Institute of Parallel and Distributed Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit Nr. 3663

Optimizing the Resource utilization of Enterprise Content management workloads through measured performance baselines and dynamic topology adaptation

Gaurav Chawla

Course of Study:	INFOTECH
Examiner:	Prof. Dr. Bernhard Mitschang
Supervisor:	Dipl.-Inf. Tim Waizenegger
Commenced:	December 01, 2013
Completed:	June 02, 2014
CR-Classification:	C.4, H.3.4, I.2.8

Abstract

To oblige with the legal requirements, organizations have to keep data up to a certain amount of time.They are creating a huge amount of data on daily basis therefore it is very difficult for them to manage and store this data due to the legal requirements. This is where Enterprise Content Management (ECM) system comes into picture. ECM is a means of organizing and storing an organization's documents and other content that relates to the organization's processes. With ECM being offered as a service, thanks to cloud computing it makes sense to offer this functionality as a shared service. There are various benefits of offering it as a shared service one of which is that it is a cheaper method to meet the needs of large organizations with different requirements for ECM functionality. ECM systems use resources like memory, central processing unit (CPU) and disk which are shared among different clients (organizations). With every client, a service level agreement is there which describes the performance criteria a provider promises to meet while delivering the ECM service. To improve the performance of the ECM by optimizing the use of resources and match the Service level agreements various techniques are used. In this thesis, heuristics technique is used. Performance baselines and utilization of resources are measured for different workloads of the clients and on the basis of that, resources of the ECM can be dynamically provisioned or assigned to different clients get the optimized resource utilization and better performance. First of all typical workload is designed which is similar to the work being performed by various banks and insurance companies using IBM ECM systems and which consists of interactive and batch type of operations. Performance baselines are being measured for these workloads by monitoring the key performance indicators (KPIs) with variable number of users performing operations on the system at the same time. After getting the results for KPIs and resource utilization, resources are being assigned dynamically according to their utilization in a way that the use of resources is optimized and clients are satisfied with better service at the same time.

Table of Contents

Figures List

Tables List

1. Introduction

"In the next five years, we'll generate more data as humankind than we generated in the previous 5,000 years.[BDHN]

Eron Kelly
General Manager Microsoft SQL Server

Our world contains a vast amount of data which is getting ever vaster more rapidly. Legal obligations and desire of providing more customer specific products and services push organizations to save data for a particular amount of time. Over the last few years, organizations have increasingly been paying attention to the concept of enterprise content management (ECM) system for managing this data, which refers to the strategies, methods, and technologies required for capturing, storing, retrieving, delivering, and retaining all types of digital information across the organization.

Depending on the organization,various different workloads are being handled by such ECM systems with providing certain amount of performance being defined in Service Level Agreements (SLAs). A service-level agreement is an agreement between two or more parties, where one is the customer and the others are service providers.SLA contains the description of services, performance , problem management , customer duties,warranties, disaster recovery and termination of agreement.The providers of ECM systems offer ECM as a service on *pay as you go* basis. This is an attractive method for both i.e. service providers and customers. Service Providers are happy because single set of resources and software are being provided to different customers,hence there is no need for individual resources and software installations for each customer which leads to less cost. This is also beneficial for the customers as they need to pay only of what they are using. They don't need to pay for all the infrastructure costs,hardware etc. They don't need premises for big servers as everything is being managed by the provider. This is similar to the electricity we use. We pay only for the electricity which we consume. We don't care about power generation ,transmission or distribution costs. But on the provider side,they have different kind of customers. Some are domestic customers using electricity for house hold purposes,some perform commercial activities and use more electricity than others. To fulfill the requirements of both type of customers, providers must manage the resources (electricity here) in such a way that each customer is getting what is being promised to him/her and also there is proper utilization with no over or under utilization. Similarly,in an ECM system,with same resources in the back,different instances (Logical Partitions LPAR) are assigned to different customers (organizations).Resources are being shared among these customers in such a way that the SLA and performance requirements are met. These resources must be shared in such a way that each customer's requirements are fulfilled with the optimized consumption of resources without under or over utilization. This is done dynamically in a multi tenant cloud environment. Resources are provisioned dynamically on the basis of workloads on the tenants.

1.1 Problem Statement

Dynamic provisioning of resources is to be implemented in a multi-tenant cloud environment based on the utilization of resources. Resources are utilized when there is workload on the system. Systems considered in this thesis are ECM systems. Typical workloads need to be designed which are performed on the ECM systems. While performing these workloads, resource utilization is monitored . The possibility of applying concept of MAPE (monitor, analyze,plan,execute) loop is being studied which consists of four phases (M,A,P and E). During the monitor phase, resources like CPU,memory and disk I/O are being monitored and their utilization is being measured for different set of ECM workloads. A test system is defined on which ECM workloads are performed and performance baseline and stress testing is performed and the performance (at minimum workload and at peak) of the test system is being measured using various key performance indicators (KPI)(e.g. number of transactions (CRUDS) per seconds, number of WS calls, data throughput, numbers of documents indexed etc) measuring tools. For the given test system, with given set of workloads, the performance is being analyzed in the analyze phase. Rules are generated for provisioning of resources for the system according to the performance and requirement of resources by the system. In the plan phase,planning of resource topology is done which is based on the rules generated during analyze phase. Provisioning of resources is done by two ways depending on the behavior (proactive or reactive). If the resources are provisioned based on the future workload assumption then it is proactive behavior. [AnBo] has presented an approach with this behavior for provisioning of resources. In this thesis, the resources are dynamically provisioned after measuring the performance baselines of actual workloads and not the assumptive workloads. That is called as reactive behavior. In future, combination of both approaches may be taken into consideration. [FFr] presented the execution phase of MAPE loop. Therefore , execution phase is not being done in much detail. But the plan phase results must be executed in the solution of execution phase of [FFr]. Also, resource topology must be created depending on client's requirements. Multi-tenancy is applied which enables sharing of resources by multiple customers. An algorithm has to be defined for the provisioning of resources which takes into consideration all these characteristics.

1.2 Related Work

There is some work being done on dynamic provisioning of resources in multi tenant cloud environment. Each has its own advantages and disadvantages. In the following section each of them are explained briefly.

Regression based analytical model [ZCS]

In this technique, regression-based approximation is made for the CPU demand of client transactions on a given hardware. Then this approximation is used in an analytic model of a simple network of queues, each queue representing a tier, and shows the approximation's effectiveness for modeling diverse workloads with a changing transaction mix over time. Using some performance monitoring tool factors that impact the efficiency and accuracy of the proposed performance prediction models are investigated. This regression-based approach provides solution for capacity planning and resource provisioning of multi-tier applications under changing workload conditions.

Model Based Resource Provisioning in web service utility [RJO]

Automated on-demand resource provisioning is done for multiple competing services hosted by a shared server infrastructure and it is offered as an utility. This utility allocates each service a slice of its resources, including shares of memory, CPU and available throughput from storage units. Slices provide performance isolation and enable the utility to use its resources efficiently. The slices are chosen to allow each hosted service to meet service quality targets for example, response time. which are mentioned in service level agreements with the utility. Workloads and content is captured to predict the user behavior and based on that resources are provisioned dynamically.

Machine Learning Adaptive Techniques

The behavior of the system is analyzed with machine learning adaptive techniques at run time. Therefore, performance testing of the system is not required. [CKF] We don't need to have system specific knowledge for provisioning of resources. The disadvantage is that training is required for applying these techniques. Resource provisioning is done either proactively or reactively [CKF]. Proactively means that the workloads are presumed and resources are provisioned on the basis of that rather than the actual workloads on the system at run time. In reactive resource provisioning, actual workloads are monitored and provisioning of resources is done on the basis of that. Various methods described in this section, consider both reactive and proactive method of resource provisioning. The important difference of proposed techniques is that some provide short-term reactive algorithms while others allow long-term proactive techniques. The benefit of proactive techniques is that the system topology can be provisioned before actual workload peaks occur. This cannot be achieved by reactive techniques, since provisioning and de-provisioning of resources takes a certain amount of time. Thus, there would be a delay between resource need and its actual applicability.

Performance models, regarding utility-based optimization techniques, have been created for multiple architectures. E.g. there are models for single tier as well as for multi tier applications. An example is shown in [RJO].

2. The Context

First of all some basic concepts are described in this chapter to better understand the background of the thesis. The context of the thesis includes multi tenant cloud environment, dynamic resource provisioning, performance testing heuristics, resources of the system etc. These concepts are explained in this chapter.

2.1 Cloud Computing

Cloud Computing can be described in two ways. Narrowly defined ,it is an updated version of utility computing basically virtual servers available over the Internet. Broadly,anything we consume outside the firewall is "in the cloud," including conventional outsourcing.[FRNL] Cloud computing comes into focus only when we think about a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities [FRNL].

2.1.1 Service Models

Cloud computing providers offer their services according to several models. These are infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) where IaaS is the most basic and each higher model abstracts from the details of the lower models. [FRNL]

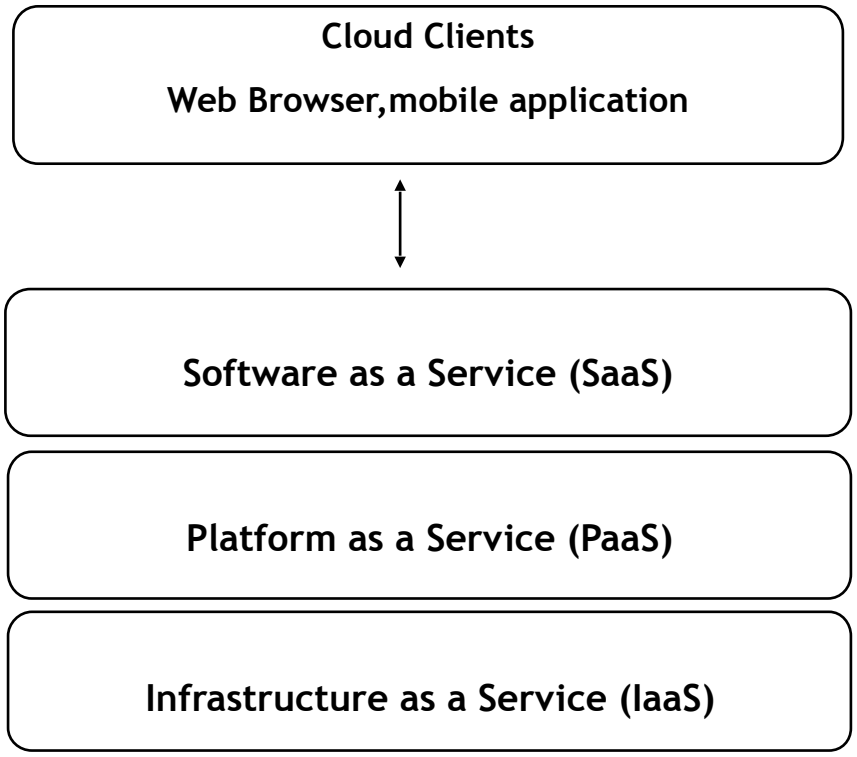

FIGURE 1. SERVICE MODELS OF CLOUD

Infrastructure as a service (IaaS)

In the most basic cloud-service model, providers of IaaS offer computers physical or virtual machines and other resources. A hyper visor runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements. IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw (block) and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or dedicated virtual private networks.

To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis cost reflects the amount of resources allocated and consumed.

Platform as a service (PaaS)

In the PaaS models, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. Some PaaS offers the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually.

Software as a service (SaaS)

In software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is also called as "on-demand software" and is priced on a pay-per-use basis. SaaS providers generally price applications using a subscription fee. In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. Cloud applications are different from other applications in their scalability which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access point. To accommodate a large number of cloud users, cloud applications can be multi tenant, that is, any machine serves more than one cloud user organization.

2.1.2 Single vs Multi Tenancy

In simple words, tenant means customers or users. Single-tenant model has a separate, logical instance of the application for each customer, while the multi-tenant model has a single logical instance of the application shared by many customers. It’s important to note that the multi-tenant model still offers separate views of the application’s data to its users. Let us see the difference between the two in the figure below.

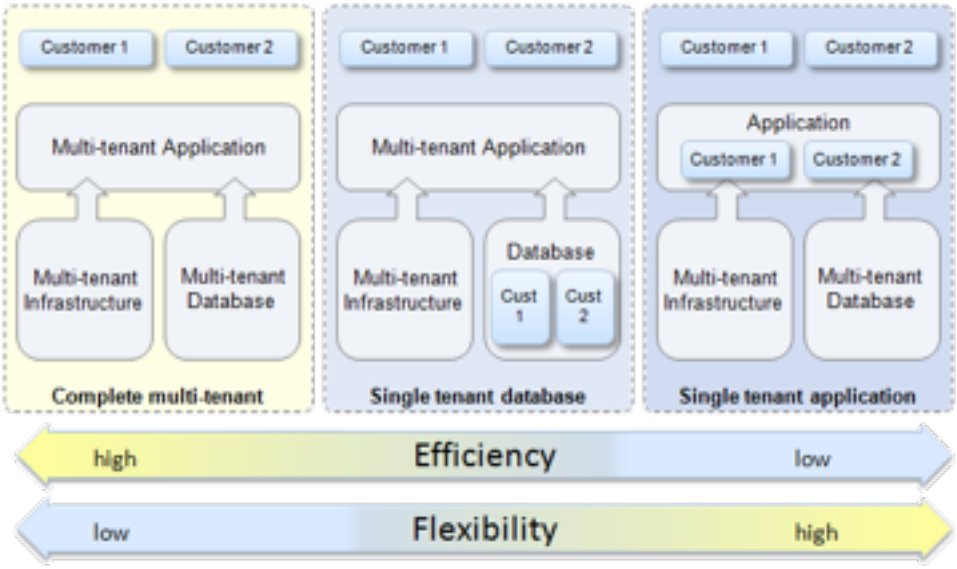

FIGURE 2. SINGLE VS MULTI-TENANCY[MALS]

In the first part of the figure,there is a complete multi tenant environment. That means that infrastructure,database and application everything is shared among the customers with high efficiency and low flexibility due to no customer specific application design. In the second part,there is multi tenant infrastructure and application but single tenant database. That means that a dedicated database is being assigned to each customer but other infrastructure and application is being shared among different customers. Last part shows a single tenant application with a dedicated application for each customer with high flexibility because applications are designed in the customer specific way. At the back end,infrastructure and database is still shared among different customers but that is invisible to the customer and s/he is happy with the application designed specifically for him/her.

2.1.3 Deployment Models

Private cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party. A private cloud requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to revaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally costly. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures.

Public cloud

In a public cloud model , services are rendered over a network that is open for public use. Technically there may be little or no difference between public and private cloud architecture, however, security consideration is very different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure and offer access via Internet.

Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. An example of hybrid cloud model is the cloud bursting. That means an application runs in a

private cloud or data center but it bursts to a public cloud when the demand increases. Advantage of this type of modeling is that companies have to only pay for extra resources when they are needed.

Distributed cloud
Cloud computing can also be provided by a distributed set of machines that are running at different locations, while still connected to a single network or hub service.

2.2 Enterprise Content Management

As the name suggests Enterprise Content Management is the Management of Enterprise Content. Enterprise Content Management (ECM) comprises the technologies used to capture, manage, store, preserve, and deliver content, and documents related to organizational processes [BDHN]. ECM tools and strategies are used to manage the unstructured information of an organization. The enterprise content can be in structured or unstructured form. A structured content is the content which has some information about the content and is classified in metadata. Unstructured content can be any random data without any information about it. For example, a structured and well defined pages web site can be seen as a structured content whereas a simple home page designed by a student can be seen as unstructured content.

The utmost importance of ECM solutions is in managing this unstructured content and making it structured so that the content can be managed efficiently and can be searched later on when required. So, back to Enterprise Content Management, it is the strategies, methods and tools used to capture, manage, store, preserve and deliver content and documents related to organizational processes. Using the ECM tools and strategies, organizations can manage their unstructured information. But actually, in an organization, it is not enough to just manage the content but also it should be easily and quickly accessed anytime. The content should be managed in such a way so that later on, when it is required, the correct version or document can be accessed. ECM systems manage the complete life cycle of content from its birth to death. In today's world with mobiles, cloud and big data the information is being created more faster than it can be read and understood. Therefore companies need Compliance, collaboration, continuity and reduced cost for this data. There are also some basic terms used in the context of ECM which are defined below.

Compliance [BDHN]
Organizations must comply with the legal requirements which force them to keep the data for a certain amount of time. There are also companies' policies which say that they will retain the data until a certain amount of time. For example, on most of the companies' job portal, when a job seeker applies for the job, he or she must accept the condition by the company that his/her data will be retained by the country for lets say 1 year. But, when we talk of legal requirements of keeping data, companies don't just think that this is the extra ongoing cost in the business for storing the data but actually they view this as an opportunity to improve common business processes. There are high costs involved in compliance for legal regulations and this cost increases with each new regulation comes over the years. To check the risks and costs involved, Enterprise Content Management strategies are developed for records management and business process management. These strategies follow proper business practices and also content is properly captured, stored, managed and disposed of at the appropriate and legal life time in its life cycle. These strategies involve many areas particularly legal, IT and records management. People from these areas collaborate with each other for better compliance. The major problem which organizations face while compliance is massive growth of unstructured

content .ECM tools help them in this problem and reduce the overall cost of compliance for the business.

Collaboration

Collaboration means working together. There are technologies like instant messaging, whiteboards, online meetings ,email etc. which allow people to work together whenever and wherever needed. Collaboration is very important because group of people can accomplish more than individuals. Collaboration helps people come together and create better results in the areas which are complementary or overlapping between them. Collaboration with the right set of technologies can save time, streamline business processes , reduce costs and improve time for marketing. There are different collaborative tools available in the market with different functionalities. For example,facilitation of communication channels by chat, instant messaging, white boarding etc which enables short-lived interaction or management of content life cycle which manages content objects involved in the business processes. There can also be project facilitation functionality where project is organized and simplified in such a way that people work towards a common business goal. So, while using collaborative tools organizations must take care about record management ,knowledge capture and compliance requirements. This is because some industries must have to keep all customer communications and for the collaborations , organizations must be sure that the results are kept as business records.

Cost

Enterprise Content Management helps organizations in reducing their costs in content management. So, if an organization is not managing its content properly, then the cost for not implementing Enterprise Content management tools can be easily measured. For example in the absence of relevant content, organizations have to go through long legal proceedings whose cost can be easily measured. Similarly, organizations will not get the repeat business if it can't perform simple customer service interactions in the absence of proper content management. This cost can be easily measured. Also, cost of business process delays with no proper content management related to process have to be beard by the organization and can be easily measured which would be cost of acquiring new customers and salaries that must be given to employees for the grace period of project. Comparing these costs with the ECM investment make organizations to think in investing ECM solutions which has immense benefits both measured and unmeasured. ECM tools make the organizations more efficient with less cost. Let us take an example of organization who process loan applications or claim processes. If the content and all the documents of customers are managed properly, these processes can speed up so much that makes customers happy on the one hand and reduce the organization's cost and time in searching, retrieving and securely disposing the unnecessary content.

Continuity

Today we live in a world which is well connected and business organizations are spread over all planets. Businesses keep going 24x7 and 365 days a year. Organizations have the task of business continuity planning for always keep the business running. This includes overall strategy for ensuring that no matter whatever happens ,business operations must continue. Disaster recovery is the part of the strategy of business continuity. It is more narrowly focused

on getting an organization's infrastructure going again after a disaster. Since documents and more specifically electronic documents are the life blood of businesses today, Enterprise Content Management plays a very important role in continuity. ECM helps in creating repositories where all corporate information can be stored. Based on the fact that how critical the content is,storage can be done as off-site back up tapes, redundant storing of information ,mirrored sites separated by geography and can be saved on different power grids. Not all of the content is very critical. Organizations prioritize their content to determine how fast does the content needed to be back online in the case of a disaster. Business continuity starts with a nice plan and high level executive support. Business critical processes and their dependencies are determined and a business impact assessment is done to understand the impact of the disruption if those processes and their dependencies are lost. Businesses also define the meaning of disaster or what they consider as disaster. Also, how important processes will be recovered are also defined in the continuity plan. By doing so,organizations can enhance their ability to recover after the disaster and improve overall ECM strategy.

Business Process Management (BPM)

BPM or workflow involves the tools which move the content throughout the business process. An example of business process can be claim processing. BPM solutions help in developing, deploying ,monitoring and optimizing various types of process automation applications including processes that involve systems and people. Any process which can be automated by ad-hoc processing or manual intervention is managed by the BPM tools. Workflow is normally associated with the manual processes of managing documents .Workflow handles approvals and prioritizes document order in the presentation. If some problem occurs, then workflow also helps in escalating decisions to the next person in the hierarchy. These decisions are based on rules which are pre defined by system owners.

Content and Documents

Organizations get unstructured content from a number of sources. The unstructured content can be email, instant message, text document, spreadsheet , electronic forms, paper documents or forms etc. Any unstructured content has a life cycle once it enters the organization. Following are the ways by which the content of any form enters into the organization's IT infrastructure.

Scanning

Papers enter the organization through a scanner or a multi functional device. In big scan operations,large volumes of paper are put into the system whereas in distributed operations, small volume of documents are scanned by low volume scanners.

Document Imaging

In document imaging, software captures the image of the paper document. In most countries of the world, nowadays an electronic document image has the same legal status as a paper document.

Forms Processing

Here business forms are ingested into the system. Most of the forms are well structured and the information about the elements of the forms are pre known. Therefore to handle business forms is much easier than handling the unstructured content.

Recognition

Today, we have technologies which read or recognize the paper information and translate it into electronic data without manual intervention. Examples of these technologies are optical character recognition (OCR) ad intelligent character recognition (ICR). These technologies are very useful In converting large amount of forms or unstructured content to usable information in a content management system.

Categorization

By doing characterization of information on the basis of needs of a business ,we can have a formal structure of information. Categorization tools automate the placement of content for future retrieval based on the category. Users can also categorize documents by themselves.

Indexing

A very important part of the capture process is to create metadata (information about data) from scanned documents so the document can be found. Indexing can be based on keywords or full-text. The example of metadata of a document can be the document title, date of creation, date of modification etc. In our test system, indexing is done by the Content Search Services server which is explained later on.

Document Management

Document management means the creation, revision,approval and consumption of electronic documents. The features of document management are document profiling,searching, check-in,check-out,version control,revision history and document security.

Records Management

There is content in the organizations which has long term business value. This content is called as record and is managed according to a retention schedule that determines how long a record is kept based on outside or inside organization's policies or requirements.

Email Management

Email management is a process of removing emails from the server and saving in the repositories. Emails are classified,stored and destroyed according to the business standards. Emails are treated as same as any other content or document.

Repositories

Repositories are the place where the content resides. Repositories can be a structured and well defined repositories or unstructured repository with no proper classification and organization of data. The purpose of repository is to store content or data. It can be a sophisticated system with the cost of thousand of dollars or it can be simply a file folder system in a small organization.

Storage

Storage is the permanent residence of the content. Storage can be optical disks,magnetic tapes,microfilm etc. Content can be stored online for rapid access or offline if the content is not needed often. Content Integration is done with different content sources that are integrated to act as a single repository. Migration is the term which means that when the storage media gets old and is prone to errors or non-functional, content is moved or migrated to new media for continued accessibility.

Backup/Recovery

This is different from migration. Backing up means copying and not moving the content in various formats and /or locations to ensure continuity of business in case of a disaster.

Search/Retrieval

One of the most important advantages of a ECM system is its ability to search and retrieve the content easily and effectively. With indexing, classification and repository services, getting the content is as simple as getting a snap.

Security

This is a very important feature of the ECM system. ECM system restricts access to content during the creation and management of content and also when the content is delivered.

2.3 Key Performance Indicators (KPI)

Key Performance indicators are the metrics for representing system performance. These include system resource utilization , I/O performance and numbers of documents loaded per second or the response time of a certain service. The system resources are Central Processing Unit, Memory ,database and Disk. An important feature in KPIs monitoring is network bandwidth. Bandwidth is the amount of data that can be received in the particular time. For example, if the bandwidth is 100 Mbps ,then this is nothing to do with how fast the connection is but it means that 100 megabits (or 12.5 mega bytes) of data can be received per second.

