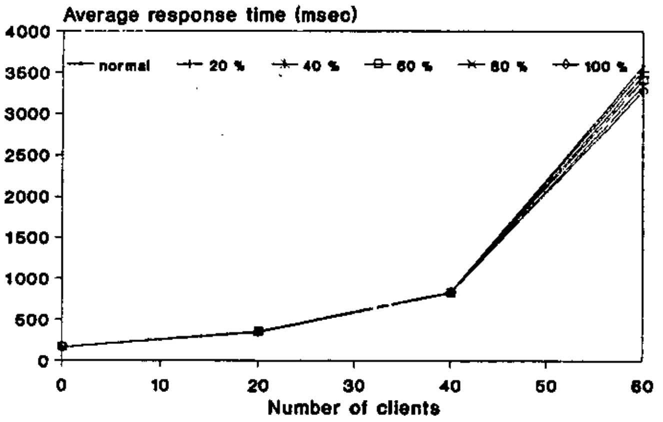


Figure C.9.1.3 : 50.7Kbytes

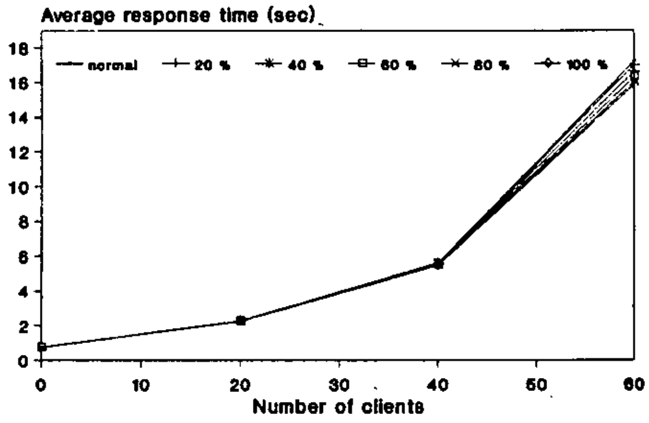


Figure C.9.1.4 : 316Kbytes(B)

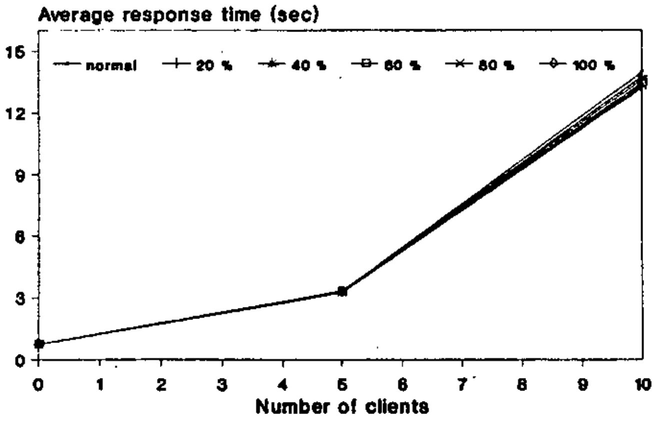


Figure C.9.1.5 : 316Kbytes

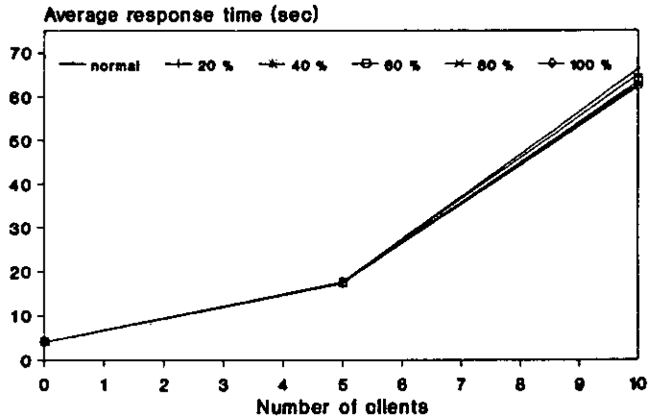


Figure C.9.1.6 : 1856Kbytes

The effect of the improved concurrency during disk I/O operations on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

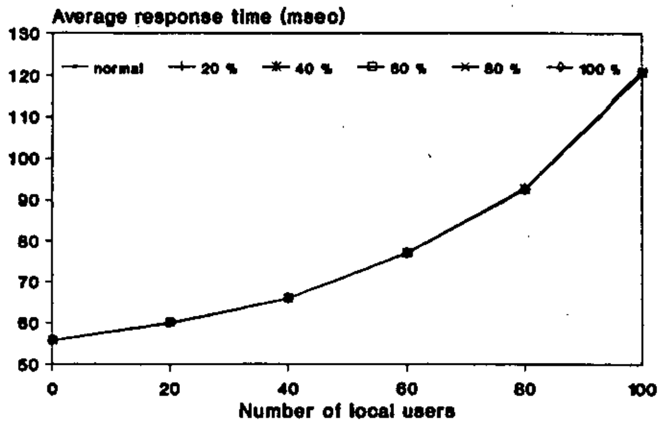


Figure C.9.1.7 : 8Kbytes

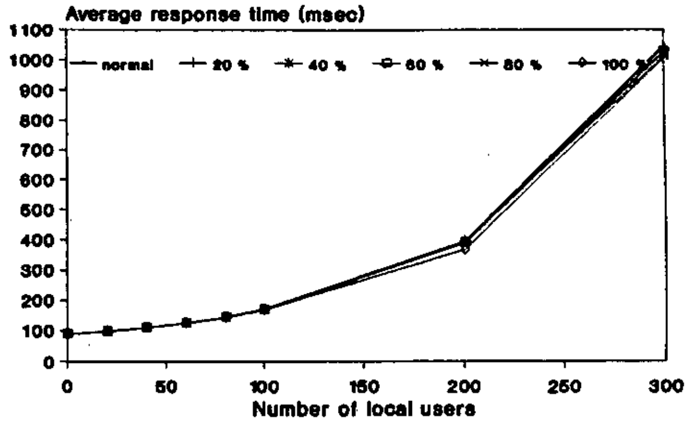


Figure C.9.1.8 : 47Kbytes

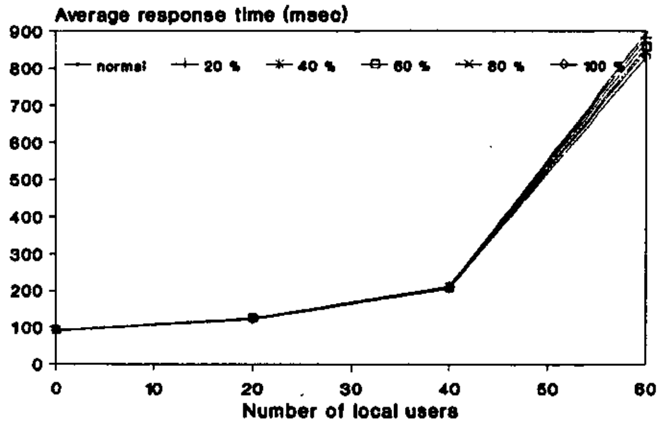


Figure C.9.1.9 : 50.7Kbytes

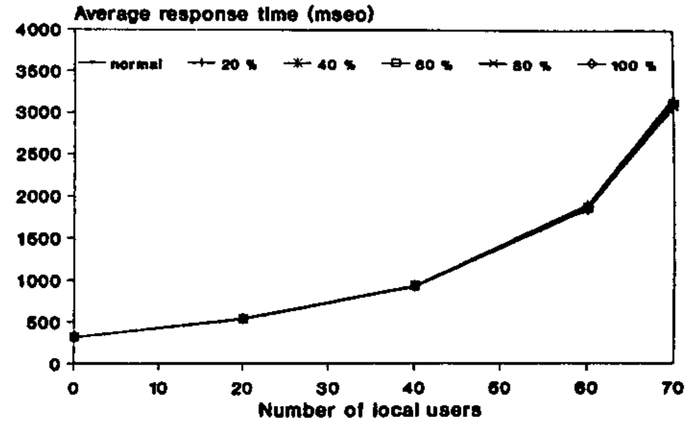


Figure C.9.1.10 : 316Kbytes(B)

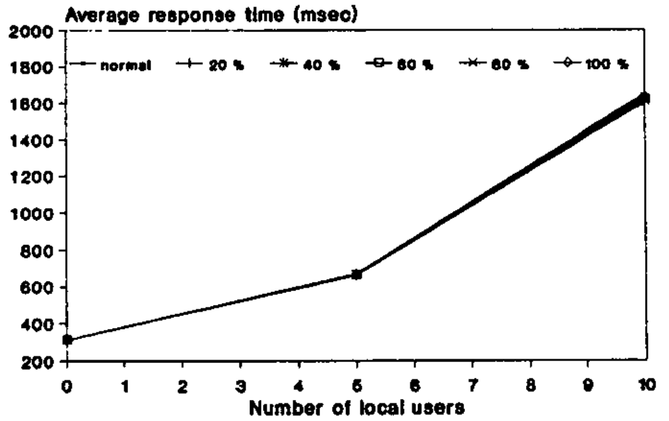


Figure C.9.1.11 : 316Kbytes

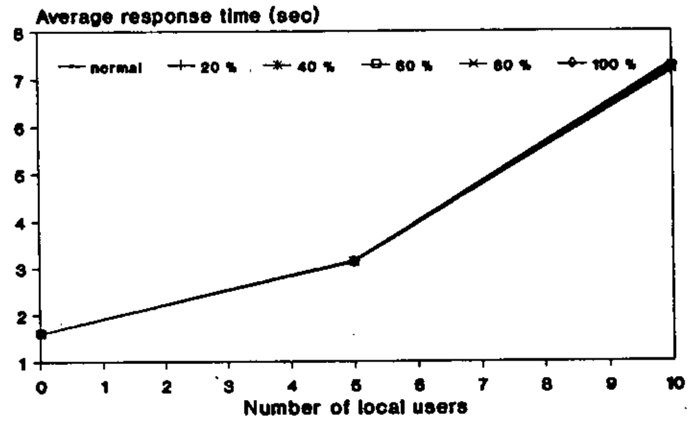


Figure C.9.1.12 : 1856Kbytes

The effect of the improved concurrency during disk I/O operations on the average response time in the Sun SPARCstation 10 workstation.

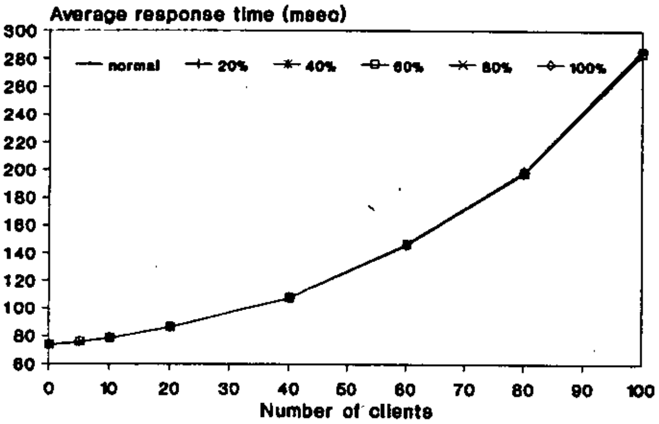


Figure C.9.2.1 : 8Kbytes

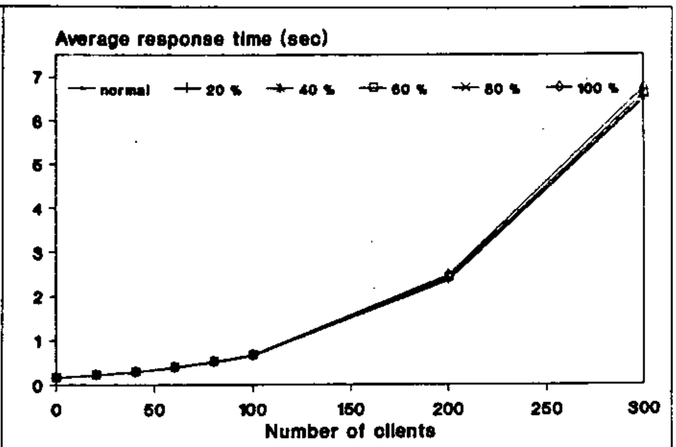


Figure C.9.2.2 : 47Kbytes

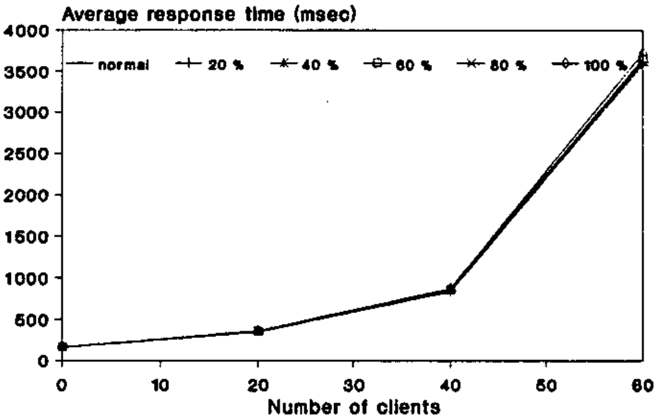


Figure C.9.2.3 : 50.7Kbytes

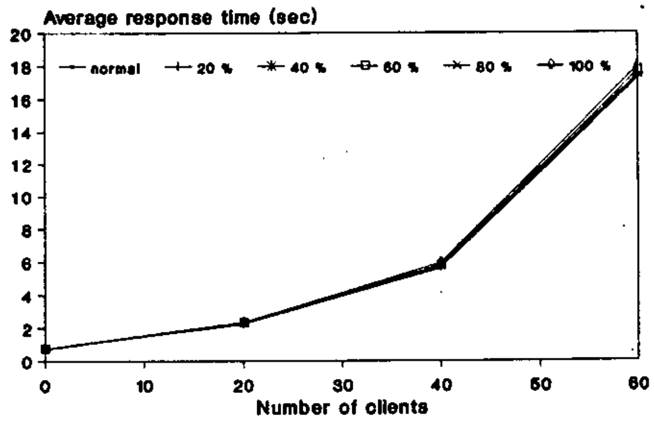


Figure C.9.2.4 : 316Kbytes(B)

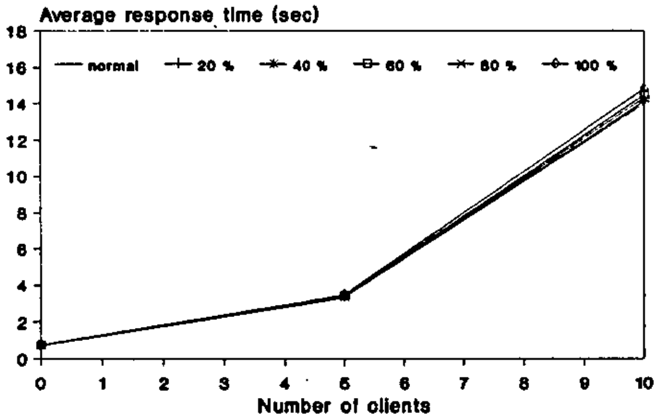


Figure C.9.2.5 : 316Kbytes

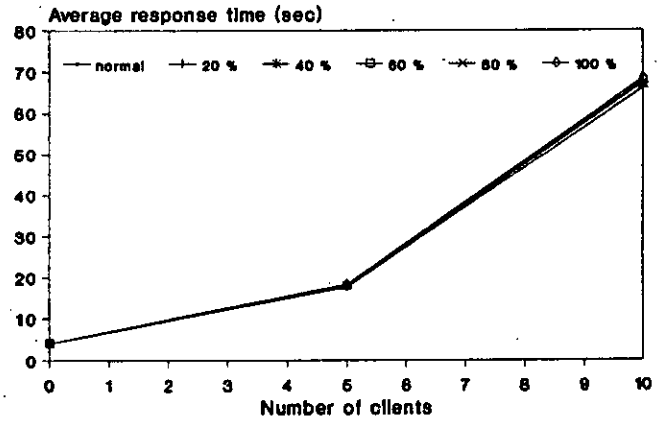


Figure C.9.2.6 : 1856Kbytes

The effect of the improved concurrency during communication operations on the average response time in the distributed file system which consists of the Sun SPARCstation 10 workstations.

