PERFORMANCE-DIRECTED SITE SELECTION SYSTEM OF AADMLSS

A Thesis

by

MIEKE PRAJUGO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Computer Science

PERFORMANCE-DIRECTED SITE SELECTION SYSTEM OF AADMLSS

A Thesis

By

MIEKE PRAJUGO

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Valerie E. Taylor (Chair of Committee)	Frank Shipman (Member)
Lynn Burlbaw (Member)	Valerie E. Taylor (Head of Department)

December 2004

Major Subject: Computer Science

ABSTRACT

Performance-Directed Site Selection System of AADMLSS. (December 2004)

Mieke Prajugo, B.S., University of Minnesota

Chair of Advisory Committee: Dr. Valerie E. Taylor

The popularity of the World Wide Web (WWW) in providing a vast array of information has drawn a large number of users in the past few years. The dramatic increase in the number of Internet users, however, has brought undesirable impacts on users, such as long response time and service unavailability. The utilization of multiple servers can be used to reduce adverse impacts. The challenge is to identify a good resource site to allocate to the user given a group of servers from which to select.

In this project, a performance-directed site selection system was developed for a web-based application called AADMLSS (African American Distributed Multiple Learning Styles System). Four different sets of experiments were conducted in this study. In order to evaluate the effectiveness of the test system, two other server selection methods, Load-based and Random-based methods, were implemented for comparative purposes. The experiments were also run during daytime and nighttime to see the impact of network load on the response time.

Experimental results indicate that the performance-directed site selection system outperforms the Load-based and Random-based methods

consistently. The response time is typically high during daytime and low during nighttime, indicating that the network load has an impact on the response time delivered. The results also show that server performance contributes to the overall response time, and network performance is the more dominating factor in determining a good resource site for the user.

ACKNOWLEDGEMENTS

I would like to express my gratitude to many people who have supported me in completing this thesis. I am especially indebted to my advisor, Dr. Valerie E. Taylor, who has provided encouragement, insight, and guidance over the course of the work presented in this thesis. I would also like to extend my appreciation to Dr. Juan E. Gilbert of Auburn University, and Dr. Roscoe Giles & Mr. Ray Giles of Boston University who have provided systems to be used as part of our testbed environment. Finally, I would like to give thanks to my husband and my family who have been understanding, and always given me unconditional support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

1.1 Introduction

The popularity of World Wide Web (WWW) in providing an inclusive resource of information is irrefutable. The existence of the Web has initiated the development of a large variety of web-based applications, ranging from e-commerce, entertainment, to educational systems. The simple use of the web interface in combination with the wide selection of services it offers has drawn a massive number of users. In the past few years, the growth in the number of its users has escalated dramatically.

The rapid escalation of users has brought about a number of adverse impacts on the users. With limited amount of network resources available, the high number of users can cause a significant slowdown in the network traffic. Furthermore, many designs and deployments of web-based applications have been completed largely without much consideration in the performance of handling a large number of users [6]. As a consequence, once the service of a site has become popular, the user response times often climbs significantly along with degraded site availability. As the number of web users increases, delivering good service response time has become a crucial issue.

____

This thesis follows the style of *IEEE Transactions on Parallel and Distributed Systems*.

A number of techniques have been introduced to overcome this problem [1, 5]. Many of these approaches often involve the utilization of multiple servers and the dispersion of these servers geographically. Besides improving scalability of service, these approaches also enhance service availability and performance. Despite the advantages it offers, the utilization of multiple machines as service providers, unfortunately, gives rise to a new problem. With a pool of servers from which to select, the question becomes: How to identify a good resource site to allocate to the user? In other words, how does the system identify a resource site that would deliver the service with good response time to the user.

1.2 Research Objective

The objective of this research project is to develop a system for performance-directed site selection for a web-based application, called AADMLSS. AADMLSS stands for African American Distributed Multiple Learning Styles System. It is an online educational system that incorporates the use of culture and the integration of sophisticated instructional tools into its learning environment as an attempt to improve a student’s learning experience and academic performance [9]. AADMLSS is a collaboration research project among several educational institutions with the goal of advancing African-American communities through the use of innovative information technologies. Auburn University, Boston University, Portland State University, and Texas A&M University are some of the institutions that are involved in this project. A group from each university has its own research core and contributes to AADMLSS in different areas of technology.

AADMLSS uses a hierarchical model to organize the educational material contained in the system. The three component layers that structure the hierarchy, starting from top to bottom, are: COURSE, MODULE, and CONCEPT. The COURSE, which represents the highest layer in the hierarchy, can be viewed as a book. The middle layer, referred to as the MODULE, can be viewed as the author of the book. Each author adopts his/her own unique teaching style to write the book. The CONCEPT, which represents the lowest layer in the hierarchy, represents a chapter in the book. Currently, AADMLSS contains a course for Algebra. The Algebra course consists of four different modules and sixteen different concepts. Additional courses will be added later. A more detailed elaboration of the system organization is discussed in the next chapter.

1.3 Proposed Solution

In order to build a performance-directed site selection system for AADMLSS, a number of contributing factors that affect web scalability and service performance were examined. In [2], Crovella and Carter suggest that poor service response time in distributed information service environment, such as the WWW, are typically due to excessive server load and congested network. Similar observations have also been made by Rajamony and Elnozahy [12].

The proposed solution selects the best site based upon the overall site performance at the time the user accesses AADMLSS. The overall site performance consists of the following two elements, the server performance and the network performance between the server and the user. The server performance represents the elapsed time for the server to respond to user request, without the inclusion of network delay. Four different experiments were conducted in order to evaluate the effectiveness of the site-selection system developed. The quality of the site-selection system developed was compared with those of two other selection methods that employ different selection criteria. The first method determines the best resource site based on the server load. The load measurement is performed dynamically and the user is directed to a server that has the minimum load at the time of access. The second method uses a random number generator to dynamically select a site. A testbed consisting of four servers was used to evaluate the proposed scheme. The testbed consists of two machines located at Texas A&M University, one machine located at Auburn University, and another at Boston University.

The sixteen AADMLSS concepts were replicated across the four different machines. A set of experiments were conducted to assess the proposed system entailing site selection based on server and network performance. Service response time is defined to be the elapsed time from the time a request is initiated by the user to the time the user’s browser completely finishes the loading of the requested service. In this study, service response time refers to the total elapsed time for each file in a specified AADMLSS concept to load completely at the user’s browser.

The experiments were also conducted during daytime (8AM-5PM ET) and nighttime (6PM-12PM ET) to observe whether the network load impacts the overall site performance. The results indicate that the network load during daytime has an effect on the resulting service response time, especially for concepts delivered by the Load-based and the Random-based server selection methods.

1.4 Thesis Organization

The remainder of this thesis outlines as follows: Chapter II provides background information of the system used in this study, AADMLSS; Chapter III describes the resource selection system; Chapter IV provides a discussion on the experimental design and results; Chapter V presents previous works that are related to this study; Chapter VI provides conclusion of the thesis and future direction.

CHAPTER II

AADMLSS

2.1 Background

The site-selection system developed in this project was intended to complement the web-based educational system called African American Distributed Multiple Learning Styles System (AADMLSS). As described earlier, AADMLSS is an online educational system that incorporates the use of culture and the integration of sophisticated instructional tools to provide a culturally-sensitive learning environment. A previous study has shown that student’s learning behaviors are strongly influenced by their social and cultural issues [17]. AADMLSS was developed based on this study as an attempt to enhance student’s learning experience and improve their academic performance.

Most of the learning environments that currently exist adopt the three traditional instructional models [4]. These include the 1-1 model where only one instructor to teach one student, 1-M model where one instructor teaches many students, and the M-M, which is a typical group study setting. As expected, each traditional model offers advantages and disadvantages.

The 1-1 instructional model is often considered to be the best learning environment for students. This is primarily because it allows the instructor to adjust his/her teaching style to the student’s personal needs. Despite the flexibility it offers to the student, this model limits the learning experience gained by the student to the knowledge of a single instructor. The 1-M instructional model represents the classic classroom environment. The advantage of this model is often viewed from the economic standpoint, especially in higher educational institutions. The disadvantage of this model is the instructor generally adopts a teaching style that is of most familiarity. Some students many not have maximum learning experience using this model.

The M-M instructional model, the group setting environment, allows each student to interact and learn from others in the group. Many educational institutions encourage the use of this model outside classroom environment as it enables the student to receive different perspectives to the problem and/or reinforce the already-gained knowledge. Unfortunately, some students may benefit more than others in this environment. Furthermore, every member in the group must be able to accommodate the meeting time into their often inflexible schedule.

AADMLSS attempts to improve the existing instructional models towards an ideal teaching environment with the additional feature of being culturally sensitive. AADMLSS uses M-1 relationship instructional model, where many instructors are available to teach a single student. The primary advantage of this model is its flexibility. AADMLSS allows the student to find a teaching style that can maximize his/her learning ability. Furthermore, it is an online educational system, enabling the student to learn anytime. The only drawback is for the student to determine which teaching style most suits his/her learning behavior, given the various different instructors available. This mapping is one area of future work with AADMLSS.

Essentially, AADMLSS employs the use of innovative technology to provide an advanced learning environment that is culturally sensitive to the students. It also contains a collection of instructional materials and constructs personalized instruction from the relevant materials to accommodate student’s individual learning style. AADMLSS facilitates an advanced online-learning environment through the use of various animated pedagogy that are different with respect to culture, ethnicity, and gender as an effort to improve students’ learning experience and academic performance.

2.2 AADMLSS System Organization

AADMLSS uses a hierarchical model to organize the educational material contained in the system. There are three levels of hierarchy: COURSE, MODULE, and CONCEPT. Figure 1 illustrates the hierarchical model used by AADMLSS system to organize educational materials in the system.

Figure 2 provides an example of an AADMLSS course that uses the hierarchical model to break the materials into finer components. As the figure indicates, the Algebra course is offered to the user by four different teaching methods. For instance, instructor 1 may teach the concept by starting with an example first before describing the concept. Instructor 2 may begin by teaching the concept first and reinforce the user’s understanding by displaying the examples afterward. Furthermore, the instructor may also use different information technology to deliver the concept. For example, instructor 3 may use media visualization to teach the material, while instructor 4

