AN ADAPTIVE QoS-BASED DYNAMIC RESOURCE MANAGEMENT IN CLOUD COMPUTING SYSTEMS

Thesis

Submitted to

Babasaheb Bhimrao Ambedkar University

(A Central University)

Lucknow

For the Award of the Degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE

By

SWATI SAXENA

Under the Supervision of

DR. NARANDER KUMAR

DEPARTMENT OF COMPUTER SCIENCE
BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY
(A CENTRAL UNIVERSITY)
LUCKNOW-226 025 (U.P.) INDIA

2018

Dedicated to my Parents…

CANDIDATE'S DECLARATION

I, Swati Saxena, solemnly declare that the research work embodied in this thesis entitled "**AN ADAPTIVE QOS-BASED DYNAMIC RESOURCE MANAGEMENT IN CLOUD COMPUTING SYSTEMS**" carried out by me under the guidance and supervision of **Dr. Narander Kumar, Assistant Professor, Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, India** is an original work and does not contain part of any work submitted for the award of any degree either in this University or any other University around the globe. It is further undertaken that the thesis is essentially free from all kinds of plagiarism.

Date: **(Swati Saxena)**

Place: Lucknow Research Scholar

Department of Computer Science

Babasaheb Bhimrao Ambedkar University, Lucknow

CERTIFICATE

This is to certify that the thesis titled **“An Adaptive QoS-based Dynamic Resource Management in Cloud Computing Systems”** submitted by Ms./Mr. **Swati Saxena** is an original research work and has not been previously submitted in part or full for the award of any other degree or diploma to this or any other university.

The thesis submitted to Babasaheb Bhimrao Ambedkar University Lucknow satisfies all the requirements as stipulated in the *Doctor of Philosophy (Ph.D.) regulations -1999 as amended in 2008/2010/2013* and it is fit for submission and evaluation for the award of the degree of Doctor of Philosophy of the University.

Date: Supervisor

Head of the Department

ACKNOWLEDGEMENTS

कायेन वाचा मनसेन्द्रियैर्वा, बुद्ध्यात्मना वा प्रकृतेः स्वभावात् ।

करोमि यद्यत्सकलं परस्मै, गुरुवरायेति समर्पयामि ॥

Whatever I do with the Body, Speech, Mind or the Sense Organs, either by discrimination of the Intellect, or by the deeper feelings of the Heart, or by the existing Tendencies of the Mind, I Do them All without Ownership, And I Surrender them at the feet of Sri Narayana – Lord Vishnu Shloka.

Doctor of Philosophy is a rewarding journey, which would not be possible without the support of many people. As this journey comes to its end, I would like to take this opportunity to thank these amazing people who inspired me during the ups and downs of this pleasant experience.

First and foremost, I wish to express my sincere gratitude to my supervisor, ***Dr. Narander Kumar, Assistant Professor, Department of Computer Science, Babasaheb Bhimrao Ambedkar University,*** *for his guidance, tremendous support and encouragement in completing this research and providing me extensive support for various research activities. He has been actively interested in my work and has always been available to advise me. His invaluable insight, into technical and professional matters, has helped me immensely throughout my research work.*

My deep and sincere thanks are due to ***Professor R. A. Khan, Dean, School of Information Science and Technology and Head, Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India.*** *His valuable suggestions made the contributions of the thesis more valuable.*

I will forever be thankful to ***Professor Vipin Saxena and Prof. S. K. Dwivedi, Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India*** *for sparing their valuable time in giving insightful suggestions in publishing quality research papers.*

My deep regards and thanks to ***Dr. Deepa Raj, Dr. Manoj Kumar and Dr. Shalini Chandra, Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India*** *for extending all possible help and support whenever required.*

I am grateful to ***Prof. R.C Sobti, Vice Chancellor, Babasaheb Bhimrao Ambedkar University, Lucknow, India*** *for providing the excellent computation facilities in the University campus. Special thanks are extended to* ***University Grant Commission, India*** *for providing financial assistance to the Central University for doing research work.*

I would also like to thank the non-teaching staff members of the ***Department of Computer Science, Babasaheb Bhimrao Ambedkar University,*** *for keeping the lab resources up-to-date and doing their work in a systematic way.*