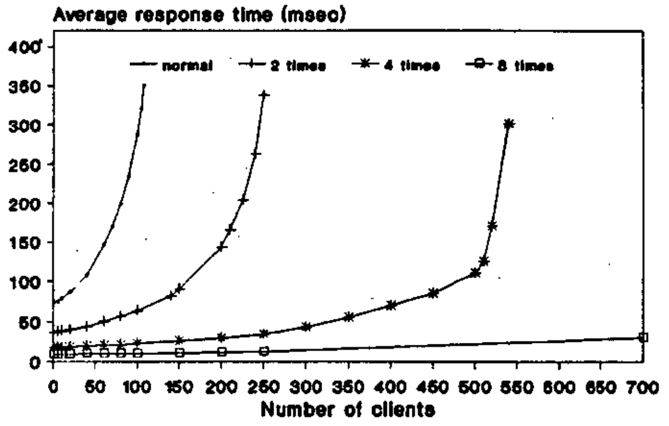


Figure C.10.1 : 8Kbytes

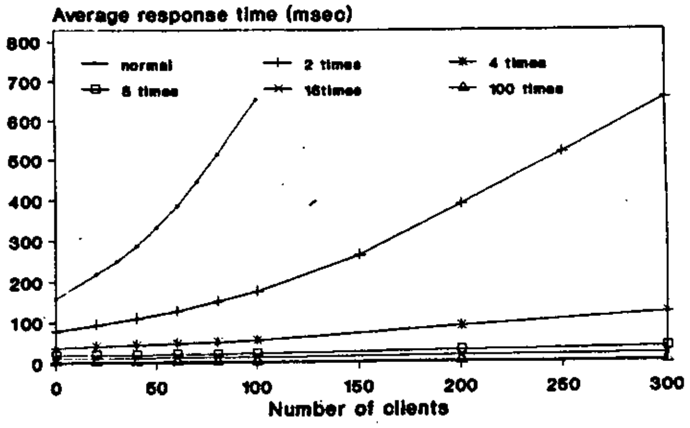


Figure C.10.2 : 47Kbytes

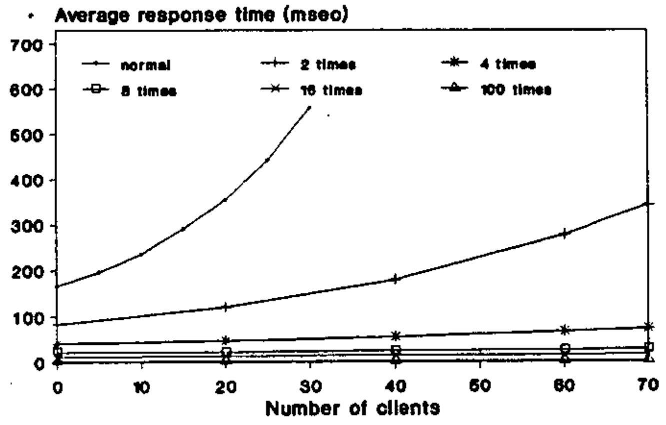


Figure C.10.3 : 50.7Kbytes

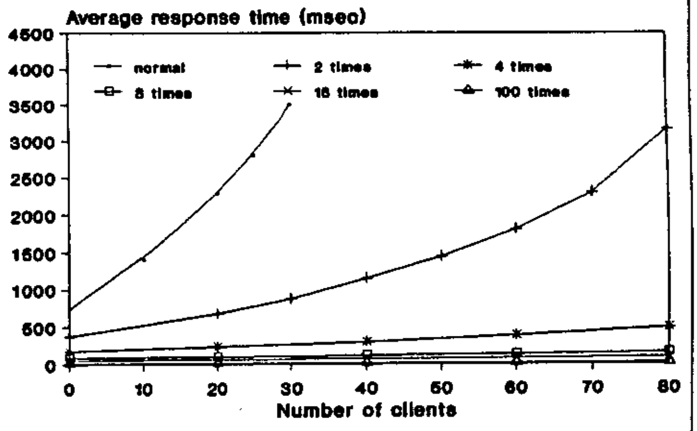


Figure C.10.4 : 316Kbytes(B)

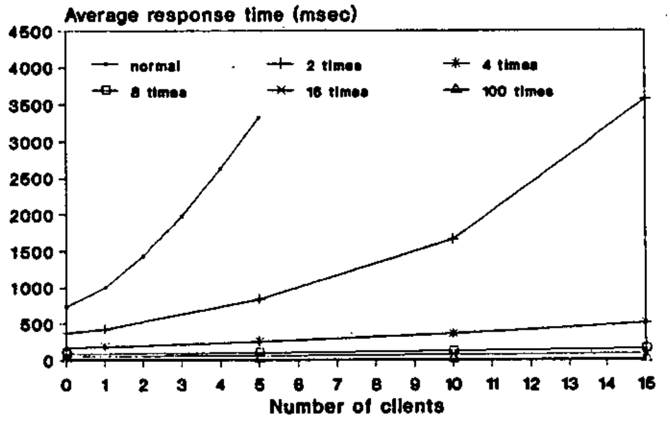


Figure C.10.5 : 316Kbytes

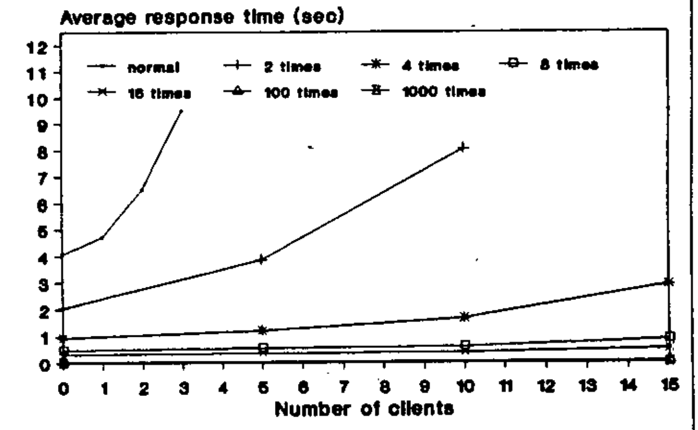


Figure C.10.6 : 1856Kbytes

The effect on the average response time of improving the power of all resources in the distributed file system which consists of the Sun SPARCstation 10 workstations.

Appendix D

The Effect of Caching

D.1 Standalone Caching in the Memory of the File Server

Figure D.1.1 to figure D.1.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure D.1.7 to figure D.1.12 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation.

D.2 Standalone Caching in the Disk Interface Unit

Figure D.2.1 to figure D.2.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations

Figure D.2.7 to figure D.2.12 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation.

D.3 Standalone Caching in the Memory of the Client

Figure D.3.1 to figure D.3.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

D.4 Standalone Caching in the Disk of the Client.

Figure D.4.1 to figure D.4.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100%. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

D.5 Combination of Caching in the Memory of the Client and Caching in the Memory of the File Server

Figure D.5.1 to figure D.5.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure D.5.7 to figure D.5.12 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system when the hit rate of the cache in the memory of the client is improved to be 20%, 40%, 60%, 80% and 100% while the hit rate of the cache in the memory of the file server is fixed to be 60%. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

D.6 Combination of Caching in the Disk of the Client and Caching in the Memory of the File Server

Figure D.6.1 to figure D.6.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure D.6.7 to figure D.6.12 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system when the hit rate of the cache in the disk of the client is improved to be 20%, 40%, 60%, 80% and 100% while the hit rate of the cache in the memory of the file server is fixed to be 60% all the time. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

D.7 Combination of Caching in the Memory of the Client, Caching in the Memory of the File Server and Caching in the Disk Interface Unit of the File Server

Figure D.7.1 to figure D.7.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in the three caches at the same time. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

D.8 Combination of Caching in the Disk of the Client, Caching in the Memory of the File Server and Caching in the Disk Interface Unit of the File Server

Figure D.8.1 to figure D.8.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in the three caches at the same time. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

D.9 Combination of Caching in the Memory and Caching in the Disk Interface Unit in a System

Figure D.9.1 to figure D.9.6 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system as the number of clients increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure D.9.7 to figure D.9.12 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the distributed file system when the hit rate of the cache in the disk interface unit of the file server is improved to be 20%, 40%, 60%, 80% and 100% while the hit rate of the cache in the memory of the file server is fixed to be 60% all the time. Except for them, all others are kept the same as the baseline distributed file system which consists of the Sun SPARCstation 10 workstations.

Figure D.9.13 to figure D.9.18 show the average response time of the 8Kbytes workload, the 47Kbytes workload, the 50.7Kbytes workload, the 316Kbytes(B) workload, the 316Kbytes workload and the 1856Kbytes workload respectively in the shared memory system as the number of local users increases gradually. In each figure, the cache hit rate is improved to be 20%, 40%, 60%, 80% and 100% in both caches at the same time. Except for them, all others are kept the same as the baseline Sun SPARCstation 10 workstation.

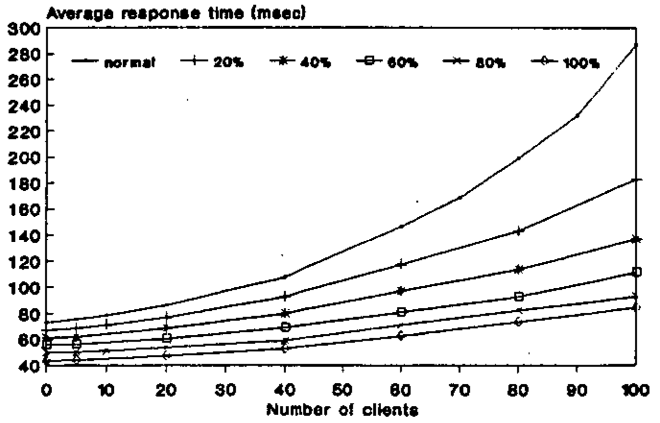


Figure D.1.1 : 8Kbytes

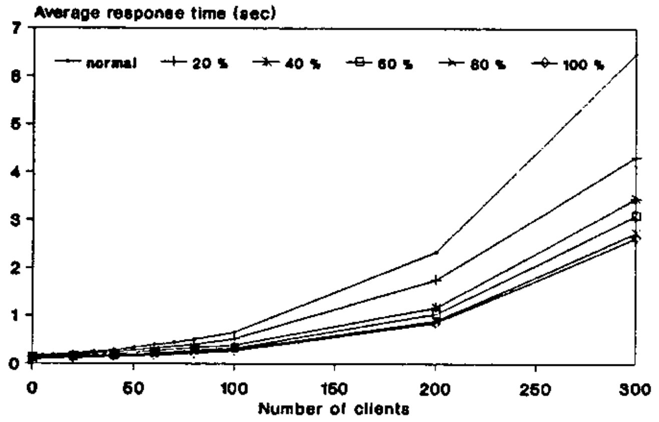


Figure D.1.2 : 47Kbytes

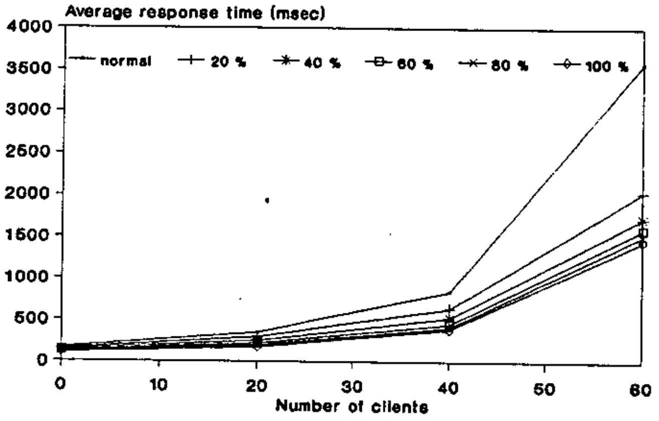


Figure D.1.3 : 50.7Kbytes

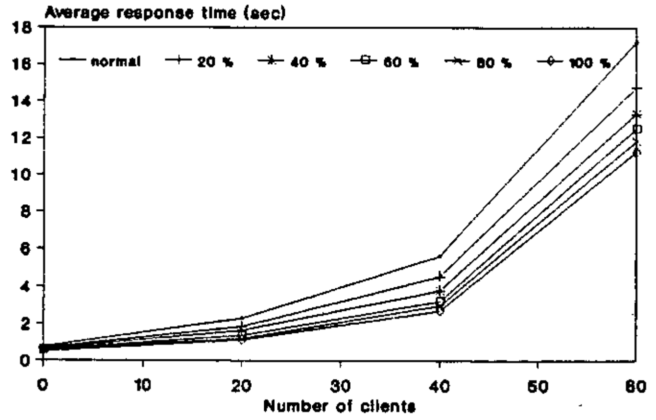


Figure D.1.4 : 316Kbytes(B)