The concept of bandwidth is similar to the cars on the highway. All the cars (data) travel at the same speed , so to get more data from the network faster, the highway needs to be wider. In other words, let say if 1Mbps is equivalent to 1 lane highway and we need to download data of 5 Mb of size ,then if the bandwidth is 1 Mbps ,then it will take 5 seconds to download that data. The network bandwidth used for tests in this thesis is 100 Mbps. Now, let us examine each of the KPI one by one.

2.3.1 Throughput

Throughput is formed of two words, through and put, as in how much can be put through. It is a rate so it is measured in an amount over time. It is calculated as rate of bytes per second. This is similar to the gallons of water flowing through a pipe.It is the number of requests per second. Every system has a maximum throughput limit. Once this is reached then it is expected by the system to start behaving badly, this usually results in longer response times but this is not the case always. There is no direct relation between through put and response time. Let us take an example of a test where 10 requests are being made in a second and each of the request is 10 Kb in size then the throughput is 100 Kbps. At the same time the response time can be measured as 300 ms. If we double the test load to 20 requests per second then the throughput increases to 200 Kbps. But this doesn't give us any idea about the response

time. We have to measure the response time. It can be 300 ms or may be 600 ms or may be anything else.

2.3.2 Response Time

Response time is the elapsed time between the end of an request and the beginning of a response. Response time is how quickly an interactive system responds to user input. Mathematically, response time is defined as follows:

The acceptable response time for the users depending on how they feel during the User interface interaction:

0.1 second: Limit for users feeling that they are directly manipulating objects in the UI. For example, this is the limit from the time the user selects a column in a table until that column should highlight or otherwise give feedback that it's selected. Ideally, this would also be the response time for sorting the column — if so, users would feel that they are sorting the table. (As opposed to feeling that they are ordering the computer to do the sorting for them.)

1 second: Limit for users feeling that they are freely navigating the command space without having to unduly wait for the computer. A delay of 0.2–1.0 seconds does mean that users notice the delay and thus feel the computer is "working" on the command, as opposed to having the command be a direct effect of the users' actions. Example: If sorting a table according to the selected column can't be done in 0.1 seconds, it certainly has to be done in 1 second, or users will feel that the UI is sluggish and will lose the sense of flow in performing their task. For delays of more than 1 second, indicate to the user that the computer is working on the problem, for example by changing the shape of the cursor.

10 seconds: Limit for users keeping their attention on the task. Anything slower than 10 seconds needs a percent-done indicator as well as a clearly signposted way for the user to interrupt the operation. Assume that users will need to reorient themselves when they return to the UI after a delay of more than 10 seconds. Delays of longer than 10 seconds are only acceptable during natural breaks in the user's work, for example when switching tasks.

2.4 Service Level Agreement

A Service Level Agreement (SLA) is a contract between a service provider and customer specifying non-functional requirements of a service the so called Quality of Service. According to [KK] ,SLA is a document that describes the performance criteria a provider promises to meet while delivering a service. It also defines the remedial actions and any penalties if performance falls below the mentioned in SLA. SLA is comprised of technical ,organizational and legal components. Technical components include service description,service objects and metrics whereas organizational components involves service periods, monitoring and reporting. Legal components are legal responsibilities, modes of invoicing and payment which are mentioned in the Service Level Agreement.

In this thesis,the technical components of the SLA are focused. Technical Components include service objects. The object classes of a IT system include hardware, software, network,storage and help desk. Most of the concern is on Software and Storage in this thesis. Software class metrics involve service time,response time and availability of the system whereas storage class metrics are response time ,availability and failure frequency [KK]. There are SLA rules for each of these metrics. For example,” the response time of a document of any size can not be greater than 1 second “ is a SLA rule for the response time. These SLA rules are used to check whether the service promised by the provider is actually being delivered. If this is not the case then the customer is compensated by the provider for example, by refunding some of the cost or providing some extra service not mentioned in the SLA. Actually Service level Agreement is the entire document that specifies what service is to be provided, how it is supported, times, locations, costs, performance, and responsibilities of the parties involved.

Another term, Service level objectives (SLO) are used for specific measurable characteristics of the SLA such as availability, throughput, frequency, response time, or quality which forms the basis of measurement of performance in this thesis. A sample SLO for a typical system is shown in table1.

Workload	Operation	Size of Load	KPI	SLO
Interactive Workload (CRUDS)	Create	100KB,250 KB	Response Time	1s
	Retrieve	100KB,250 KB	Response Time	1s
	Update	100KB,250 KB	Response Time	1s
	Delete	100KB,250 KB	Response Time	1s
	Content Based Search	Archive consisting 1 million documents	Response Time	5s
Batch Loading	Bulk Load	Amount of Documents	Throughput	50 documents/sec

Table1. Sample SLO

2.4 System Architecture

IBM® SmartCloud™ Content Management (SCCM)* provides a cloud-based data archiving solution that index, search, retrieve and store archived content cost effectively. It provides reliable, cloud-based archiving service that is designed to classify, index, search and retrieve data in a security rich manner while automating regulatory monitoring and reporting. It improves availability and ease access to information to enhance decision making and better realize the value of information as an asset. It provides an end-to-end information life-cycle management solution for archived data based on a utility priced cloud service to help more cost effectively address the data management issues.

2.4.1 Three Tier Architecture

In multi tenancy the software must appear to each tenant as if he was the sole tenant of the application. So, tenants have to be segregated at some part of the application providing each tenant with its own data. This separation can be done at many different levels of the application. SCCM* differentiates between three tiers of tenant separation, web tier, application layer and data (database) tier. Following is the architecture for the SCCM which is being used in this thesis. A single Logical Partition (LPAR) is being used as the test system . In general, the system is represented in the 3-tier or 3 parts. These are Web tier, application tier and database tier.

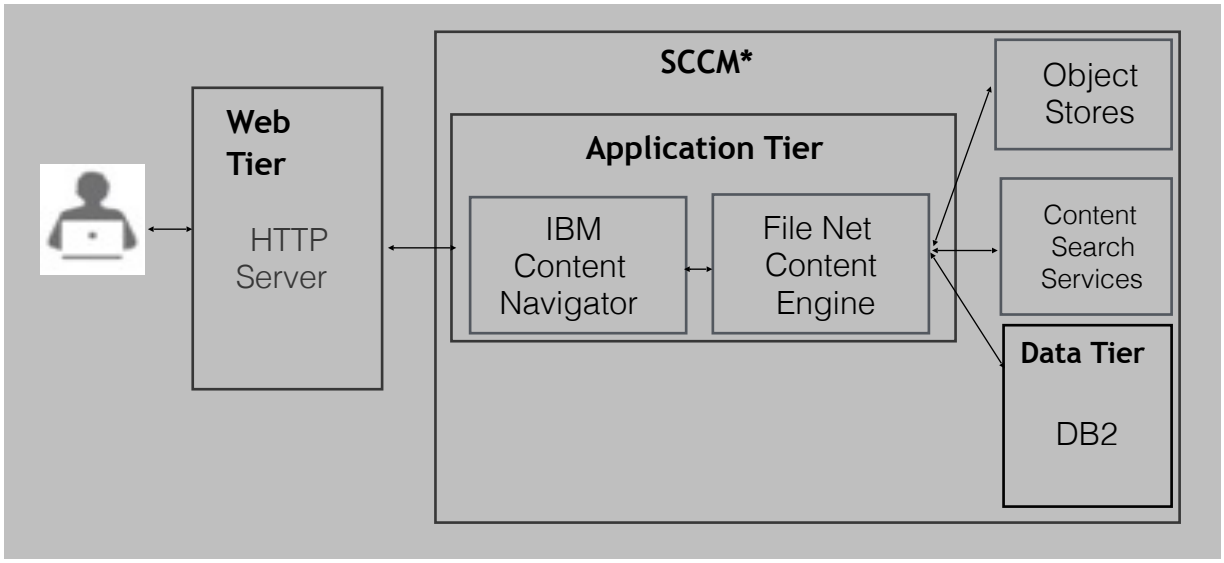

FIGURE 3. THREE-TIER ARCHITECTURE

Web Tier (Presentation Tier)

Web tier or presentation tier displays information and personalized content and makes site content available to the user. It manages the communication between web browser and a web server. http server is used as the web server for the test system in this thesis. Web server receives requests from browser and returns responses to it. A web server can also have web services that communicate with browser.

Application Tier (Logic Tier)

Application tier contains the application logic. It is the tier which actually runs the application. It can also be seen as the engine of the application. It contains IBM Content Navigator* and File Net Content Engine* for the test system in this thesis. The functionality of application tier is that if a user adds a document using the web client i.e. IBM Content Navigator from his browser, then content engine saves that document in the repository,sends the request to content search services server (CSS) to index the document if the repository in which the document is saved has indexing enabled for Content based search (CBR). Finally, the document is saved in the db2 of the database tier.

IBM Content Navigator (ICN)

IBM Content Navigator is a user interface that providers users with a console by which users can work with content from multiple content servers. This is very important component for the interactive workloads being performed directly by the user. ICN enables users to create custom views of the content on the web client (UI) by creating team spaces. IBM Content Navigator helps companies provide managed access to large volumes of electronic data. IBM Content Navigator enables users to create, retrieve, update, delete or search for (CRUDS) and work with documents that are stored in content servers and can be located around the world from a web browser.

Data Tier

Data tier consists of the database (db2 here), to save the data and metadata.Data tier manages the data. Data is inserted, updated and deleted from the database (db2).

Object Stores

Object stores are the specialized type of database designed to handle data created by web applications. For ECM systems, documents are saved in the object stores.

Content Search Services (CSS)

IBM Content Search Services is configured as an IBM Content Search Services* server on IBM File Net Content Engine*. The IBM Content Search Services server contains the connection and configuration information for a single IBM Content Search Services server and is associated with a site. Any object store in the same site can use the server to create and search indexes. Depending on the size and configuration of the system, single or multiple IBM Content Search Services server instances can be used. Our test system contains single CSS server which fulfill indexes and searches requests. In a single server configuration, a single instance of the IBM Content Search Services server performs both the indexing and searching tasks for the Content Engine. In a multiple server configuration, the server instances are assigned different roles to distribute the load for both indexing and searching tasks. IBM Content Search Services calls IBM Content Collector P8 Content Search Services Support with a set of document IDs. Content Search Services Support preprocesses each Content Collector document in the following way:

- Creates an empty XML template for each document ID

- Retrieves the archived document from the repository and adds items such as the subject and body to the XML template
- Retrieves the document attribute values and adds these values to the XML template
- Attachment data is processed by the IBM Content Search Services text extraction service and the extracted textual data is added to the XML template
- The generated XML content is indexed by IBM Content Search Services. If warnings or errors occur during this process, these messages are attached to the indexed document.

An index belongs to an index area, and an index area belongs to an object store. An index can be accessed by only one IBM Content Search Services index server at a time. Content Engine assigns indexes to the available index servers so that the work load is roughly the same for all servers. Whenever an object is created, updated, or deleted in an object store that has been enabled for full-text indexing using IBM Content Search Services, an index request is started. One or more index requests are part of an index job that is configured to run within defined time slots.

2.5 WebSphere Application Server*

WebSphere Application server is a very important component in our test system as well as in the dynamic provisioning of resources. This is because of the fact that the application servers are shared among different tenants and therefore we must know the architecture of WebSphere application server and how it is shared among different tenants. An application server is a server program in a distributed network that provides the execution environment for an application program. The application server is where an application actually executes. All WebSphere Application Server configurations can have one or more application servers. With Network Deployment, we can build a distributed server environment consisting of multiple application servers maintained from a central administration point. In a distributed server environment, we can cluster application servers for workload distribution. Following is the general overview of a web sphere application server which is part of our system's application tier. [JP]

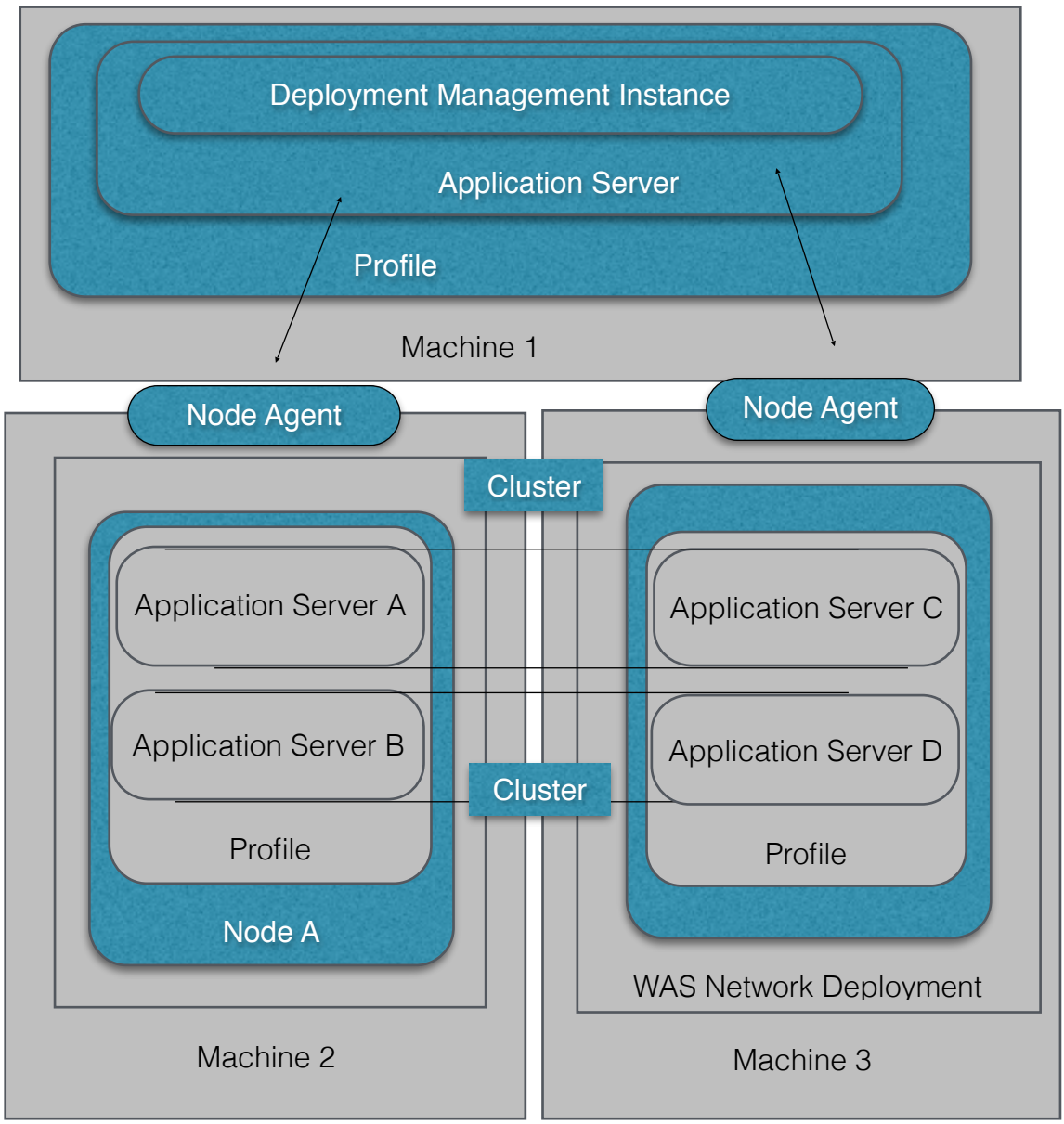

FIGURE 4. WEBSPHERE APPLICATION SERVER ARCHITECTURE

* Trademarks of IBM in USA and/or other countries

Cell

In the figure 4, cell is a grouping of nodes into a single domain. A cell can consist of multiple nodes and node groups, which are all administered from a single point, the deployment manager. If the cell configuration contains nodes running on the same platform, then it is a homogeneous cell. We can also have a cell with nodes on mixed platforms. Then it is called a heterogeneous cell. [JP]

Cluster

A cluster is defined as a group of application servers that come together for workload balancing .That means that if one application server has more workload than the other in the same cluster then, workload can be balanced .Also if one application server fails then its workload is divided among the other application servers in the same cluster. Application servers that belong to a cluster are members of that cluster and must all have identical application components deployed on them. Other than the applications configured to run on them, cluster members do not have to share any other configuration data. For example, one cluster member might be running on a large multi-processor server while another member of that same cluster might be running on a small mobile computer. The server configuration settings for each of these two cluster members is very different, except in the area of the application components that are assigned to them. In that area of configuration, they are identical. The members of a cluster can be located on a single node which is called as vertical cluster, across multiple nodes which is called as horizontal cluster, or on a combination of the two. When the application is installed, updated, or deleted , the updates are automatically distributed to all members in the cluster. [JP]

Deployment Manager

A Deployment Manager is the central administration point of a cell that consists of various nodes and node groups in a distributed server configuration. The deployment manager uses the node agent to manage the application servers within one node. [JP]A deployment manager provides management capability for multiple nodes and can manage nodes that can be on multiple systems and platforms. A node can only be managed by a single deployment manager and must be federated to the cell of that deployment manager. The configuration and application files for all nodes in the cell are centralized into a master configuration repository. This centralized repository is managed by the deployment manager and synchronized with local copies that are held on each of the nodes.

Node

A node is a logical grouping of managed servers. It is an administrative grouping of application servers for configuration and operational management within one operating system instance but by virtualization we can have more than one OS on the same machine. [JP]It is possible to create multiple nodes inside one operating system instance, but a node cannot leave the operating system boundaries. In a stand-alone application server configuration, there is only one node. With Network Deployment, a distributed server environment with multiple nodes can be configured whereas nodes are managed from one central administration server.

Node Agent

A Node Agent is an administrative agent that manages all application servers on a node and represents the node in the management cell. In distributed server configurations, each node has a node agent that works with the deployment manager to manage administration processes... A node agent is created automatically when a stand-alone node is added to a cell. The node agent's purpose is to pass information between the deployment manager and the application server.

Profile

A profile is an instance of a WebSphere Application Server configuration.Actually these are collections of user files. They share core product files. A profile contains its own set of scripts, its own environment, and its own repository. Each profile is stored in a unique directory path selected by the user at profile creation time. Profiles are stored in a subdirectory of the installation directory by default, but they can be located anywhere. An important advantage of profiles is that they allow an administrator to have multiple application servers on a single machine that all use the same binaries from one install of WebSphere Application Server. Administration is greatly enhanced when using profiles instead of multiple product installations. Also, creating new profiles is more efficient and less prone to error than full product installations. Therefore developers create different profiles for development an testing. Templates for several types of profiles are provided with WebSphere Application Server Network Deployment.

Overall System Architecture in one system stack

This is the overall system architecture from the hardware up to the application layer.[CKF]

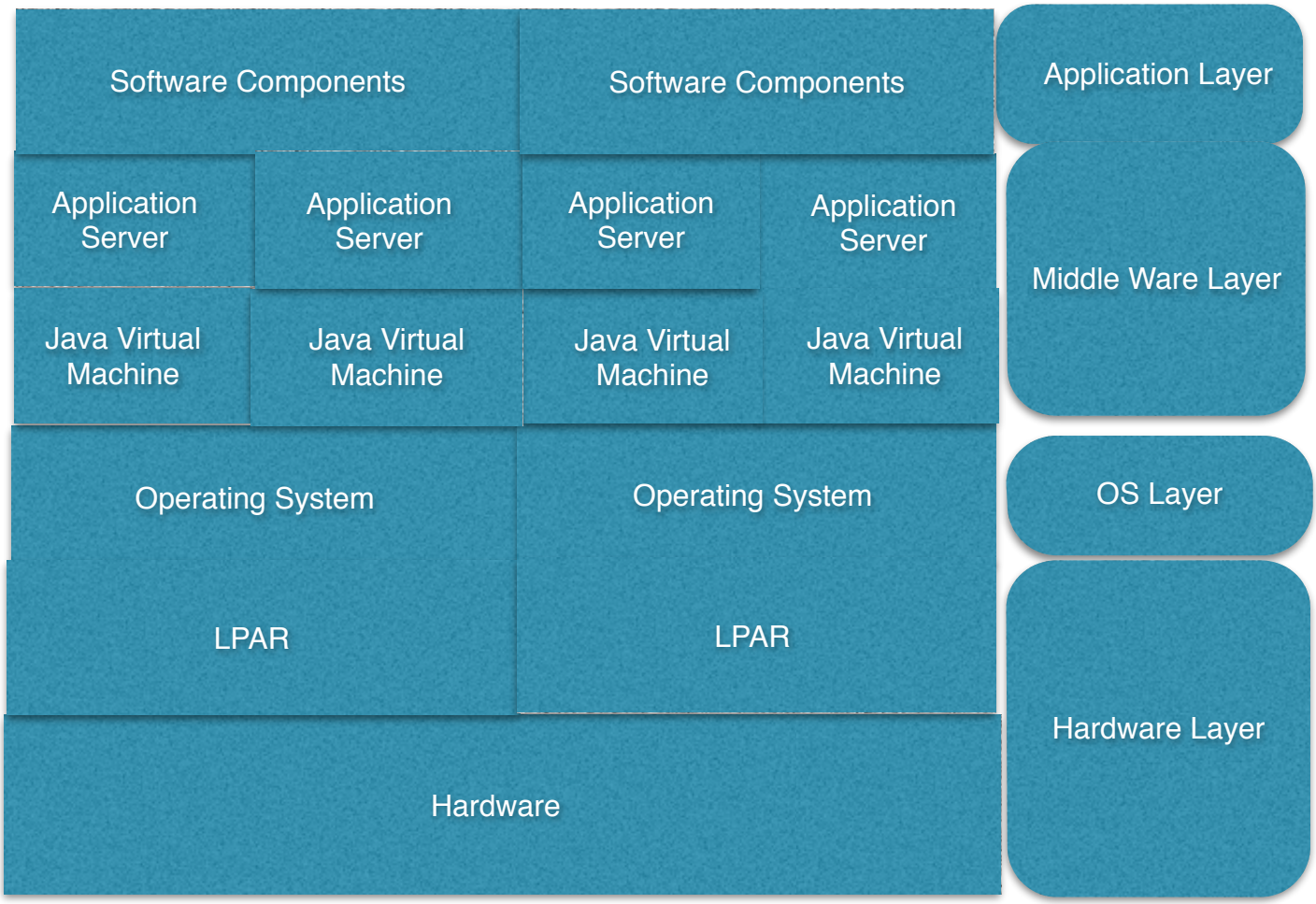

FIGURE 5. OVERALL SYSTEM ARCHITECTURE

Layers of the System Stack

Application layer, middle ware layer, operating system layer and hardware layer. In the following these layers are explained [CKF]

1. **Enterprise Java Code** This layer consists of all SCCM* application components (software components). They have been implemented using the Java programming language.

2. **Application Server** The application server is used as a container for SCCM* application components. WebSphere application servers provide the environment for the application software components. Multiple application components can be hosted by one application server.

3. **Java Virtual Machine** The Java Virtual Machine (JVM) layer is needed, since application servers hosted by instances of this layer are running Java code. One instance of this layer is used solely for one application server.

4. **Operating System** Exactly one operating system (OS) is running on an underlying logical partition. It provides the fundamental environment for the hosted middle ware. AIX is used as the operating system for the SCCM.

5. **Logical Partition** Logical partitions (LPAR) are used to separate the underlying hardware into virtual systems. Therefore, an LPAR provides a certain percentage of the power provided by the underlying hardware to its hosted operating system. Thus, an LPAR can be scaled up and down.

6. **Hardware** This layer provides all hardware resources used by software resources located on higher levels of abstraction. Provided power can be partitioned between multiple hosted LPARs.

Summarizing, figure above, illustrates all layers of the application tier system stack embedded into layers of separation. Additionally, it shows whether the multiplicity of hosted components can be greater than one.

Analogous to the design of the application tier an LPAR layer is placed on the hardware layer. One OS is hosted on this LPAR. Furthermore, exactly one database management system is running on the OS and managing databases of multiple tenants.

2.6 Test System Resources

Any system has many types of resources. They include the CPU, hard disk and memory. As we need to dynamically provision the resources according to their utilization in this thesis, following is the overview of resources and their utilization.