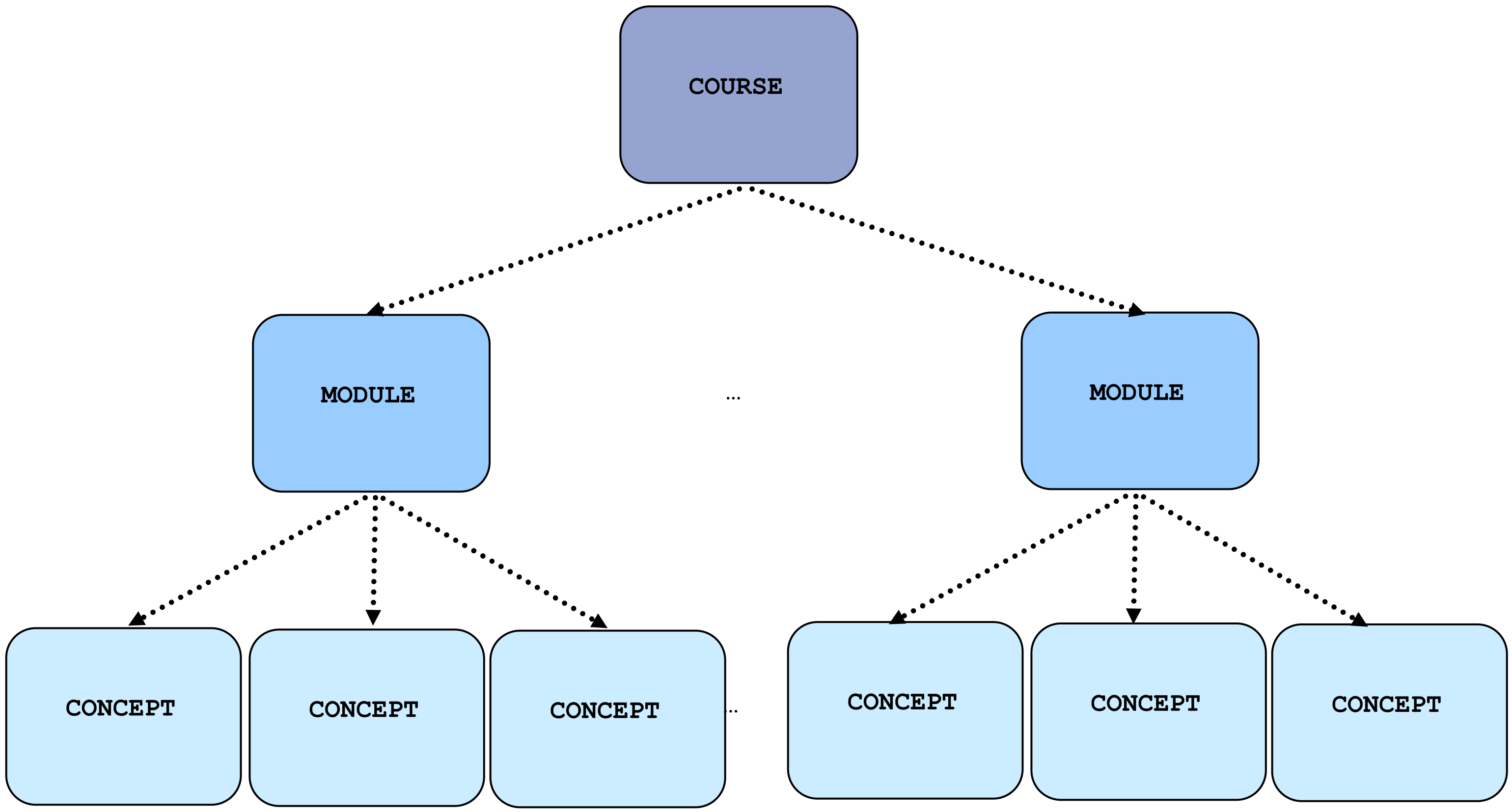


Figure 1: Hierarchical Model for AADMLSS Educational Material

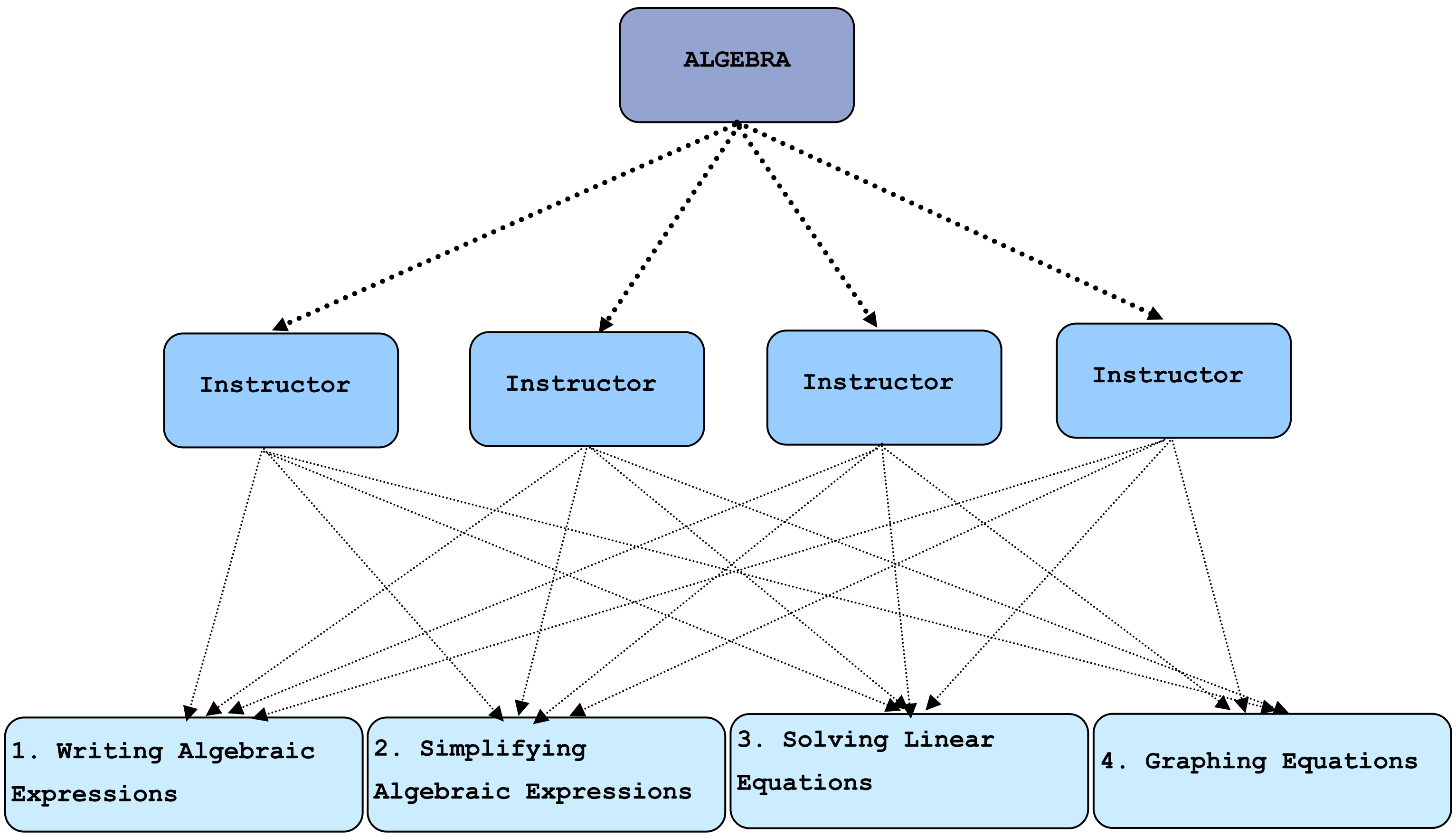


Figure 2. Examples of AADMLSS Course, Modules, and Concepts

may use information technology involving audio technology to deliver the concept.

2.3 AADMLSS System Flow

The user, in this case the student, can access the concepts in AADMLSS with a regular web browser, such as Internet Explorer, Netscape, or Mozilla. A number of AADMLSS concepts may require software installation (e.g Haptek software) on the user machine to guarantee proper execution of the subject materials. AADMLSS uses a database to store information about the user, the instructor, and the educational materials, e.g. courses, modules, and concepts. It also records the user’s progress in the subject material.

In order to access AADMLSS, the user must have a valid username and password. The system connects to the database server, performs username and password authentication, and checks which concept material that was last associated with the user. This information is used to determine which course, module, and concept to be displayed next for the user. If the user has never accessed AADMLSS system, the system will use a default course and assign the first concept to the user. The instructor for the concept will be chosen at random.

At the end of each concept, the system displays a quiz to verify the user’s understanding about the material presented earlier. The user’s score on the quiz will determine whether or not the user passes the concept material. If the score on the quiz is beyond the threshold value assigned for that concept, the system will allow the user to move ahead with the next concept in the course, taught by the same instructor. Otherwise, the student must repeat the previous concept using a different teaching style selected at random by the system. AADMLSS guarantees that although the selection for the teaching style is performed in a random manner, the next instructor chosen for the user will teach the concept with a different teaching style than the instructor the user had previously.

In the near future, AADMLSS will adopt the use of adaptive system, where intelligent agents assist in determining the choice for the next instructor when a student needs to repeat a concept. In addition, the system will also incorporate a groupware environment tool, such as animated chat rooms and interactive video, to facilitate real time interaction among the students and between the student and the instructor.

Currently, the existing AADMLSS system utilizes a single machine to host the web server, the application server, and the database server. The limitation on the hardware capacity of a single machine to perform all the processing required to respond to user requests will result in poor service response time [8]. With increasing number of clients, there is no doubt the system performance would degrade very rapidly. Hence the need to replicate AADMLSS on multiple servers and utilize a performance-directed site selection system.

CHAPTER III

RESOURCE SELECTION SYSTEM

3.1 Overview

The use of a single machine to respond to every user request directed to AADMLSS will immediately create a bottleneck. This is especially true when a large number of users are accessing the website. Furthermore, should the server fail or become unreachable, the service availability of a single machine will diminish very rapidly. In short, relying on a single machine to handle every user requests is undesirable.

The use of multiple machines to respond to high user requests has been known to offer many advantages. This technique improves the system's ability to provide enhanced performance such that the user no longer needs to experience a long wait before he/she receives the requested service. The use of multiple servers also improves service availability. Through the implementation of an appropriate mechanism the system can direct the user to a different server should the default server experience problems in delivering the requested service. The utilization of multiple machines increases the flexibility in the arrangement and distribution of such servers geographically. In addition, the system facilitates its service performance to scale as the number of users increases.

Essentially, a system that is structured from multiple servers is more advantageous than that of a single machine. This type of system offers a great potential to be a powerful service provider for its users. However, there is also a vital challenge implicated in building a multiple-server system. The greatest challenge is to find the server to achieve good performance. To address the challenge, issues related to constructing such systems were examined. The fundamental issue dealing with a multiple-server system appears to be the fact that there is a pool of servers from which the user can select. Given this condition, the question becomes: How to identify a good resource site to allocate the user? Thus, in the replicated system, correct identification of a resource site becomes crucial. The system must be able to select a resource site that delivers good service response time to the user at any time.

3.2 Methodology

The proposed solution focuses on building a system that considers such performance metrics as the selection criteria used to identify the server. The selection of the resource site is determined based on the overall site performance at the time the user accesses AADMLSS. The overall site performance is composed of two components, the server performance and the network performance between the server and the user. In this study, server performance refers to the elapsed time for the server to respond to user request, minus the network delay.

3.2.1 AADMLSS Site-Selection Process Flow

Figure 3 illustrates a process flow of AADMLSS site-selection system. The process starts with the user connecting to an AADMLSS server, which further connects to the central database. This server retrieves performance data from the database and performs the necessary computation to identify a resource site that has good service response time. Next, the server sends user request to the selected site. The site handles the request and sends response back to the user. Further details about each step are given below.

When the user accesses an AADMLSS server, the user sends the request information to the server containing his/her username, password, and IP address of the user's machine. AADMLSS uses the username to find the identification number of course, module and concept materials last associated with the user. This information is used to determine which course, module, and concept to be displayed next to the user.

Service response time is defined to be the elapsed time from the time a request is initiated by the user to the time the user's browser completely finishes the loading of the requested service. In this study, service response time refers to the elapsed time required by an AADMLSS concept to load completely at the user's browser. In order to calculate service response time, the server retrieves the previous measurement of the server performance, collected from each AADMLSS server, for that particular concept material. Server performance is defined to be the elapsed time for the server to respond to user

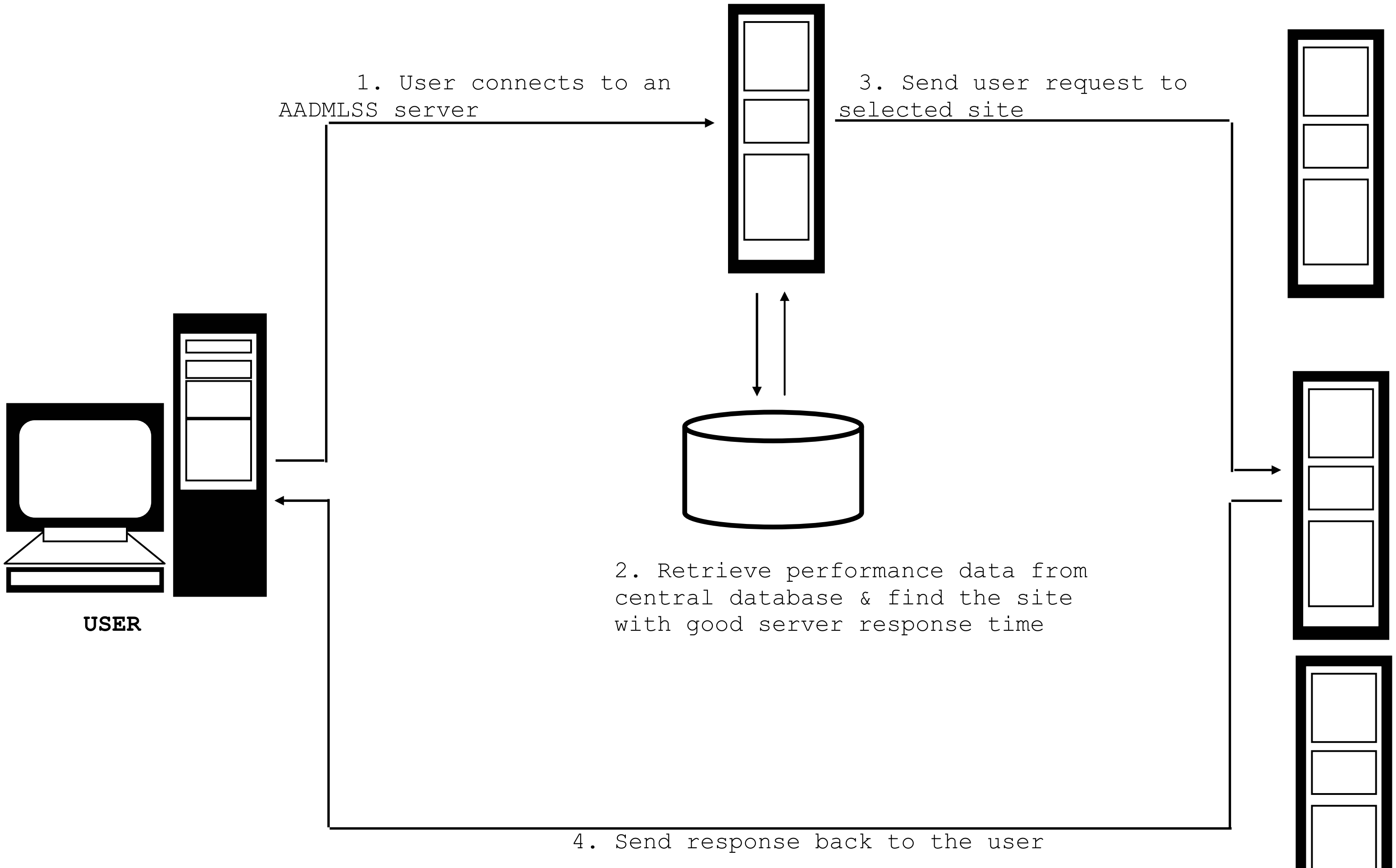


Figure 3. Process Flow of AADMLSS Site-Selection System

request, minus the network delay. The estimated network delay between the user and each candidate server is computed dynamically by using the IP address information contained in the initial user request to the server. Next, the server combines both information, the server performance and the network delay, to correctly identify a good resource site that would give good response time for the specified concept. Once the site has been identified, the system automatically directs the user to the selected site. Detailed description on how the server performance and the network performance were measured, which together compose the service response time, are given in the following sections.

3.2.2 Measuring Server Performance

To measure the server performance, every file in the AADMLSS concepts was instrumented. Instrumentation of the file permits correct measurement of the time required for the requested file to complete loading on the user's browser, e.g. *file response time*. The HTML event-handler mechanisms and client-side scripting languages, such as JavaScript, facilitate the measurement and collection of the actual file response time. This technique has been known to generate low overhead [20]. In our case, the overhead is less than 5%.

The measurement of file response time for each file is stored into a central database. The sum of file response time for all the files in a given AADMLSS concept reflects both the server performance and the network delay involved in accessing that concept. Note that the variable Network Delay stated in the following equation represents the network delay for the requested concept, rather than for a single file. The quality of server performance is computed dynamically whenever a user needs to access that concept. The computation is performed by using the following formula:

$$\textit{Server Access Time}(t) = \Sigma \textit{ File Response Time}(t) - \textit{Network Delay}(t) \qquad (1)$$

Server Access Time: the elapsed time for the server to respond to the requested concept material, without the inclusion of network delay

File Response Time: the elapsed time from the time the user sends his/her request to the server to the time the requested file completes loading on the user's browser.