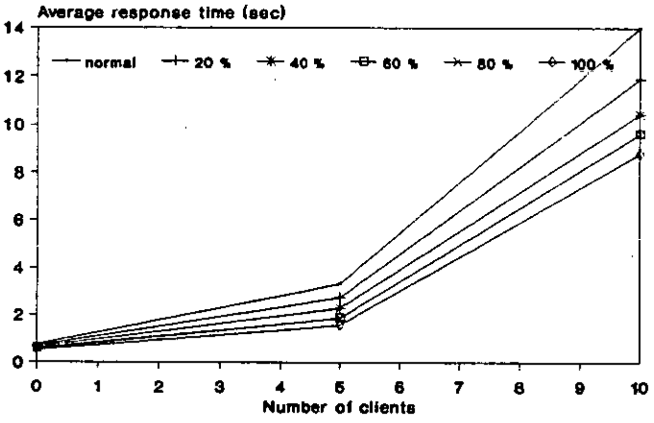


Figure D.1.5 : 316Kbytes

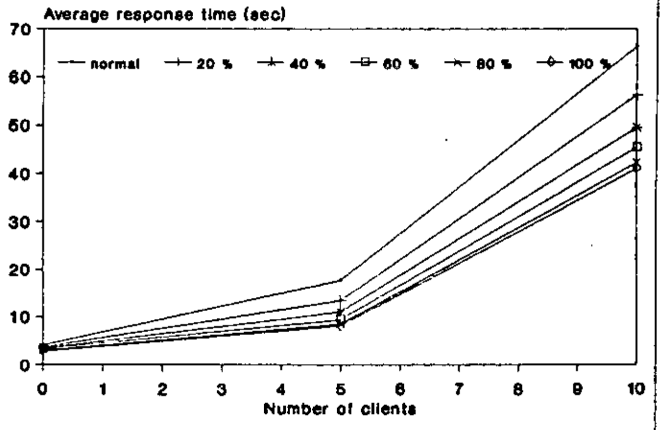


Figure D.1.6 : 1856Kbytes

The effect on the average response time of caching in the memory of the file server of the distributed file system which consists of the Sun SPARCstation 10 workstations.

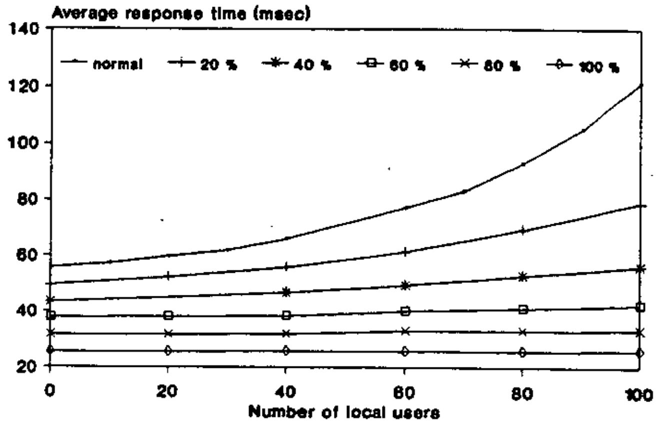


Figure D.1.7 : 8Kbytes

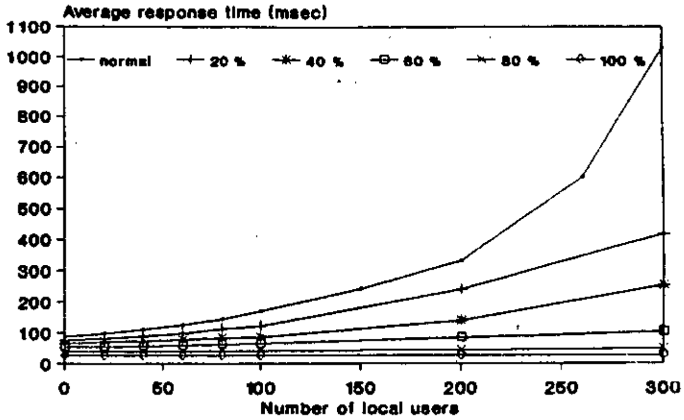


Figure D.1.8 : 47Kbytes

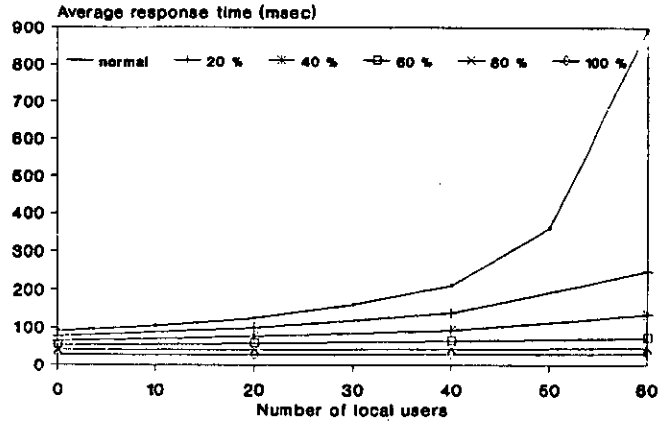


Figure D.1.9 : 50.7Kbytes

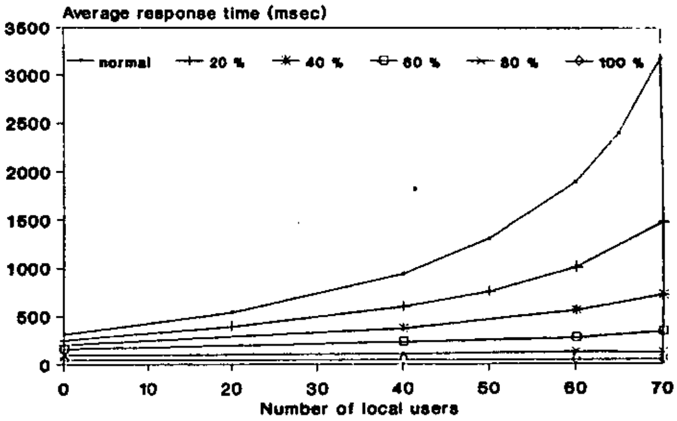


Figure D.1.10 : 316Kbytes(B)

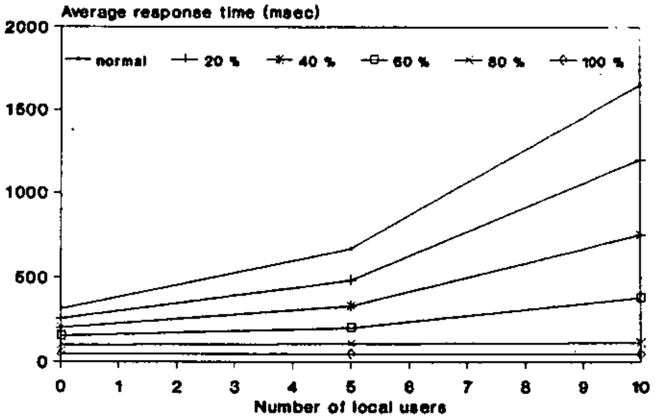


Figure D.1.11 : 316Kbytes

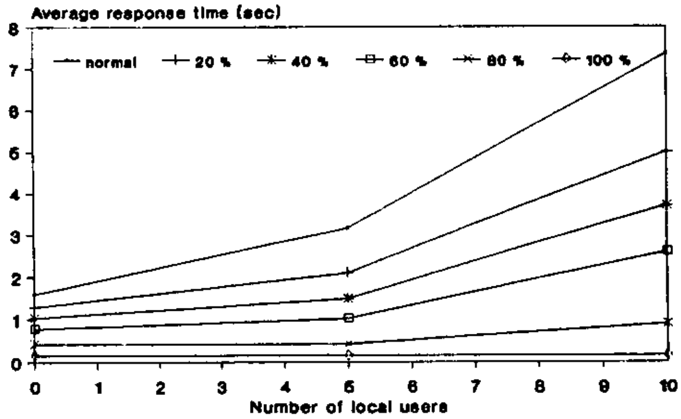


Figure D.1.12 : 1856Kbytes

The effect on the average response time of caching in the memory of the file server in the Sun SPARCstation 10 workstation.

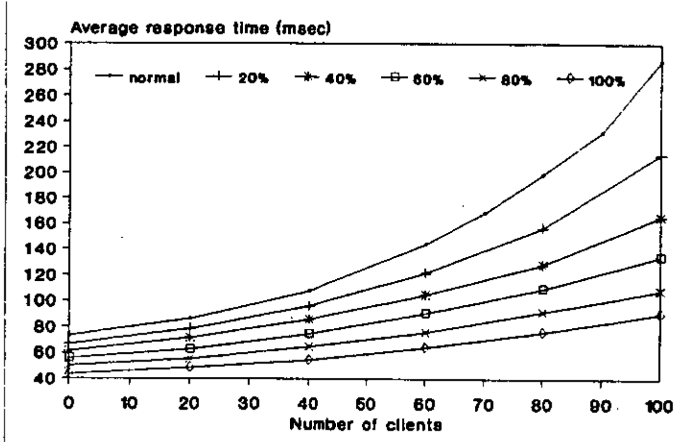


Figure D.2.1 : 8Kbytes

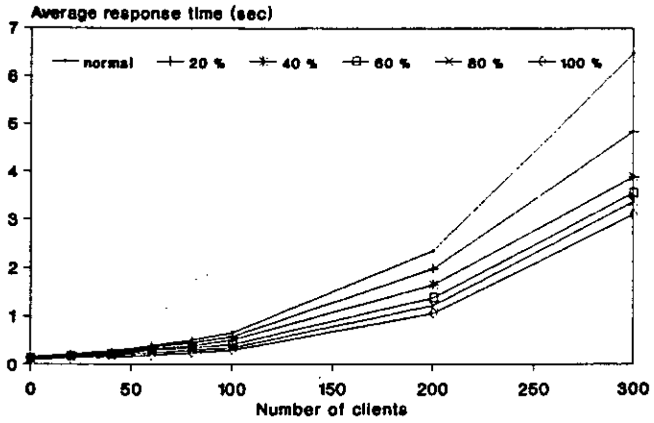


Figure D.2.2 : 47Kbytes

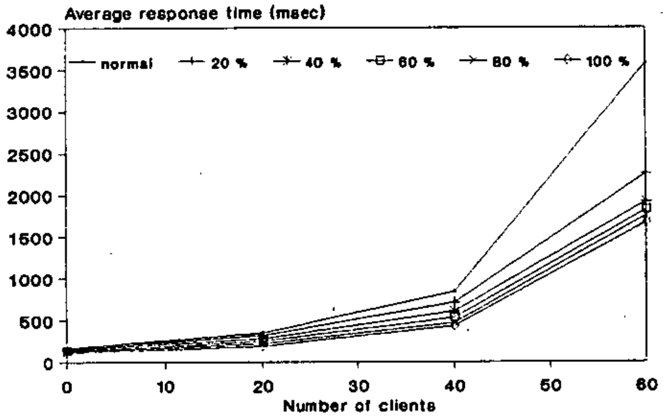


Figure D.2.3 : 50.7Kbytes

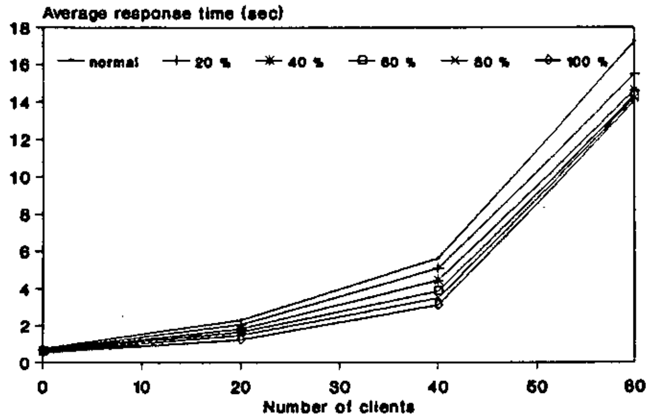


Figure D.2.4 : 316Kbytes(B)

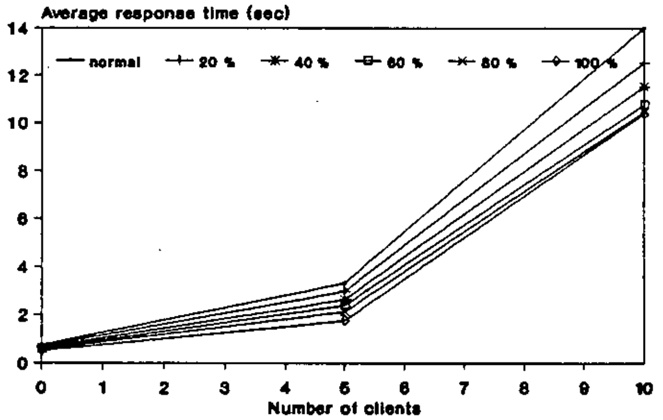


Figure D.2.5 : 316Kbytes

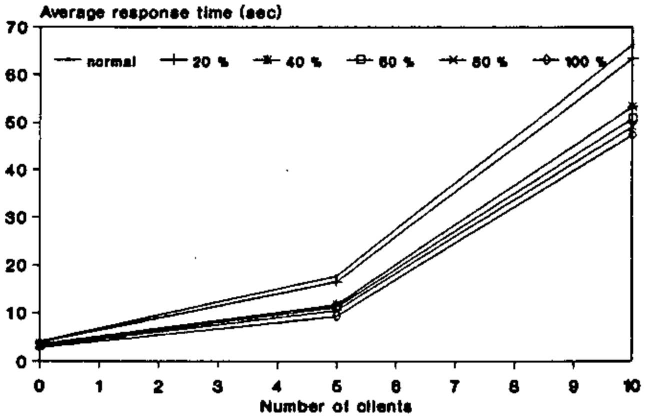


Figure D.2.6 : 1856Kbytes

The effect on the average response time of caching in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations.