2.6.1 AIX Test System Central Processing Unit (CPU)

The processor core executes instructions. The amount of time a processor core is busy executing instructions is called the core utilization. A Processor core executes instructions at a rate approaching one instruction per processor cycle. A cycle is the inverse of the processor frequency. Let us take an example. If we have a processor with frequency 4GHz I.e. 4 billion cycles per second,then that means one-fourth nanosecond of the processor cycle. Execution of instructions is done through pipelining i.e. there is no start and end of execution of instructions in a cycle but multiple stages are involved and each stage is one cycle long. The instructions do not execute from beginning to end within the time of one cycle. We can understand this by the analogy of an assembly line with multiple stages. Each stage which is one cycle long does something with the instruction before passing it on to the next stage. Once the instruction is gone through one stage,another subsequent instruction can execute using this stage.

Single Pipe Architecture

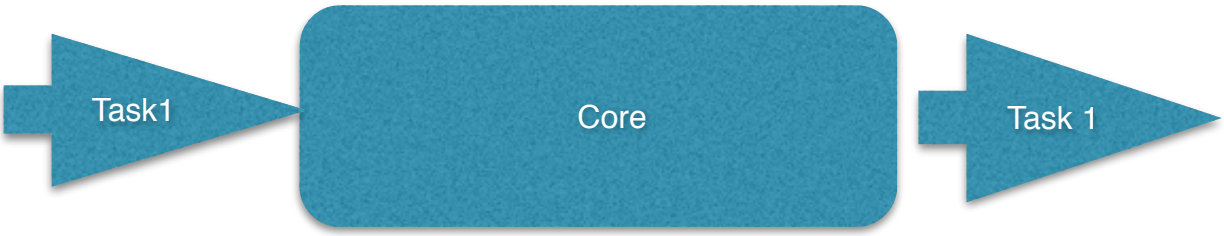

FIGURE 6. SINGLE PROCESSOR ARCHITECTURE

A single pipe architecture is shown in figure 6. In a single pipe architecture, the processor core executes instructions from the same task throughout the execution time. Therefore,the aggregation of time of tasks execution represents the Processor Core utilization. In a single pipe architecture,it is very simple to calculate CPU utilization because it is being used by only one task at a time.[IRB]

Multi Processor Architecture

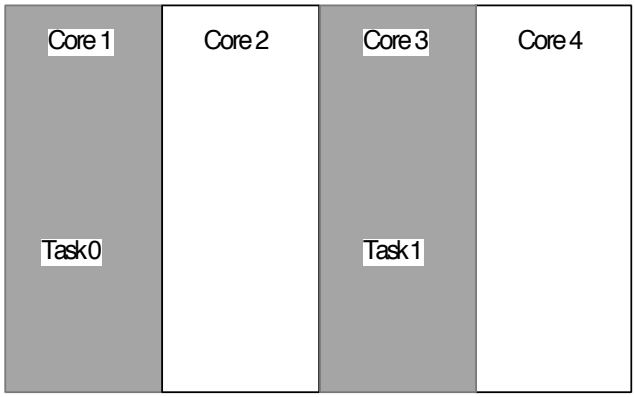

FIGURE 7. MULTI-PROCESSOR ARCHITECTURE

In a multi processor architecture ,let us assume that there are four processor cores as shown in figure 7. Core 1 and 3 are busy in executing tasks 0 and 1 respectively where as core 2 and 4 are idle with nothing to run. So, the overall CPU utilization is 50 % since core 2 and core 4 are not running anything with 0 % busy whereas core 1 and core 3 have tasks to execute with 100 % busy. This averages to the overall utilization of 50 %.[IRB]

Until now, we have calculated the processor utilization by simply taking into account the task execution.

Latest IBM POWER processors support multiple pipe lining stages for independent execution of instructions in parallel. It is shown in figure 8.

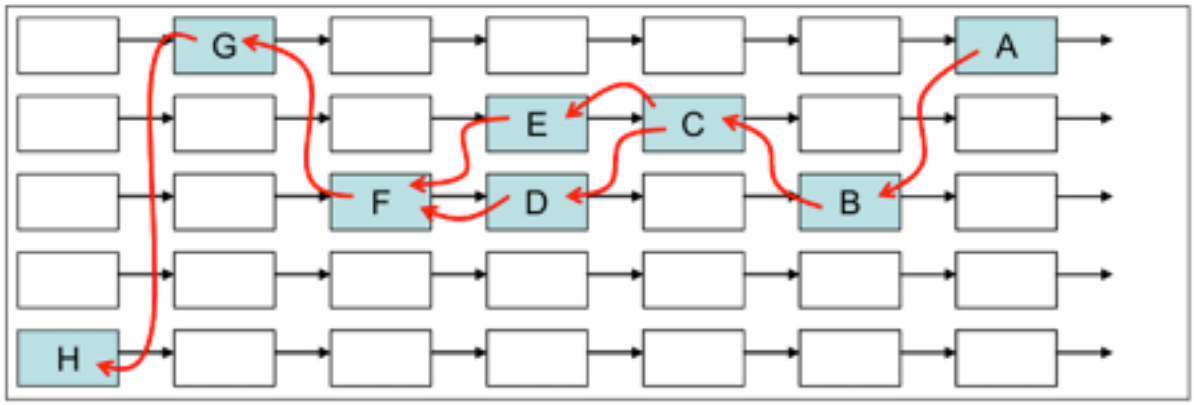

FIGURE 8. PIPE LINING EXAMPLE[BMBM]

Each row represents an execution unit and each column represents a processor cycle. Colored boxes show that execution unit is busy during that cycle. Let us take an example of a task which comprises of instructions from A to H. As we can see in the figure 8,instructions E and D can be executed in parallel but still a single task can't efficiently utilize all pipeline stages. To efficiently utilize all the pipeline stages,latest IBM processors use the concept of Simultaneous Multi-threading (SMT). SMT enables the concurrent execution of multiple instruction stream from multiple tasks on the same core. **[IRB]**

To provide SMT, the processor cores are provided with multiple hardware thread contexts. Latest (Power v7) processor's core supports one , two or four thread contexts per core. AIX operating system treats each hardware thread as a logical CPU.

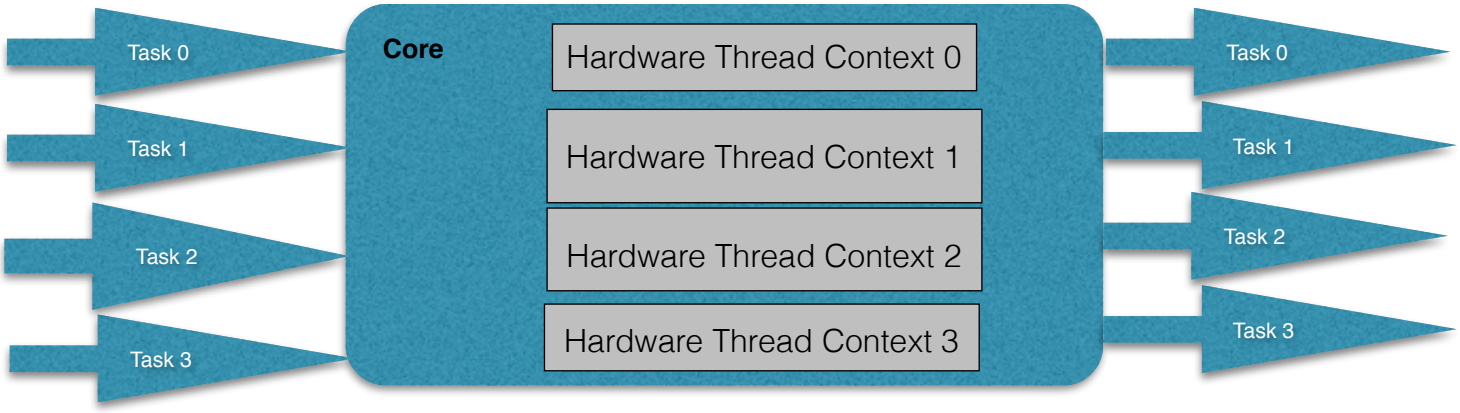

FIGURE 9. HARDWARE THREAD CONTEXTS

Now we can execute four tasks by four threads of a single core. But the throughput may not be equal when four tasks executed on four cores. Since AIX OS supports SMT by treating each hardware thread context as a logical processor ,therefore,scheduling the tasks to multiple hardware thread contexts is simple. [**IRB**]

Let us assume a web application running in SMT4 mode (I.e. Simultaneous Multi-threading with four hardware thread contexts) in a POWER7 system. Since the core is running with four hardware thread contexts,processor core capacity is increased to support four concurrent tasks execution.

If the CPU utilization from nmon is as follows (irrelevant information removed for better readability) :

cpu	%usr	%sys	physc
0	100	0	0.63
1	0	0	0.12
2	0	0	0.12
3	0	0	0.12

The last column physc shows the physical core utilization which is 63 % for the web application.

A Micro - Partitioned environment is an environment where the CPU is shared among logical partitions (LPAR) .In micro-partitioned environment the processor core is shared between various logical partitions. Each logical partition at a minimum gets 1/10th of the processor core. Therefore, a single LPAR can run with fractions of core allocated to it. A LPAR can be capped to run with the entitled fraction of cores or it can run in uncapped mode, where it can get additional fraction of cores based on the need and availability.

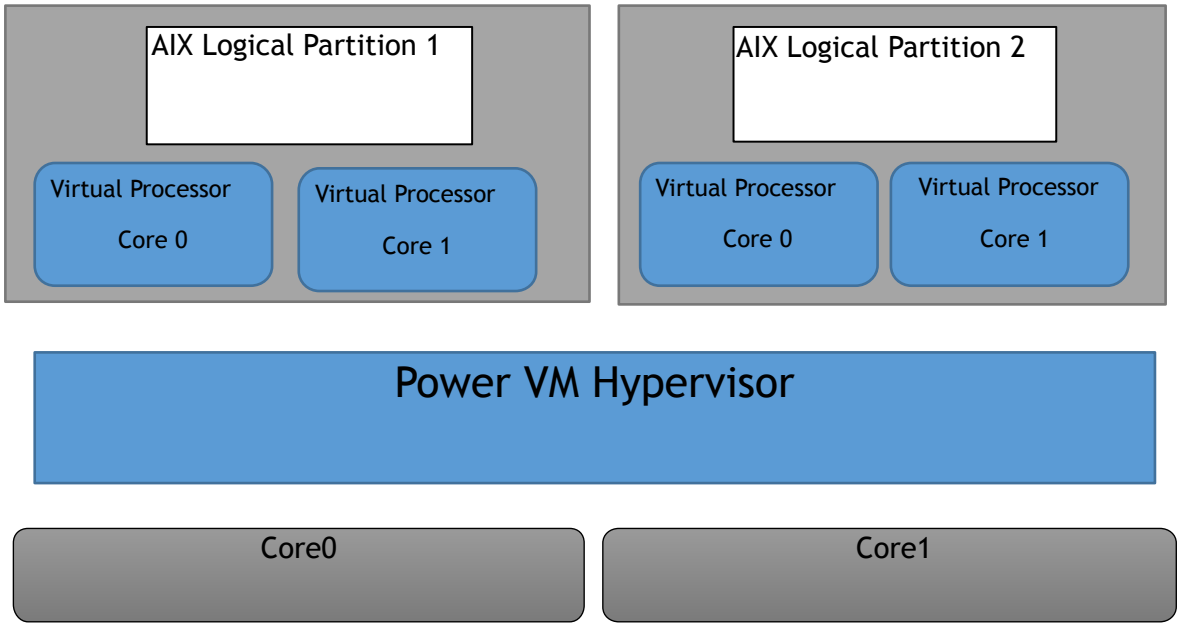

FIGURE 10. CPU IN MICRO-PARTITIONED ENVIRONMENT

2.6.2 AIX* Test System Memory

We have the IBM Power system used as the test system. Operating System is AIX 7.1. Therefore, AIX system memory is explained here. [IRB]

The physical memory of an IBM Power system can be shared or assigned to multiple logical partitions in two modes.

1. Dedicated mode
2. Shared mode

The system administrator has capability to assign some physical memory to a logical partition and some physical memory to a pool that is shared among logical partitions. [JE]

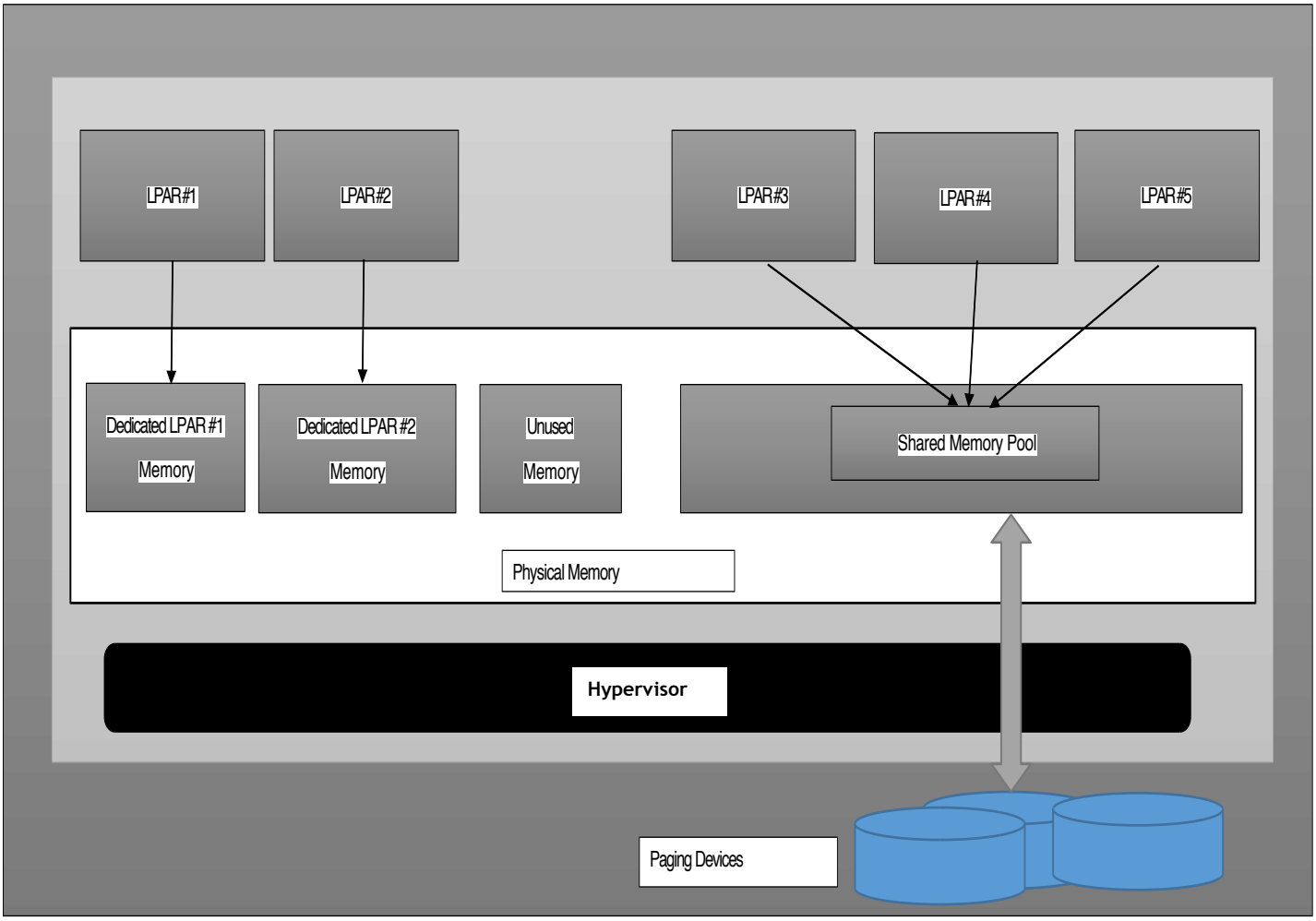

FIGURE 11. POWER VM VIRTUALIZATION ACTIVE MEMORY SHARING

A single partition can have dedicated or shared memory. [IRB]

1. **Dedicated mode:**

IBM Power system platform supports the dedicated mode. Here, the memory is distributed among the logical partitions. System administrator distribute the available memory among the logical partitions in a way that the memory utilization is optimized. If one logical partition needs more memory and another logical partition has been assigned memory which is of no use then, system administrator can assign the memory from one logical partition to another manually.

2. **Shared mode:**

Memory is distributed automatically in a shared mode. System decides the optimal distribution of the physical memory to logical partitions and adjusts the memory assignment based on demand of memory pages. The job of administrator is to just reserving physical memory to a shared memory pool and assigns logical partitions to the pool.

IBM Power System server can host multiple logical partitions, each using a part of system resources. The Operating System is inside the LPAR and hence is dedicated to a single LPAR. The OS provides access to the resources configured by system administrators. Resources include processors, memory and disk I/O.

Every resource is assigned to the Logical partitions as a dedicated resource or shred among different Logical partitions. The choice depends on performance expectations, resource optimization and cost. Typically, a single system is configured with both modes (dedicated and shared) of resources.

A LPAR has exclusive access to all the dedicated resources assigned to it. This give performance advantages on resource access time depending on the amount of resources utilization by the LPAR workload. On the server side, some LPARs can be in high stress with dedicated resources while others are using very less resources dedicated to them.[IRB]

This disadvantage is removed in the shared mode of resource assignment. Here, multiple LPARs share the same resource under the control of a hypervisor who monitors, applies allocation rules and shares the access of the resource. Each LPAR uses the shared resource as it has complete access to the resource. It is the hypervisor which manages the real access and avoids conflicts and interferences. In a micro partitioned environment explained in the CPU utilization section, this type of sharing of resources takes place. Administrator defines a pool of physical processors or memory and logical partitions can be created with a set of virtual processors and memory assigned to them through the resource pool. The system hypervisor assigns physical processors and physical memory to the virtual processors and memory for a period of time that depend on access rules and the load on all LPARs. This assignment is not visible to the OS that assigns threads to virtual processors or memory as though they were physical processors or memory.[IRB]

The Active Memory Sharing Feature allows sharing of system memory. In addition to dedicated memory assignment to single LPARs, a memory pool can also be shared among a set of LPARs.

A shared memory pool is a collection of physical memory blocks that are managed as a whole by the hypervisor. The memory in the pool is reserved and can’t be allocated to dedicated memory partitions. As an instance, we have a system with 20 GB of real memory, with 15 GB shared memory pool. Then the remaining 5 GB of memory can be assigned as the dedicated memory to the LPARs. The 15 GB of memory of shared pool is reserved for sharing even if there are no LPARs which are sharing it at the moment. [IRB].

Virtual Memory

Virtual Memory is the memory management technique which is implemented at both hardware and software level. Memory addresses used by a program called virtual addresses are mapped into physical addresses in computer memory. Main storage as seen by the process appears as a collection of contiguous segments or address space. The operating system manages virtual address spaces and the assignment of real memory to virtual memory. Memory management unit (MMU) translates virtual addresses to physical addresses. Software in the Operating system can extend these capabilities to provide a virtual address space that can exceed the capacity of real memory and hence reference more memory than is physically present. The benefit of virtual memory is that it frees applications from having to manage a shared memory space ,increases security because memory is isolated now and conceptually more memory can be used that might not be available physically with the technique of paging. [JE].

Paging

The management of memory pages is handled by the Virtual Memory Manager (VMM).[BAC] Virtual-memory segments are partitioned in units called *pages*. A *paging space* is a type of logical volume with allocated disk space that stores information which is resident in virtual memory but is not currently being accessed. This logical volume has an attribute type equal to paging, and is usually simply referred to as paging space or *swap space*. When the amount of free RAM in the system is low, programs or data that have not been used recently are moved from memory to paging space to release memory for other activities.

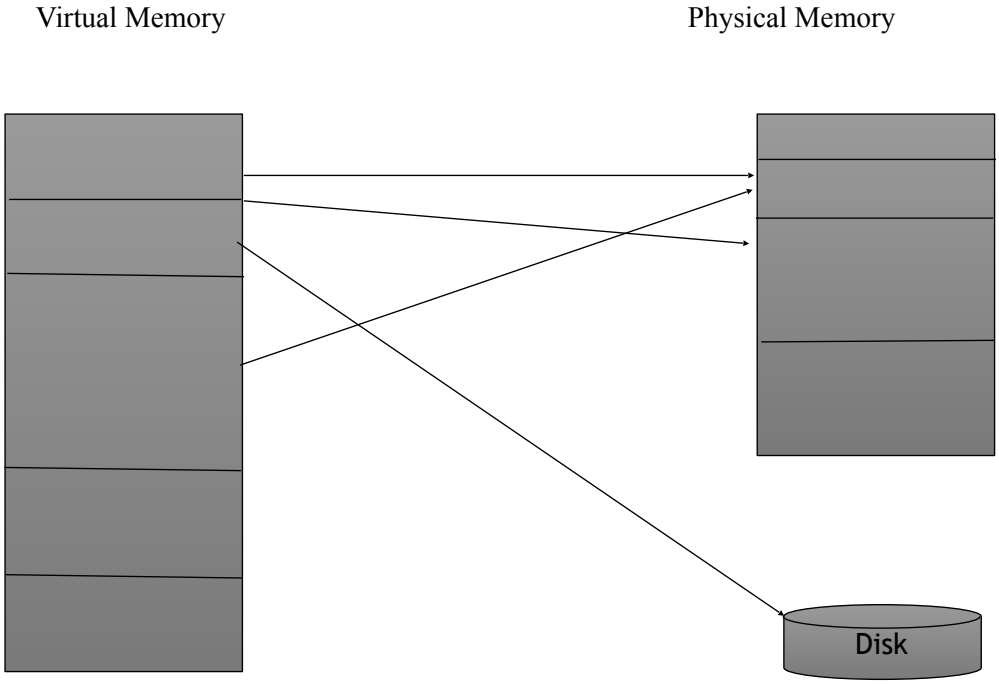

FIGURE 12. VIRTUAL VS PHYSICAL MEMORY

Computational Memory
Computational memory is used while the processes are working on computing. These working segments are temporary and only exist up until the time a process terminates or the page is stolen. These processes have no real permanent disk storage location. When a process terminates, both the physical and paging spaces are released in many cases. When free physical memory starts getting low, programs that have not used recently are moved from RAM to paging space to help release physical memory for more real work.

File Memory
File memory uses permanent segments and has a permanent storage location on the disk. Data files or executable programs are mapped to permanent segments rather than working segments. The data files can relate to file systems which is JFS2 in our test system. They remain in memory until the file is un-mounted, a page is stolen, or a file is unlinked. After the data file is copied into RAM, VMM controls when these pages are overwritten or used to store other data.

Active Memory Expansion
Active Memory Expansion (AME) compresses data within memory and helps in keeping more data in memory, and reduce the amount of page swapping to disk as data is loaded.[BAC] Active Memory Expansion is not enabled in our test system. We can improve the performance of the system with enabling the AME for the database partition.The compressed amount is defined using a compression ratio. For example if the compression is 2.0 and memory size is 4 GB then the effective memory capacity would be 8 GB.

2.6.3 File System

A file system is a set of files, directories and other structures. The file systems maintain information and identify where the data is located on the disk for a file or directory. In our test system we are using the Journaled File system 2 (JFS2) of size 200 GB.JFS2 contains a super block, allocation maps and one or more allocation groups. Each file system occupies one logical volume.