Network delay: the estimated network time required for the user to access the requested concept material from the selected server.

3.2.3 Measuring Network Performance

The current implementation of AADMLSS system determines the network performance between the user and each server dynamically. As mentioned earlier, network delay is defined to be the estimated network time required for the user to access the concept material from the selected server. The estimation on the network delay is accomplished by using the standard UNIX networking utility: ***ping***. This utility uses ICMP protocol to send and request data and places a timestamp in each packet to facilitate easy computation of packet round trip-time [11]. The central database server used in this study consists of the database used with the Prophesy infrastructure. Prophesy provides a web-based performance analysis and modeling infrastructure for distributed and parallel applications. Prophesy database stores the information related to AADMLSS learning environment. For example, it contains information in regards to the user's learning progress, the instructor that a particular student uses, and the location of educational materials. In addition, Prophesy stores information that is necessary for performing performance-directed site-selection in AADMLSS. It contains information about each of the servers composing AADMLSS, and its IP address. The database is also used to archive performance information, in particular data with respect to server and network performance. Prophesy assists the performance-directed site-selection system in generating an appropriate analytical model for the network performance between the user and each candidate server.

In order to estimate for the network delay, four different packet sizes were sent from each candidate server to the user machine. The IP address of the user machine is obtained from the initial user request to the server, as previously mentioned. The server IP address is retrieved from the database server. The four packet sizes used in this study are 64B, 128B, 256B, and 512B. The packet sizes were chosen as an attempt to minimize the overhead. The ping command was executed three times consecutively for each packet. The average packet round-trip time is recorded and sent to Prophesy for further analysis. Using the collected performance data, Prophesy produces an appropriate mathematical model for the network performance using least squares fit to the affine function. The resulting model equation is used to compute the estimated network delay. The general form of the model equation is

$$\text{Network Delay}(t) = \alpha + \beta * \text{Concept Size} \qquad (2)$$

as follows:

α: a constant

β: a line gradient

Concept Size: total file size in an AADMLSS concept.

It was acknowledged that there are a number of network forecasting tools that can readily be used to collect and gives reasonably accurate prediction regarding end-to-end network bandwidth and latency [15]. However, it is not suitable for the project. The users of AADMLSS are not a fixed group of users, but rather arbitrary users. The requirement of software installation at the user end hinders us from integrating this tool into AADMLSS. Furthermore, the network monitoring tools require all the replicas to be located outside the firewall.

3.2.4 Combining Server and Network Performance

Once the performance data with respect to server and network performance have been obtained, the service response time is computed using the following formula:

Service Response Time(t)=Server Access Time(t-1)+Network Delay(t) (3)

Service Response Time: the elapsed time from the time a request is initiated by the user to the time the user's browser completely finishes loading the requested AADMLSS concept.

Equation (3) shows that the service response time is computed by using the estimated network delay and the previous measurement of server access time on the specified concept material. The service response time at any time is calculated for each of the candidate servers. The server, which the user initially connects to, selects the system that displays the best overall site performance and automatically directs the user to access the concept material from that server.

CHAPTER IV

EXPERIMENTAL DESIGN AND RESULTS

4.1 Testbed Environment

In order to assess the quality of the site-selection system developed, a testbed consisting of four servers was constructed. The testbed consists of two machines located at Texas A&M University, one machine located at Auburn University and another at Boston University. Each AADMLSS concept was replicated across these four different servers. The hardware specification of each replica is given in Table 1.

As shown in Table 1, the two local experimental servers, Loner and Prophesy, are fairly different in terms of their hardware characteristics. While Loner is a typical PC desktop with CPU speed of 997 MHz, Prophesy is a server designed for high-performance usage with CPU speed of 3056.85 MHz.

Table 1. Hardware Specifications of Testbed Server Replica

SPECIFICATION	Loner (Local)	Prophesy (Local)	Interact (AL)	Tina (MA)
CPU speed (MHz)	997.62	3056.85	697.87	1993.56
Bus Speed (MB/s)	205	856	214	638
Memory (MB)	256	2048	256	256
Hard Disk (GB)	30	146	10	40

Furthermore, Prophesy has superior memory and hard disk size of 2048 MB and 146 GB respectively compared to Loner's 256 MB and 30 GB. The other two remote servers, Interact and Tina, are more similar to Loner both in terms of CPU speed and memory size. However, the hard disk size of the Interact machine is extremely low compared to the other three servers.

Software installed in the three server replicas: Loner, Tina, and Interact, include Redhat Linux 9.0 and Apache Web Server 2.0. Loner and Tina use PHP 4.2., while Interact uses PHP 4.1. Software installed in Prophesy includes Redhat Linux Enterprise 3.0, Apache Web Server 2.0, PHP 4.3, and MySQL 5.0.

4.2 Experimental Design

Currently, there are sixteen different concepts with the AADMLSS system. The profile information of all the sixteen concepts are displayed in Table 2. Four different experiments were conducted in order to evaluate the effectiveness of the site-selection system developed. The first set of experiments involved all four servers, e.g. 4-servers. The 4-server experiments were conducted both during daytime (8AM – 5PM ET) and nighttime (6PM - 12PM ET). The purpose is to observe whether there is an impact on the server performance due to changes in network traffic. The fact that Prophesy is a local machine and has superior hardware capacity causes the inclusion of Prophesy in this set

Table 2. AADMLSS Concept Profile Information

Teacher	Technology	Description	Concept ID	Number of Files	Avg. Size (KB)
Steve	Animated Agent	1. Writing Algebraic Expressions	3/0/0	24	6.29
		2. Simplifying Algebraic Expressions	3/0/1	17	9.35
		3. Solving Linear Equations	3/0/2	37	17.46
		4. Graphing Equations	3/0/3	64	12.47
	Video	1. Writing Algebraic Expressions	3/1/0	1	25,204
		2. Simplifying Algebraic Expressions	3/1/1	1	13,732
		3. Solving Linear Equations	3/1/2	1	60,654
		4. Graphing Equations	3/1/3	1	50,951
Dwight	Animated Agent	1. Writing Algebraic Expressions	3/2/0	10	12
		2. Simplifying Algebraic Expressions	3/2/1	13	10.08
		3. Solving Linear Equations	3/2/2	19	10.65
		4. Graphing Equations	3/2/3	24	11.54
	Video	1. Writing Algebraic Expressions	3/3/0	1	26,284
		2. Simplifying Algebraic Expressions	3/3/1	1	28,310
		3. Solving Linear Equations	3/3/2	1	27,788
		4. Graphing Equations	3/3/3	1	39,904

of experiments to result in the SRT-based method choosing Prophesy consistently.

In order to appreciate the effectiveness of the SRT-based method in identifying the best resource site, a second set of experiments was conducted. The set of experiments included only three servers, e.g. 3-servers, that have comparable hardware specifications, which excludes Prophesy. In order to study the effect of network delay on the service response time, two servers consisting of a local machine (Loner) and a remote machine (Interact) of comparable hardware capacities were used, e.g. 2-Servers Local-Remote. To study the effect of server performance on the service response time, the two remote servers that have comparable network performance were used, e.g. 2-Servers Remote-Remote.

The performance-directed site selection system developed for AADMLSS uses a combination of both server performance and network delay as the selection criteria to dynamically determine the best resource site at any time. In order to evaluate the service performance delivered by this service response time-based method, two other selection methods were implemented for comparative purpose.

The first selection method merely uses server load as selection criteria. Thus, it dynamically identifies a resource site based on the load and directs the user to a server that has minimum load at any time. In order to do this, the standard UNIX resource monitoring utility ***uptime*** was used. Load average has been known to provide valuable information in identifying a "busy" server [5]. The value reported is computed by using an exponentially weighted moving average technique where more weight is given to the latest measurement value. The value represents the number of active processes running on the server at any given time. Using this method, the site which gives the minimum one-minute load average at the time of the measurement is selected.

As reported by Dinda and O'Hallaron, the use of the last measurement value to perform load prediction does not provide good estimate of future machine load [3]. We acknowledged a number of performance forecasting tools that include load average data collection, which can readily be used to give reasonably accurate prediction. Unfortunately, the tool requires all server replicas to be located outside the firewall, which poses security risk to the system.

The second comparative selection method implemented in this project uses a random number generator to select a site dynamically. To avoid using the same sequence of random numbers for each run, the number generator function uses a random seed value. The seed value is generated by a built-in function in PHP. For each user invocation of the system, a different seed value is used. The purpose is to guarantee that the selection of resource site is performed in a random manner.

Each of the sixteen AADMLSS concepts was executed 350 times using the three different selection methods. As mentioned previously, the purpose of implementing the two selection methods described earlier is to provide a means to appreciate the performance gain obtained from using server performance and network delay as the selection criteria. The three server selection methods are compared in terms of service response time encountered by the user. Their results are discussed in the next section. The user for these experiments originated from a machine located at Texas A&M University.

4.3 Results

4.3.1 4-Server Experiments

The 4-Server experiments include all four servers in the testbed. Figure 4(a) and 4(b) show the average service response time for concepts with animated pedagogical agent as instructors during daytime and nighttime, respectively. Figure 5(a) and 5(b) show the average service response time for concepts taught via video file during daytime and nighttime, respectively.

As shown, Figure 4(a) & 5(a) clearly indicate SRT-based method, which based its selection on the combination of server performance and network delay, outperforms the Load-based and Random-based selection methods. Note that in the figures and tables that follow, the following notations will be used:

SRT: Service Response Time-based selection method (proposed method)

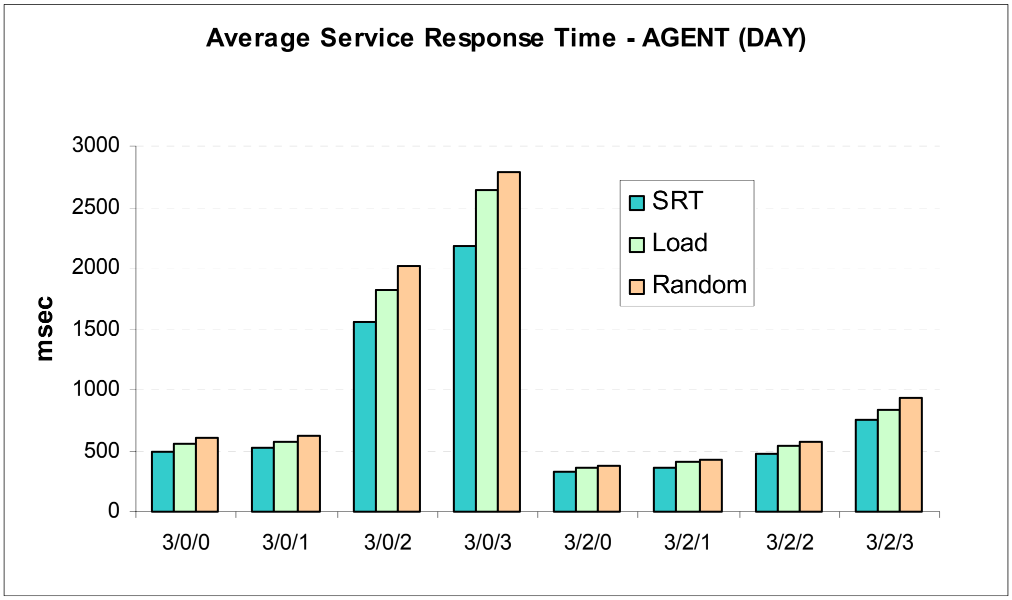


Figure 4(a). 4-Servers Average Service Response Time – AGENT (DAY)

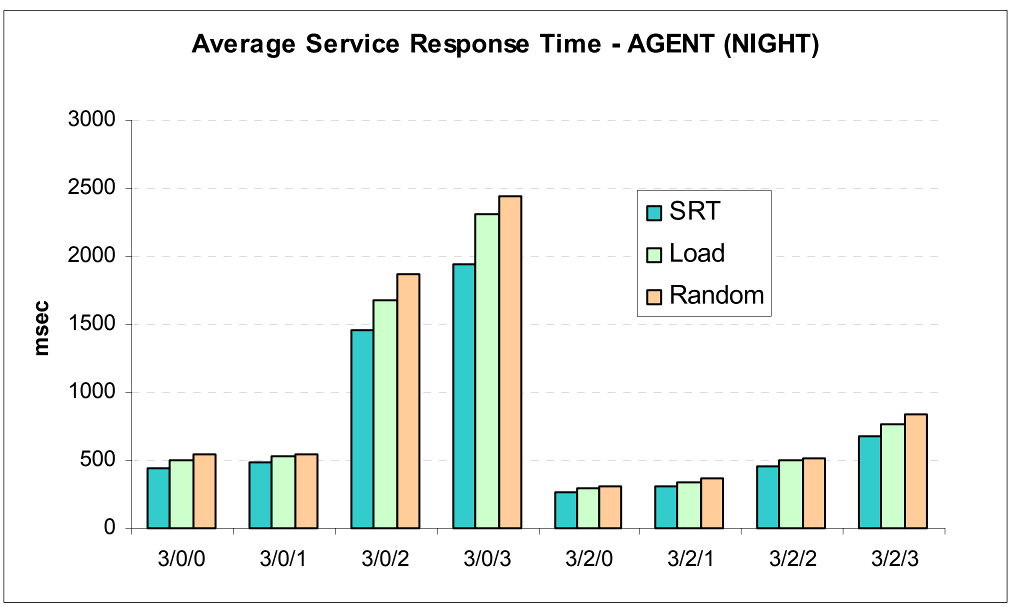


Figure 4(b). 4-Servers Average Service Response Time – AGENT (NIGHT)