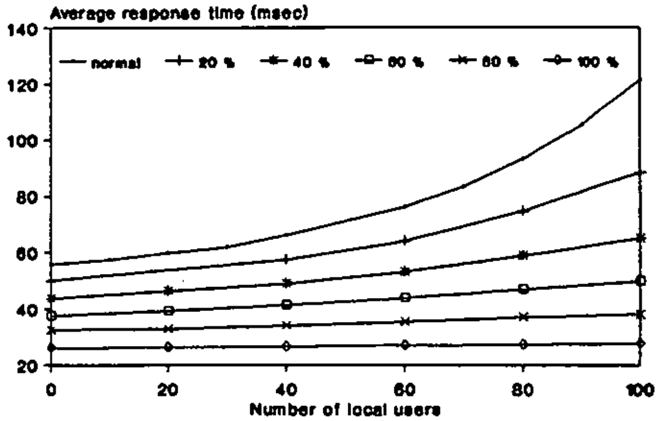


Figure D.2.7 : 8Kbytes

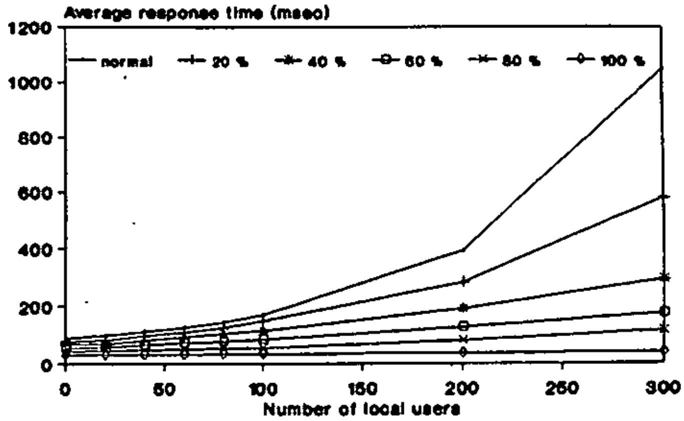


Figure D.2.8 : 47Kbytes

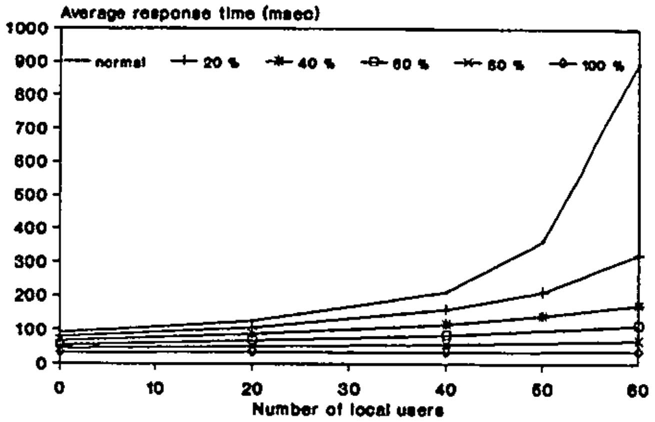


Figure D.2.9 : 50.7Kbytes

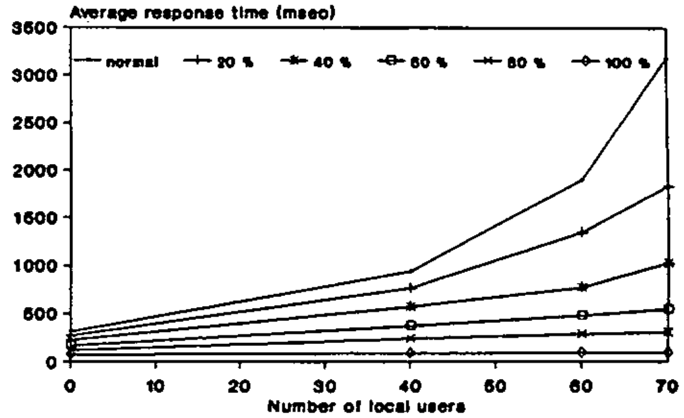


Figure D.2.10 : 316Kbytes(B)

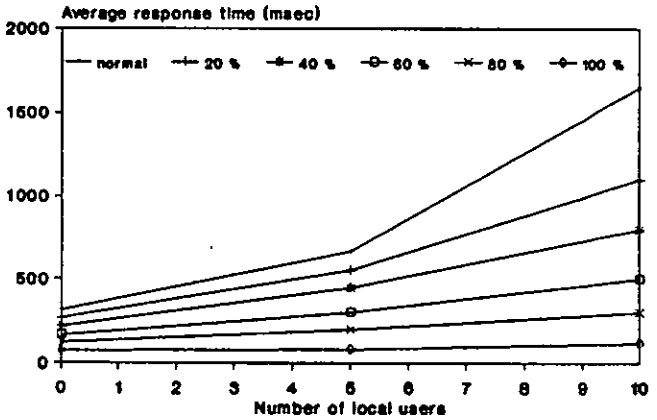


Figure D.2.11 : 316Kbytes

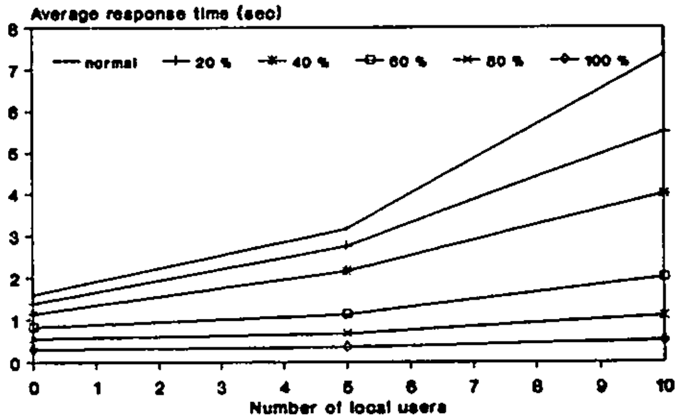


Figure D.2.12 : 1856Kbytes

The effect on the average response time of caching in the disk interface unit of the Sun SPARCstation 10 workstation.

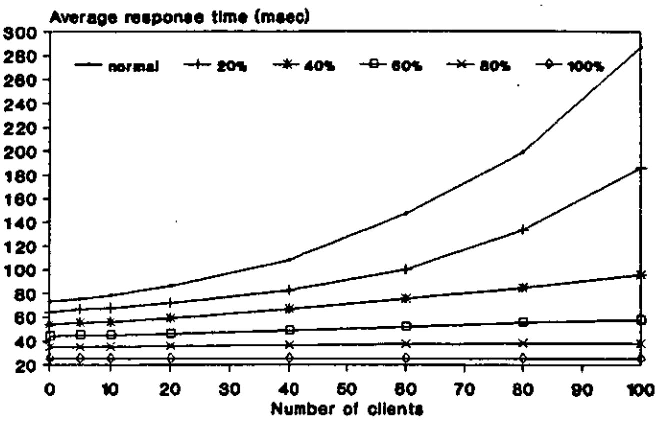


Figure D.3.1 : 8Kbytes

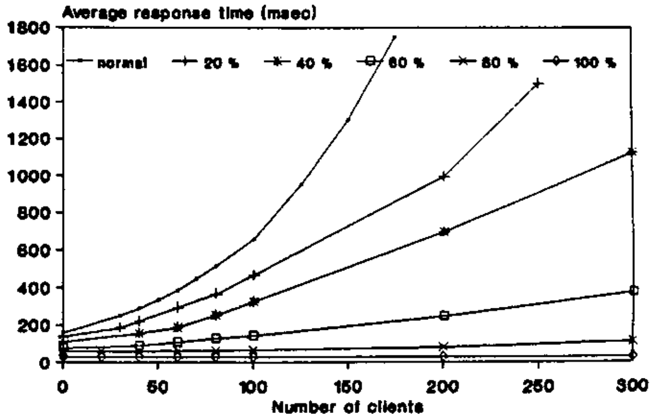


Figure D.3.2 : 47Kbytes

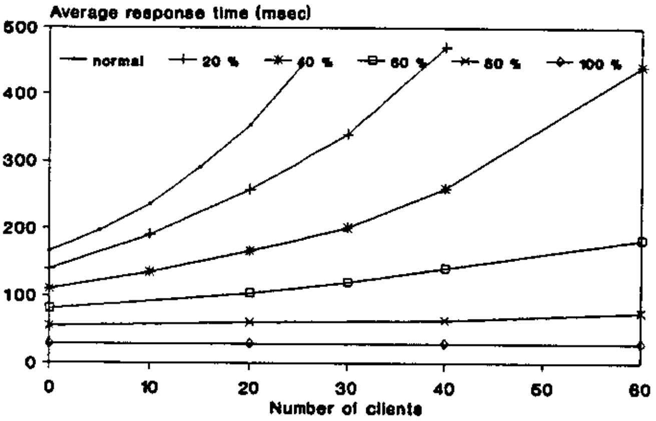


Figure D.3.3 : 50.7Kbytes

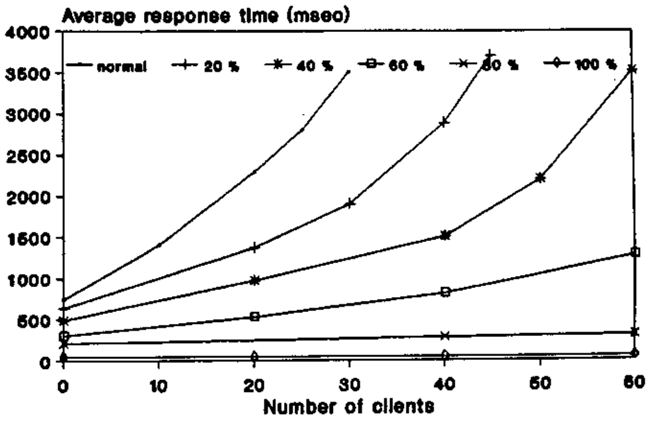


Figure D.3.4 : 316Kbytes(B)

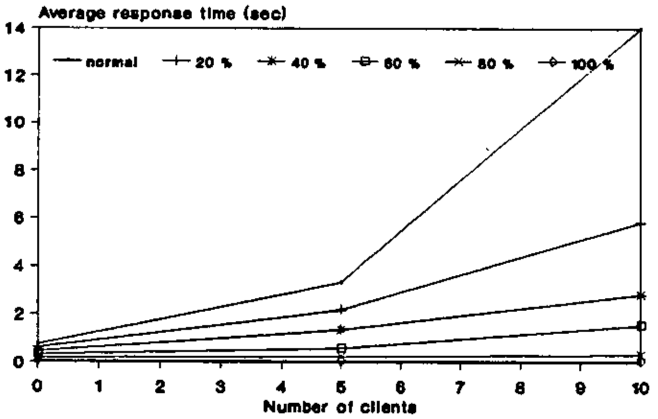


Figure D.3.5 : 316Kbytes

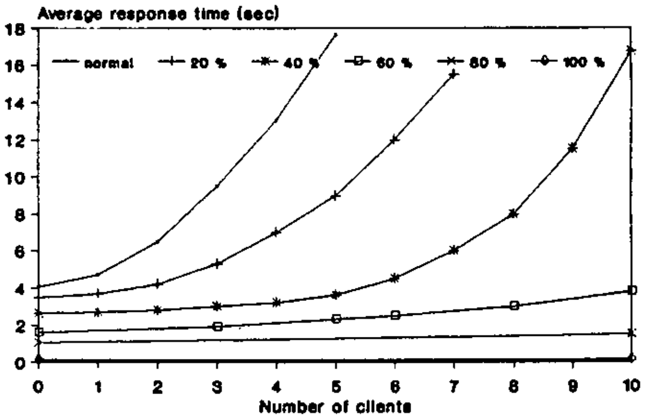


Figure D.3.6 : 1856Kbytes

The effect on the average response time of caching in the memory of the client in the distributed file system which consists of the Sun SPARCstation 10 workstations.

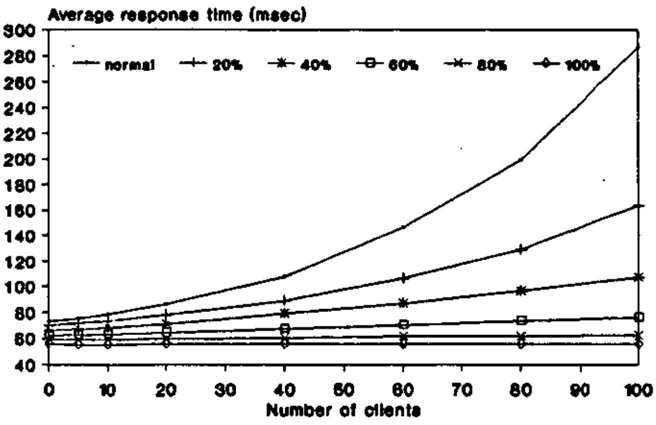


Figure D.4.1 : 8Kbytes

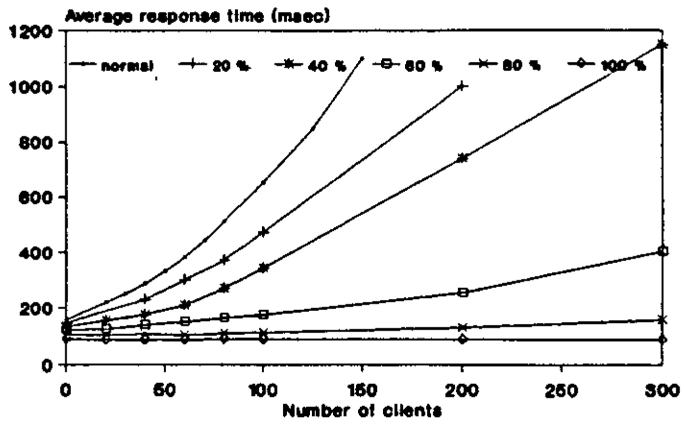


Figure D.4.2 : 47Kbytes

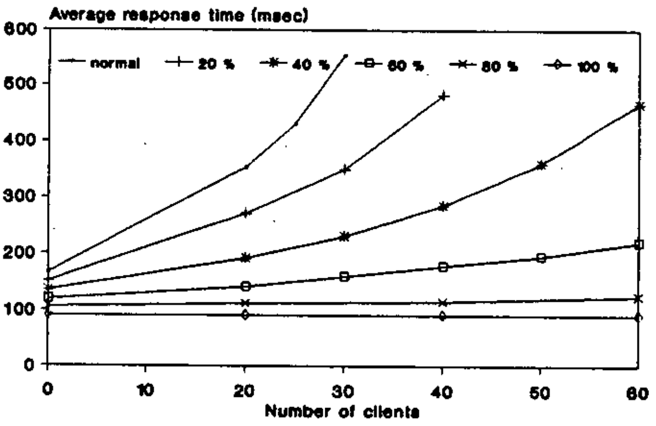


Figure D.4.3 : 50.7Kbytes

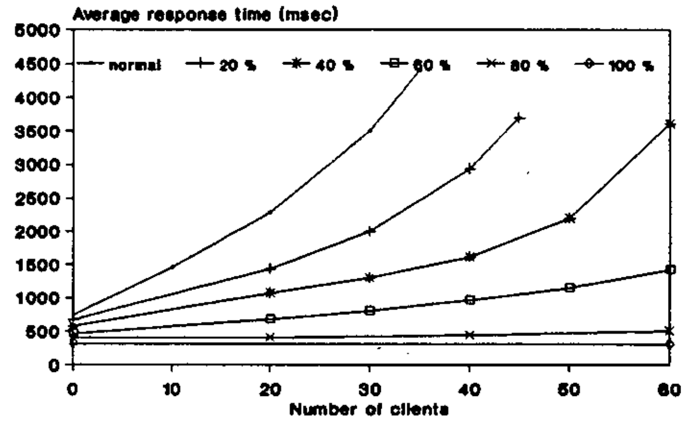


Figure D.4.4 : 316Kbytes(B)