JFS2 Superblock
The superblock is 4096 bytes in size and starts at byte offset 32768 on the disk. The superblock maintains information about the entire file system and includes the fields:

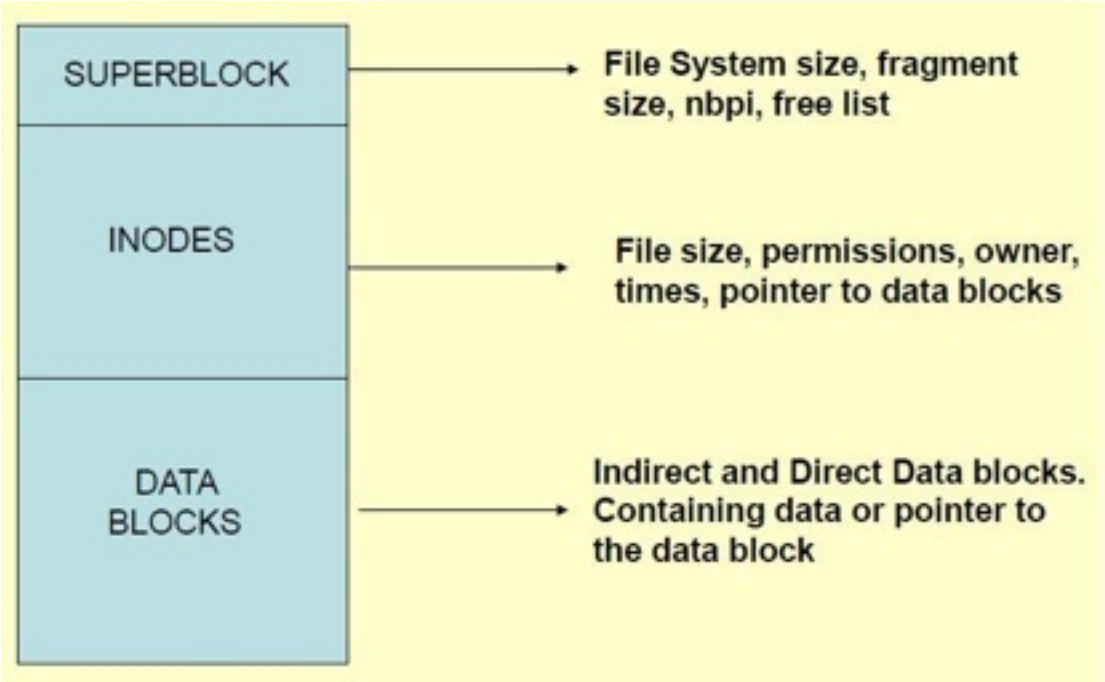

FIGURE 13. JFS BLOCKS

1. Size of the file system
2. Number of data blocks in the file system
3. A flag indicating the state of the file system
4. Allocation group sizes
5. File system block size
6. JFS2 Allocation Maps

Allocation Maps
The i-node allocation map records as (shown in the figure 13)the location and allocation of all i-nodes in the file system. The block allocation map records the allocation state of each file system block.

JFS2 Disk i-Nodes

A logical block contains a file or directory's data in units of file system blocks. Each logical block is allocated file system blocks for the storage of its data. Each file and directory has an i-node that contains access information such as file type, access permissions, owner's ID, and number of links to that file.

JFS2 Allocation Groups
Allocation groups divide the space on a file system into chunks. Allocation groups are used only for a problem-solving technique in which the most appropriate solution, found by attempting alternative methods, is selected at successive stages of a program for using in the next step of the program. Allocation groups allow JFS2 resource-allocation policies to use well-known methods for achieving optimum I/O performance. First, the allocation policies try to cluster disk blocks and disk i-nodes for related data to achieve good locality for the disk. Files are often read and written sequentially and the files within a directory are often accessed together. Second, the allocation policies try to distribute unrelated data throughout the file system in order to accommodate disk locality.

Allocation groups within a file system are identified by a zero-based allocation group index, the allocation group number. Allocation group sizes must be selected that yield allocation groups that are large enough to provide for contiguous resource allocation over time. Allocation groups are limited to a maximum number of 128 groups. The minimum allocation group size is 8192 file-system blocks.

Partial Allocation Groups

A file system whose size is not a multiple of the allocation group size will contain a partial allocation group; the last allocation group of the file system is not fully covered by disk blocks. This partial allocation group will be treated as a complete allocation group, except that the nonexistent disk blocks will be marked as allocated in the block allocation map.

2.6.4 Test System DB2*

DB2 is the database server produced by IBM. An instance of DB2 is used which include memory areas ,processes or threads. A DB2 instance is the copy of DB2 code running on a server. A DB2 instance stores information specific to that instance that includes node directory,database directory,db2diag.log,db2 notification log and dump files,database manager configuration file and db2node.cfg .i.e. configuration file. By using DB2 instances we can separate DB2 operating environment from one set of installation code.

Table spaces

All the data is for database is stored in different table spaces. Table space is a sort of sub set or child of the database. That means that we can have more than one table spaces for a database but we can't have more than one parent of a table space. Table spaces are named and divided according to their usage and management. Table spaces support page sizes of 4K, 8K, 16K and 32K.

Catalog Table space: In a database,there can be maximum one catalog tablespace and it is created automatically when the database is created by the command “Create Database”.

Regular Table space: This table stores all permanent data which includes tables and indexes. It can also have LOBs (Large objects) if LOBs are not stored in large table space separately. A table and its indexes can be divided into separate regular table spaces if the table spaces are database managed space (DMS) for Non – partitioned tables or system managed space (SMS) for partitioned tables.

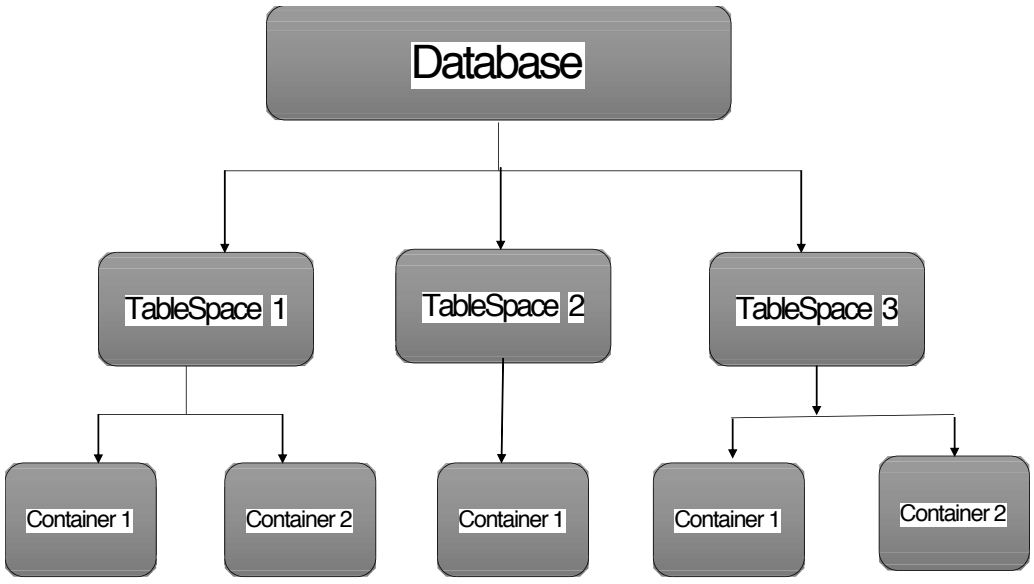

FIGURE 14. DB2 STRUCTURE

Actually, DMS and SMS are two ways of managing table spaces.

Database Managed Space

In DMS, DB2 manages DMS table spaces. Containers are defined as either files, which will be fully allocated with the size given when the table space is created, or as devices. DB2 manages as much of the I/O as the allocation method and the operating system allow. Extending the containers is possible using the ALTER TABLESPACE command. Unused portions of DMS containers can be released.

System Managed Space

In SMS, the OS manages SMS table spaces. Containers are defined as regular OS files and they are accessed using OS calls. Therefore, I/O is buffered by OS,space is allocated according to OS conventions and the table space is extended automatically when required.

Large Table space

A large table space stores all permanent data just as a regular table space does, including LOBs. This table space type must be DMS, which is the default type. A table created in a large table space can be larger than a table in a regular table space.

System temporary table space

A system temporary table space stores internal temporary data required during SQL operations such as sorting, reorganizing tables, creating indexes, and joining tables. At least one system temporary must exist per database.

All table spaces in our test system are database managed space (DMS) and we can see the type of database i.e. is it a large table space or regular space .

User Temporary Table Space:

A user temporary table space stores declared global temporary tables. No user temporary table spaces exist when a database is created. At least one user temporary table space is created to allow definition of declared temporary tables. User temporary table spaces are optional.

Containers

Each table space has one or more containers. Containers can be added to or removed from a DMS table space , and their sizes can be modified. Containers can only be added to SMS table spaces on partitioned databases in a partition that does not have a container allocated for the table space. When new containers are added, an automatic re-balancing distributes the data across all containers.

Buffer pools

Buffer pools are areas of storage in which DB2 temporarily stores pages of table spaces and indexes. When an application program accesses a row of a table, DB2 retrieves the page containing that row and places the page in a buffer. If the needed data is already in a buffer, the application program does not have to wait for it to be retrieved from disk, significantly reducing the cost of retrieving the page. A buffer pool is associated with single database and can be used by more than one table space. But there is the restriction of using buffer pools by table spaces that is all the table spaces page size should be same as the buffer pool page size. A table space can only use one buffer pool. When the database is created, a default buffer pool named IBMDEFAULTBP is created, which is shared by all table spaces. More buffer pools can be added using the CREATE BUFFERPOOL statement. The buffer pool size defaults to the size specified by the BUFFPAGE database configuration parameter, but it can be overridden by specifying the SIZE keyword in the CREATE BUFFERPOOL command. A reasonable buffer pool size is necessary for good database performance, because it will reduce disk I/O, which is the most time consuming operation. Large buffer pools also have an effect on query optimization, because more of the work can be done in memory.

2.7 Summary

The title of this thesis is “Dynamic provisioning of resources for ECM system using performance baselines in multi tenant cloud environment”. The long title contains so many terms which must be explained before doing the main work for the thesis. All the terms in title are made and defined on the top of cloud computing. Cloud computing is the basic thing which need to be understood before moving further. The concept of cloud computing is similar to any service we enjoy in our daily lives. Let us take the example of mode of public transport (lets say public bus). We just pay for our own travelling costs. We don’t care about the diesel being utilized by the public bus or the salary of the bus driver. Similarly, cloud computing offers to its clients infrastructure (hardware), platform (operating system) or software as a service and clients need to pay just what they use and don’t need to care about the costs of hardware, OS or software installation. Multi tenant cloud environment is an environment where there are multiple tenants or we can say multiple customers for the same service which has resources on its backend. These resources can be either dedicated or shared by the customers. For a typical system the resources are central processing unit, memory and disk. The test system used in this thesis is an IBM Power7 AIX* system (system with OS AIX). Therefore resources of this system i.e. Power7* CPU, Power7 Memory are described. The disk used in the test system is Journaled File system2 (JFS2). We need to describe the structure of all these resources to better understand and measure their performance and utilization. Without knowing the structure of these resources , it would be difficult to understand their utilization and performance. A term ECM system is also used in the title. ECM is enterprise content management system which is a system used to manage the content of enterprises or organizations. Performance baselines are the best performance of the system when some basic functions are performed by the system with minimal workload and minimizing the network latency. By monitoring these performance baselines provisioning of resources can be done. But how? The answer is in chapter 3.

3. The Concept

The context of the thesis is explained in chapter 2. In this chapter, the concept of thesis is described for dynamic provisioning of resources. As described in chapter 1 , heuristics approach is being used in this thesis. Performance of the system is being measured for the single interactive operations (Create, Retrieve,Update,Delete ,Search) which is called as performance baselines. Performance baselines means the performance of the system when there is no hops between client and the server to have minimum latency and there is no other operations being done on the system so that the system is only doing these operations. The response of the system in the form of KPIs i.e. response time and throughput are being monitored and these are called performance baselines. Now, once we have performance baselines, we can develop a solution on top of this for provisioning of resources with the performance baselines. This is done by doing stress testing first. The workload on system is increased to such a level that system stops responding and shuts down gracefully. Here, gracefully means that the system doesn't crash but is extremely slow. If further workload will be increased, the system will be crashed for sure. So, by doing this test, the peak limit of the workload for system is found. This is very helpful in modeling the resources for a definite workload. The concept of MAPE (monitor ,analyze,plan,execute) loop is applied in this thesis.

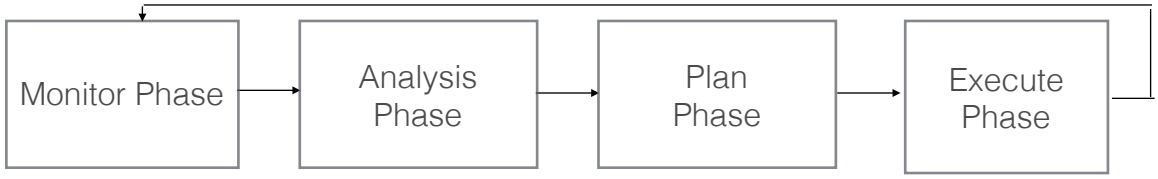

FIGURE 15. MAPE LOOP

It consists of 4 phases i.e. monitor,analysis,plan and execute phase.In the monitor phase performance baselines and resource consumption is monitored for 2 hours for a given workload which is typical of the workload by banks and insurance companies on the ECM system. By doing baseline and stress performance testing, the best and worst performance of system is measured with key performance indicators (KPIs). These performance baselines and resource consumption result is then matched with the typical service level objectives (SLOs) for any tenant or customer. Based on the performance baselines and resource utilization result, rules are generated in the analysis phase based on which the planning of the new topologies for the system resources can be designed. In the planning phase, a topology has been planned for a sample mixed workload and results of the performance baselines.Execution phase is not discussed in much detail.It is being executed in the thesis of Florian Fritz[FF].

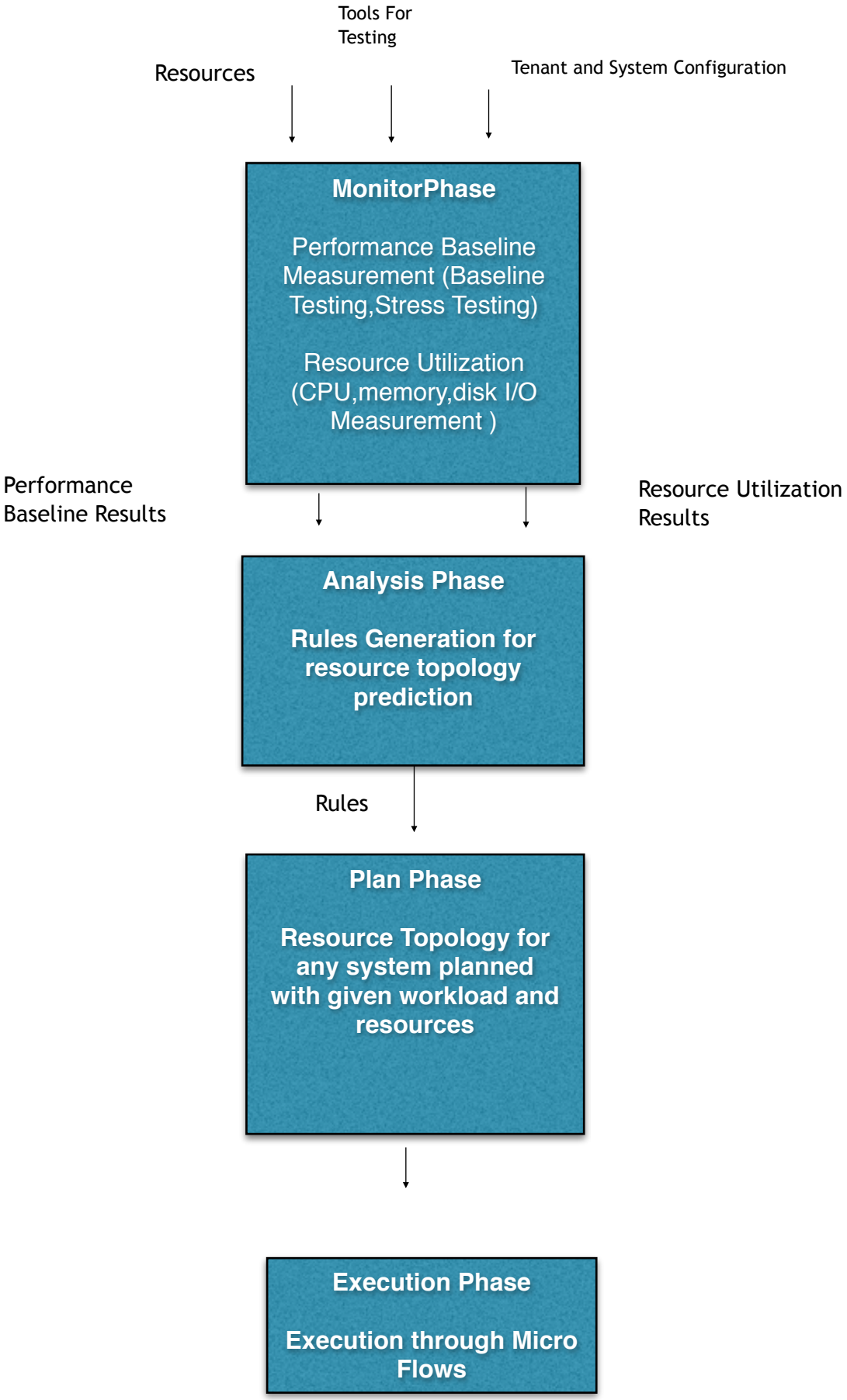

FIGURE 16. MAPE LOOP EXPLAINED

3.1 Monitor Phase

Monitoring of the system is done in this phase with monitoring of KPIs and resource utilization. KPIs are response time and throughput. This is done by performing performance testing.

3.1.1 Performance Testing

Performance refers to the way in which a system performs or functions given a particular workload. Performance Testing is a testing process that is performed to determine how fast some aspect of a system performs under a particular workload. It can also be done to validate and verify other quality attributes of the system such as scalability,reliability and resource usage. Basically,performance testing is done to identify bottlenecks of the system,and to determine whether the performance goals and requirements are being met or not by the system. This helps the stakeholders in making decisions related to the overall quality of the system.

Types of Performance Testing

Baseline Testing
Base line is a line that is the basis for any measurement. Once we measure the baselines, we can compare other measurements with the baselines and find the problems and solve them. It is done with a variable number of users to determine baselines for mainly response times. In this thesis,base lines for Create,Retrieve,Update,Delete and Search Operations are being measured with variable number of users and their response times are being monitored.

Load Testing
Load testing means testing in a test system with multiple users and determining performance under the given workload.

Stress Testing
This checks the breakpoint of the system. Here,the system is loaded to its breakpoint. This is done to determine the system breakpoint or threshold. How the system breaks and recovers is also being monitored in Stress Testing. In this thesis,stress testing is also being performed by loading the system with increasing workload until the breakpoint of the system is reached and the system stops gracefully without breaking down unexpectedly.

3.1.2 Activities in Performance Testing

1. **Identifying Test Environment :** For doing performance baseline testing a test or reference system is defined. This system has a particular platform and optimized for the workloads of the ECM system. In our case, physical test environment and the production environment(reference system) are in the same network so that the network traffic is minimized and we get the performance baselines without any nodes in between the test environment (load generator) and the system under test (production environment). Our test environment comprises of AIX 7.1 Operating System, 2 CPUs 3.55 GHz and 8 virtual processors,RAM 48 GB (+ 4 GB swap space),disk 200 GB. For load generating, we have a Windows Server operating system with 2 CPU s AMD Opteron 2,6 Ghz,memory is 8GB and the tool IBM Rational Performance Tester 8.5.1 for measuring KPIs.

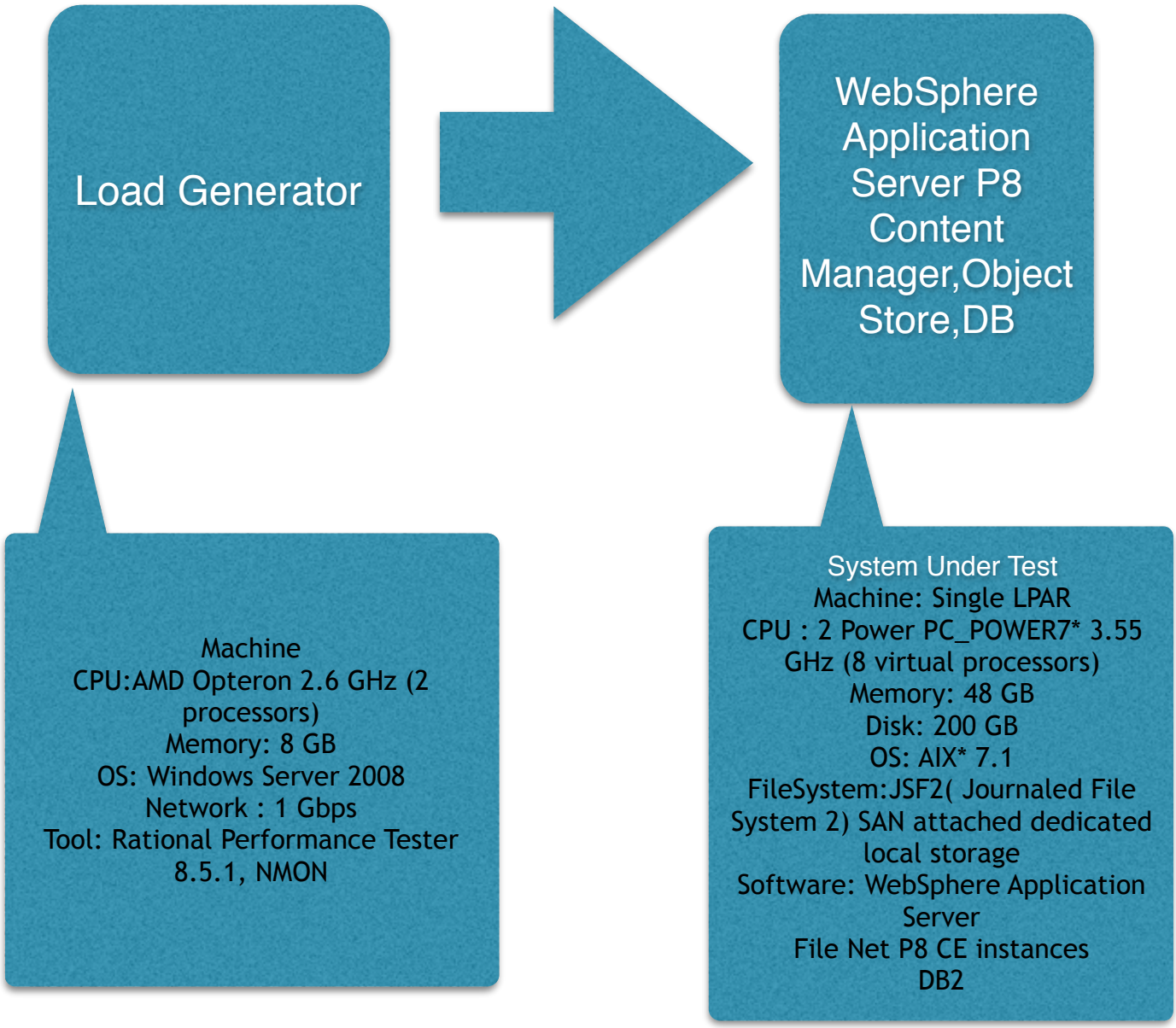

FIGURE 17. SYSTEM UNDER TEST

IBM Rational Performance Tester (RPT) is a tool for performance testing of web and server based applications . We can create tests with RPT which plays back against the server when executed and give the impression of real world scenario. Workloads or number of users to the server can be increased or decreased through RPT schedule feature. We can also monitor the system resources like db2 through RPT for a given workload. Also, server response times and throughput is measured and collected in the performance report generated automatically in RPT to identify the presence of any application performance bottlenecks. There are no hops in between test system and load generator to minimize the network latency and its effect on the test execution which is shown in the figure 17. We have also used the AIX tool nmon to measure the resource utilization like CPU,Memory and Disk I/O.

2. Identifying Performance Acceptance Criteria(Goals): Performance Acceptance Criteria is the fulfillment of the non -functional requirement of performance which is being mentioned in the Service level agreements with the customers. IBM Enterprise Content Management is being offered as the service as explained before. This service is based on the utilization of SLAs.

3. Plan and Design Tests: Tests are being recorded using RPT only once. Since,we want the performance base lines, we record the test only for one user,but we can change data during running the recorded test by creating a data pool in RPT and accessing data from that and not the data recorded during test recording. Then we can find the average of response times and the range of response times for a range of content size (for example,100 Kilobytes to 10 Megabytes)

4. Configuring Test Environment: This involves configuring the RPT tests recorded according to the operation. One modification, which is needed in each test is the substitution of security token of each request of the test recorded. This is because when we run the recorded test the security token is expired which was saved during recording. We can do this by creating a reference of the security token sent by the server in the response of the request "getdesktop.do" (for example in the case of IBM Content Navigator).We must also configure the tests depending on the different workloads to give a sense of real life scenario.

5. Implementing test design: Performance Test scripts are developed according to mixed workloads.These scripts are recorded only once by the option of Test from recording of the RPT and later on,they can be executed any number of times and anywhere. For recording a test ,the URL of the portal being test is given.In this thesis a single LPAR system is used.This is a test system with configuration defined in.