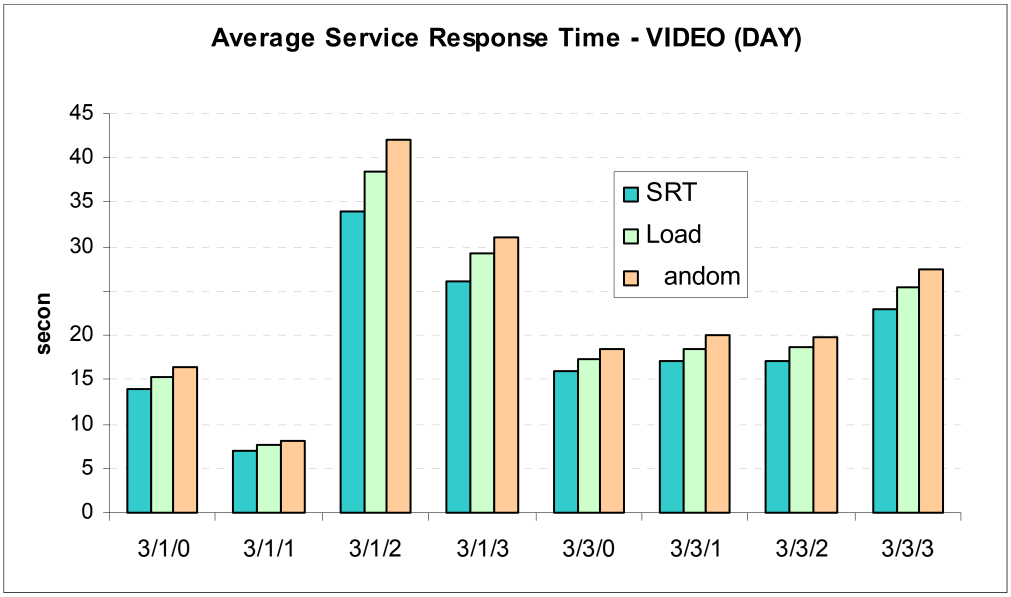


Figure 5(a). 4-Servers Average Service Response Time – VIDEO (DAY)

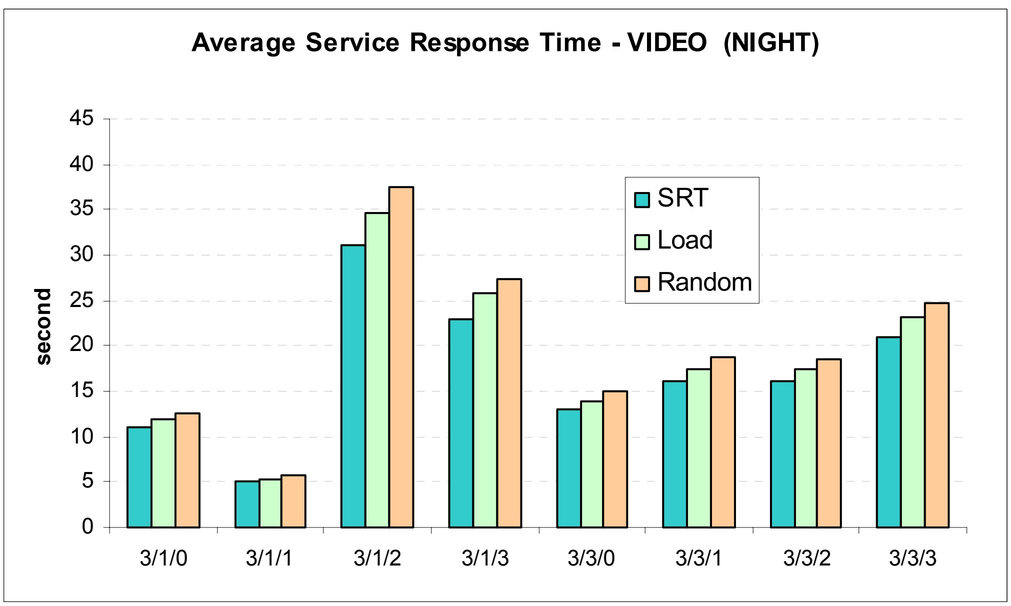


Figure 5(b). 4-Servers Average Service Response Time – VIDEO (NIGHT)

LOAD: Load-based selection method

RANDOM: Random-based server selection method.

Table 3 shows the percentage difference in the service performance for the 4-server experiment among the three selection methods for the sixteen AADMLSS concept. The notations used on the first row are explained below:

SRT-LOAD: percent difference between SRT and LOAD performance

SRT-RANDOM: percent difference between SRT and RANDOM performance

LOAD-RANDOM: percent difference between LOAD and RANDOM performance.

Thus, for example, SRT-LOAD represents percent difference in the performance between Service Response Time-based method (SRT) and the Load-based method (LOAD). The values reported on column two of the table are calculated using the following formula:

$$\text{SRT-LOAD} = \frac{\text{Response Time (LOAD)} - \text{Response Time (SRT)}}{\text{Response Time (SRT)}} \times 100\% \quad (4)$$

Table 3 shows the SRT-based method outperforms the Load-based and the Random-based. Results from the study show that the Load-based method consistently performs better than Random-based method. The performance gain of the SRT-based method with respect to the other two comparative

Table 3. 4-Servers Percent Difference in Service Response Time

	DAY			NIGHT		
CONCEPT	**SRT-LOAD(%)**	**SRT-RANDOM(%)**	**LOAD-RANDOM(%)**	**SRT-LOAD(%)**	**SRT-RANDOM(%)**	**LOAD-RANDOM(%)**
3/0/0	9.75	16.97	6.58	8.76	13.54	4.39
3/0/1	12.58	24.76	10.82	12.30	22.54	9.12
3/0/2	16.75	29.70	11.10	15.75	28.95	11.40
3/0/3	20.54	27.10	5.44	18.75	25.54	5.72
3/1/0	9.14	16.92	7.13	8.76	13.96	4.78
3/1/1	8.67	15.76	6.52	8.01	14.15	5.68
3/1/2	13.38	23.57	8.99	11.94	20.67	7.80
3/1/3	12.16	19.76	6.00	11.87	19.11	6.47
3/2/0	8.95	15.15	5.69	8.64	15.09	5.94
3/2/1	11.57	17.40	5.22	9.95	15.54	5.08
3/2/2	10.95	19.75	7.93	9.60	15.27	5.17
3/2/3	11.04	23.08	10.84	12.54	22.84	9.15
3/3/0	8.91	15.94	6.45	7.69	15.91	7.63
3/3/1	9.07	17.90	8.10	8.47	16.95	7.82
3/3/2	9.46	16.77	6.68	9.31	15.76	5.90
3/3/3	10.55	19.57	8.16	9.87	17.95	7.35
AVERAGE	11.47	20.01	7.60	10.76	18.36	6.84

methods is attributed to the fact that one of the replicas used in the testbed environment, Prophesy, has hardware specifications that give the server a considerable advantage in terms of CPU speed, memory and hard disk size among the remaining server replicas. Furthermore, Prophesy is located within the campus permitting it to have another advantage in terms of network performance.

The difference in the performance between the SRT-based and the Random-based method is larger when the majority of files used in the concepts contain embedded images, rather than plain text. The larger the file, the larger the difference in the service response time between the two methods.

The difference between the SRT-based and the Load-based selection method was not as large as that observed between the SRT-based and the Random-based selection. Figure 4(b) and 5(b), displaying average service response time during nighttime, clearly show that despite the reduced network traffic during night time, the Load-based method cannot outperform the SRT-based selection method.

During the experimental study, the load values for each server were recorded into the database. Table 4 shows the average of load for each replica for the four set of experiments conducted in the study. As shown in Table 4, the load average of Prophesy, in the 4-server experiment, was lower compared to those of other replicas. However, the fact that the load average value reported by the UNIX uptime utility may fluctuate considerably over a short period of time causes the Load-based method to select Loner and Tina occasionally during the experimental study.

As shown in Table 3, the percent difference between each selection method was often higher during daytime than the nighttime. The results indicate that the network load during daytime seems to have an effect on the service response time, especially for concepts delivered by Load-based and Random-based selection methods. The number of processes running on the servers was also observed to decrease during the night. Figure 5(a) and 5(b), which show that the average service response time for each video file, also indicate similar observation to those using agents as instructors. The average service response time for concepts using video files was typically high during daytime and low during nighttime. The difference in response time among the three selection methods observed to be larger for larger video files.

Table 4. Server Load Average Values

SERVER NAME	4-Servers		3-Servers		2-Servers (Local-Remote)		2-Servers (Remote-Remote)
	Day	Night	Day	Night	Day	Night	Day
Loner	0.38	0.36	0.37	0.35	0.38	0.36	N/A
Prophesy	0.33	0.28	N/A	N/A	N/A	N/A	N/A
Tina	0.35	0.31	0.36	0.33	N/A	N/A	0.35
Interact	0.42	0.39	0.39	0.36	0.39	0.37	0.38

A more elaborate data analysis with respect to individual file response time from each concept displays similar findings. The average file response time is typically higher during the day, even for those that do not contain embedded images. The figures resulting from experimental results with regards to file access time for each concept can be viewed on pages 44-51. The corresponding content type and file size are also listed with each figure.

Performance data has also been carefully analyzed to determine how well each of the server selection method chooses its replica. Figure 6, which shows the site selection distribution for each method during daytime and nighttime, indicates that the Random-based method selects its replica most fairly. The distributions of site selection for the other two methods are moderately skewed, especially those of SRT-based selection method. Both figures indicate that the performance gain obtained from using SRT-based selection method was due to the consistent selection of Prophesy machine as the best resource site. As mentioned earlier, besides its superior hardware capacity, Prophesy is a local machine located within the campus. Choosing Prophesy has the advantage of both the server load and the reduced network delay. Figure 7(a) and 7(b) provide the performance comparison of each AADMLSS server in terms of average file access time for concepts using animated pedagogical agents. Both figures show that Prophesy gives the minimum file access time on average, followed by Loner, Tina and lastly, Interact. The contributing factors to the poor performance of the latter replica may be due to the higher load average both during the daytime and nighttime and the lower CPU speed compared to the remaining three replicas. Similar observations were also made in the case of concepts using video technology, shown in Figure 8(a) and 8(b).

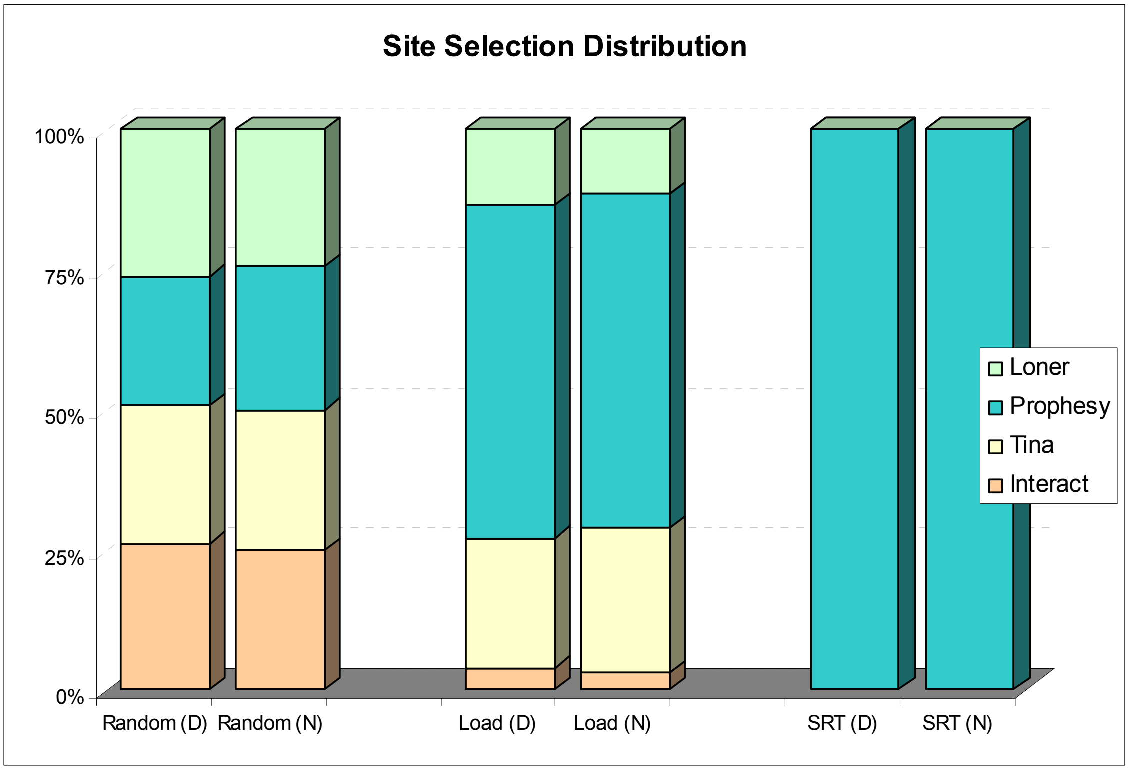


Figure 6. 4-Servers Site Selection Distribution

4.3.2 3-Server Experiments

The 3-Server experiments include the three servers with comparable hardware specifications, which exclude Prophesy. The SRT-based method outperforms the Load-based and Random-based methods by 10.04% and 17.24% on average, respectively (see Table 5). Figure 9, which displays the site selection distribution for each site-selection method during daytime and nighttime, shows that the SRT-based method chose the local machine for the majority of the time. The fact that the SRT-based method chose Loner indicates that in our experiment, network delay is a dominating factor in identifying a good resource site to allocate to the user.