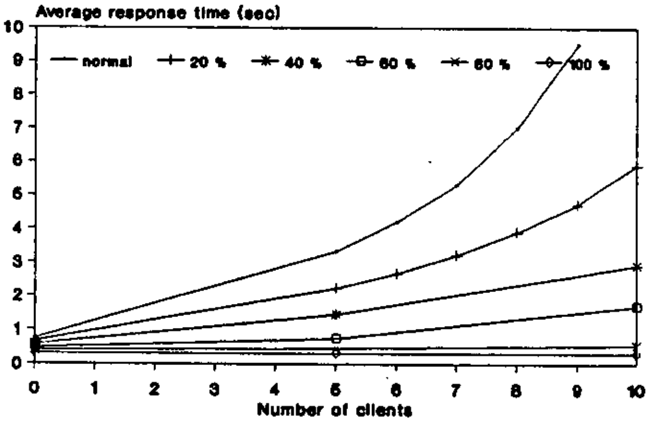


Figure D.4.5 : 316Kbytes

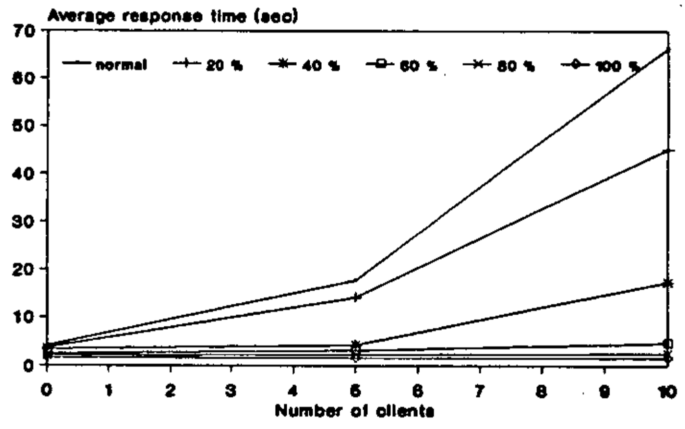


Figure D.4.6 : 1856Kbytes

The effect on the average response time of caching in the disk of the client in the distributed file system which consists of the Sun SPARCstation 10 workstations.

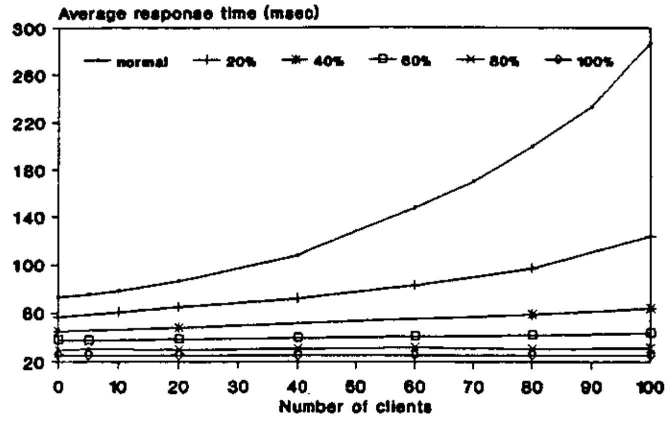


Figure D.5.1 : 8Kbytes

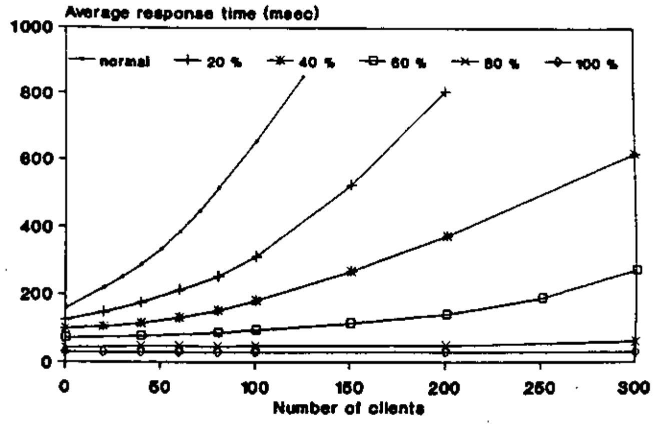


Figure D.5.2 : 47Kbytes

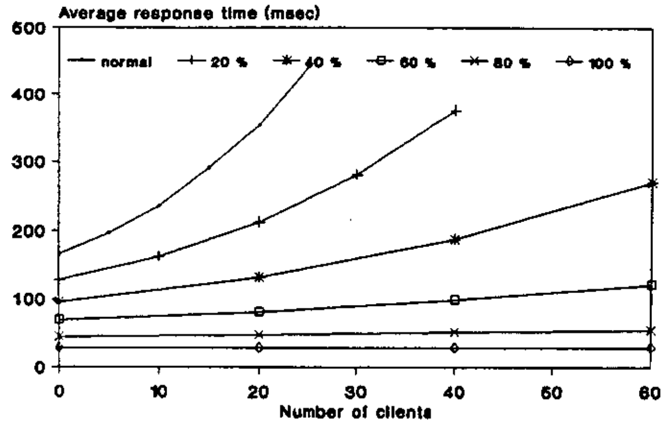


Figure D.5.3 : 50.7Kbytes

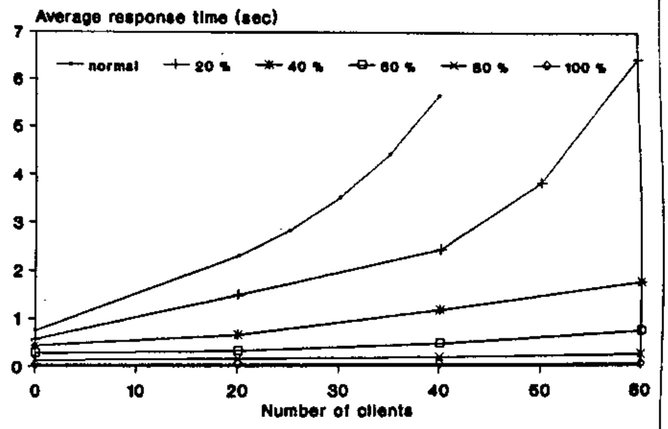


Figure D.5.4 : 316Kbytes(B)

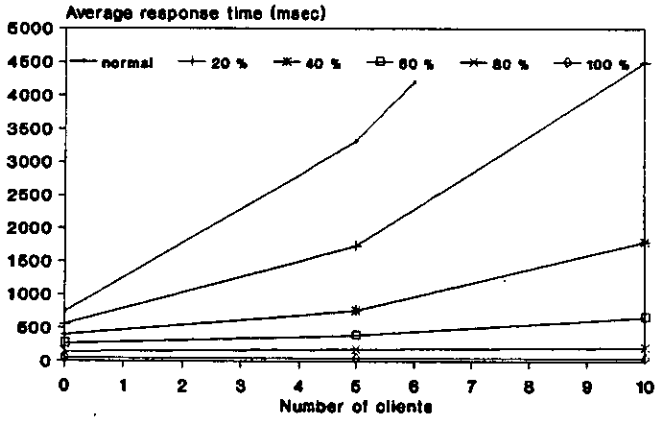


Figure D.5.5 : 316Kbytes

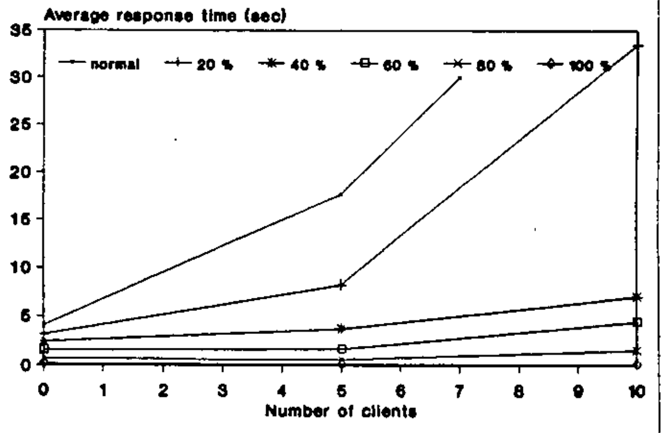


Figure D.5.6 : 1856Kbytes

The effect on the average response time of caching both in the memory of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the both caches improves at the same time.

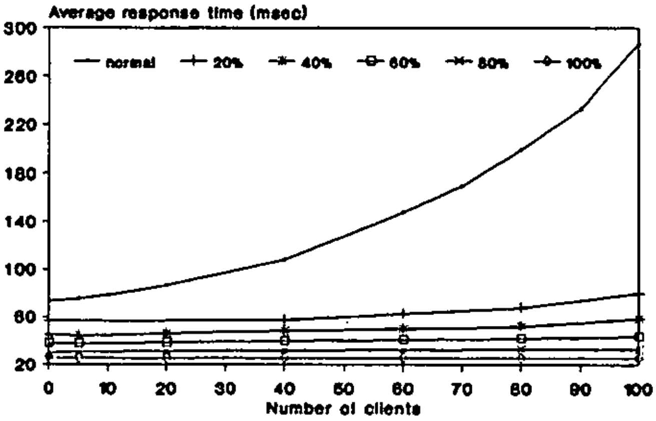


Figure D.5.7 : 8Kbytes

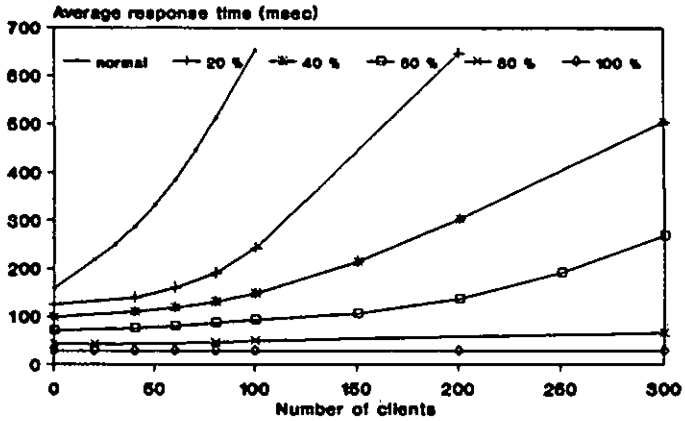


Figure D.5.8 : 47Kbytes

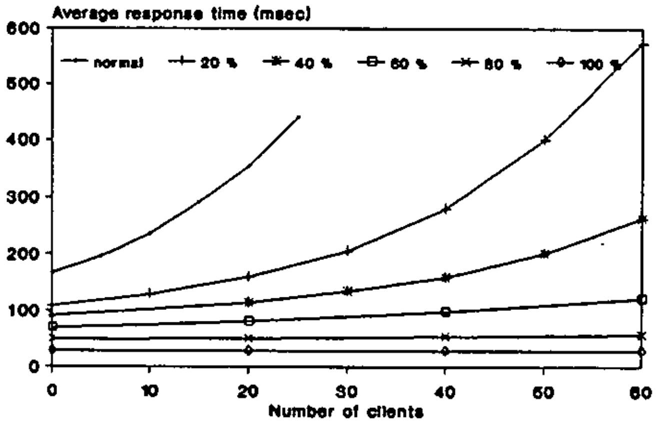


Figure D.5.9 : 50.7Kbytes

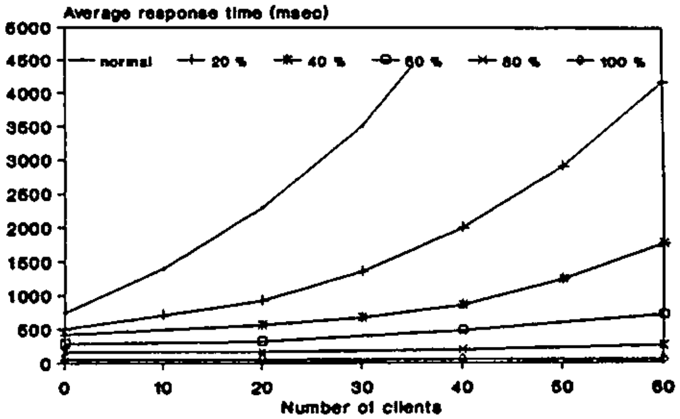


Figure D.5.10 : 316Kbytes(B)

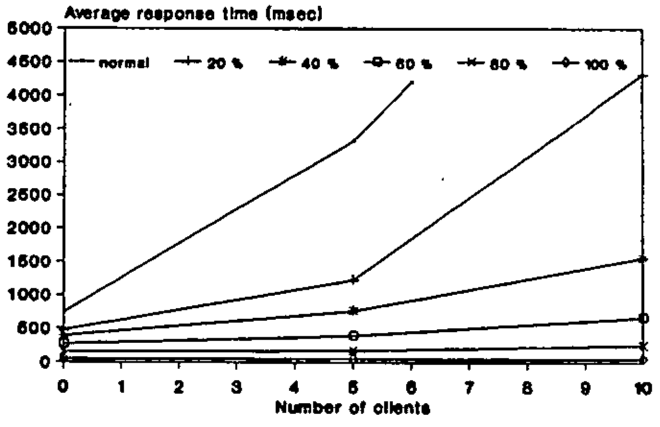


Figure D.5.11 : 316Kbytes

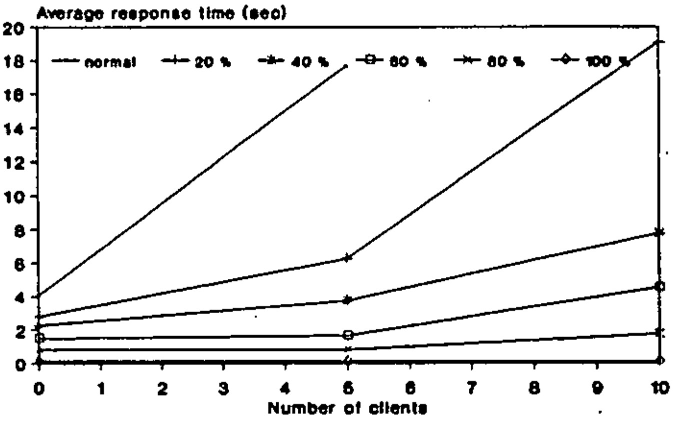


Figure D.5.12 : 1856Kbytes

The effect on the average response time of caching both in the memory of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the cache in the memory of the client improves while the hit rate of the cache in the memory of the file server is fixed to be 60%.