6. Execute test: This involves running the tests and monitoring of various KPIs like response time, server throughput,db2 sql statements etc. Validating tests, seeing requests' status
and result collection can be done in the Test Execution perspective where the requests headers, response headers and response content in the Protocol Data window can be seen.

7. Analyze Results, Report and Retest: Analyzing the results and checking
whether it satisfies the performance goals and objectives (SLO). Performance Reports are
generated automatically in the RPT which can be used to further analyze the results
for key performance indicators (response time, server through put time, page
throughput, server health, caching details etc.).Tests are executed again with changing the test data and think time which is set in the range of 0-10 seconds initially.

3.1.3 Performance Tuning

Performance Tuning is the activity of removing of performance bottlenecks, increasing existing throughput, improving response time and increasing existing cost. Performance tuning is a complex and iterative process. It involves establishing quantitative objectives, constant system monitoring, and selective and careful tuning to ensure that the objectives are met over time.[BMBM]
Steps Involved in doing Performance Tuning in this thesis are:

1. Establishing a base line by Base line performance testing described above.
2. Doing Stress Test and monitor the system for the breakpoint.
3. Identifying the bottlenecks
4. Tune the system (Removing bottlenecks)
5. Repeat from Step 2.

Our test system is AIX* Version 7.1 with 2 CPUs of 3.55 GHz (8 virtual CPUs), 48 GB RAM,4 GB swap space and 200 GB disk. Following techniques are implemented on our test system for tuning for better performance.

Memory Tuning

The most immediate performance improvement method is to change (more specifically, increase) the page size. This is because of the fact that now a much larger virtual memory range can be mapped to the physical memory. Large pages in a shared memory can be set by setting the v_pnshm parameter

value to vmo -p -o v_pinshm=1.We don't want to page working segments because this causes a lot of paging and decreases performance. This can be achieved by the parameters minperm and maxperm .maxperm is set to a high value (e.g. >90) and minperm to a low value (e.g. 6). There is another parameter lru_file_repage which indicates whether or not the VMM re-page counts should be considered and what type of memory it should steal (computational or file).To tune the memory, this is set to 0 (default value is 1) because then, it will tell VMM to steal only file pages and not computational pages.Also there are two other parameters minfree and maxfree.If our number of pages on the free list falls below minfree, then VMM starts to steal pages which is not good.It will continue to do so until the free list has atleast maxfree number of pages. [BMBM]

Parameters described here are set on the test system by following commands.

vmo -p -o minperm%=5
vmo -p -o maxperm%=90
vmo -p -o minfree=960
vmo -p -o maxfree=960

Shared Memory Segments Tuning
The 'EXTSHM' environment variable defines the maximum number of memory segments shared by all user-mode applications.DB2 relies on this environment variable to support large workloads.

Buffer pool tuning
The purpose of more than one user table spaces is to manage the buffer pool utilization. The goal of buffer pool tuning is to help the DB2 make use of the memory available, to its best for buffers. The decision of overall buffer size is very important for DB2 performance, because a large number of pages can significantly reduce I/O, which is the most time-consuming operation. But if the total buffer size is too large, and there is not enough storage to allocate them, a minimum system buffer pool for each page size is allocated, and performance is sharply reduced. So there is a trade off between size of the buffer pool and performance to an extent. For calculating maximum buffer size, DB2 considers all other storage utilization, the operating system, and any other applications. Once the total available size is determined, this area can be divided into different buffer pools to improve utilization. If there are table spaces with different page sizes, there must be at least one buffer pool per page size.
Having more than one buffer pool can preserve data in the buffers. For example, we can have a database with many very-frequently used small tables, which would normally be in the buffer in their entirety to be accessible very quickly. We can also have a very large table that uses the same buffer pool and involves reading more pages than the total buffer size. When the SQL query runs for reading more pages from the large table then the pages from the small, very frequently used tables are lost and make it necessary to re-read them when they are needed again. If the small tables have their own buffer pool, thereby making it necessary for them to have their own table space, their pages cannot be overwritten by the large query. This can lead to better overall system performance, albeit at the price of a small negative effect on the large query. Often tuning is a trade-off between different functions of a system to achieve an overall performance gain. It is essential to prioritize functions and keep total throughput and usage in mind while making adjustments to the performance of a system. Buffer pool size can be changed by the SQL query ALTER BUFFERPOOL with the IMMEDIATE option that takes effect right away, unless there is not enough reserved space in the database-shared memory to allocate new space. This is done automatically by DB2 self-tuning memory manager which tune the database performance according to periodic changes in use, such as switching from daytime interactive use to nighttime batch work.

Disk I/O Performance Tuning
The main bottleneck of our test system is the disk. There is a single disk for db2, CSS server etc. and all the components and this is not good because disk I/Os are enormous for each component on the single disk and the performance of the system suffers since it is always waiting for the I/Os to finish. This is recommended that a system should be such that it has different disks for CSS server, db2 and each component for better performance. Better performance means low response times as there will be different disks and hence there will not be disk I/O waiting times as each component has its own disk.

3.1.4 Performance Baselines

After tuning the system as explained in the previous section, performance baselines are measured by measuring the response time for basic create,retrieve,update,delete and search operations with the tool RPT.The requests whose response times are being measured with the RPT are:

Operation	Request Monitored by RPT
Create	Additem.do
Retrieve	Getdocument.do
Update Content	(Checkout.do, getdocument.do,checkin.do)
Delete	Deleteitem.do
Search	Search.do

Table2. Requests monitored for CRUDS

3.1.5 Workload Design

A workload is the representative mix of basic operations performed on any system.These represent the typical behavior of ECM users performing various operations.The typical workload which are designed for enterprise content management systems comprises of three types as follows.

Interactive Workloads

Interactive workloads are the basic operations and mixture of them where user interacts with the ECM system. S/he creates the content (documents in this thesis), retrieves it by downloading the document as it is or as in the portable document format (PDF).S/he can also view the document in the file viewer provided by the ICN, modify the content of document, search for the document in repositories. User can either search in the content of documents or can search for the document properties like document title,date of document creation. In this thesis, the tests are performed with content based search as it involves the functioning of CSS server.

Administrative Workloads

SCCM provides the administrative portal for the users where they can perform administrative tasks like creating, modifying or deleting users, groups and roles for the archive portal. As part of the information life cycle governance, the life cycle of content is decided by the retention policies which tells that until when the content should reside in the repository,after that the content is securely disposed.

Batch Loading Workload

Batch loading is loading the documents in bulk into the repositories. This is part of the administrative portal of the SCCM but considered as a separate workload. Bulk Documents can be uploaded as the zip file. This is performed by loading the content, archiving documents in the folder,indexing using the predefined document classes and declaring record filters into record category. Finally, the batch to be loaded, is associated with a disposition schedule (retention policy). For the tests performed in this thesis, a zip file with 1000 documents is used . The average size of document is 200 KB and the total size of batch file is 800 MB.

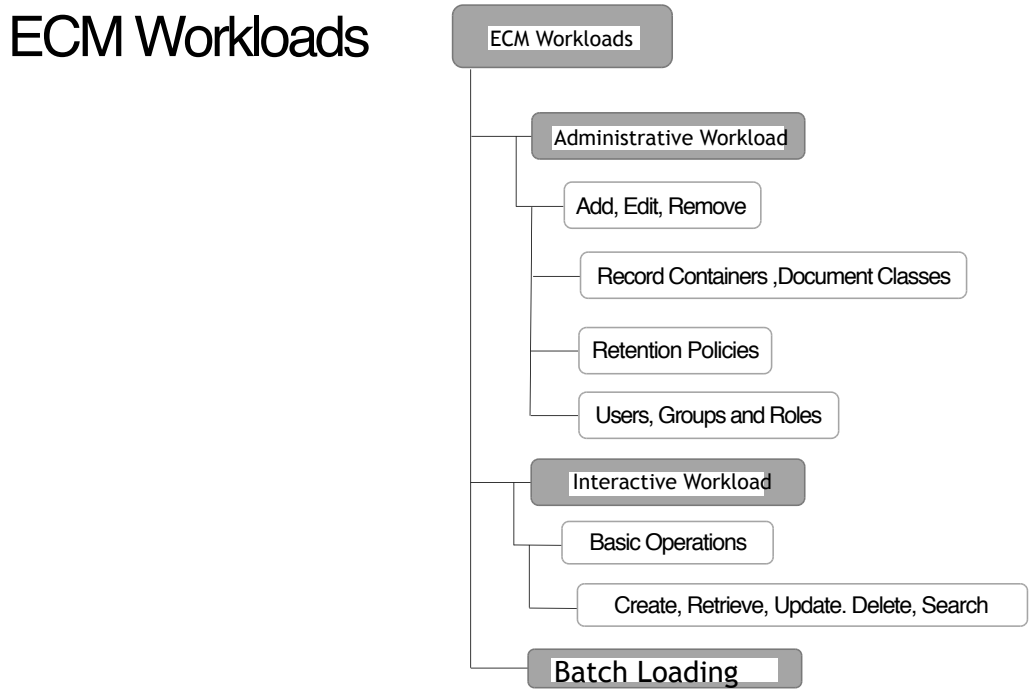

FIGURE 18. ECM WORKLOADS

3.1.5 Mixed Interactive Workload

Mixed workload is designed on the basis of experience with different customers of IBM ECM Solutions which are mainly banks and insurance companies. The mixed workload mainly consists of the functions performed by the ECM users in banks and insurance companies. We chose the one as shown in figure 19. This involves adding the document to repository, then waiting for some time (think time), then searching document for name or title (document property), retrieving document in file viewer, then changing the content of document. This is performed by checking out the document and downloading it. Then, the user modifies the content locally and then check in the document again. A typical bank user then again search for any document with a keyword (content based full text search) and finally downloading the document. Obviously, the sequence can be different and also these operations are not performed continuously.

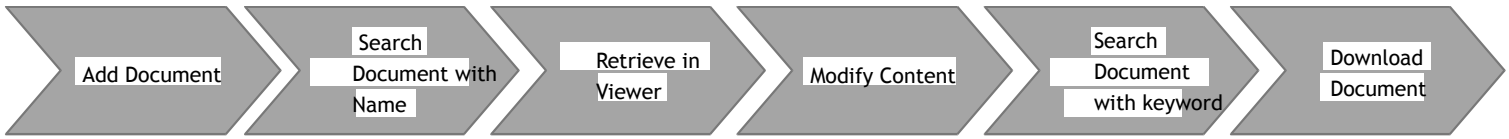

FIGURE 19. MIXED INTERACTIVE WORKLOAD

Therefore, a think time is selected between 0-10 seconds between each operation which is randomly being selected by the Rational performance tester. When user retrieves a file in file viewer,then a delay of 0-30 seconds is included, the time when user reads the document. This is done to give a feel of the real production environment. A data pool is used for these operations of figure 19. The data pool

contains the locations of the files to be added with size varying from 100 KB to 10 MB with average size of 2.8 MB. The documents being used are of the sizes 100 KB,250 KB,500 KB,1MB,5MB and 10 MB. The format of documents is also varying. It is the mixture of presentation files (ppt),spreadsheet (xls) and Portable Document Format (PDF)files. The RPT is configured in such a way that it can use this data for different operations randomly with each test execution. Figure 20 shows the average response time for single user. The first bar is p8_additem which is the request for adding a document into the repository. As shown the response time for adding a document with single user is 563 ms whereas the document size is variable . Similarly, full_text_search is the response time for full text search into the repository DOCUMENT01 and the average response time is 125 ms. Downloading the document takes 62 ms. The reason why this is very fast is explained below in the section of single download operation. Updating metadata of document i.e. (title) takes 141 ms and searching for metadata (properties) takes 47 ms. The response time for search depends on the number of documents found. The data for the test is taken from data pools. All these operations are described more accurately in next section.

Datapool

	FileLocation:String	FileName:String	DocId:String	UpdatedFileName:String	SearchString:String
0	C:\Datapoolt\Datapool\100KB.pdf	100KB.pdf	43F09093-58FC-4700-84AR-R57621F9F88F	100K.pdf	goals
1	C:\Datapoolt\Datapool\250KB.ppt	250KB.ppt	8FAAF32A-642A-4C29-9249-0768F4A6A14F	250K.ppt	specifications
2	C:\Datapoolt\Datapool\500KB.xls	500KB.xls	0D8A5805-1871-4A05-AC9D-62F8A2747DA8	500K.xls	oil
3	C:\Datapoolt\Datapool\1MB.pdf	1MB.pdf	556D2588-D285-4CF0-B6A4-4970AFR8327C	1M.pdf	clustering
4	C:\Datapoolt\Datapool\5MB.pdf	5MB.pdf	AD3C618C-D901-4314-8F22-A47716BC7209	5M.pdf	expert
5	C:\Datapoolt\Datapool\10MB.pdf	10MB.pdf	D4A15ACF-DE36-4335-9585-81F81FFB2295	10M.pdf	xerox

Table3. Datapool for RPT

Each time the user performs these operations, they use different documents or text. For example, not the same document is added each time. A data pool is created which contains the location of different documents ranging from size 100 KB to 10 MB with average size of 2.6 MB. Each time the test is run, a document is randomly selected from the data pool and is added to the repository. Similarly, for the search operation, not the same text is searched each time, the user performs Content based full text search. A data pool is created which contains different strings which are searched each time the test is executed. This whole process is done as follows using RPT.

A test is recorded with interactive mixed workload as defined in figure 19. User logs in the Archive Portal and a particular document is added and a particular string is searched on the Archive Cloud Portal during the test recording. Finally the user logs off the portal. Now RPT has all the requests from the user and responses from the server recorded. The string of data being added is substituted by the column of the data pool which contains the location of different documents. Also, the string being searched during recording is replaced by the column of data pool which contains the different strings. The access mode of columns can be set as sequential or random. Random access is being selected in this thesis. We can even substitute the host name, user name and password with the data pool entries but it is not done in this thesis since there is only a single test system.

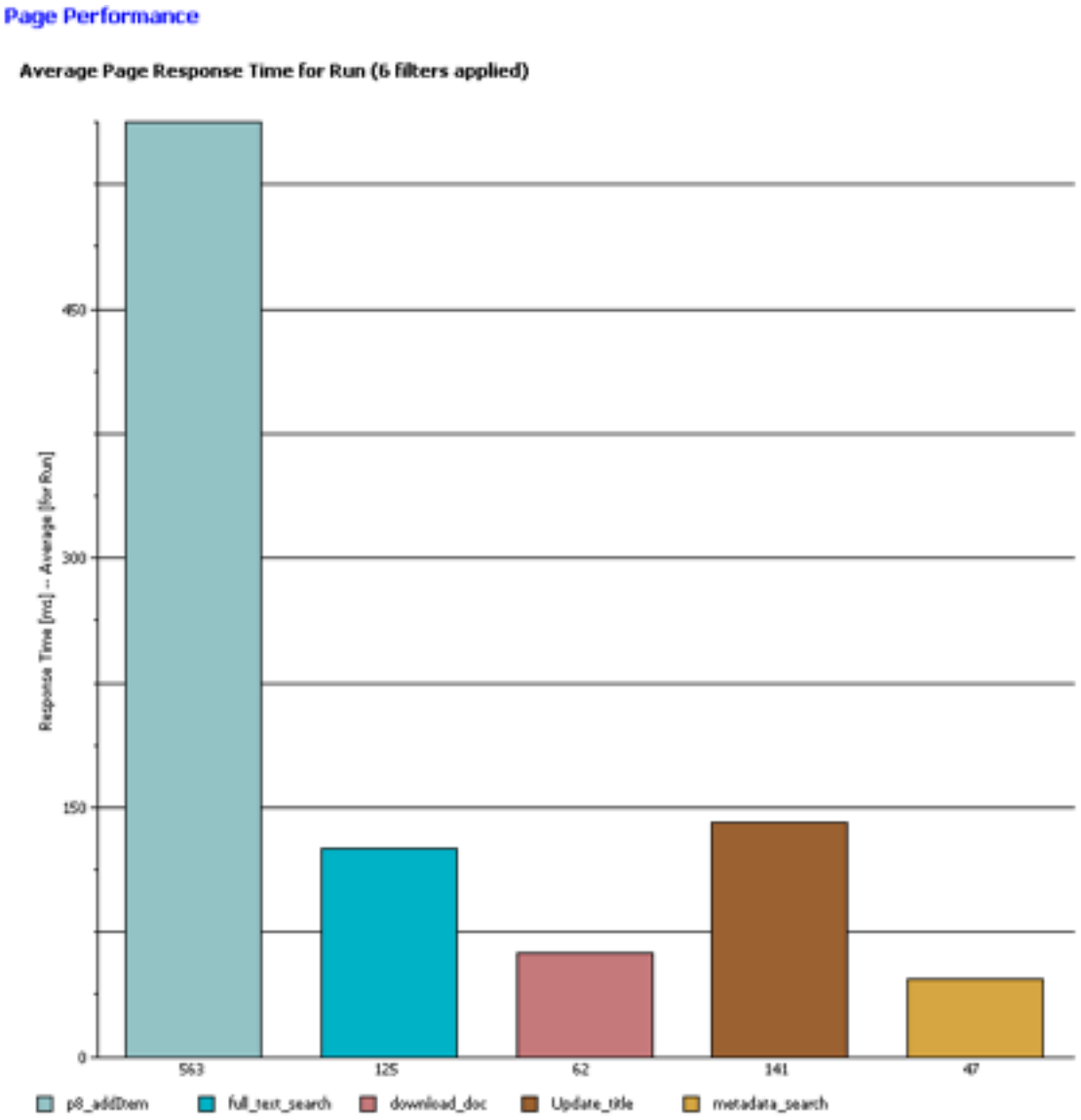

Figure 20. Response Time for different pages with mixed interactive workload with single user

3.1.5 Performance baselines for CRUDS operations

The interactive workloads designed for the SCCM comprises of basic operations like create, retrieve,update,delete and search for the documents. For measuring performance baselines, single operation response time is being measured with single user with range of data being used during the test execution. Tests are being executed with Rational Performance Tester and system under test is as explained in the performance testing

Create Operation

User adds document to repository. Various sizes of documents will be tested as explained in table3. A test is recorded using RPT and a single document is created in the repository. Then, the binary data chunk of the created document in the request "*additem.do*" is substituted with the column name of data pool which contain locations of the documents as shown in the figure above. Then, the recorded test is being run repeatedly which create a document from the location of data pool each time (sequentially, randomly or in shuffled mode) in the repository. By parsing the p8_server_trace.log files for node 1 and node 2 of the Websphere Application Server, it is found that during a create operation document is added to the repository and the information about the document (document ID,title,created

on etc.) is saved in the DB2. The request and response calls go in a sequence as shown in the three tier architecture in chapter 2 figure 3.

If the Context based Retrieval is enabled for the repository in which document is saved,then indexing of document is done when it is added to the repository. However,this indexing doesn't effect the response time for the create operation. This is because of the reason that indexing is done with the Content Search Services (CSS) Server which queues the requests of indexing to it. So, Content Engine responses back once the document is added to the repository and DB2 and doesn't wait for the CSS Server to complete the indexing job.
When we add a document, it is uploaded in the repository .Metadata is inserted into db2 table docversion.

File Name	File Size(KB)	Response Time (ms)	Throughput (KB/s)
100KB.pdf	100	188	396.83
250KB.ppt	250	331	543.48
500KB.xls	500	391	1012.15
1MB.pdf	1024	610	1294.56
5MB.pdf	5120	2703	1603.01
10MB.pdf	10240	4625	1947.88

Table4. Create Operation KPI

The response time and throughput of the create operation when the file of given size is added to the repository are shown in the table 4. It can be seen that the response time increases with the increase in size of data. Minimum response time is for adding 100 KB file whereas maximum response time and throughput is for adding file of size 10 MB. If content based retrieval is enabled then,indexing is done when a document is created or updated as described in chapter 2. The object stores and index areas available in our test system are as follows. During indexing of documents,the index belongs to the index area and index area belongs to the object store.

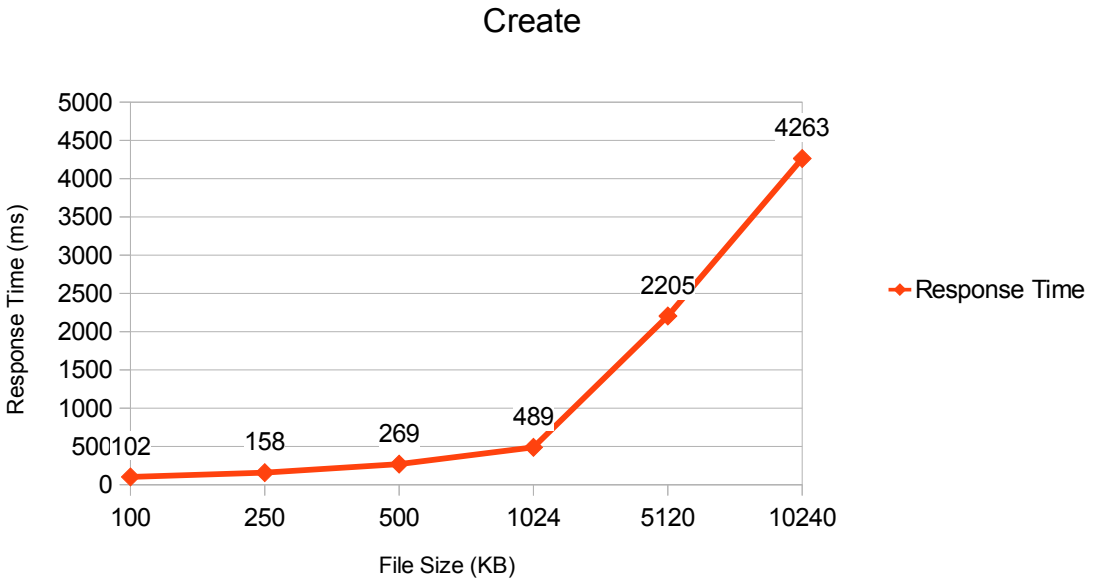

FIGURE 21. CHART FOR SINGLE CREATE OPERATION

Object Stores:

```
root@sccm-c053 # ls /sccm/was
DISCOVERY01    EMAIL01        SCCMREGISTRY
DOCUMENT01     FILEPLAN01     export
```

Index Area:

```
DISCOVERY01_IndexArea    EMAIL01_IndexArea        SCCMREGISTRY_IndexArea
DOCUMENT01_IndexArea     FILEPLAN01_IndexArea
```

Retrieve Operation

A test is recorded using RPT in which a document is retrieved (downloaded and directly viewed in the file viewer) from the repository. The test is automated by having two data pool columns which consists of document ids and titles of the documents to be retrieved (data pool shown in table 3). Then, we substitute the initial document id and title in the request recorded "*retrieve.do*" with the column names of data pool which contain document ids and titles .Then, the recorded test is being run repeatedly which search for the particular document id (and corresponding title) from the column of data pool each time (sequentially, randomly or in shuffled mode).

Retrieving is done from repository (File Content Area). ICN calls CE and CE calls the WAS object store (repository) where the document is stored. Tracing the retrieve operation in the Content Engine trace files of nodes 1 and 2, it is found that when the end user downloads the document , the document

is recognized by the document ID and it is accessed or downloaded from the repository. Caching is disabled so that actual results can be seen.

Retrieving faster than Creating

As we can see from the graphs that downloading or retrieving is faster than uploading. This is because of the reason that Internet service providers(ISP) design their systems to give priority to downloading. So upload speed is always slower than the download speed. They design the system like this, because most of the time data is being downloaded than uploaded. There is a ratio of 1:3 of the uploading and downloading speed. That means that in the time it takes to upload a particular size of data, it can download the same size of data 3 times. This is proved by measuring the upload and download speed of the network of the test system. The network used in this thesis has upload speed of 31,71 Mbps and the download speed is 87.37 Mbps.

Also, the upload and download speeds will almost never match the maximum speed of the connection. It is up to get 80-90% of the maximum. This is because of the reason that the connection may be shared with other people, so if a lot of other people are using the connection a slowdown is experienced.