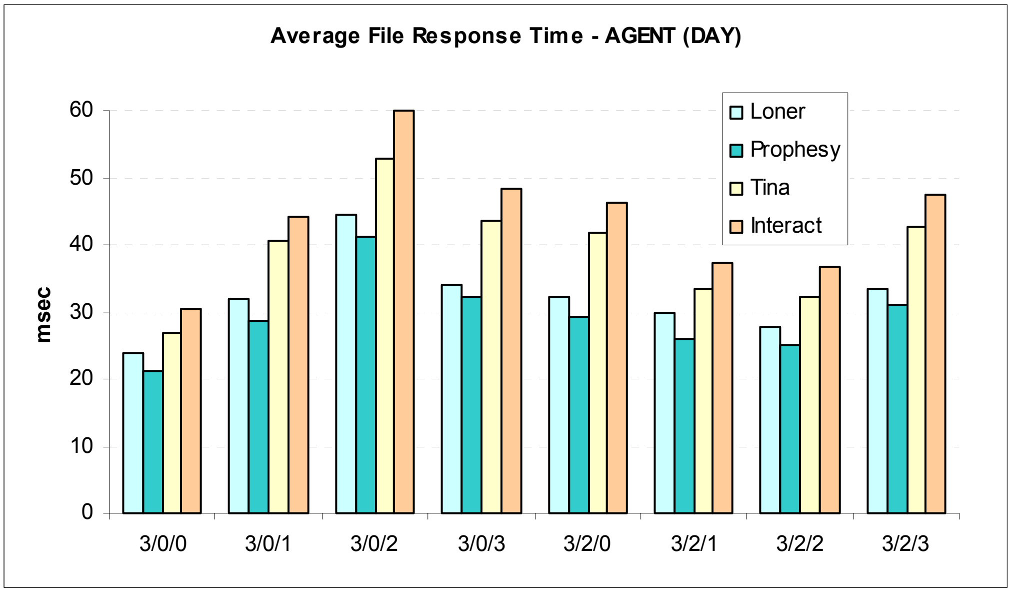


Figure 7(a). 4-Servers Average File Response Time – AGENT (DAY)

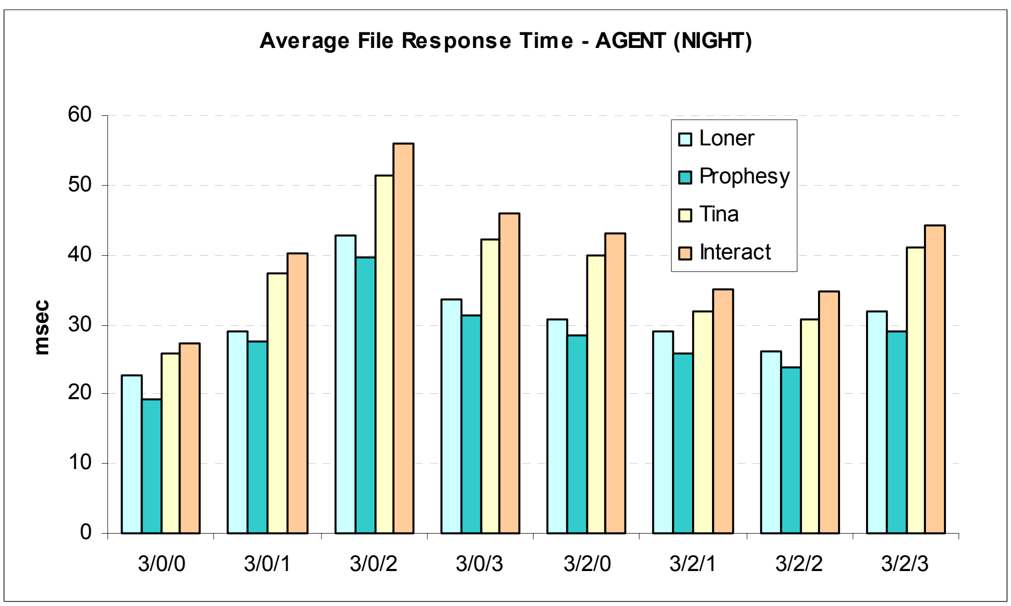


Figure 7(b). 4-Servers Average File Response Time – AGENT (NIGHT)

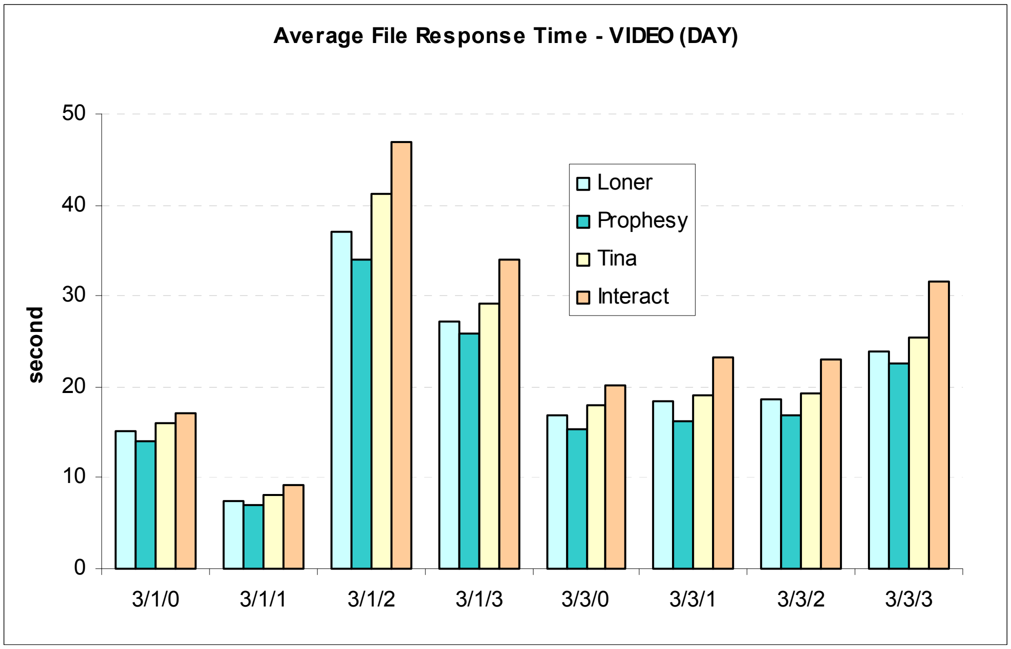


Figure 8(a). 4-Servers Average File Response Time – VIDEO (DAY)

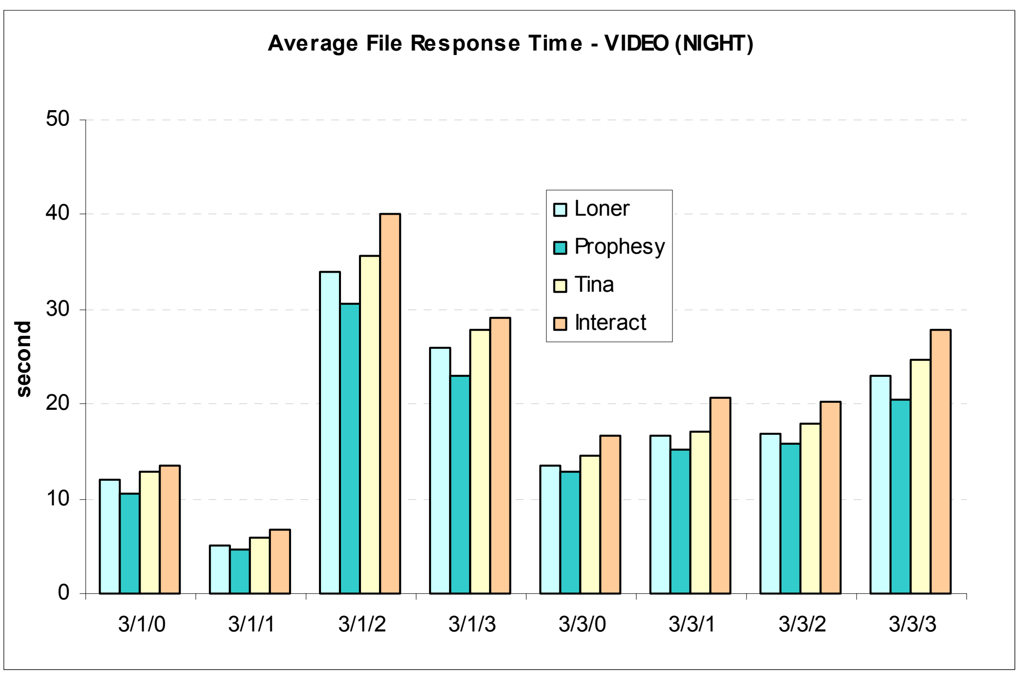


Figure 8(b). 4-Servers Average File Response Time – VIDEO (NIGHT)

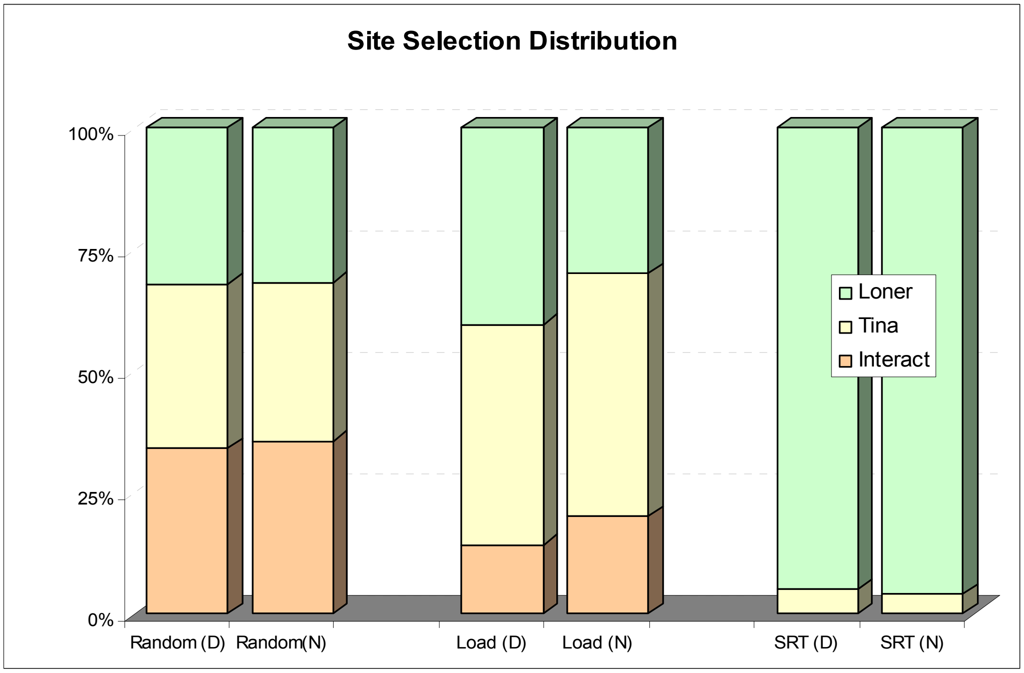


Figure 9. 3-Servers Site Selection Distribution

The fact that the server loads for all the three servers were relatively low allow Loner to give a comparable server performance when compared to Tina. Even the server performance of Interact, the most inferior server of all the three replicas, was still found to be fairly comparable to the other two machines. The Load-based method suffered poorly as it selects Tina for the majority of the time merely due to its low server load during the experimental study.

Figure 10 shows the fraction of server and network performance attributed to the overall response time. As shown, the network delay between the user and the remote machines gives the local machine a considerable advantage in terms of overall response time. The advantage on the network performance is more apparent for larger file sizes.

Table 5. 3-Servers Percent Difference in Service Response Time

CONCEPT	SRT-LOAD(%)	SRT-RANDOM (%)	LOAD-RANDOM (%)
3/0/0 D	6.21	14.05	5.76
3/0/1 D	12.13	21.94	9.73
3/0/2 N	14.02	25.83	10.36
3/0/3 N	18.12	23.52	5.90
3/1/0 N	8.05	12.04	3.69
3/1/1 N	7.31	12.25	4.60
3/1/2 N	12.60	18.74	5.45
3/1/3 N	10.96	19.11	7.34
3/2/0 N	7.93	12.58	5.64
3/2/1 N	8.05	14.25	5.74
3/2/2 N	9.14	15.97	6.26
3/2/3 D	9.79	20.58	9.83
3/3/0 D	8.94	13.64	3.33
3/3/1 D	8.26	16.74	7.83
3/3/2 D	9.21	15.21	4.06
3/3/3 D	9.97	19.36	8.54
AVERAGE	10.04	17.24	6.50

Thus, the results indicate that in our experiments, the network delay is an influential factor in determining the response time experienced by the user compared to server performance.

4.3.3 2-Server Experiments (Local-Remote)

The 2-Server Experiments (Local-Remote) include a local machine, Loner, and a remote machine, Interact, that have comparable hardware specifications. The purpose is to study the impact of network delay on the service response time. Table 6 shows that the performance gain obtained by the SRT-based method to be 10.83% and 11.34% on average

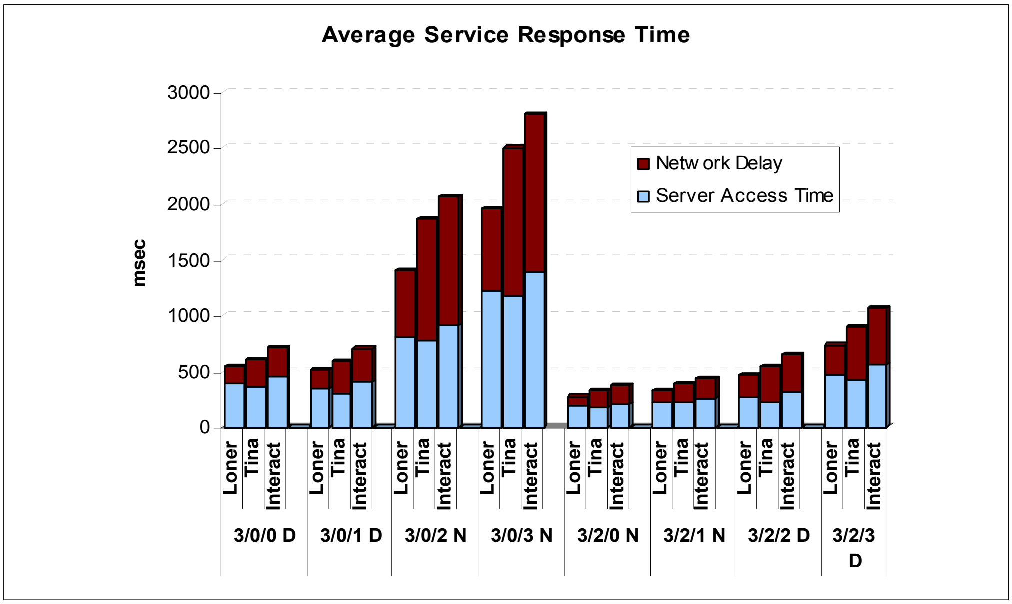


Figure 10. Fractions of Server Access Time and Network Delay

with respect to the Load-based and the Random-based methods, respectively. These values confirms the statements made earlier regarding network delay between the user and the server as an important part in selecting a good resource site to allocate to the user.