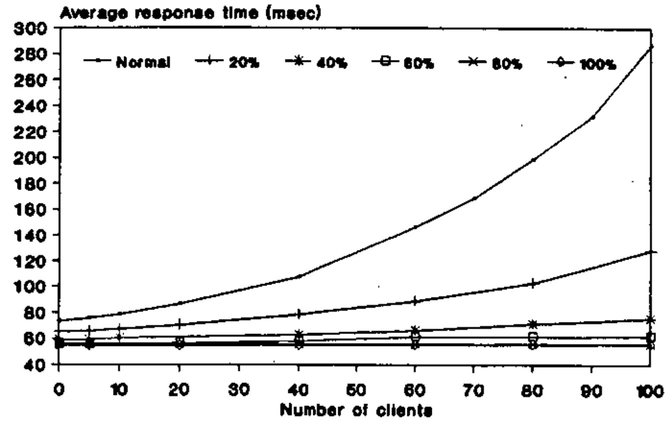


Figure D.6.1 : 8Kbytes

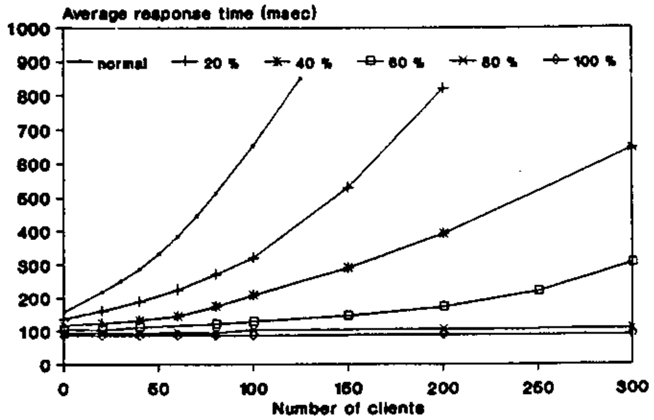


Figure D.6.2 : 47Kbytes

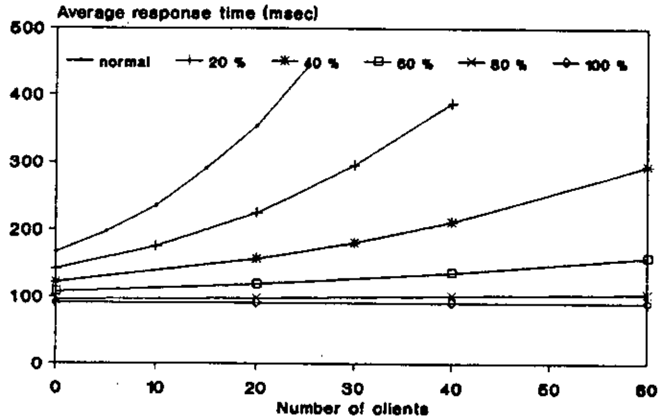


Figure D.6.3 : 50.7Kbytes

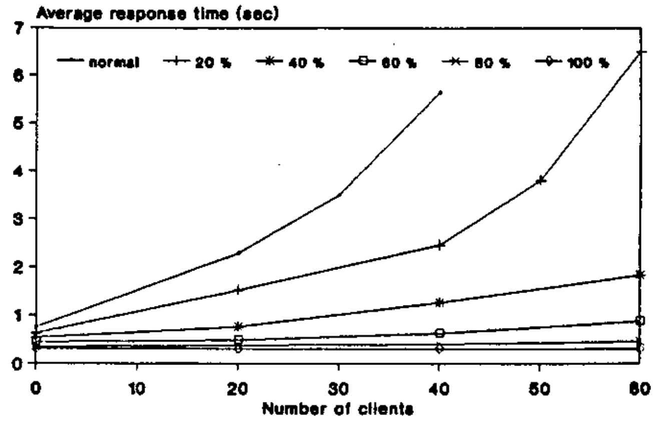


Figure D.6.4 : 316Kbytes(B)

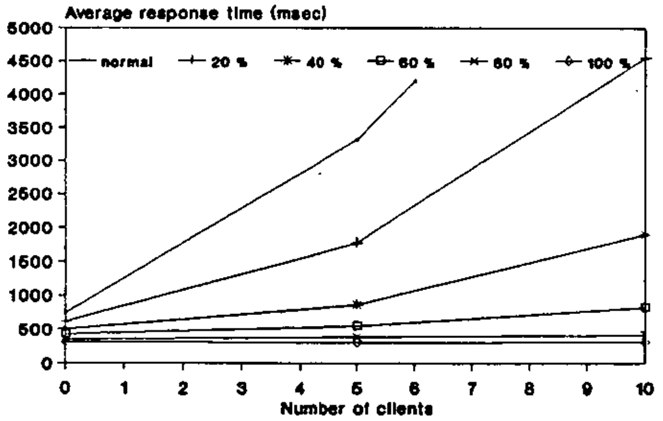


Figure D.6.5 : 316Kbytes

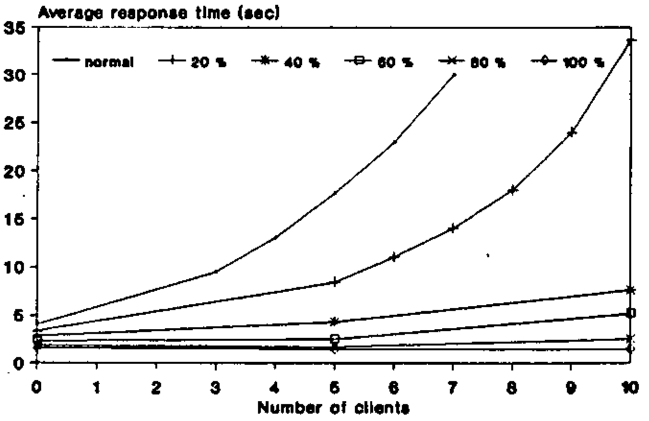


Figure D.6.6 : 1856Kbytes

The effect on the average response time of caching both in the disk of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the both caches improves at the same time.

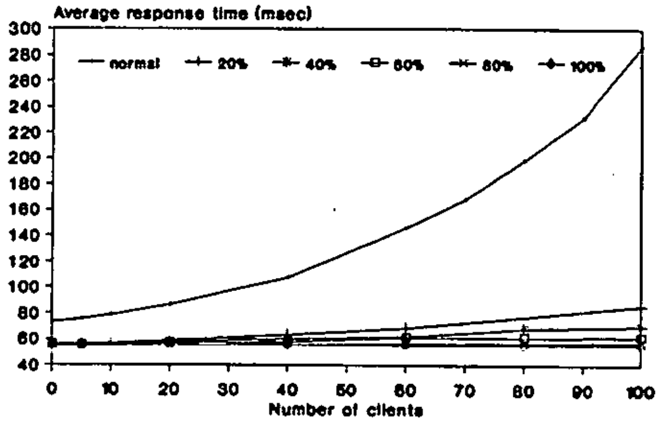


Figure D.6.7 : 8Kbytes

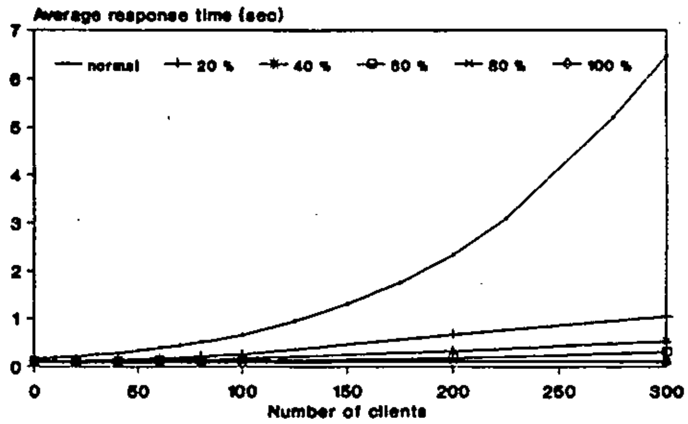


Figure D.6.8 : 47Kbytes

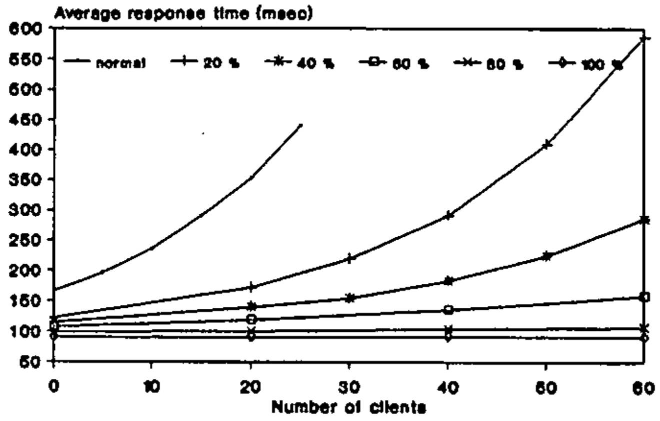


Figure D.6.9 : 50.7Kbytes

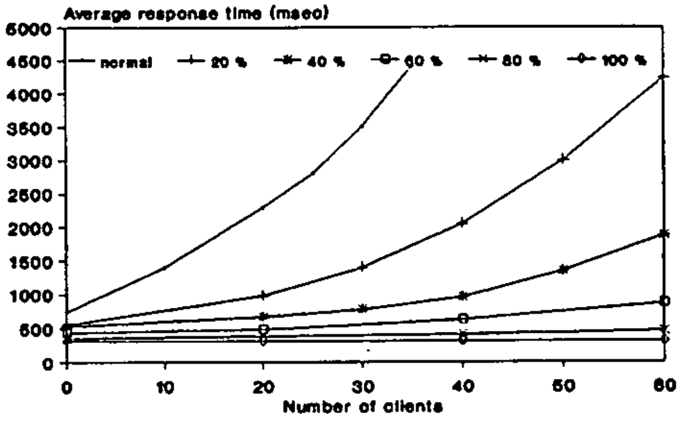


Figure D.6.10 : 316Kbytes(B)

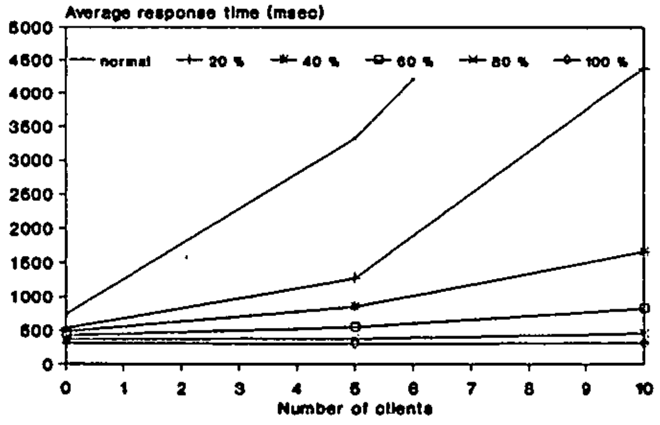


Figure D.6.11 : 316Kbytes

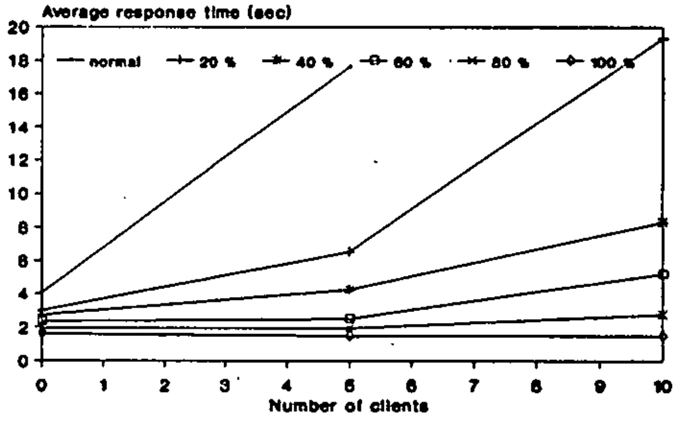


Figure D.6.12 : 1856Kbytes

The effect on the average response time of caching both in the disk of the client and in the memory of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the cache in the memory of the client improves while the the hit rate of the cache in the memory of the file server is fixed to be 60%.