File Name	File Size (KB)	Retrieve Response Time (ms)	Throughput (MB/s)
100KB.pdf	100	36	2.7
250KB.ppt	250	49	5.1
500KB.xls	500	60	8.3
1MB.pdf	1024	78	13
5MB.pdf	5120	125	40
10MB.pdf	10240	165	62

Table5. Retrieve Operation KPI

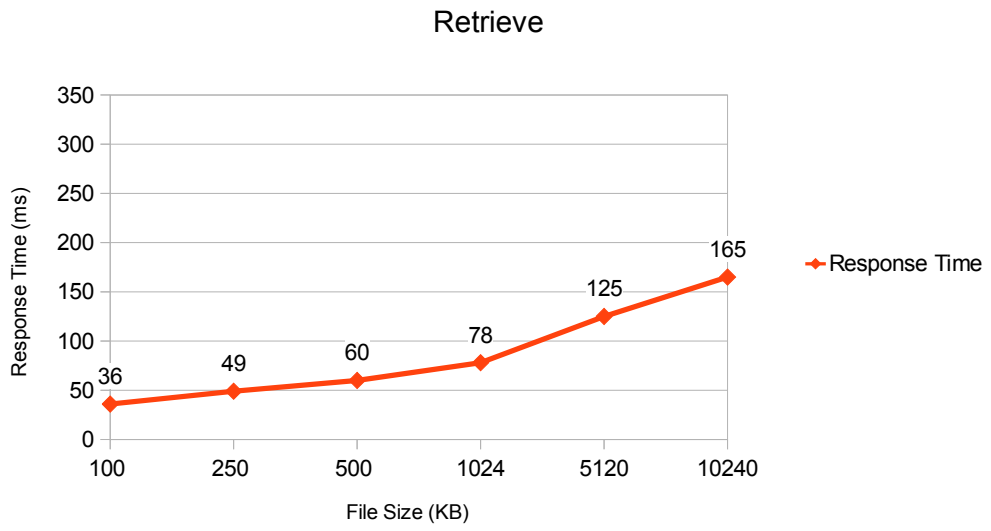

FIGURE 22. CHART FOR SINGLE RETRIEVE OPERATION

Update Operation

Updating involves checking out of the document and downloading it. Then,changing the content of data locally and then checking in the document again. The time user takes to modify data locally is being put in the range of 0-20 seconds. Response time recorded ,is consistent with the individual response times of downloading document,checking out and checking in operations. By monitoring the response time for check-in operation it is found that it is equivalent to the adding a document.

Response Time (Update)= Res. Time (Retrieve) + Res. Time (Checkout only) + Res. Time(Add)

File Name	File Size (KB)	Response Time (ms) (check out+ download+checkin)
100KB.pdf	100	354
250KB.ppt	250	476
500KB.xls	500	607
1MB.pdf	1024	821
5MB.pdf	5120	2880
10MB.pdf	10240	4784

Table6. Update Operation KPI

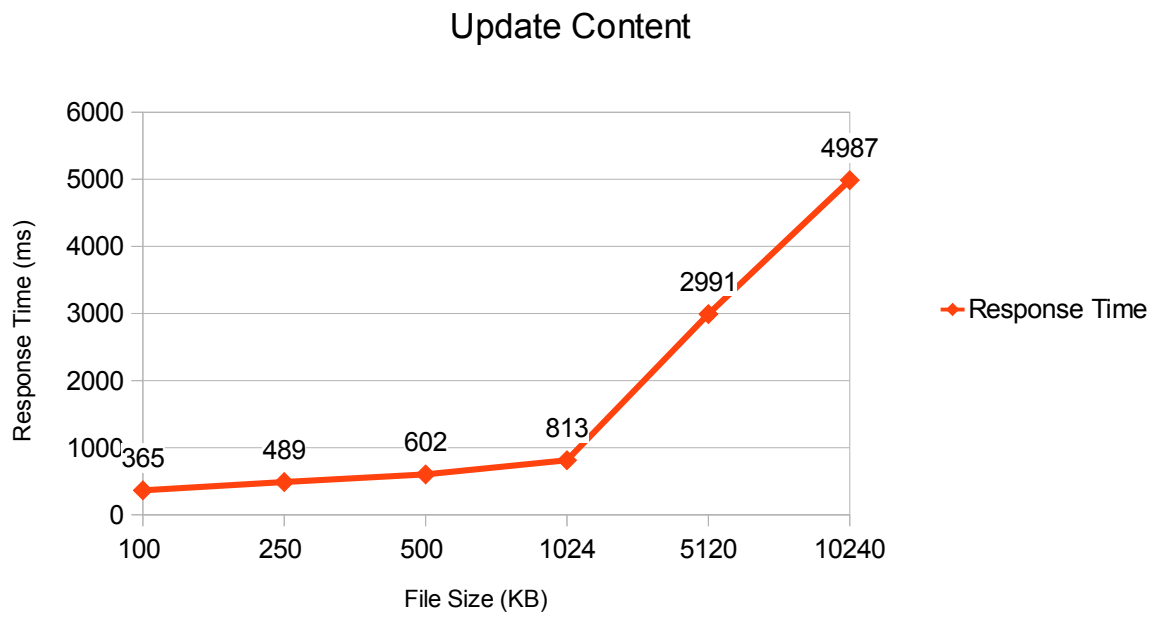

FIGURE 23. CHART FOR SINGLE UPDATE OPERATION

Delete operation

Deleting document form repository will also update the CBR index. A document from repository of 100 KB size is deleted during the test recording with RPT. To automate this test, document id is substituted in the request "*delete.do*" (with the column name of data pool which contains document ids) which was recorded initially. Each time, the test is run it will delete the document (of corresponding document id in data pool) in the repository. If document of the particular document id doesn't exist or deleted before in the repository, it will give an error. By going through the log files, it was found that in delete operation,document is deleted from WAS storage area and from database. CSS index entry is also deleted from the CSS server. CSS server queues this request of deleting. So,the user gets the response of deletion once,the document is deleted from WAS repository and database. The document is deleted from the database table docversion and also the storage area where document was stored is cleared. It intends to delete the resource or move it to an inaccessible location. As it is seen from the table below, delete operation is quite fast and almost same response time is there irrespective of the document size.

File Name	File Size (KB)	Delete Response Time (ms)
100KB.pdf	100	46
250KB.ppt	250	63
500KB.xls	500	47
1MB.pdf	1024	78
5MB.pdf	5120	62
10MB.pdf	10240	62

Table7. Delete Operation KPI

Full Text Search Operation

There are 1 million documents in the repository and the indexed object size is also 1 million. CBR searches are performed using RPT .A test is recorded using RPT in which a single keyword is searched in the repository. We can automate this test afterwards by adding a data pool column which consists of all the strings we want to search.

Then, we substitute the initial search string in the request recorded first with the column name of data pool which contains strings. Then, the recorded test is being run repeatedly which search for the particular string from the column of data pool each time (sequentially, randomly or in shuffled mode).

Expected response time with search results is up to 3 seconds which is typical for a search operation for a web application. This operation can only be performed in a repository where Content Based Retrieval (CBR) is enabled. When the user enters a keyword and searches for the documents containing that keyword, then the CE calls CSS with the given keyword. CSS has all the XML file templates for all document Ids. It checks in all the XML text area and finds all the document IDs which contains that text. Once the document IDs are received by the CE, it retrieves the list of documents with the given IDs from the repository and shows to the user. As it can be seen in the figure 24, response time is directly proportional to the number of files found. As the number of files found increase, so is the response time . The tests are performed for the content based full text search therefore CSS server searches in the repository for the keywords in the documents and then access these documents by DB2 sql queries.

String	Files Found	Response Time (ms)
goals	406	1734
specifications	181	875
oil	240	1063
clustering	179	907
expert	287	1344
Xerox	10	172

Table8. Full Text Search Operation KPI

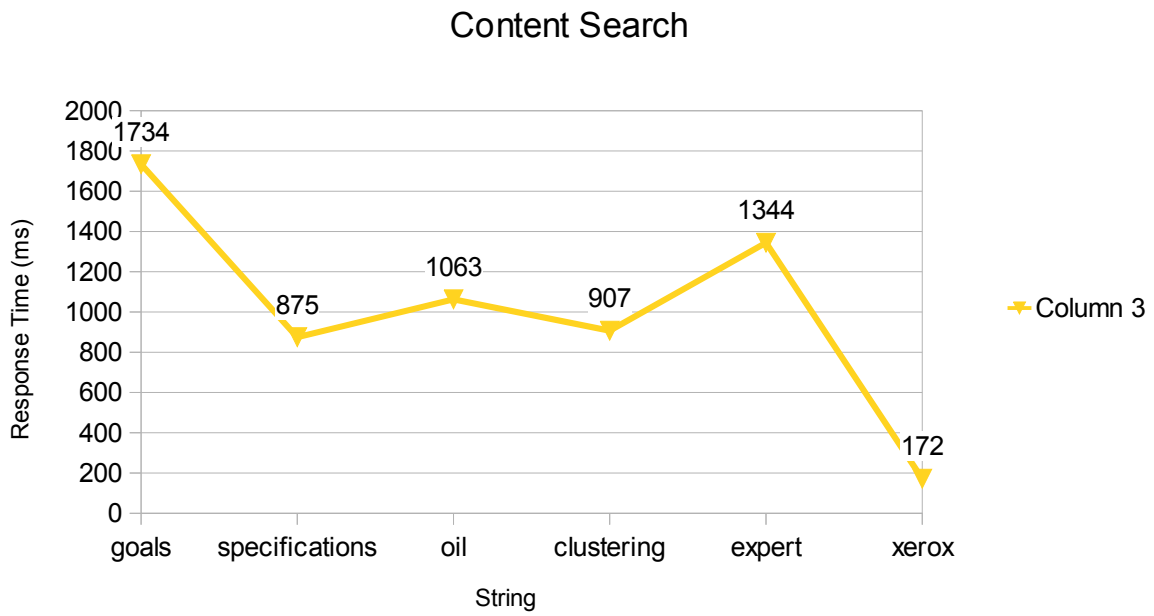

FIGURE 24. CHART FOR SINGLE FULL TEXT SEARCH OPERATION

3.1.6 Multiple Users with Mixed Interactive Workload

The test with mixed workload is performed with different number of users also ranging from 0-200. In this thesis since we have only a single test system. Number of users are from 0-200 ,starting from 0 and increasing by 20 every 10 minutes. These are the virtual users created in RPT whereas each virtual user is approximately equals 20 real users. Therefore the system peak limit is 200* 20= 4000 concurrent users. This is done to have the sample data (response time, throughput) with these number of users. At 200 virtual users, the system becomes extremely slow and when the number of users are increased from 200, the system stops responding. At 200 users the resource utilization is optimum with 92 % CPU utilization, and only 10 % free available memory with typical average response time 2 seconds

which is optimum for a typical web application as after 2 seconds user assumes that the system is not working.

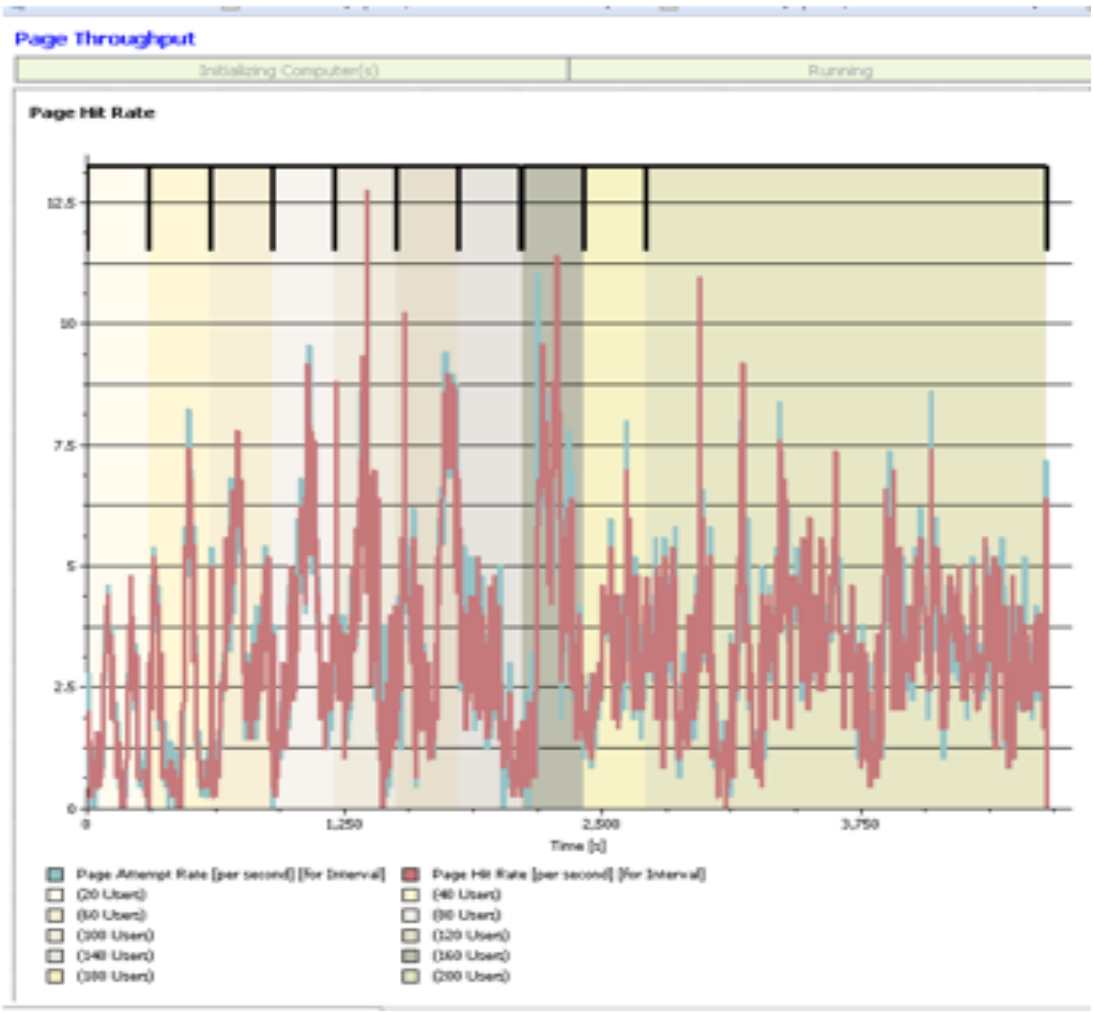

FIGURE 25. PAGE THROUGHPUT FOR 0-200 USERS

In the figure 26, page throughput is shown with the number of users. While executing the test, in RPT, (since RPT playbacks the recorded tests with the defined number of users) nmon data is also recorded for CPU, memory utilization and disk I/O. This effects the performance of the system with higher response times and low throughput. This is examined and proved by the tests performed through the RPT for the test system and also for the system with different disks for different components but the configuration is same. That means the hard disk is still 200 GB,memory is still 48+4 GB and so on and so forth. Figure 27 shows the ideal response time for the system with users from 0 to 200. As it is seen the response time is almost constant from the number of users 0 to 180. When the number of users are increased from 180 to 200, the response time increases steeply and from 600 ms to almost 2 seconds.

As described in the performance tuning of disk, the test system has a single disk for all its components.

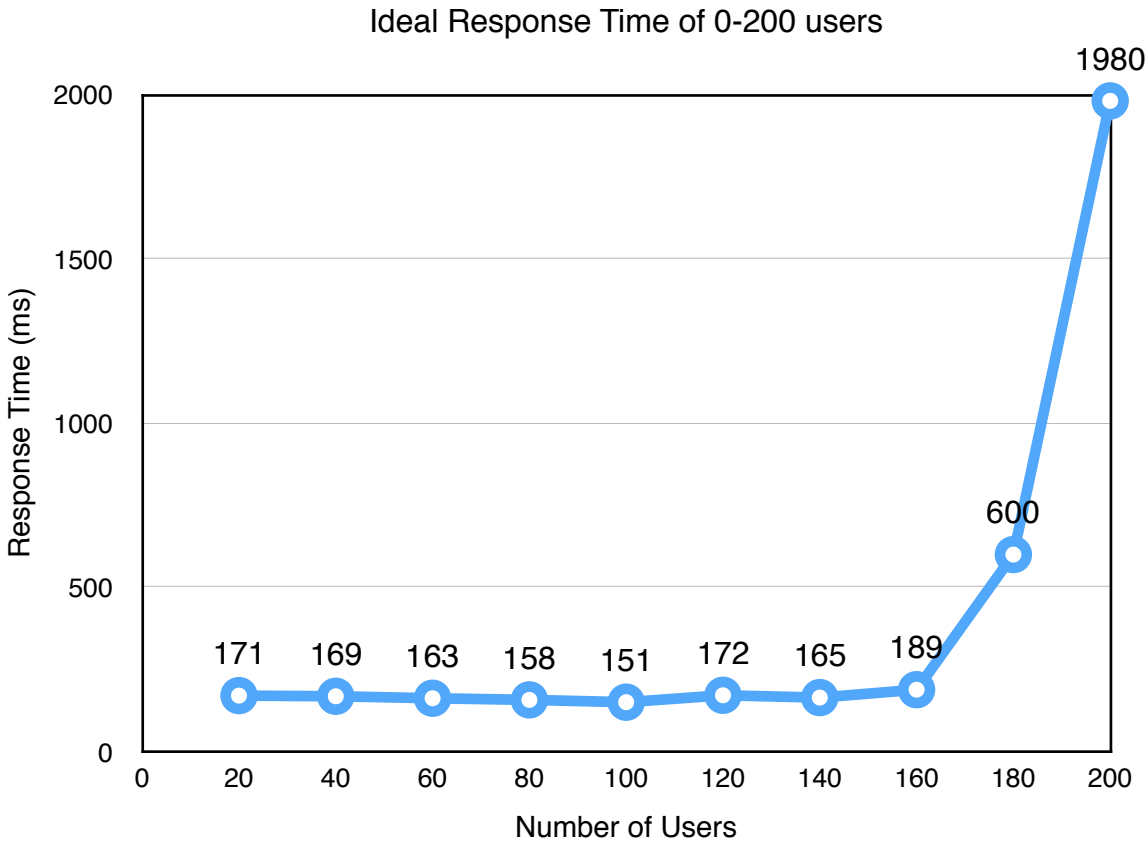

FIGURE 26. IDEAL RESPONSE TIME FOR INTERACTIVE MIXED WORKLOAD WITH 0-200 USERS

That means the system is behaving abnormally with the extreme workload and stops responding or responding too late. In our test system the actual response times are shown in figure 28 for the number of users from 0 to 200. As it is seen , the response time is increasing each time the number of users are increasing That means there is already a bottleneck of resources when number of users are 20. At 160 users, the system is behaving normally but when the workload is increased from 160 to 180 the system

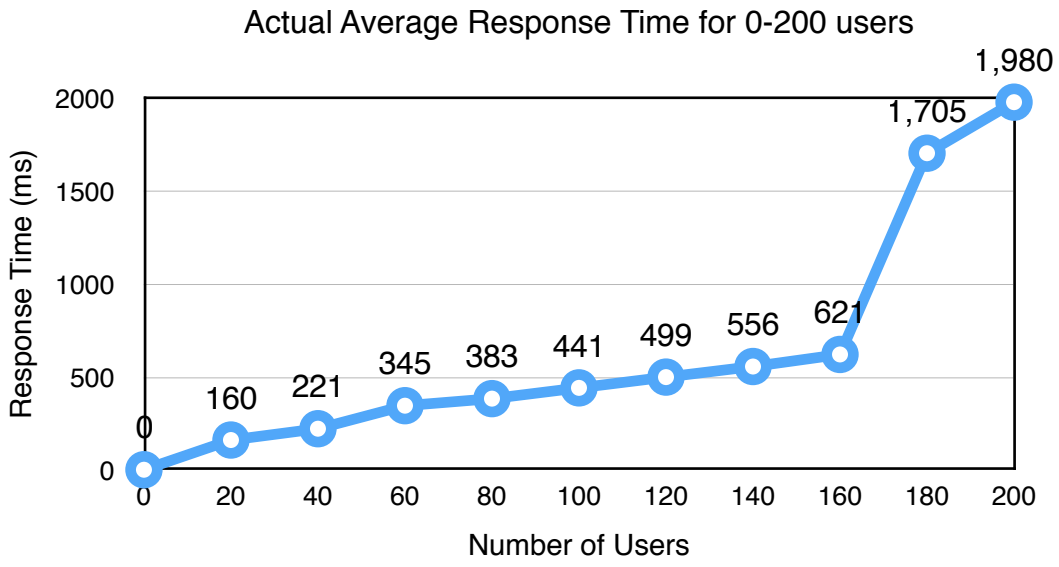

FIGURE 27. ACTUAL AVERAGE RESPONSE TIME FOR INTERACTIVE MIXED WORKLOAD WITH 0-200 USERS

3.1.6 Monitoring of resource utilization

The test system configuration is explained in the performance testing section. According to that the test system consists of 2 CPUs of 3.55 GHz each with 4 threads, 52 GB of memory and 200 GB file system. The utilization of these resources are explained below.

3.1.7 CPU Utilization with 0-200 users

To measure the CPU utilization a mixed interactive workload is designed (figure 19), and this workload is assigned to the system with number of users from zero to 200 increasing by 20 in every 10 minutes with each user performing the same workload with variable data accessing through data pool. All this is done through the RPT tool. The following figure shows the maximum CPU utilization as 1.867 where 2 CPUs are entitled which is almost 92 %. To find out that, how much of CPU is being utilized by WAS processes and db2 processes, top processes utilizing CPU are also recorded and based on the top processes consuming most of the CPU following graph is made which says that WAS processes are consuming almost 80 % of CPU whereas almost 12 % of CPU is being consumed by the db2 processes when the number of users become 200 and the throughput is maximum 46 . The vertical axis in the graph below,suggests the % CPU (Core) consumption where as horizontal axis shows the number of users.

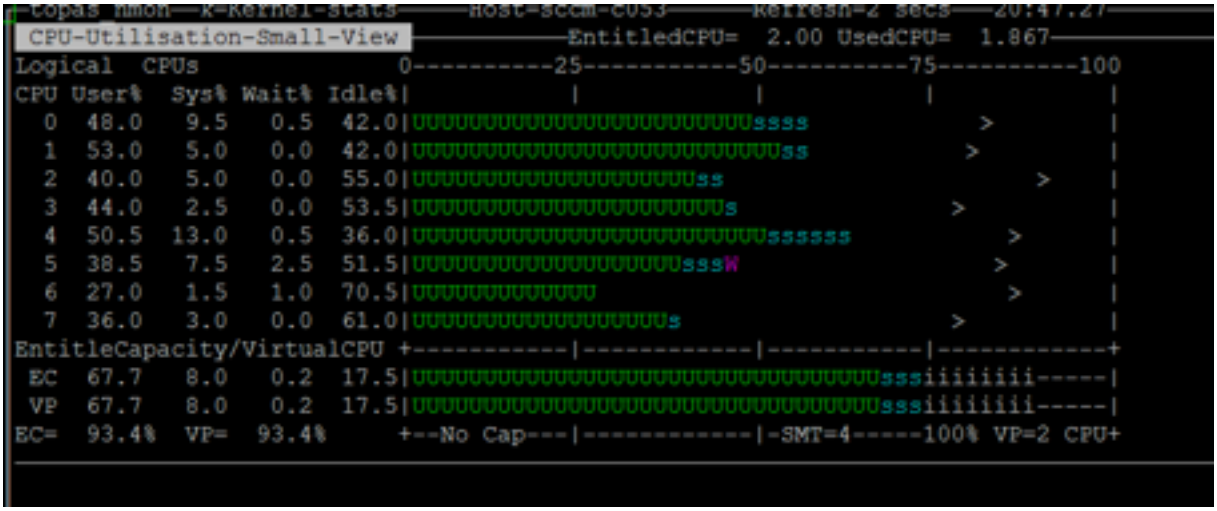

FIGURE 28. NMON ENTRIES FOR CPU UTILIZATION

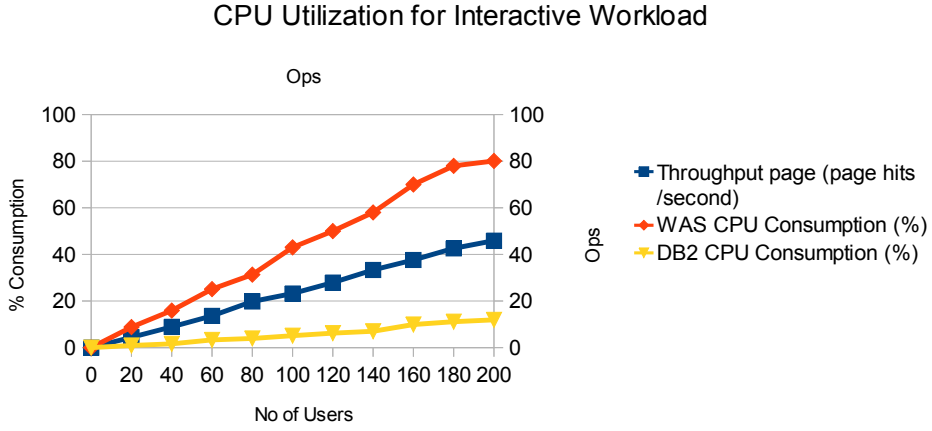

FIGURE 29. CPU UTILIZATION FOR INTERACTIVE WORKLOAD

For batch loading operation also, the CPU consumption is being monitored using nmon. Here RPT is not used for test recording or execution, since it will be of no use. Batch files are loaded from the archive portal and nmon data is recorded at the same time during the operation. How does the batch loading takes place is defined in section 3.1.4. It is seen from the graph that 80 % WAS processes and 10 % db2 processes are consuming CPU at the through put of 48 documents per second.