Table 6 also shows that there is a negligible difference in the performance gain from using the Load-based method compared to that of Random-based method. This result was largely due to the fact that the server load values between the two machines were very closed in comparison. Since the load average may fluctuate considerably over a short period of time, the Load-based method chooses the Loner and Interact almost in alternate. The site selection distribution for this set of experiment is shown in Figure 11.

Table 6. (Local-Remote) Percent Difference in Service Response Time

CONCEPT	SRT-LOAD (%)	SRT-RANDOM (%)	LOAD-RANDOM (%)
3/0/0 D	9.91	10.24	0.00
3/0/1 D	13.04	15.06	0.01
3/0/2 D	18.06	19.16	0.01
3/0/3 D	20.54	21.29	0.01
3/1/0 N	9.81	9.58	0.00
3/1/1 N	7.02	7.91	0.02
3/1/2 N	11.35	12.15	0.01
3/1/3 N	10.47	10.36	0.01
3/2/0 D	8.56	8.67	0.00
3/2/1 D	8.75	9.75	0.01
3/2/2 D	10.06	10.92	0.01
3/2/3 D	10.15	10.50	0.00
3/3/0 N	8.41	9.56	0.02
3/3/1 N	8.58	8.08	0.01
3/3/2 N	8.31	7.95	0.01
3/3/3 N	10.21	10.19	0.01
AVERAGE	10.83	11.34	0.01

4.3.4 2-Server Experiments (Remote-Remote)

The 2-Server Experiments (Remote-Remote) include the two servers located remotely. From previous experiments, Tina and Interact were observed to have a comparable network performance. The purpose of this experiment is to study the impact of network delay on the service response time. Table 7 shows that the performance gain obtained by the SRT-based method were 4.71% and 5.62% on average with respect to Load-based and Random-based methods, respectively.

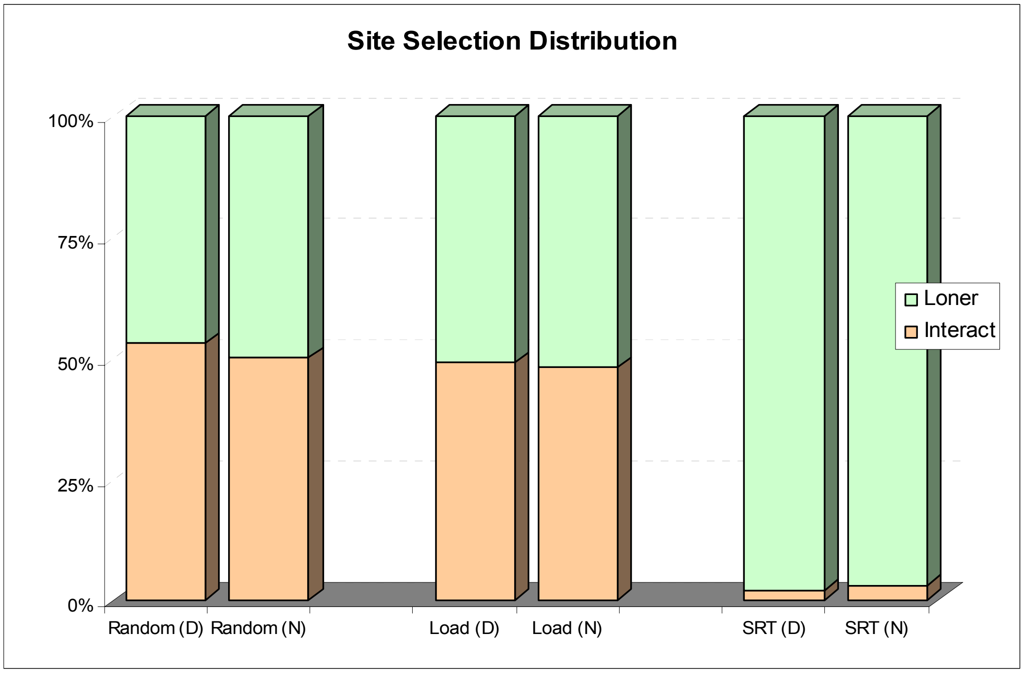


Figure 11. 2-Server (Local-Remote) Site Selection Distribution

These values indicate that the hardware capacity of a server has influential impact on the overall service response time, although the impact was not as large as in the case of network delay. The site selection distribution for this set of 2-servers experiment is shown in Figure 12. Following Figure 12 are Figures 13-20, which show the results of file access time for each concept from the 4-Servers experiment.

Table 7. (Remote-Remote) Percent Difference in Service Response Time

CONCEPT	SRT-LOAD (%)	SRT-RANDOM (%)	LOAD-RANDOM (%)
3/0/0 D	3.13	4.03	0.01
3/0/1 D	4.26	5.97	0.02
3/0/2 D	7.02	8.28	0.01
3/0/3 D	8.64	9.02	0.00
3/1/0 D	3.25	4.94	0.02
3/1/1 D	3.27	4.10	0.01
3/1/2 D	3.93	5.97	0.02
3/1/3 D	3.64	4.08	0.01
3/2/0 D	3.15	3.32	0.02
3/2/1 D	4.39	5.20	0.01
3/2/2 D	5.80	5.97	0.01
3/2/3 D	6.52	6.95	0.00
3/3/0 D	4.39	5.64	0.02
3/3/1 D	4.16	5.20	0.01
3/3/2 D	4.81	5.73	0.01
3/3/3 D	5.02	5.58	0.00
AVERAGE	4.71	5.62	0.01

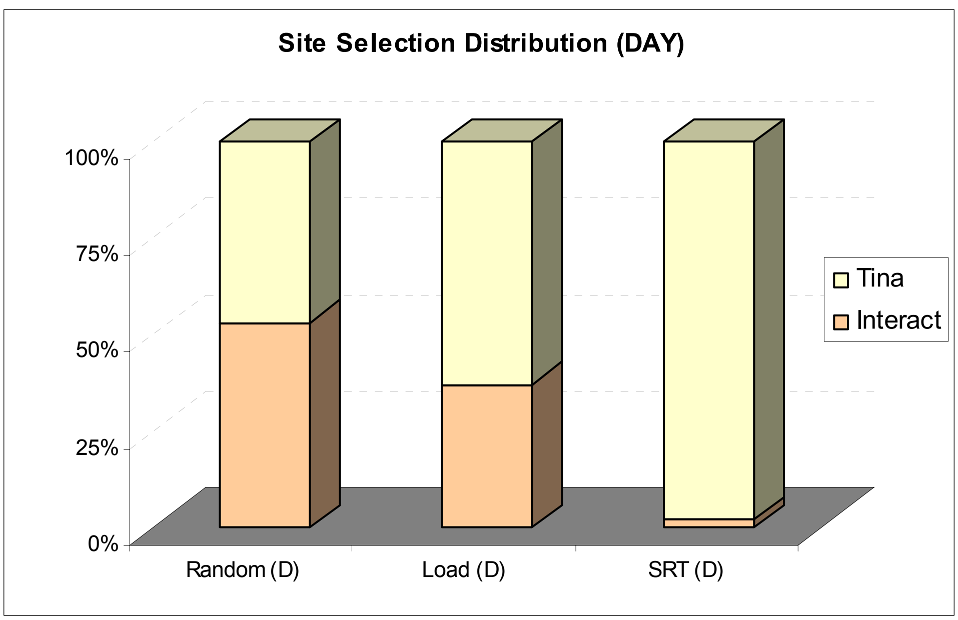
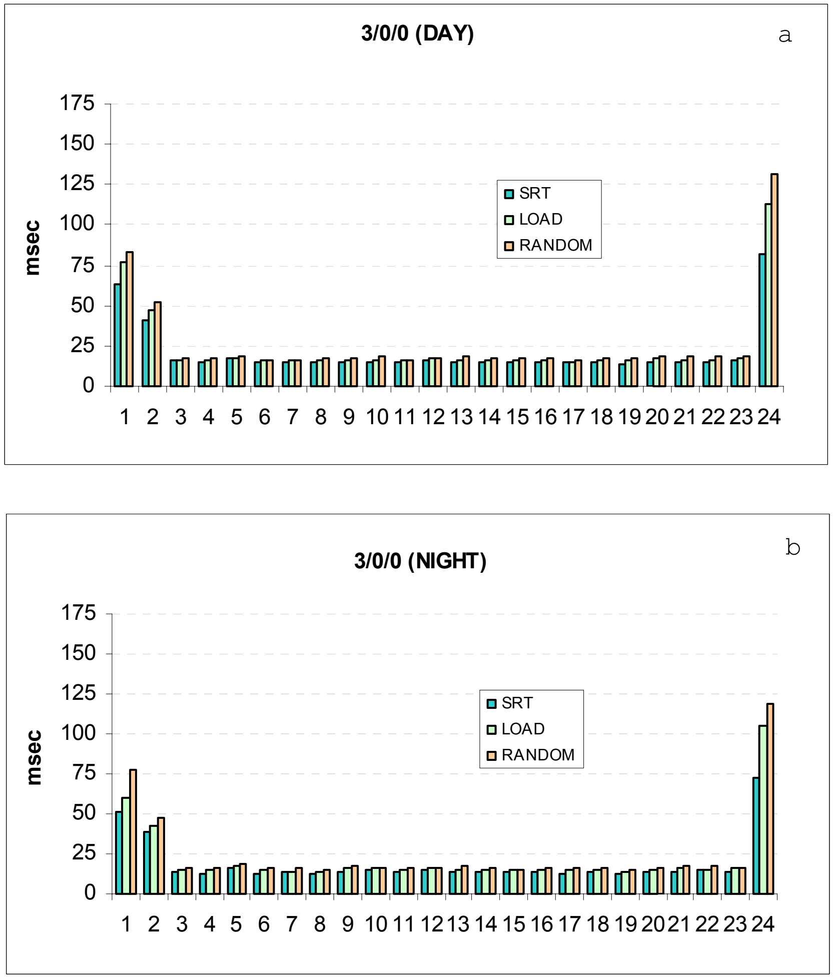
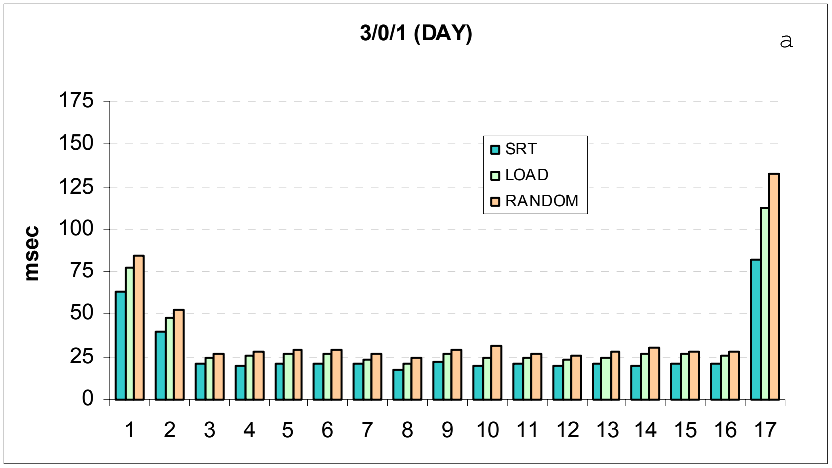


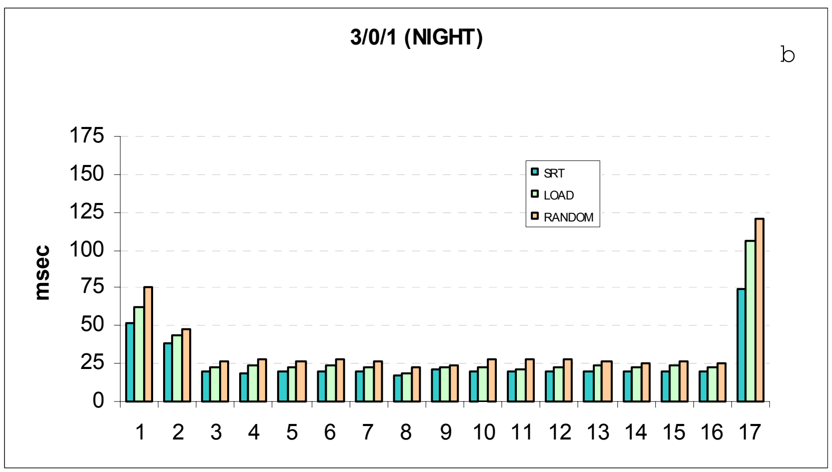
Figure 12. 2-Server (Remote-Remote) Site Selection Distribution



File ID	Content	Total (KB)
1	Text + Img	32
2	Text + Img	17
3	Text	2
4	Text	2
5	Text	4
6	Text	2
7	Text	2
8	Text	2
9	Text	3
10	Text	3
11	Text	2
12	Text	3
13	Text	3
14	Text	2
15	Text	2
16	Text	2
17	Text	2
18	Text	2
19	Text	2
20	Text	3
21	Text	3
22	Text	3
23	Text	3
24	Text + Img	50