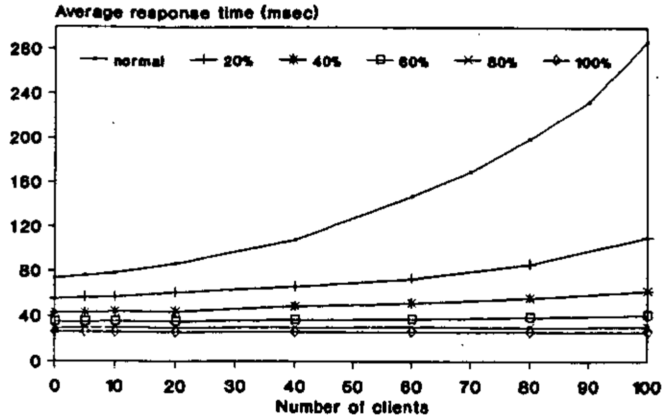


Figure D.7.1 : 8Kbytes

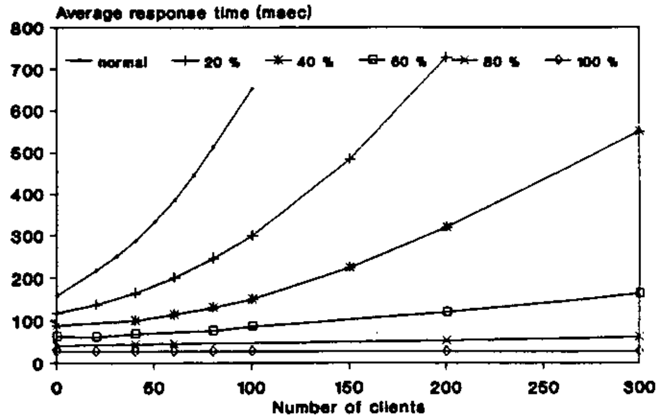


Figure D.7.2 : 47Kbytes

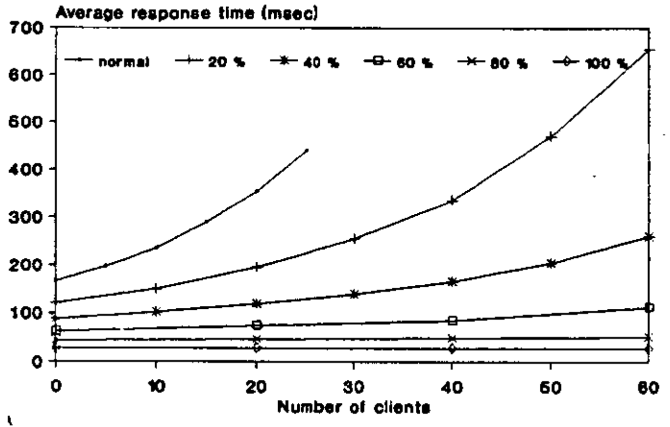


Figure D.7.3 : 50.7Kbytes

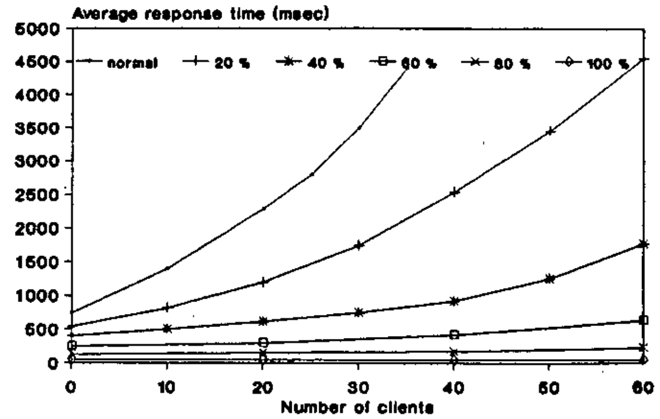


Figure D.7.4 : 316Kbytes(B)

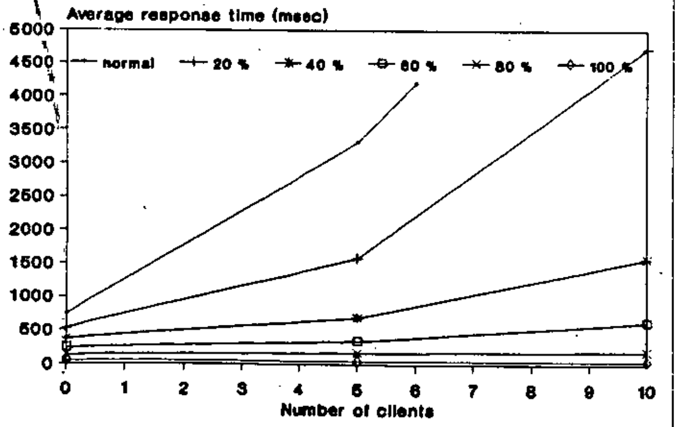


Figure D.7.5 : 316Kbytes

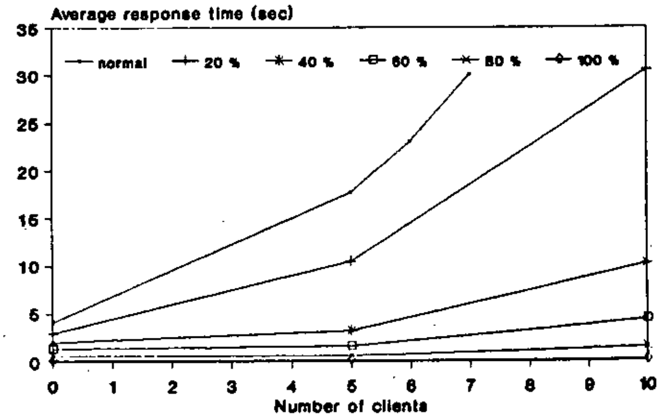


Figure D.7.6 : 1856Kbytes

The effect on the average response time of the combination of caching in the memory of the client, caching in the memory of the file server and caching in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations.

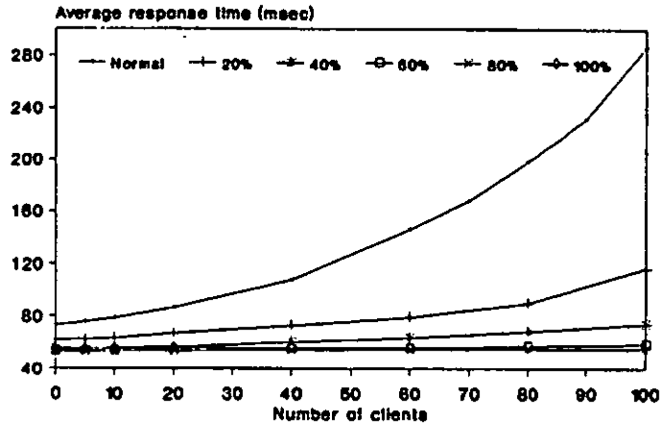


Figure D.8.1 : 8Kbytes

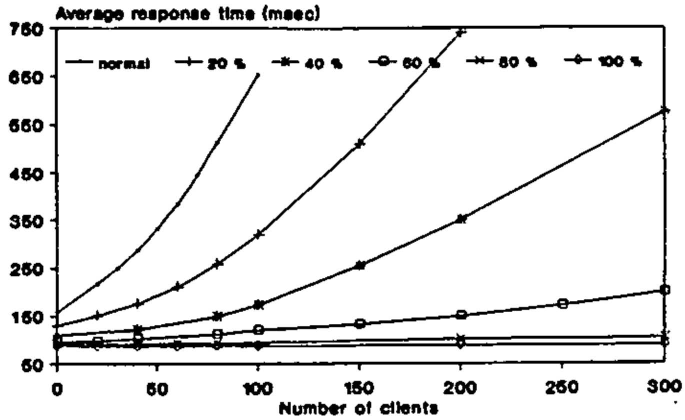


Figure D.8.2 : 47Kbytes

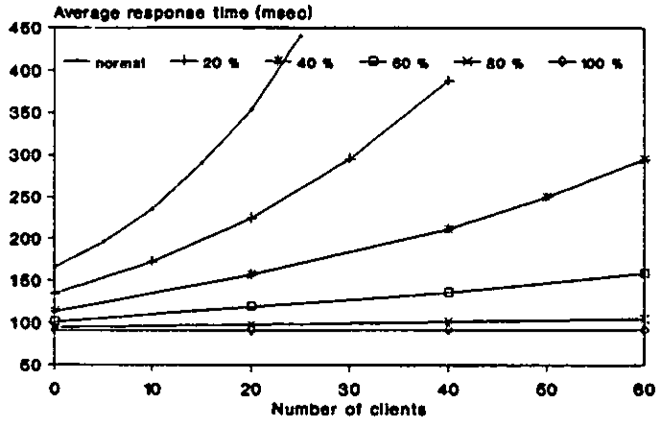


Figure D.8.3 : 50.7Kbytes

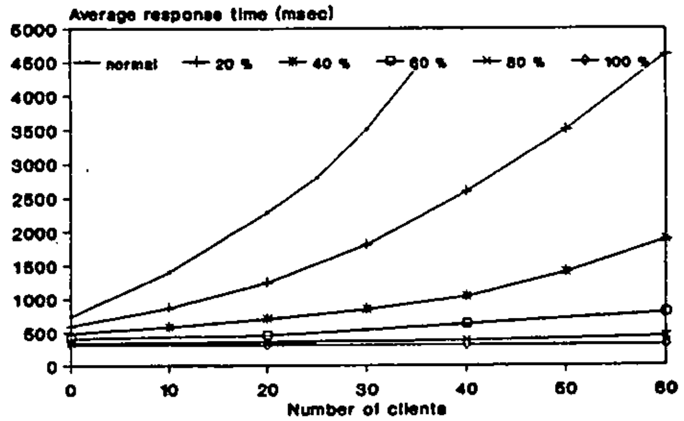


Figure D.8.4 : 316Kbytes(B)

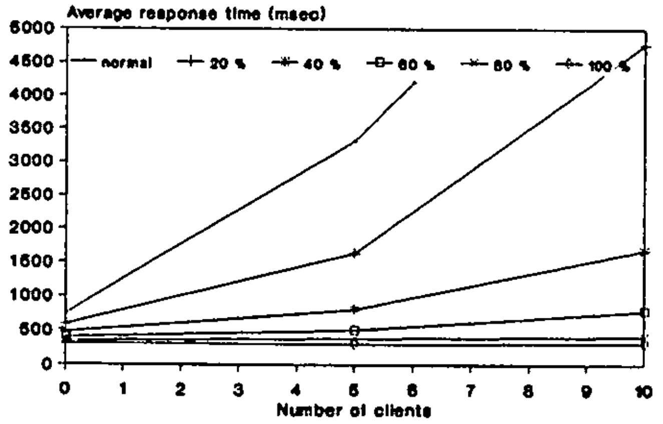


Figure D.8.5 : 316Kbytes

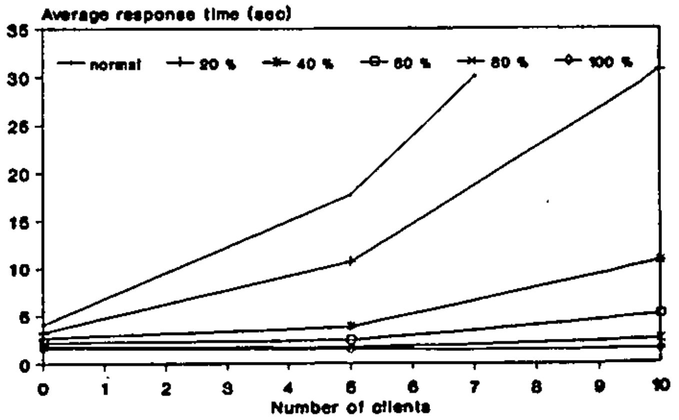


Figure D.8.6 : 1856Kbytes

The effect on the average response time of caching in the disk of the client, in the memory of the file server and in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations.

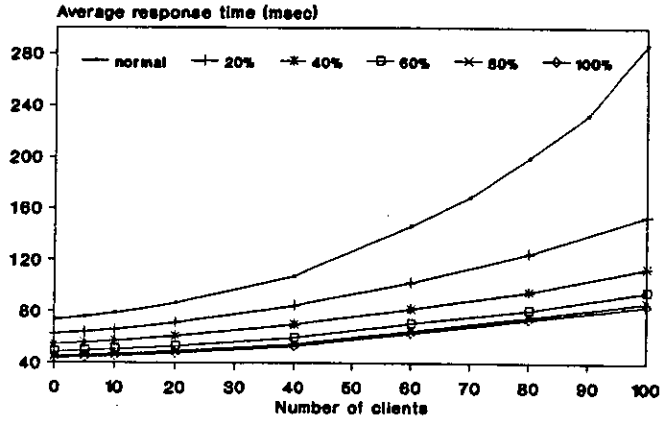


Figure D.9.1 : 8Kbytes

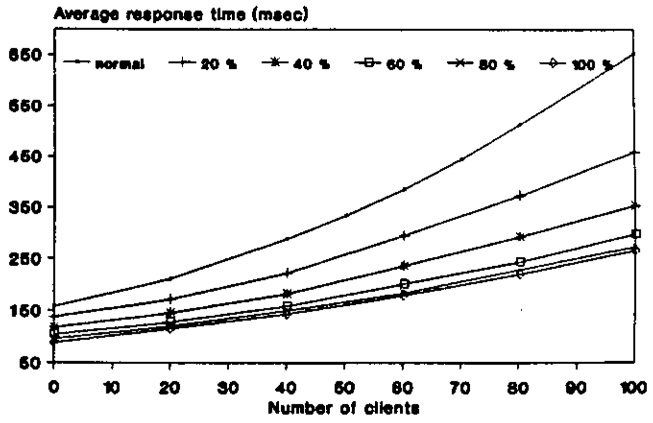


Figure D.9.2 : 47Kbytes

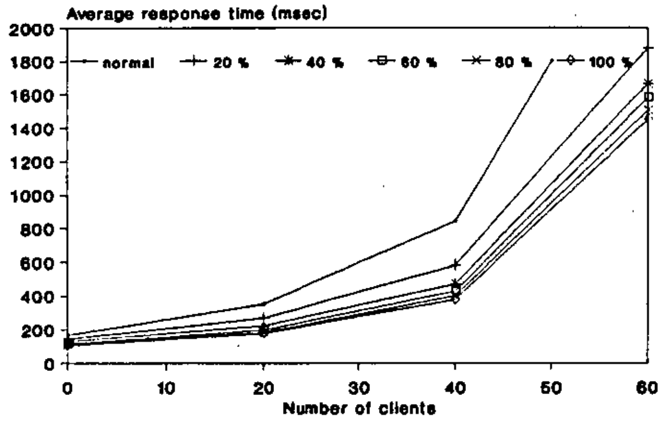


Figure D.9.3 : 50.7Kbytes

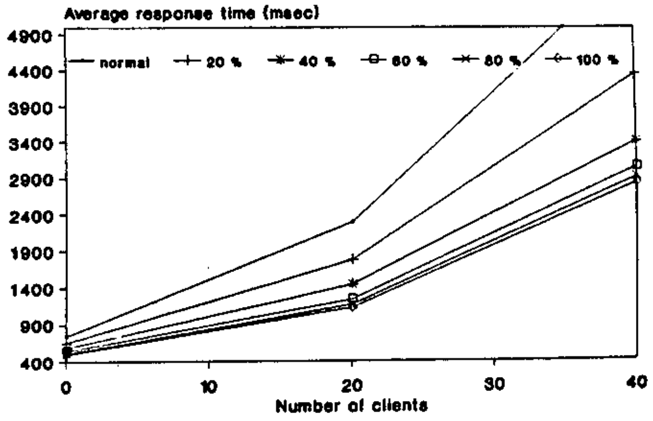


Figure D.9.4 : 316Kbytes(B)

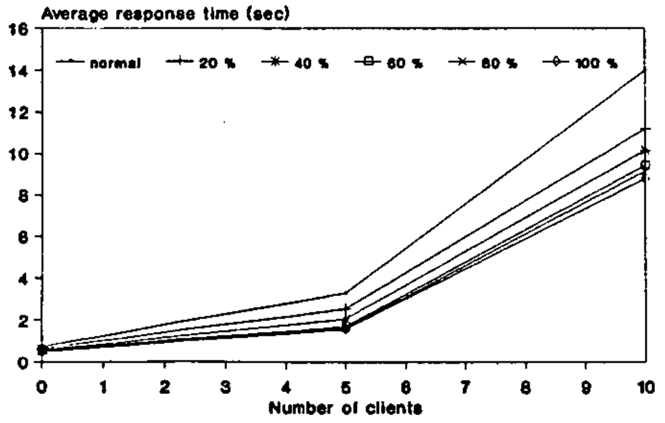


Figure D.9.5 : 316Kbytes

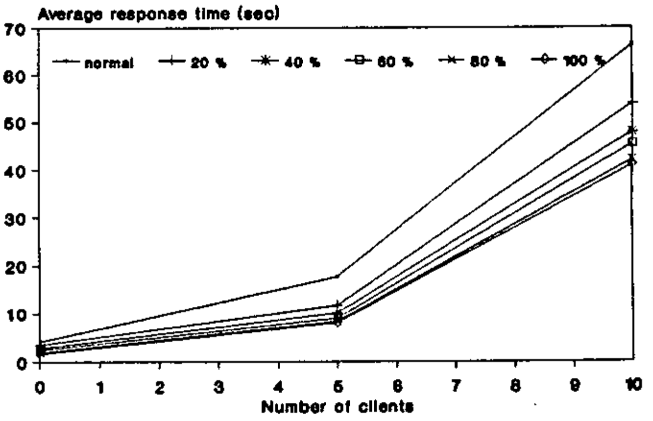


Figure D.9.6 : 1856Kbytes

The effect on the average response time of caching both in the memory of the file server and in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the both caches improves at the same time.