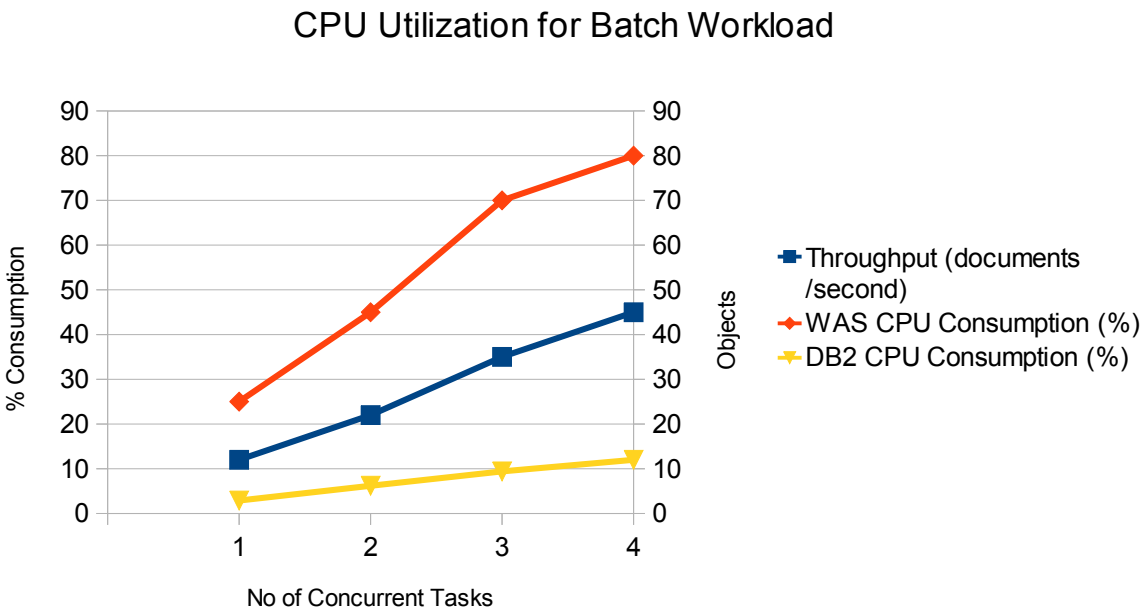

FIGURE 30. CPU UTILIZATION FOR BATCH WORKLOAD

3.1.8 Memory Utilization

AIX systems assign free memory to the file system cache. This is the reason most of the time the memory utilization is more than 80% and sometimes up to 90 % even without any workload because memory is being assigned to file system cache. To find the actual available (free) memory which includes file system cache memory command *svmon –G –O unit=auto* is used which shows the following result.

```
               size       inuse        free         pin     virtual   available    mmode
memory       48.0G       43.3G       4.73G       6.48G       27.1G       19.4G     Ded
pg space      4.00G       69.7M

               work       pers        clnt       other
pin           4.82G          0K       3.54M       1.66G
in use        27.1G        100K       16.1G
```

Here available memory is 19.4 GB (which includes file system cache) and the total memory is 48 GB and 4 GB page space (swap space). This means almost 40 % of memory is available. Virtual memory available is 27.1 GB.Total virtual memory is 52 GB (48+4).The terms virtual memory, page space etc are explained in the chapter 2. The page space is 4 GB. So the total amount of memory space we have is 52 GB (48+4) and the total virtual memory can and is of the size 52 GB.The virtual memory manager (VMM) manages the allocation of RAM and virtual pages. If the workload increases and memory is busy most of the time, more paging activity will occur to free the memory and save unused data into page space. Hence, Memory utilization can be seen by the available memory. If available memory is equal to or less than 10 % (5.2 GB), that is the peak situation and is shown below when we have number of users equal 200.

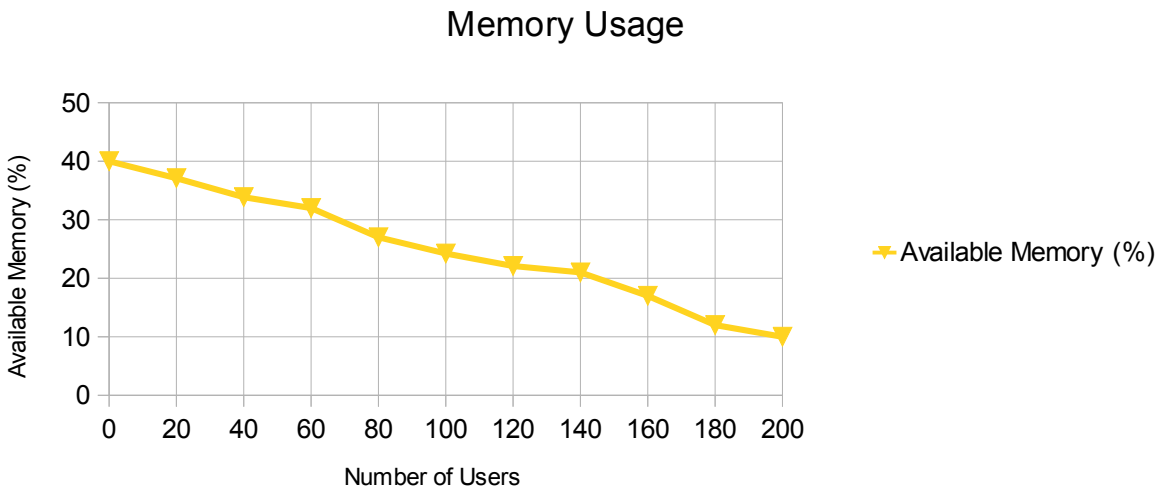

FIGURE 31. MEMORY USAGE CHART

3.1.9 Disk I/O

Disk I/O means the input/output operations on the physical disk. It can also be said as the read and write operations on the disk. If the data is being read from a file on the disk, the processor needs to wait for the file to be read. Same is for writing. Since we have single disk in our test system disk I/O is the major bottleneck of our test system. Disk I/O is done for single page of size 4KB. The important thing for disk I/O is access time. This is the time required for a machine to process a data request from the processor and then retrieve the required data from the storage device. Since disk is not as fast as CPU and memory, caching data in memory is so important for performance as the difference in latency between RAM and the hard disk is so much. For our test system hdisk1 is assigned to the SCCM File system. The IO/sec is shown in figure 31. It is seen from the graph that write operations are more than the read operations.

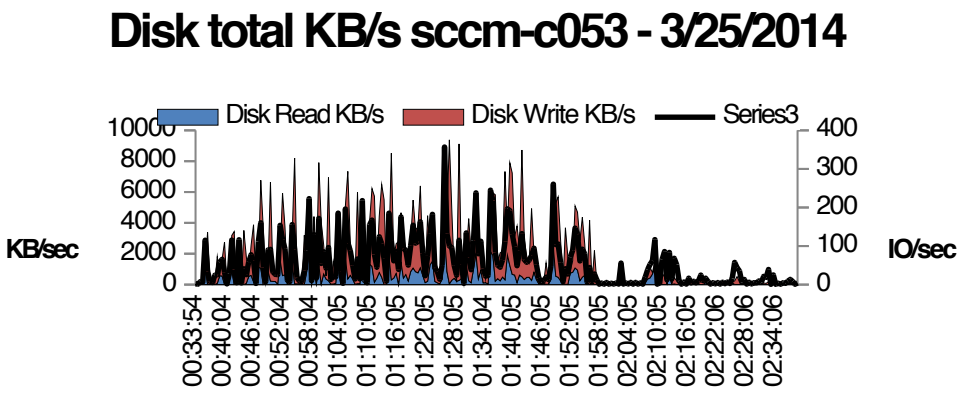

FIGURE 32. DISK I/O GRAPH

Following figure shows the reads,writes and other IO operations per second. As mentioned,disk performance is the major bottle neck with increase in number of users.

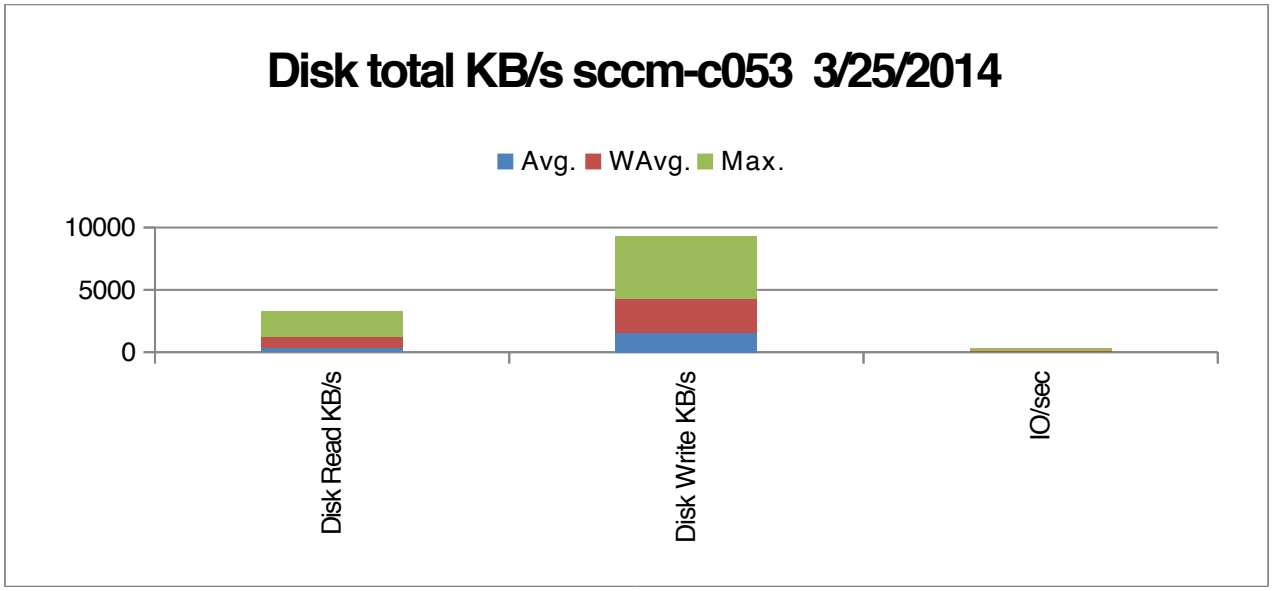

FIGURE 33. DISK READ/WRITE GRAPH

System Summary

Overall test system performance is shown as follows. The Physical CPU utilization is at peak near to 2 CPUs being consumed (92 %) and the disk transfer (I/O) per second is 300.

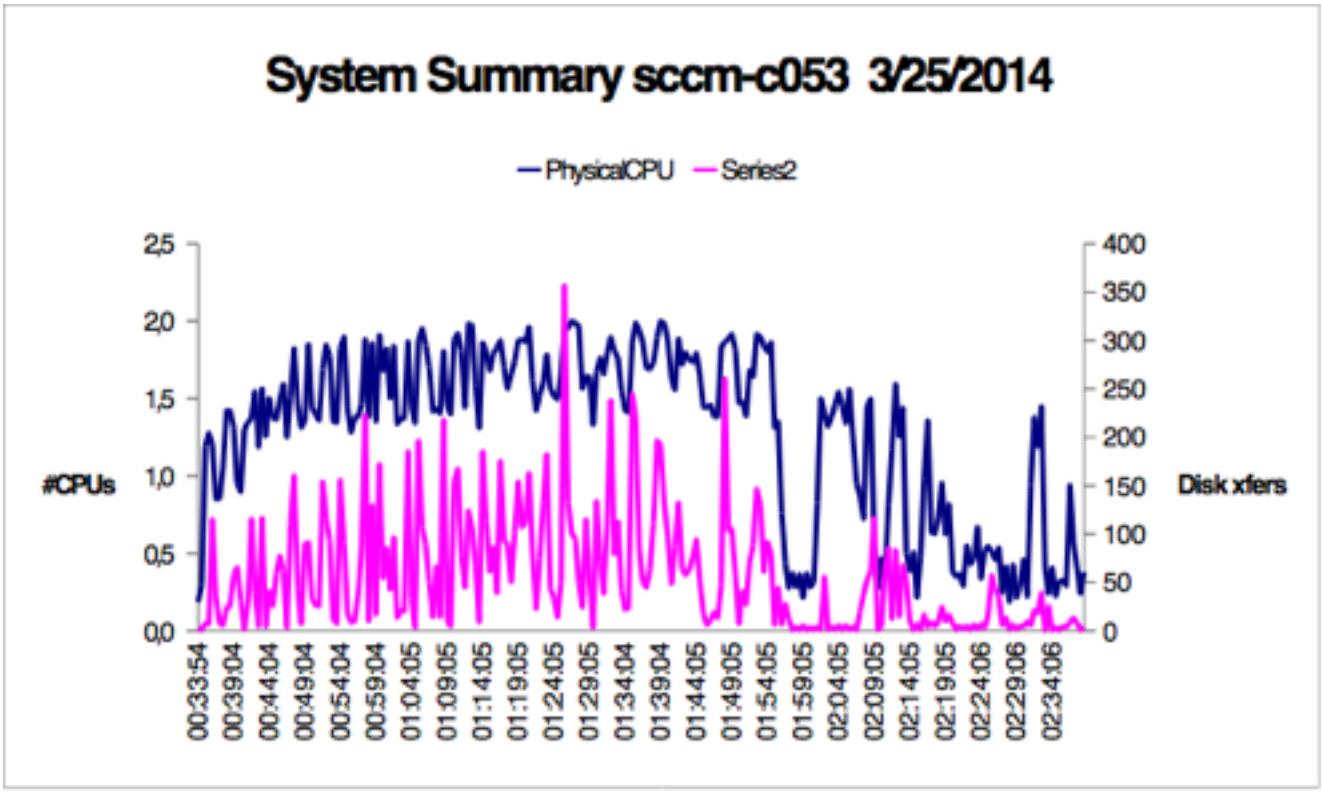

FIGURE 34. OVERALL SYSTEM SUMMARY GRAPH

3.2 Analyze Phase

In the second phase of analyze of MAPE, heuristics (performance base lines results) is applied to determine the optimal set of system resources required for a particular set of workloads and environment.The rules are generated on the basis of the results we got in the monitoring phase through the IBM ILOG* tool which acts as the rules engine. Having the information from heuristics, it is possible to determine which resources are required to meet the SLOs given the current system workload and then to de-provision (de allocate) the required resources. This task is performed by the four phases of the MAPE-loop that are continuously executed. After performing all the tests and resource monitoring in the first phase,now we have response time, throughputs (KPI) of the system for particular workloads, the peak value of them. Also we know the limit of resource consumption i.e. at 200 users we have the peak resource (CPU, memory etc) consumption. On the basis of that, now we can derive certain rules for system resource topology. These rules are similar to the concept from [MWLS]. These are as follows.

Rule 1: When number of virtual users become greater than 200, start adding application servers.

Rule 2: When the number of concurrent virtual users become less than 200, reduce the number of application servers by one if the number of application servers is larger than one.

Rule 3: The database to application server ratio for mixed Interactive workload is 1:7

Rule 4: In a batch document loading scenario, the database to application server CPU ratio is 1:7.

Rule 5: It is better to separate application server nodes for batch loading and mixed interactive workload tasks. This is because of the reason that mixing them and performing them on single application server node degrades the performance.

Rule 6: When splitting a mixed workload application server node, it is beneficial to split the node in such a way that the new nodes process pure workloads (separate interactive and batch loads)and not mixed workloads.

Rule 7: Two application servers for pure batch loading are necessary if more than four tasks are scheduled.

The components of the system which are shared among tenants are memory, db2, CPU, websphere application server (WAS) and disk.These rules are generated on the basis of CPU utilization by WAS and db2 processes with the KPIs measured by the RPT. If we consider the memory utilization and disk I/O for generating the rules,we get the similar rules.

These rules are derived by taking the measurements described in section 3.1. The term virtual users is something which is offered in RPT.RPT simulates the virtual users in such a way that one virtual user is equivalent to the 20 real users One virtual user represents a certain amount of operations or workload. The CPU utilization curve shows that at 200 simulated users, the Websphere Application Server (WAS) on the test reference system reaches saturation (92%) .Response Times suffer (almost 2 sec ,above SLO) and hence SLOs can no longer be met.Analysis of log files shows the number of exception thrown is so much and the system is shutting down. Therefore, it is concluded that when the system hits 200 concurrent users, another application server needs to be deployed for reducing the amount of users per application server.This is illustrated in rule number 1 and 2 .

At 200 concurrent virtual users the system load on the database system is approximately 12%, therefore the database to application server CPU ratio is almost 1:7 with a reserve of 10 % to tolerate workload spikes.This means that the performance of a two CPU database node is enough to support up to seven application server nodes running mixed interactive workload. System utilization above 75% is considered as a high load and the system might not be able to tolerate workload spikes. Interactive mixed workloads response time measurements show that at 150 concurrent users the response time is optimal in relation to one application server CPU reserves are sufficient for the OS to run smoothly.

The batch loader is another component of the SCCM service used to ingest large amount of documents into the repository. CPU utilization during batch loading shows the performance during load/ingest operations. It shows that the application server reaches saturation (CPU>80%) at around four concurrent batch loader tasks.These four tasks generate a throughput of 46 documents per second.Analysis of log files show that exceptions are generated and system has started degenerating.Therefore ,the test system should not be pushed above four concurrent load tasks,as expressed by rule 7.For document ingestion workloads, the database to application server CPU ratio is 1:7.This means that a two CPU database node can serve up to seven application server nodes running exclusively document ingest workloads. This is the rule 4

3.3 Plan Phase

Each system has different resources like CPU, memory etc with different configurations. In this thesis, we use the term performance unit (PU) for the capacity of resources by the system for resource planning. Performance unit is the amount of resource capacity being provided by a system.This is needed because by doing this, we can say that our measured performance data is for our test system with so and so resources (x amount of CPU and y amount of memory etc) or for one performance unit. We can then transfer these results to the other systems and their resources as well and then predict the topology of system resources for them based on the performance results from our reference (test) system.The performance baselines we measured, enable the characterization of the system and its respective components under different workloads. After analyzing these baselines specific provisioning rules are derived which are used to find a satisfactory resource topology in a short period of time. We assume that the capacity of our test system (LPAR) described in monitoring phase constitutes 1 performance unit (PU). Based on the rules generated in section 3.2 and typical SLOs for web applications, the capacity calculation is done. The workloads as explained in the section 3.1 are of interactive and batch load types which are considered and measured for KPIs in this thesis.

Assuming that we are applying dynamic provisioning techniques for long term adaption of the ECM cloud service. This means that we are trying to adjust the system such that it can tolerate the workloads for next four hours.During the last weeks the workloads are monitored regarding all operations of the service. A tenant represent a company or organization with multiple users.Since we have a single tenant (LPAR) system ,we generated the workloads of single basic operations (CRUDS) on the system with increasing the number of users from 1 to 200. For example,how does the system behave (throughput, response time of the system) when a single user adds the document to the repository,or a single user searches for the keyboard in a content based retrieval enabled repository.Similarly,what is the behavior of the system when a single user updates the content of the documents in the repository, or when he deletes the document. All this is measured in the monitor phase of the MAPE loop approach. We then assume that the workload of the service consists only of simple search operations. Also when the number of users are increased for these basic operations up to 200 (which is the limit of the system tolerance and found by the CPU and memory utilization graphs), we see that application servers can be shared by multiple tenants while for the database the multi tenancy is applied on database instance level.This means that multiple tenant use the same database servers but each tenant has its own instance.

We have an SLO from test plan for simple searches stating a maximum response time of three seconds in a archive repository consisting of 1 million documents which is consistent with the size of our test system. In total, the test for 200 concurrent users performing simple search operations on the service is conducted. Referring to the performance baselines we know that we need a system with a total capacity of 1 PU hosting seven application servers (above 200 virtual users) .Furthermore, we know that a system of an overall capacity of 1 PU is needed for the database tier in order to host the database.Using the workload prediction techniques, the capacity needed for the next four hours is calculated .

We take this simple example to illustrate this planning.Let us say we have a system with two tenants on the system with 1PU.For tenant A there are 100 users performing mixed interactive workload operations (as explained in the performance testing section) whereas on tenant B there are 300 users performing the same operation. Therefore , an increased overall capacity of approximately 1.5 PU is needed for the application servers for tenant B whereas for tenant A, following the rule number 2, a decreased capacity of approximately 0.5 PU is sufficient for the application servers. In total,this constitutes 400 users with the same SLO performing simple search operations on the system so that the overall capacity of the application tier needs to be approximately 2 PU.

Also, we know that one database instance running on the system with the capacity of 1 PU can tolerate the mixed interactive workload running on the system. Consequently, for the application tier, systems providing 2 PU s need to be provisioned No changes have to be applied to the database server in order to tolerate the workload of the next four hours.

Using the heuristics (performance baselines) and rules described in analysis phase, we can derive the resources (PUs) needed to tolerate given i.e. actual or predicted workloads of different types.This knowledge is then used for the calculation of a new resource topology, which in turn is the first step for dynamic provisioning of resources in multi tenant cloud environment. This also enables to collaborate exactly those tenants on a physical system that combine best in order to reach the optimal utilization for the system.

3.4 Execute Phase

Execute phase is not described in this thesis. This is already done in [FF]. The results from the plan phase are executed in the execute phase solution of [FF] therefore, these both must be compatible with each other.

3.5 Summary

MAPE loop concept is being used for the dynamic provisioning of resources in this thesis. MAPE i.e. monitor, analysis,plan, execute and again monitoring is done in this concept. The emphasis of this thesis is on first two phases only i.e. monitor and analysis phase. First of all, the performance of the system is monitored for workloads of ECM systems like interactive and batch workloads. Interactive workloads consist of some basic operations like CRUDS (create, retrieve,update,delete, search) and mixture of these. Similarly, batch workload is the loading of bulk data for example in the form of a zip file. These workloads are designed for variable data. Variable data means that data of different sizes (100 KB-10 MB) and different formats (PDF,PPT,XLS etc) is being used for the interactive workloads of the system. These workloads are then monitored for the KPIs like response time and throughput through the tool Rational performance tester (RPT). The workload is also varied through RPT by varying the number of users (or workload as each user is performing same set of workloads) and when the system has this varying workload, the utilization of resources CPU ,memory and disk is also monitored through the NMON. The peak limit is found when the response time is almost 2 seconds and the CPU utilization is 90%. This means at this workload (or number of users) the system performance is worst and with further increase in workload, the system will crash for sure. Therefore, for the given test system whose configuration is 48+4 GB memory,CPU, 200 GB disk, it is found that when the number of users reach 160 performing mixed workload comprising of create, retrieving the document,content based search for the document for different keywords and then modifying the content of the document, then the system responds normally but when the number of users increase from 160 to 180, the system response time increase abruptly and with further increase in users, ultimately the system stops responding and crashes down since there are no resources available to serve the increased workload. By these tests and monitoring in the monitor phase , rules are generated by analyzing the results from the monitor phase. This is done in analyze phase. Rules are generated for provisioning of resources. For a given set of system and workload, it can be analyzed that how much resources like CPU, memory or disk are further needed. In our test system the main bottle neck is the disk since for all the components of the system there is a single disk. Therefore there is always waiting for the disk I/Os ultimately leading to the slow response and bad performance of the system. Therefore it is recommended that for all components of the system like DB2, CSS server etc. there should be separate disks , so that there should be no waiting for disk I/Os and hence the system performance is better.