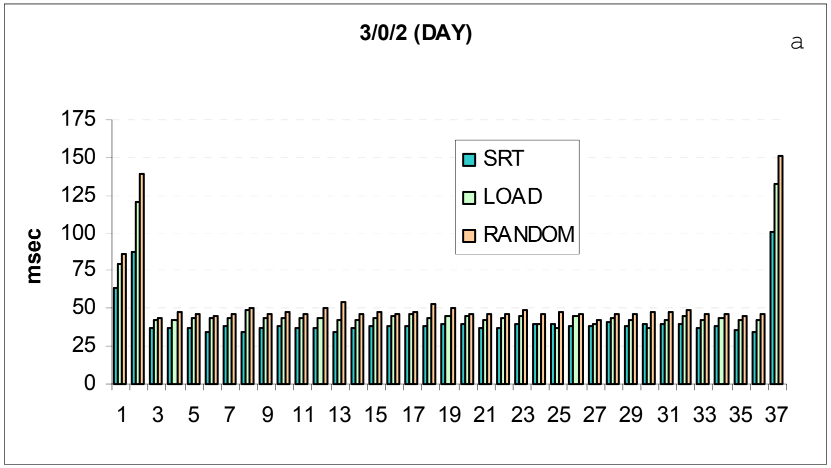
Figure 13(a). 4-Servers File Response Time-3/0/0(DAY) (b) 4-Servers File Response Time-3/0/0(NIGHT)

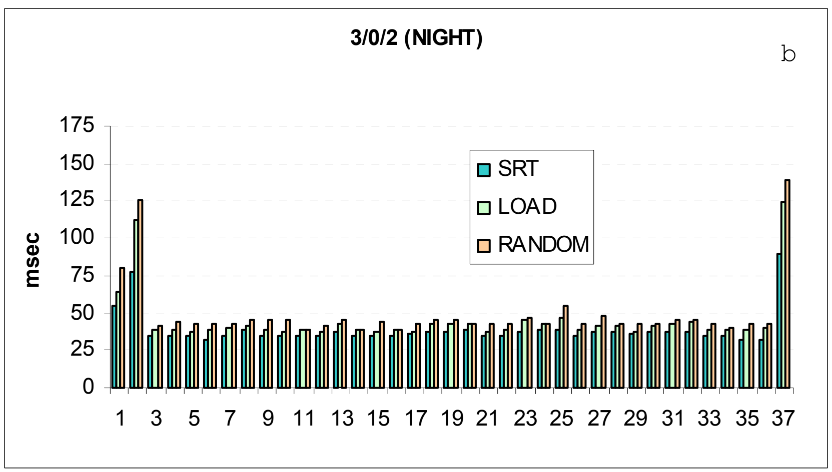




File ID	Content	Total (KB)
1	Text + Img	32
2	Text + Img	17
3	Text + Img	4
4	Text + Img	4
5	Text + Img	4
6	Text + Img	5
7	Text + Img	4
8	Text + Img	4*
9	Text + Img	5
10	Text + Img	4
11	Text + Img	4
12	Text + Img	4
13	Text + Img	4
14	Text + Img	4
15	Text + Img	4
16	Text + Img	4
17	Text + Img	52
*: cached		

Figure 14(a). 4-Servers File Response Time-3/0/1(DAY) (b) 4-Servers File Response Time-3/0/1(NIGHT)

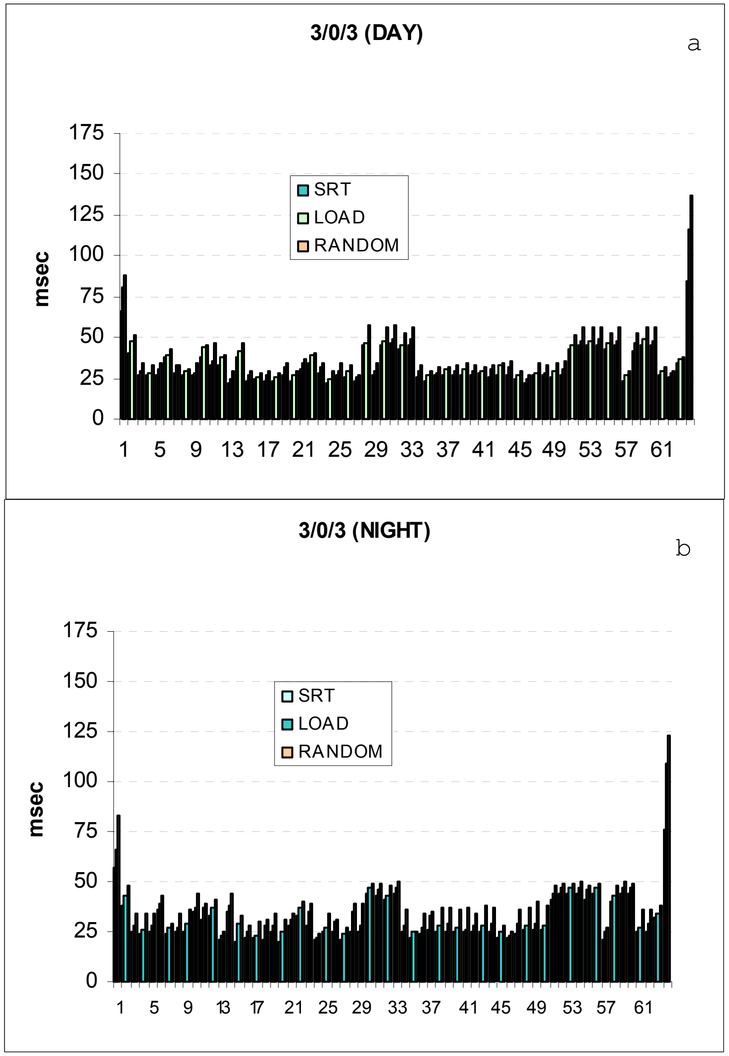




File ID	Content	Total (KB)
1	Text + Img	35
2	Text + Img	59
3	Text + Img	12
4	Text + Img	12
5	Text + Img	12
6	Text + Img	10
7	Text + Img	12
8	Text + Img	21
9	Text + Img	12
10	Text + Img	12
11	Text + Img	12
12	Text + Img	12
13	Text + Img	15
14	Text + Img	12
15	Text + Img	12
16	Text + Img	12
17	Text + Img	12
18	Text + Img	15
19	Text + Img	15
20	Text + Img	17
21	Text + Img	12
22	Text + Img	12

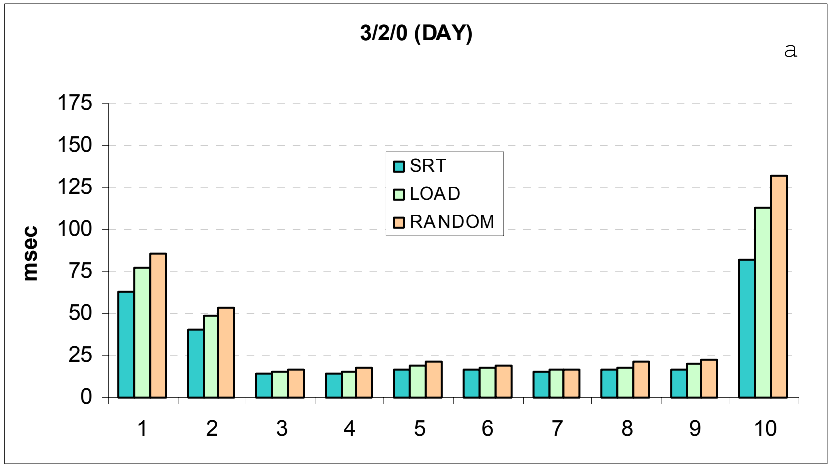
File ID	Content	Total (KB)
23	Text + Img	15
24	Text + Img	17
25	Text + Img	17
26	Text + Img	12
27	Text + Img	17
28	Text + Img	15
29	Text + Img	12
30	Text + Img	15
31	Text + Img	17
32	Text + Img	15
33	Text + Img	12
34	Text + Img	12
35	Text + Img	10
36	Text + Img	10
37	Text + Img	95

Figure 15(a). 4-Servers File Response Time-3/0/2(DAY) (b) 4-Servers File Response Time-3/0/2(NIGHT)



File ID	Content	Total (KB)	File ID	Content	Total (KB)	File ID	Content	Total (KB)
1	Text + Img	39	23	Text + Img	10*	45	Text + Img	8*
2	Text + Img	17	24	Text + Img	6	46	Text + Img	6
3	Text + Img	8	25	Text + Img	8	47	Text + Img	8
4	Text + Img	8	26	Text + Img	8	48	Text + Img	8
5	Text + Img	8	27	Text + Img	8*	49	Text + Img	8
6	Text + Img	12	28	Text + Img	21	50	Text + Img	8
7	Text + Img	8	29	Text + Img	8	51	Text + Img	21*
8	Text + Img	8	30	Text + Img	21	52	Text + Img	21
9	Text + Img	8	31	Text + Img	21	53	Text + Img	21
10	Text + Img	12	32	Text + Img	21*	54	Text + Img	21
11	Text + Img	10	33	Text + Img	21	55	Text + Img	21*
12	Text + Img	10	34	Text + Img	8	56	Text + Img	21
13	Text + Img	6	35	Text + Img	6	57	Text + Img	6
14	Text + Img	12	36	Text + Img	8	58	Text + Img	21*
15	Text + Img	8*	37	Text + Img	8	59	Text + Img	21
16	Text + Img	8*	38	Text + Img	8	60	Text + Img	21
17	Text + Img	8*	39	Text + Img	8	61	Text + Img	8
18	Text + Img	8*	40	Text + Img	8	62	Text + Img	8
19	Text + Img	8	41	Text + Img	8	63	Text + Img	10
20	Text + Img	8*	42	Text + Img	8	64	Text + Img	56
21	Text + Img	12*	43	Text + Img	8	*: cached		
22	Text + Img	10	44	Text + Img	8			

Figure 16(a). 4-Servers File Response Time-3/0/3(DAY) (b) 4-Servers File Response Time-3/0/3(NIGHT)



File ID	Content	Total (KB)
1	Text + Img	32
2	Text + Img	17
3	Text	2
4	Text	2
5	Text	3
6	Text	3
7	Text	3
8	Text	3
9	Text	3
10	Text + Img	52

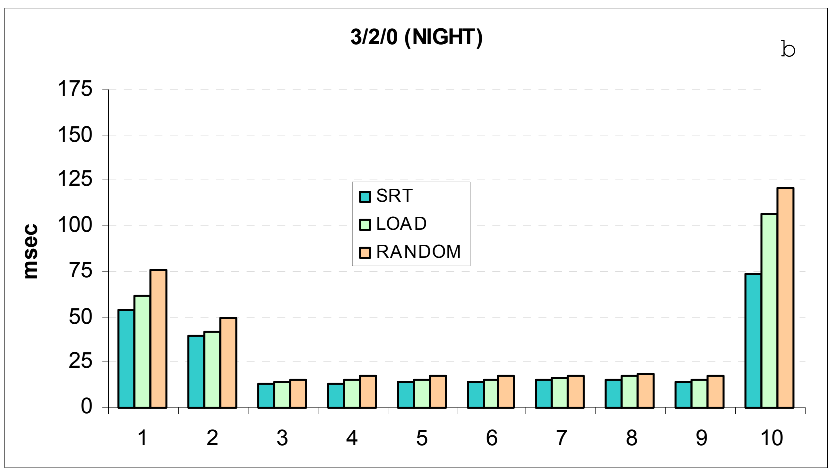
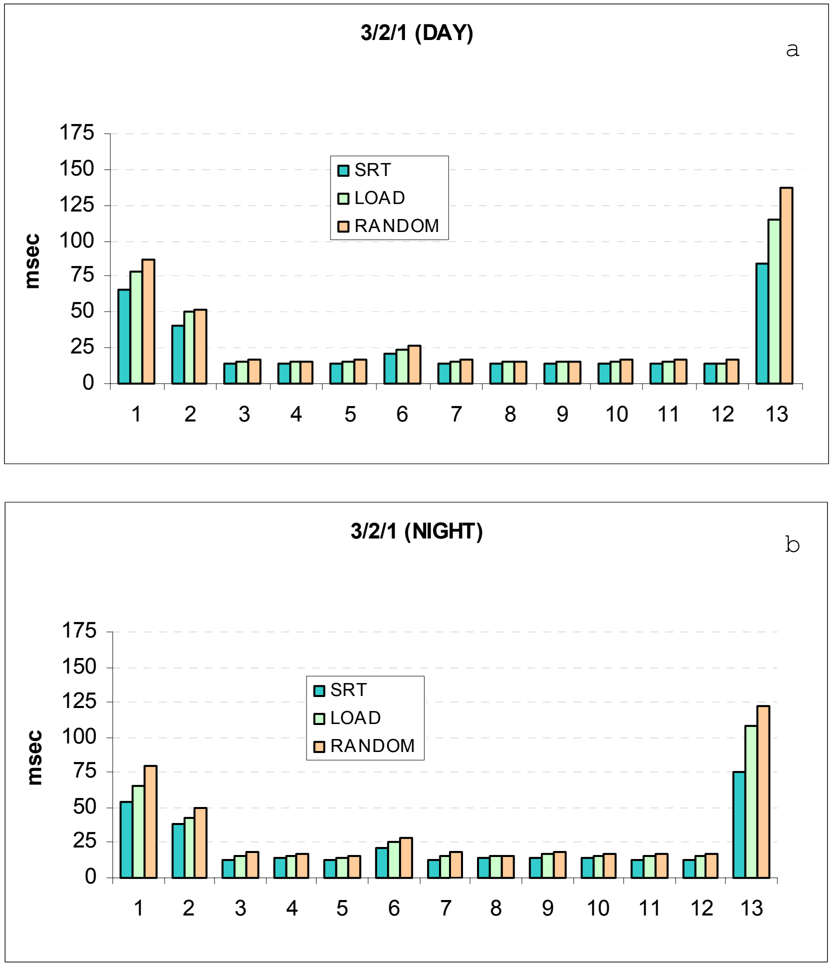
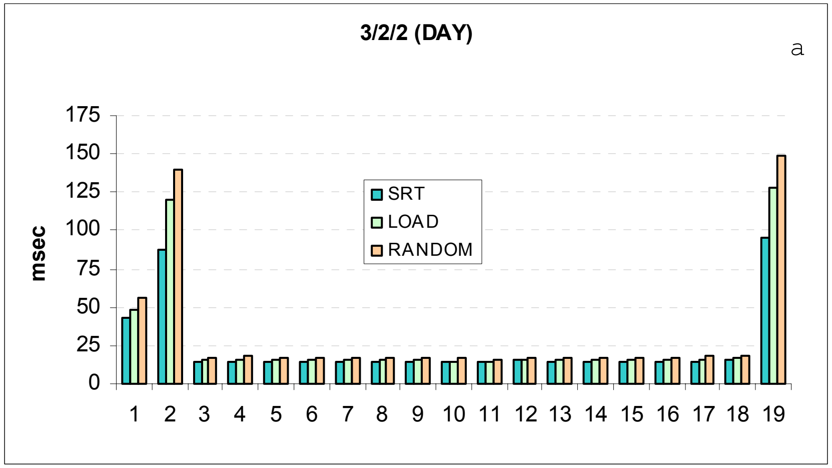