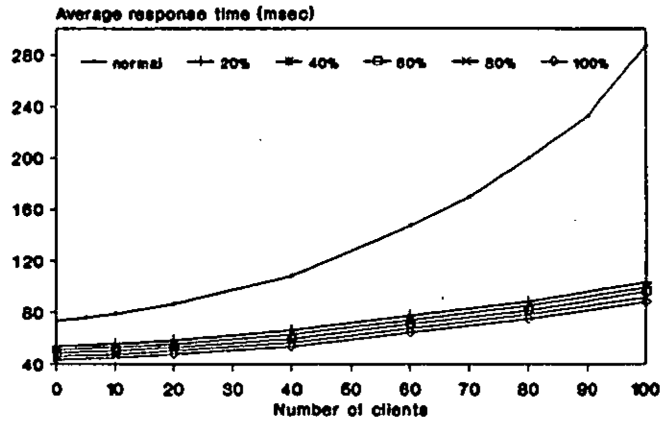


Figure D.9.7 : 8Kbytes

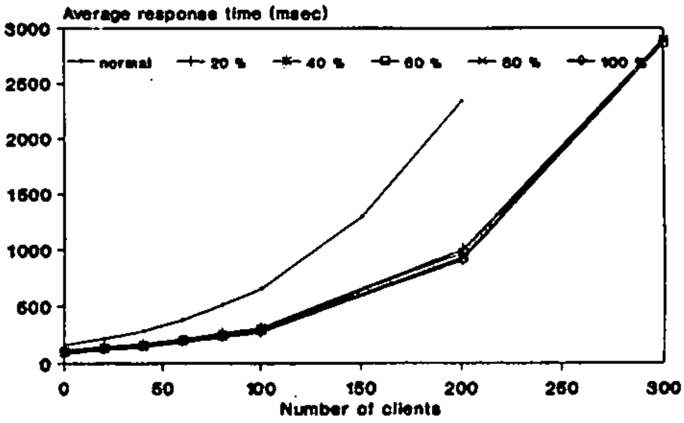


Figure D.9.8 : 47Kbytes

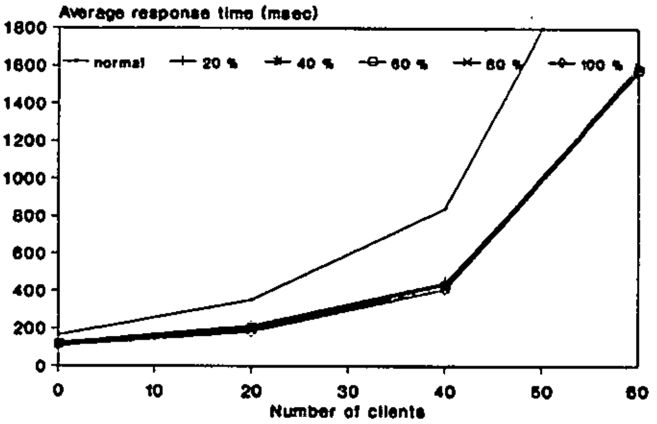


Figure D.9.9 : 50.7Kbytes

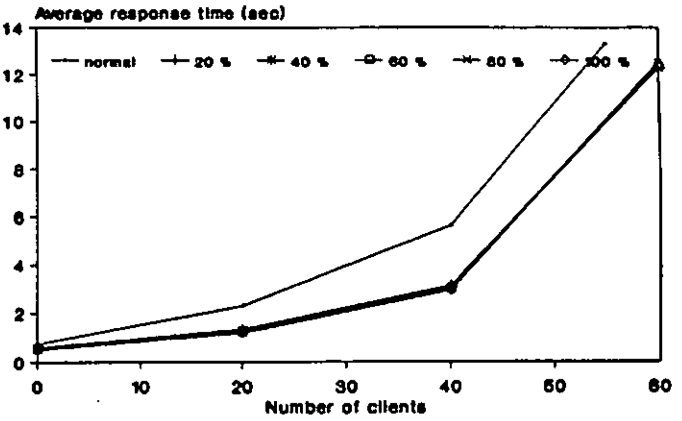


Figure D.9.10 : 316Kbytes(B)

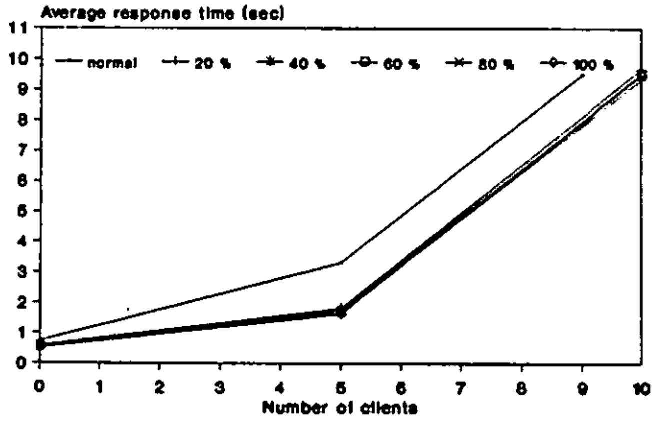


Figure D.9.11 : 316Kbytes

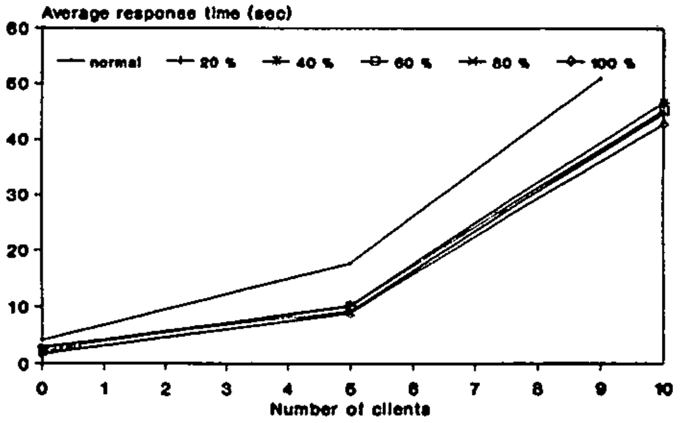


Figure D.9.12 : 1856Kbytes

The effect on the average response time of caching both in the memory of the file server and in the disk interface unit of the file server in the distributed file system which consists of the Sun SPARCstation 10 workstations when the hit rate of the cache in the memory of the client improves while the the hit rate of the cache in the memory of the file server is fixed to be 60%.

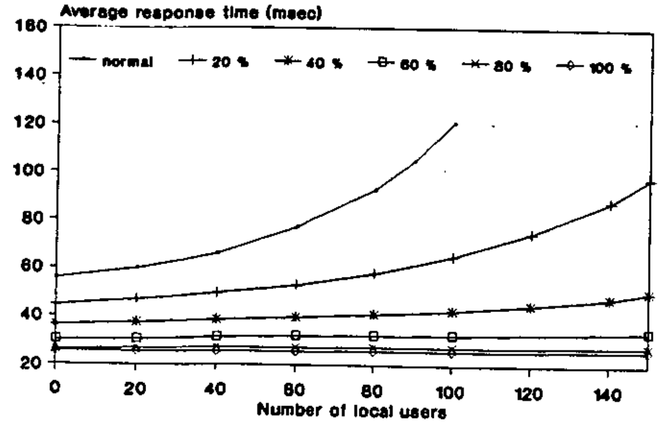


Figure D.9.13 : 8Kbytes

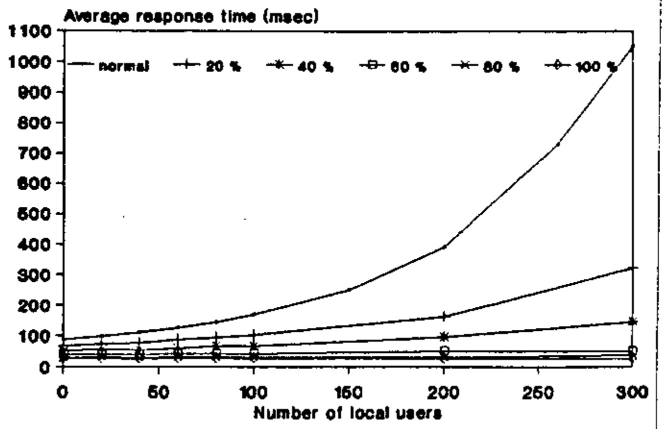


Figure D.9.14 : 47Kbytes

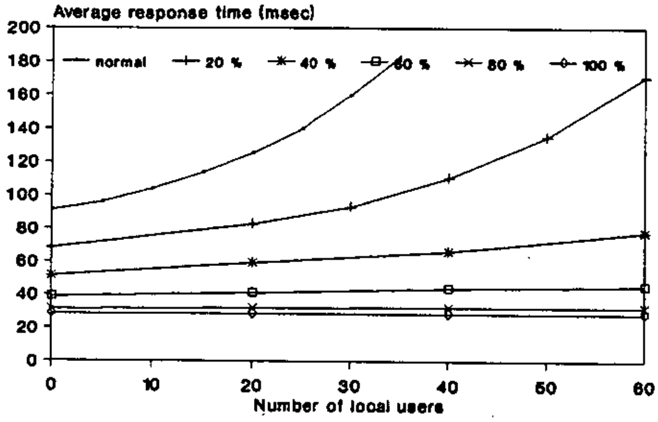


Figure D.9.15 : 50.7Kbytes

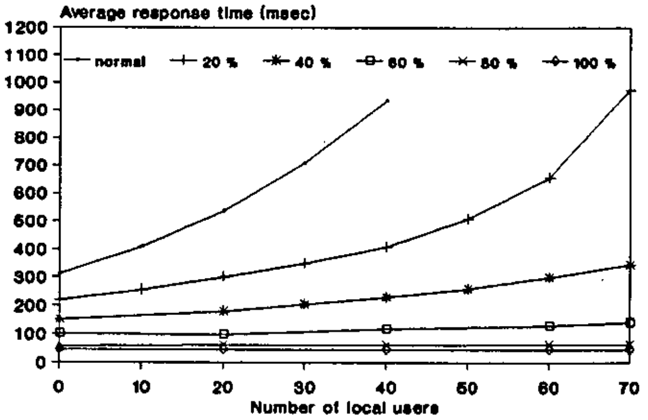


Figure D.9.16 : 316Kbytes(B)

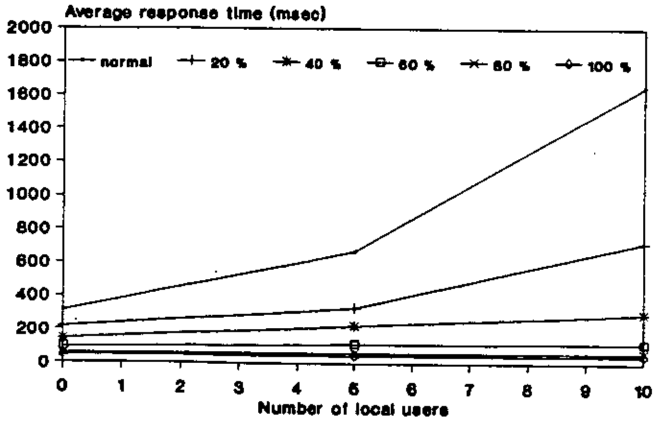


Figure D.9.17 : 316Kbytes

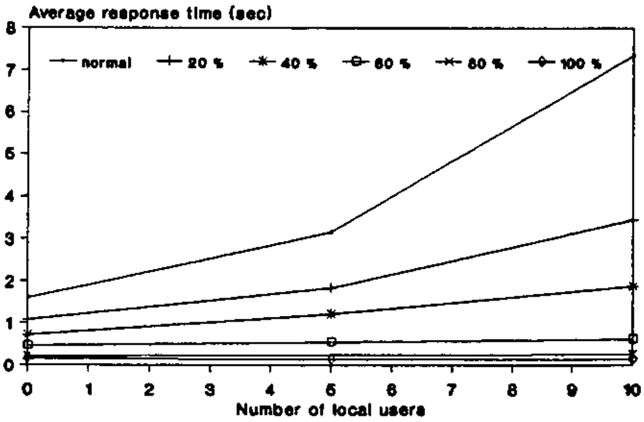


Figure D.9.18 : 1856Kbytes

The effect on the average response time of caching both in the memory and in the disk interface unit of the Sun SPARCstation 10 workstation when the hit rate of the both caches improves at the same time.