4. Implementation of Prototype

In the chapter 3, performance baseline testing technique is explained which is used in this thesis for optimizing of system resources.It comprises of the monitor phase of the MAPE loop. In chapter 3 ,analysis, plan and execute phase is explained. In this chapter, a basic sample working model is described. A sample model of MAPE loop is designed and implemented and an overview of interfaces between the different phases of loop are also explained.

Overview

An overview of the structure is shown as below. Basically, the results of the monitor phase which are based on the performance testing and nmon tool are stored in the database and from there, analyze phase can access the results and observations of the monitor phase. This database is different from the database of the system which is described in the chapter 2. An overview of the prototype architecture is shown in figure 34.

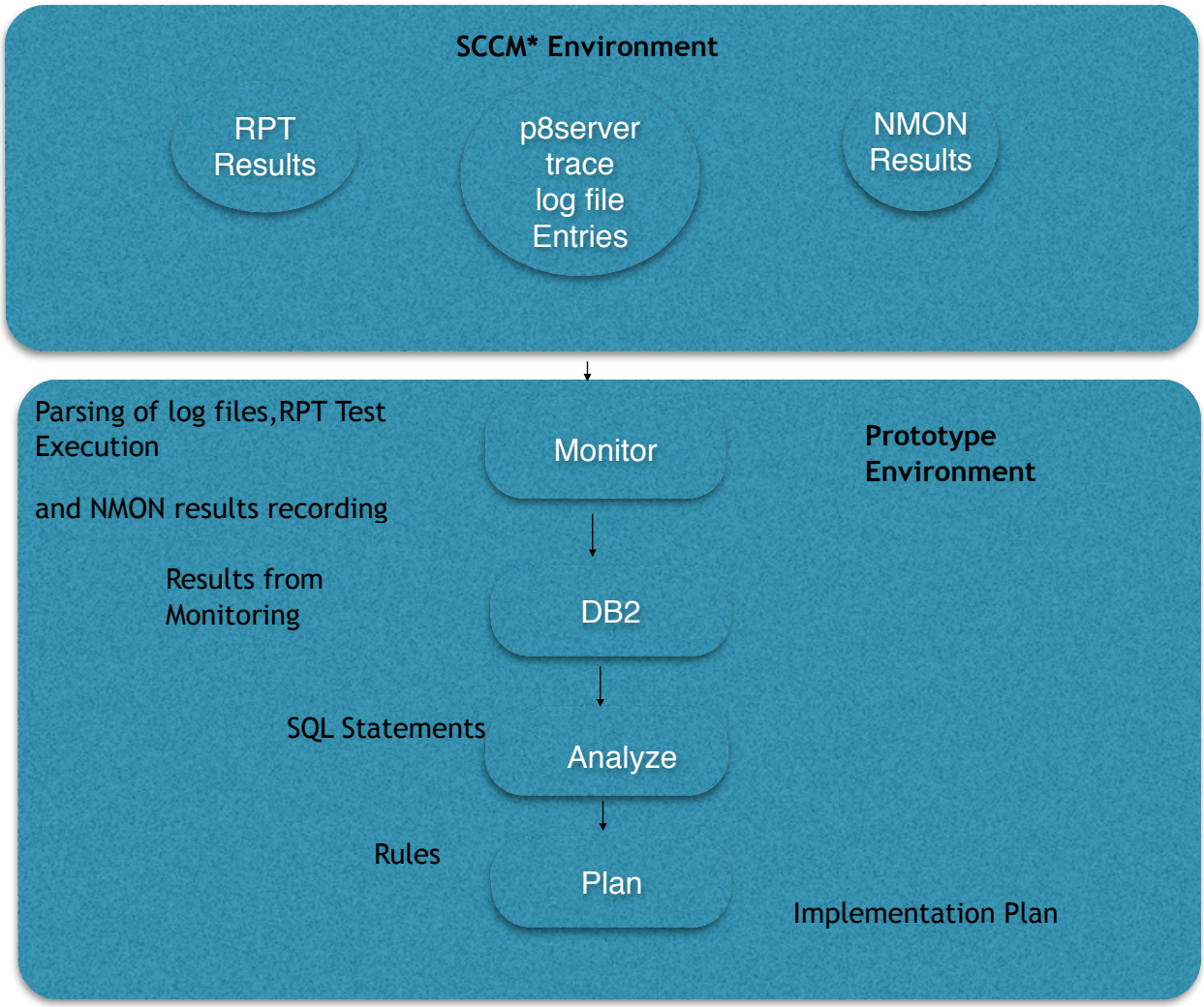

FIGURE 35. PROTOTYPE OVERVIEW

This is divided into two parts. The upper part shows the IBM® SmartCloud ContentManagement* environment whereas the lower part shows the implementation of prototype. As the SCCM application components are running on WebSphere application servers, these are monitored through RPT, parsing of p8 server trace log files for both CEs, use of nmon of system for the monitoring of the resources. DB2* is used as the storage solution, database tier related monitoring information is retrieved through

* Trademarks of IBM in USA and/or other countries

SQL statements in system trace log files and db2top tool for the AIX machine.All the data from all these methods of monitoring is stored in DB2* database which is between monitor and analyze phase. SQL statements are used to access it.

4.1 Monitor Phase Prototype

Monitor phase consists of monitoring the resources of the system through various tools. First of all RPT is used to record the test scripts according to the workload designed typical of banks and insurance companies. Those tests are run and response times and throughput of the system for these workloads are monitored for different number of users and at different point of time. Single operations like create, retrieve,update, delete and search operations are also recorded and tests are run for monitoring the response times and throughputs for these single operations being performed on the server. The tests recorded through RPT are run again and again. When we run the tests through the RPT it play backs the recorded tests to the system and number of users are increased from 1 to 200 to see the performance of system for different workloads for 2 hours. CPU utilization, memory utilization and disk I/Os are monitored during this play back of test through NMON tool and is recorded in excel file. Utilization of these resources at different points of time can be then seen through nice graphs which can be drawn by these excel files very easily. All these results are stored in the DB2* database which is situated between monitor and analyze phase. These results are stored in the single table with following columns:

1. **UID (Integer)** : It contains a unique identification number for each entry. This is the primary key for the table.

2. **Tenant_Info (VarChar)**: This column contains the information about the tenant for which the results are monitored. This specifies the tenant identifier. The datatype is varchar which is unique string for each tenant.

3. **Resource (string)** : This column contains the information about the resource. Typical examples of data in this column are CPU, Memory, Disk, DB2* etc. The datatype is string.

4. **Function (string)** : This column contains the information about the function for which the results are being stored. Typical examples of data for this column are Create, Retrieve,Update ,delete, search etc.

5. **Time_of_Recording(TimeStamp)**: This column contains the timing information for which the resources are being monitored.

6. **Value (VarChar)** : This column contains the value of the monitored workload.

Other than this main table, different tables are created for each resource. For example a table for CPU utilization is created which contains the CPU utilization percentage for every 10 seconds for 2 hours. Similar is the case with memory and db2 utilization. Each of these tables has a unique key and referenced and connected with the main table through foreign keys.

4.2 Analyze Phase Prototype

Stress Tests are also performed on the system through RPT and it is analyzed that at what workload peak the system reaches its maximum throughput and response time and then stops gracefully. Now since there are performance baselines and stress testing results are available in the database db2. These results are accessed through SQL statements and rules are created through the IBM ILOG* tool or rule engine. In the rule engine, the SLOs are inserted, monitoring information is also inserted and by these, the rules are generated which decides about the topology of resources like application servers and db2 etc. As this is shown that the main bottle neck for the test system is disk, it is highly recommended that different disks should be there for each component of the system. For example separate disk should be

there for db2, CSS servers and WAS etc. This will increase the I/O per seconds and hence other resources will not be waiting for the disk I/Os to be completed. The rules created during the analyze phase decides about the topology of resources for the given workload. For example, for a given system with a certain set of resources system reaches at peak utilization of resources with workload of say 200 users with SLO of response time of 1 second for the given set of resources for the particular system. So the rule is generated that resources let say db2* must be added when the number of users increase more than 200. Similarly different rules are created in the basis of monitoring results and these rules then decide the resource topology of the system for any number of users for the given workload. SQL statements are used to access the monitoring data stored in the db2 database during the monitoring phase.

4.3 Plan Phase Prototype

All the resources of the system are available in the resource pool. In the resource pool, the resources are represented in the boolean matrix. The rows and columns of the matrix tells the information about the resources. For example, a resource (e.g. application server) is represented by an index a and if it is deployed, the position (a,a) is set to 1 (otherwise 0). Using this, we can find all the resources which are deployed diagonally. To find the position of deployed resource , other index say b, is used which tells the position of the resource. So, basically the resources are arranged in rows and columns in the matrix. Rows tell about the deployment information of resources whereas column (b here) tells the position of deployed resources. So , for example a resource 'a' is deployed and the parent container of this resource is 'b', then the boolean values at (a,a) and (a,b) must be 1.We have resources like web sphere application servers, CPU,OS,memory etc. These are arranged hierarchically. That means OS will be at the lowest index followed by, web sphere nodes, applications server and application stacks. Each resource has also an index which tells about the provisioning status of the resource as explained before with booleans values 0 and 1 . If the resource is not provisioned i.e. the boolean value of resource at its position is 0 then only, the resource can be provisioned or assigned to a tenant. Many more constraints can be defined for the resources and tenants for the implementation. Here the prototype is defined that does not include all the constraints for resources.

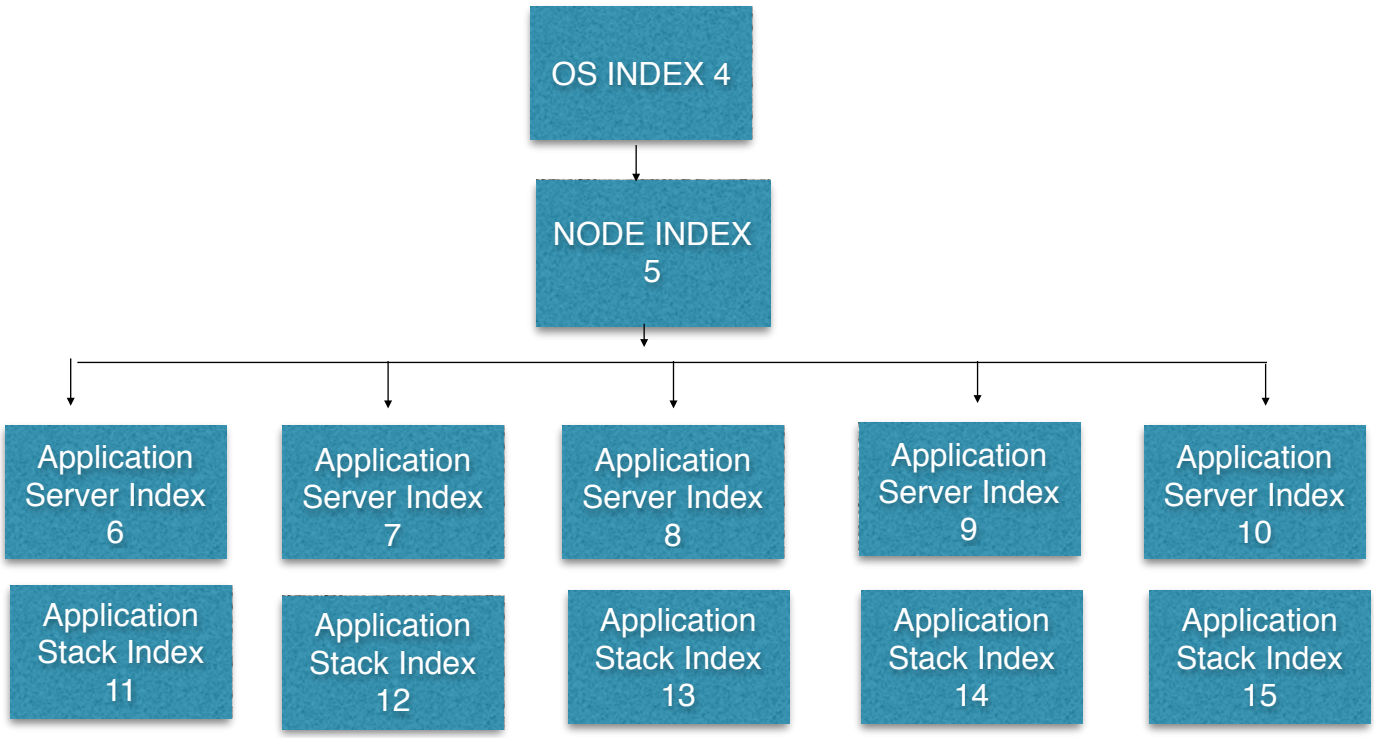

FIGURE 36. ILLUSTRATION OF PLAN PHASE PROTOTYPE

Let us take an example of figure 37 to understand this. Lets say that an OS at index 4 needs two application servers and node at index 5 can be deployed on the OS at index 4. On this node, five application servers are there out of which 3 are hosting application stacks and these can be deployed on

the OS at index 4. Now we know the parent container of these servers (i.e. node at index 5). Now since each resource instance can have only one parent container resource, the matrix defined above, is used to assign the application servers to the OS respecting the hierarchy of resources. As discussed in (Florian thesis) micro flows are executed in the execution phase for provisioning and de-provisioning of resources. Each resource has a state of provisioned or de-provisioned as explained in chapter 1. Two tasks are created as provision and de-provision to provision and de-provision the resources to the tenants. Provisioning can only be done of a resource if it is free i.e. it is an de-provisioned state. Once, it is provisioned then it is in provisioned state and can't be provisioned again but de-provisioned if necessary. That means if a tenant doesn't want it anymore than de-provision task is executed and the resource is de-provisioned from that tenant. For a single resource a single task (provision or de-provision) is executed only once. But since the resources are assigned to different tenants, it is possible that the application server is or application stack is assigned to different tenant and therefore, the provision task must be executed twice. More variables can be defined for the provisioning and de-provisioning of resources to speed the process of provisioning of resources. Also, it is important to note that the resources which are deployed should have a single container resource. Resources are only deployed when their container resource is deployed to the OS. The resource matrix keeps the hierarchical order. Therefore , all the deployed resources must be shown on the diagonal of the matrix or on the left side of the matrix. That means that there should be all zeros on the right side of the matrix.

Summarizing the plan phase of prototype implementation, following constraints are used for dynamic provisioning of resources:

1. Resource matrix cell index is 0 if index is not deployed and 1 if it is deployed.

2. Resources must be ordered hierarchically starting from the OS.

3. If a parent resource at index a has a child resource at index b then (a,b) must be 1 otherwise 0.

4. Resources can be in provisioned or de-provisioned state. Provisioning can only be done if the resource is in de-provisioned state and vice -versa.

5. For each resource type, boundaries must be defined for the resource instances.

6. No capacity should be provided to the tenant from an de-provisioned resource.

7. Capacity provided by a container resource should be sum total of the capacities provided by its child resources.

8. Sharing of resources can only be done by a limited number of tenants. There should not be unlimited number of tenants sharing the same resource.

9. Capacity requirement of resources by tenants must be fulfilled. Total capacity of resources assigned to a particular tenant must be equal to its total requirement.

10. If a resource is never required like OS, web server etc. then it is never being shared among the tenants and therefore its index must always be zero in the resource matrix.

5. Conclusion and Future Work

There are various techniques for dynamic provisioning of resources depending on the proactive or reactive behavior. The main thing is that resources should be provisioned automatically according to the customer's requirements. From the customer's perspective, the service should be available for use without any limit. This is normally achieved by a pay as you go mode. This model has the advantage that customer needs to pay what he is using while provider has the same set of services being sold to different customers hence more revenue with single installation rather than installation of services for each customer . The resources should be available adequately such that there should be no scarcity of resources even at the peak workloads. IBM provides the enterprise content management as the service in the cloud. ITs implementation is done in SmartCloud Content Management*. In this project, a solution need to be developed for dynamic provisioning of resources. This is achieved through MAPE loop. M,A and P are explained in this thesis. Execution is not done as this is done and explained before by [FF]. But the plan phase results of this thesis must be executed and compatible with the solution for execute phase from [FF]. M phase has performance baselines monitoring of requests to the server and monitoring of utilization of resource for the peak workload. Workloads are designed for the system which are typical of banks and insurance companies. Heuristics of workloads are then fed to the analyze phase of the MAPE loop where the results are analyzed and rules are generated from IBM ILOG*. Historical workload traces helps in predicting future workloads. This can be used in the plan phase and the resource topology for the service can be calculated. This will be probably contain less resources but more utilization which would meet the SLOs and make customer happy with decreased costs at the same time. For the system, the concept of performance unit (PU) is used that means the capacity of the test system in this thesis. Depending on the workloads, the resulting system topology can have more or less PU . This will also depend on the time of the day. This is because of the fact that there are certain hours in the day when there is peak workload and there is also certain hours in the day when the system is ideal. This is also the case in various organizations that only some particular workloads are being performed on the system repeatedly which leads to the intensive writing or intensive reading to the disk. Before provisioning the resources all these concepts must also be taken into account. This is done by creating the rules during analyze phase which can then be applied to the plan phase while planning for the resources.

Further work need to be done for the solution. The integration of the results of plan phase into the execution phase need to be done. Also, the communication between user and the system can be done without http server. One possible way for doing this can be directly communicating with the FileNetCE by directly uploading or downloading the data from the system. The heuristics technique described in this thesis is a proactive technique.

Methods from reactive technique should also be combined and a solution can be developed which consist of mixed behavior of the both. The prototype implementation of the results is shown in this thesis. The solution can be tested further and a real implementation can be tested in a real environment with provisioning of resources. If the results of performance of the system doesn't match with the one promised to customer in SLO then, it must raise the alarm immediately before testing further components. This can be done using RPT tool by setting the verification points for each single test. Also, single disk is used for every component of the test system. Therefore, the main bottle neck of the test system is its disk. Different components must be installed on different disks to minimize the I/Os for a single disk and hence a better performance of system will be there since it will not be always waiting to other components to finish. Implementation can be researched more and be done more effectively.

* Trademarks of IBM in USA and/or other countries

6. References

[RJO]:	Ronald P. Doyle, Jeffrey S. Chase, Omer M. Asad, Wei Jin, Amin M. Vahdat, Department of Computer Science , Duke University,*Model-Based Resource Provisioning in a web service utility* URL http://ftp.uwo.ca/courses/CS9843b/papers/model_paper.pdf (cited on page 2,3,6)
[LP]:	Ludmila Cherkasova, Peter Phaal, *Session Based Admission Control: A mechanism for peak load management of websites* URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1009151 (cited on page 669,671,674,678)
[FF]:	Florian Fritz, Master Thesis University of Stuttgart 2011, *Maximization of resource utilization through dynamic provisioning and deprovisioning in cloud*. (Cited on page 8,10,18,19,43,70)
[FRNL]:	Frank Leymann, *Cloud Computing: The next revolution in IT,* URL http://www.ifp.uni-stuttgart.de/publications/phowo09/010Leymann.pdf
[CKF]	Andreas Boerner, *Orchestration and Provisioning of Dynamic System Topologies* (cited on page 2,4,5,8,45,38,49,65)
[IRB]	IBM RedBook, *PowerVM Virtualization Active Memory Sharing*, URL https://www.redbooks.ibm.com/redpapers/pdfs/redp4470.pdf
[VY]	Vivian Yeo, *Controlled Cloud: The way to go for security* (cited on page 1,2)
[MB]	M. Boniface et al., *Platform as a service Architecture for real time quality of service management in clouds cited on page 155-160*
[GK]	G. Kousiouris et al., *Dynamic, behavioral based estimation of resource provisioning based on high level application terms in cloud platforms*
[BDHN]	vom Brocke,Derungs, Rene, Herbst, Navotny, Simons , *The drivers behind enterprise content management: A process oriented perspective* (cited on page 2,3,8)
[JP]	Jessica Piziak, *Overview of IBM Websphere Application server concepts for IBM Lotus connections administrators* (cited on page 4,5,7,8)
[BMBM]	Brian Hall,Mala Anand,Bill Buros,Miso Cilimdzic, *Maximizing the Value of an IBM Power7 and IBM Power7+ Environment through tuning and optimization*
[JE]	Jessen Elke, *Origin of the virtual memory concept* (cited on page 71,72)
[BAC]	Bensoussan, Andre,Clingen, *The Multics Virtual Memory :Concepts and Design* (cited on page 308-318)
[RW]	The White Paper , Rick Wilson, *Implementing Information Governance* (cited on page 2,3,4)
[LNK]	L. Ramachandran, N. Narendra, K. Ponnalagu, *Dynamic Provisioning in multi-tenant service clouds*
[SRMS]	Sagar Girase, Rahul Samant, Mayank Sohani, Suraj Patil, *Resource Provisioning in Cloud computing environment* (cited on page 1,2,4,8)

[TKB]	Tsung-Ju Lee, Kuo Chan Huang,Bo-Jyun Shen,Hsi Ya Chang,Yuan Hsin Tung,Pin Zei Shih , *Resource allocation and dynamic provisioning for service oriented applications in cloud environment,* (cited on page 2,106,387)
[TLH]	Hung Yi Teng, Wei Ru Lee, Ren Hung Hwang, *Optimization of cloud resource subscription policy* (cited on page 449-455)
[JMF]	J. Oriol Fito, Jordi Guitart, Mario Macias, Ferran Julia, *Business driven IT management for cloud computing providers* (cited on page 193-200)
[HRCK]	Kai Hwang, Chunming Rong, Erdal Cayirci, MAciej Koczur, *A multi-criteria design scheme for service federating inter cloud applications* (cited on page 129,131,133,134)
[LWTE]	Wu Li, Wenjun Wu, Wei Tek Tsai, Babak Esmaeili, *Model driven tenant development for PaaS based SaaS* (cited on page 821,823,824,826)
[SAJ]	Richard O. Sinnott, Jemal Abawajy, Bahman Javadi, *Hybrid Cloud resource provisioning policy in the presence of resource failures* (cited on page 10,12,13,14,17)
[MALS]	Dominik Muhler, Vasilios Andrikopoulos, Frank Leymann, Steve Strauch, *ESBMT: Enabling multi-tenancy in enterprise service buses*
[CM]	S.G. Chavan, R.P. Mogre, *Resource Provisioning for Elastic Applications on Hybrid Cloud Environment (*cited on page 620,621)
[BYV]	R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, *Cloud Computing and emerging IT platforms:vision hype and reality for delivering IT services as the 5th utility*
[VRC]	L. Vaquero, L. Rodero-Marino, J. Caceres, M. Lindner, *A break in the clouds : towards a cloud definition*, (cited on page 137,140,141)
[BZT]	J. Bi, Z.Zhu, R. Tian, Q. Wang, *Dynamic Provisioning modeling for virtualized multi tier applications in cloud data center* (cited on page 2,3)
[CMKS]	T.C. Chieu, A. Mohindra, A.A. Karve and A. Segal, Dynamic scaling of web applications in a virtualized cloud computing environment (cited on page 7,8,10)
[TDM]	Ibrahim Takouna, Wesam Dawoud, Christoph Meinel, *Dynamic Scalability and contention prediction in public infrastructure using internet application profiling* (cited on pages 208,209,211,216)
[LW1]	Yen-Ting Lee, Ching- She Wu, *Automatic SaaS test cases generation based on SOA in the cloud service* (cited on page 349,350)
[FME]	Eugen Feller, Christine Morin, Armel Esnault, *A case for fully decentralized dynamic VM consolidation in clouds* (cited on page 26,27)
[KL]	Kai Liu, Diplomarbeit, University of Stuttgart , Development of TOSCA service templates for provisioning portable IT services
[MWLS]	Cataldo Mega, Tim Waizenegger, David Lebutsch, Stefan Schleipen, J.M. Barney ,*Dynamic cloud service topology adaption for minimizing resources while meeting performance goals* (cited on page 1,3,4,5,6,7,8)
[KK]	Kathlen Krebs, *Content Management as a service* (cited on page 2,3)

ZCS	Qi Zhang,Ludmila Cherkasova, Evgenia Smirni, *A regression based model analytic model for dynamic resource provisioning of multi-tier applications* cited on page 1,2,3

Declaration

I hereby declare that the work presented in this thesis is entirely my own.

I did not use any other sources and references that the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations.

Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.

The electronic copy is consistent with all submitted copies.

Gaurav Chawla

Stuttgart, 2.6.2014