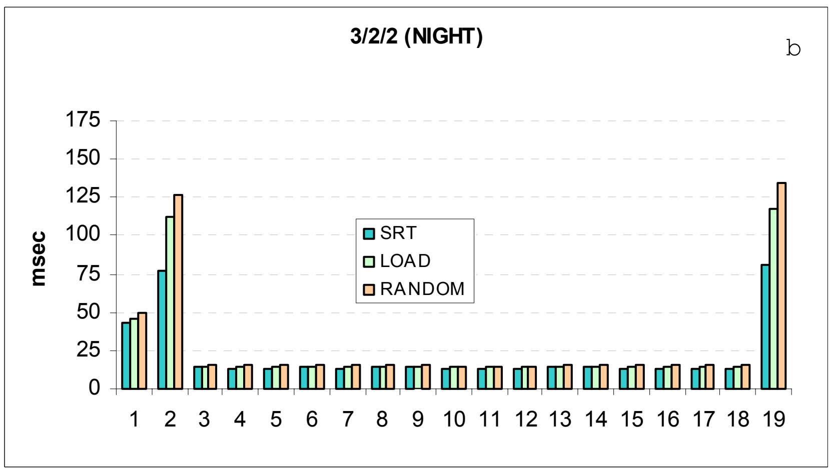
Figure 17(a). 4-Servers File Response Time-3/2/0(DAY) (b) 4-Servers File Response Time-3/2/0(NIGHT)



File ID	Content	Total (KB)
1	Text + Img	36
2	Text + Img	17
3	Text	2
4	Text	2
5	Text	2
6	Text + Img	5
7	Text	2
8	Text	2
9	Text	2
10	Text	2
11	Text	2
12	Text	2
13	Text + Img	55

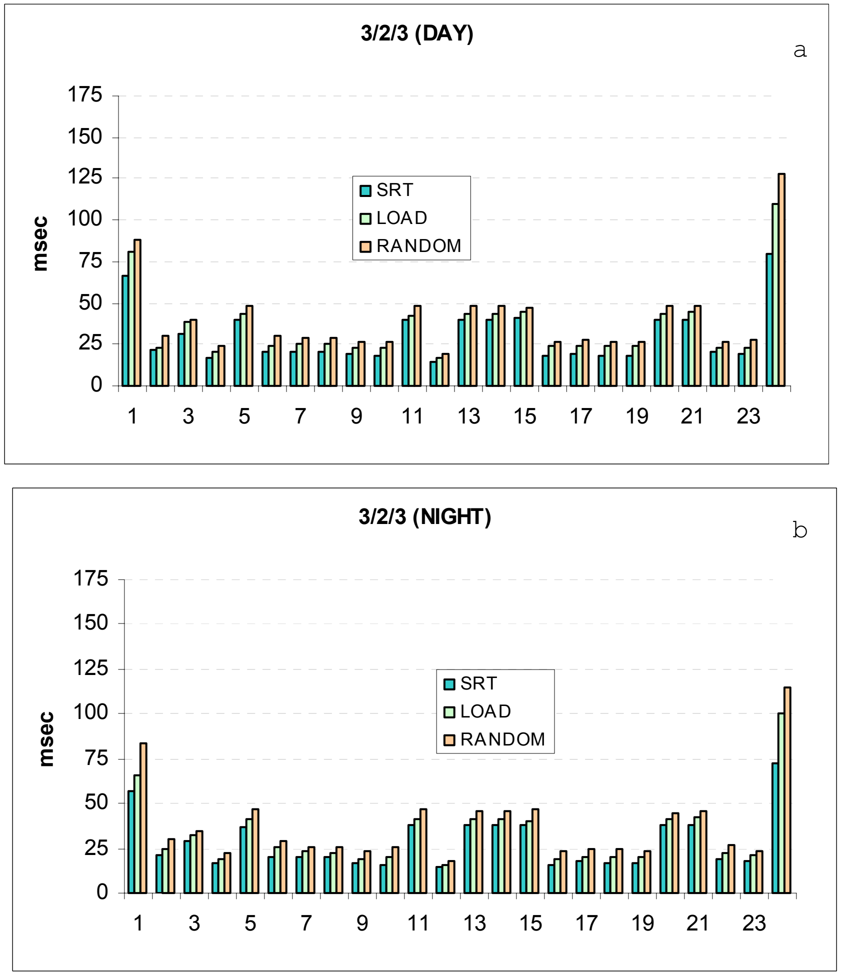
Figure 18(a). 4-Servers File Response Time-3/2/1(DAY) (b) 4-Servers File Response Time-3/2/1(NIGHT)





File ID	Content	Total (KB)
1	Text + Img	20
2	Text + Img	59
3	Text	2
4	Text	2
5	Text	2
6	Text + Img	2
7	Text	2
8	Text	2
9	Text	2
10	Text	2
11	Text	2
12	Text	2
13	Text	2
14	Text	2
15	Text	2
16	Text	2
17	Text	2
18	Text	2
19	Text + Img	80

Figure 19(a). 4-Servers File Response Time-3/2/2(DAY) (b) 4-Servers File Response Time-3/2/2(NIGHT)



File ID	Content	Total (KB)
1	Text + Img	38
2	Text + Img	6
3	Text	15
4	Text	4
5	Text + Img	17
6	Text + Img	4
7	Text + Img	4
8	Text + Img	4
9	Text + Img	3
10	Text + Img	3
11	Text + Img	17
12	Text	2
13	Text + Img	17
14	Text + Img	17
15	Text + Img	17
16	Text + Img	3
17	Text + Img	3
18	Text + Img	3
19	Text + Img	3
20	Text + Img	17
21	Text + Img	17
22	Text + Img	4
23	Text + Img	3
24	Text + Img	45

Figure 20(a). 4-Servers File Response Time-3/2/3(DAY) (b) 4-Servers File Response Time-3/2/3(NIGHT)

CHAPTER V

RELATED WORK

The issue on how to select the "best" server has been the subject of research for the past few years [13]. The major issue involves identifying the proximity metric to use in order to estimate the network performance between the user and each candidate server correctly. There has been a number of researchers that study the effectiveness of proximity metrics in reflecting the actual network. The metrics are described below.

Geographic distance, one of the oldest proximity metric known, has been known to be a poor indicator of the resulting response time [7]. In majority of the cases, the geographic distance between the client and the server has negligible effect on the network performance between the two ends.

Other researchers have also reported on using the number of hops to measure network distance between the client and the server [3]. Since the changes in the network topology over a period of time is generally negligible, using the number of hops as proximity metrics is very attractive because of its relatively stable value. Unfortunately, this metric does not reflect the variation in the link speed and the current load on the network path between the client and the server [13]. Other studies have also reported that the number of hops is not an appropriate metric to use for estimating network performance between a client and a server [7].

Another popular metric that is commonly used to evaluate the quality of network performance is round-trip time. Recent work done by Hanna and Natarajan has shown that round-trip time provides a fairly good indicator of network performance, especially for small file size [7]. Research study by Crovella and Carter has indicated that the bandwidth size of the network path between a client and a server has a significant effect to the final response time encountered by the client [2]. Their report shows that network bandwidth causes substantial impact on the file transfer time as the file size exceed 10 KB.

Two of the widely known server selection techniques are called static server selection and dynamic server selection [2]. Static server selection technique employs the use of a proximity metric that only changes negligibly over the years to determine the best server. Common metrics used with this technique are geographic distance, the number of hops, and the use of previous server discovery step [8]. The opposite of static server selection is dynamic server selection. Similar to static server selection, dynamic server selection also uses a proximity metric to identify the best server. The difference is the dynamic server selection uses metrics that may change considerably over a short period of time. Examples of such metrics are packet round-trip time. Previous work has shown that dynamic server selection consistently performs better than static server selection [2].

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

Experimental results from this study indicate that the performance-directed site-selection system developed for AADMLSS consistently performs better than the Load-based and Random-based selection methods that are used for comparative purpose. The Random-based method, which has the worst performance among the three methods, chose resource sites most fairly regardless the load on the server and the network performance between the user and the server. The performance of the Load-based selection method also suffered from merely considering the load on the server at the given time. The experimental results clearly indicate the Service Response Time-based selection method consistently identifies the best resource site at any time. This selection method takes advantage of the knowledge regarding network performance between the server and the user and combines this information with the historical server performance measurement as selection criteria to identify the best resource site. Thus, it is clear that network performance plays an important role in determining the best resource site. The benefits are especially more apparent when dealing with large size files. The experimental results also indicate that server performance becomes crucial when network delay between the user and the candidate servers are comparable.

The study has also shown that despite the reduced network traffic during nighttime, the service performance delivered by the load-based method cannot outperform the service response time-based selection method. This indicates that network performance between the user and the server plays an important role in determining the best resource site at any time. Furthermore, the differences in the service performance among the three selection methods were more apparent for larger file size.

At this point, the framework that has been developed for this project uses a centralized database management system. Any HTTP requests with regard to user profile, concept location, and statistics of student's learning progress are obtained from this single database server. In addition, performance data collected primarily with regards to network performance and server latency are stored into the database. As the number of replica increases, a single database server will become overloaded.

For this reason, the implementation of distributed database is very desirable, especially with increasing number of server replicas in the near future. Distributed database facilitates easy scalability by placing data across multiple different machines. However, there are a number of fundamental issues related to building such system. This includes the mechanism required to synchronize data across all servers to guarantee data consistency. Other issues that must be carefully addressed are finding the means in preserving both local and global concurrency and the architecture required at the network layer to support such system [14].

Another issue that may arise in the future involves the technique used in estimating the network performance between the user and a candidate server. At this point, the framework measures the network delay between the two ends by sending packets of different size to each candidate server and generates an appropriate equation model for each server in the system. As the number of server replicas increases, this technique may create a bottleneck in the overall system performance. Therefore, it is desirable to implement a technique where the network delay equation model is only generated for a fixed number of servers, rather than for each server in the system. This can be achieved by assigning a rank for each server based on their historical measurements of network performance and selecting the top five servers from the list. To give a fair opportunity to servers that are not selected, the rank can be re-evaluated after a certain period of time that is found to be effective.

REFERENCES

[1] R. Caceres, F. Douglis, A. Feldmann, G. Glass, and M. Rabinovich, ”Web Proxy Caching: The Devil Is in the Details,” in *Proc. 1998 Workshop on Internet Server Performance*, pp.32-36, June 1998.

[2] M. Crovella, R. L. Carter, ”Dynamic Server Selection in the Internet,” in *Proc. Third IEEE Workshop on the Architecture and Implementation of High Performance Computer Systems*, pp. 158-162, Aug. 1995.

[3] P. Dinda, D. O’Hallaron, “An Evaluation of Linear Models for Host Load Prediction,” in *Proc. Eighth IEEE International Symposium on High Performance Distributed Computing*, pp. 10-15, Aug. 1999.

[4] J. E. Gilbert and Han, C. Y., ”Arthur: A Personalized Instruction System”, *Journal of Computing in Higher Education*, vol. 14, no. 1, pp. 21-25, Oct. 2002.

[5] J. N. Gunther, ”Performance and Scalability Models for a Hypergrowth e-Commerce Website,” *Performance Engineering*, vol. 2047, pp. 267-282, June 2001.

[6] J. Guyton, M. F. Schwartz, ”Locating Nearby Copies of Replicated Internet Servers”, in *Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, vol. 25, pp. 288-298, Oct. 1995.

[7] K.Hanna, N. Natarajan, and B.N. Levine, ”Evaluation of a Novel Two-Step Server Selection Metric,” in *Proc. Ninth International Conference on Network Protocols*, pp. 290-295, Nov. 2001.

[8] J. Offutt and M. Lee, ”Web Architectures”, http://www.isse.gmu.edu/~lili/642/WebArchJ2EE.pdf, 2001.

[9] N. Parks, T. Simmons, K. Sapp & J.E. Gilbert, “Culturally Influenced E-Learning: An Introduction to AADMLSS”, in *Proc. E-Learn World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education*, pp. 1960-1965, 2003.

[10] M. Rabinovich and O. Spatscheck, *Web Caching and Replication*. Boston, MA: Addison-Wesley, 2002.

[11] J. Peek, T. O'Reilly, and M. Loukides, *UNIX Power Tools,* second ed. Sebastopol, CA: O'Reilly & Assoc. Inc., 1997.

[12] R. Rajamony and M. Elnozahy, ”Client-Perceived Response Times on the WWW”, in *Proc. Third USENIX Symposium on Internet Technologies and Systems*, pp. 185-196, March 2001.

[13] M. Sayal, L. Breitbart, P. Scheuermann, and R. Vingralek, ”Selection Algorithms for Replicated Web Servers”, *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, pp.44-50, December 1998.

[14] G. Thompson and L. Breitbart, “Design Issues in Distributed Multidatabase System”, in *Proc. 1986 Workshop on Applied Computing,* pp. 38–46, Oct. 1986.

[15] R. Wolski, ”Dynamically Forecasting Network Performance Using the Network Weather Service”, *Journal of Cluster Computing*, vol. 1, pp. 119-132, Jan. 1998.

VITA

Name: Mieke Prajugo

Home Address: 134 Innis Ave #C4

Poughkeepsie, NY 12601

Education: B.S. Computer Science University of Minnesota 2001