Special thanks are due to my family for supporting, helping and providing me each and every facility during the period of research. My parents, ***Shri A. K. Saxena and Smt. Rajni Saxena****, have been the source of my inner strength and self-confidence. They are the reason I could see the light of this day. I would like to express my deep hearted thanks to my husband,* ***Rajat Nigam****, for supporting me in fulfilling my professional dreams. Last but not the least, my love and thanks to my kids,* ***Ivaan*** *and* ***Hrivaan,*** *for giving me reasons to smile everyday amidst various struggles and responsibilities.*

Swati Saxena

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PUBLICATIONS

1. Narander Kumar, **Swati Saxena**, "Enhancing Performance of Data-centers using Location-aware Live VM Migration", **Springer** series on **Lecture Notes in Networks and Systems (LNNS),** vol. 9, pp. 119-129, 2018. (ISSN: 2367 – 3370)

2. Narander Kumar, **Swati Saxena**, "Energy-efficient Load-Aware VM Placement using Multi-Metrics Analysis", **Indian Journal of Science and Technology (IJST)**, vol. 10 (32), pp. 1-8, 2017. (ISSN: 0974-6846)

3. Narander Kumar, **Swati Saxena**, "An Energy-efficient Topological Framework for Data center Networking", **BRIS Journal of Advances in Science & Technology (BRISJAST),** vol. 4 (2). pp. 80-87, 2017. (ISSN: 1444-8939)

4. Narander Kumar, **Swati Saxena**, "Dynamic Resource Provisioning in Datacenters using Profitability-aware VM Placement", **BRIS Journal of Advances in Science & Technology (BRISJAST)**, vol. 4 (1), pp. 22-33, 2017. (ISSN: 1444-8939)

5. Narander Kumar, **Swati Saxena**, "Energy-Efficient Multi-Criteria based Admission Control in Cloud Data-Centers", **International Journal of Innovations & Advancement in Computer Science (IJIACS)**, vol. 5 (6), pp. 110-115, 2016. (ISSN: 2347 – 8616)

6. Narander Kumar, **Swati Saxena**, "A Semantic Framework to Standardize Cloud Adoption Process", **Springer** series on **Advances in Intelligent Systems and Computing (AISC)**, vol. 434, pp. 179-187, 2016. (ISSN: 2194 – 5357)

7. Narander Kumar, **Swati Saxena**, "A Preference-based Resource Allocation in Cloud Computing Systems", **Elsevier Procedia Computer Science Journal**, vol. 57, pp. 104 – 111, 2015. (ISSN 1877 – 0509)

8. Narander Kumar, **Swati Saxena**, "Migration Performance of Cloud Applications-A Quantitative Analysis", **Elsevier Procedia Computer Science Journal,** vol. 45, pp. 823 – 831, 2015. (ISSN 1877 – 0509)

9. Narander Kumar, **Swati Saxena**, "Token-based Predictive Scheduling of Tasks in Cloud Data-enters", **Research Journal of Recent Sciences (RJRS)**, vol. 4, pp 29-33, 2015. (ISSN 2277-2502)

10. Narander Kumar, **Swati Saxena**, "An Efficient Live VM Migration Technique in Clustered Datacenters", **Research Journal of Recent Sciences (RJRS)**, vol. 3, pp. 13-20, 2014. (ISSN 2277-2502)

(Swati Saxena)
Research Scholar

Summary

SUMMARY

Cloud computing is a utility-based computing paradigm. It provides a seamless acquisition of computing, storage and network resources to users on a payment basis, in a similar fashion like that of water, gas and electricity. Cloud computing is highly embraced by individuals and business organizations due to the advantages it offers, few of them be like-

- Zero or no opening investment for accessing computing resources like computing, memory and/or network as these are measured in metrical units and provided to cloud users on a demand basis.
- Cloud resources or services are available round the clock with negligible down-time and can be accessed from any location.
- Cloud users' privacy and security features are well-maintained. Moreover, cloud functioning is transparent to its users.
- Cloud users are charged according to their utilization of cloud resources, thereby, extending the advantage of scalability.
- Cloud services are made available on a payment basis and are categorized as software, platform or infrastructure services.

Cloud computing still lies in its infancy stage. Its potential is yet to be unleashed in new business capabilities and advancements. However, like any other advancing technology, cloud also suffers from certain issues and trials which must addressed in time to make cloud adoption a seamless process for all.

The key mechanism behind cloud computing is virtualization which creates an abstraction of actual cloud resources, so that more and more number of users can utilize the potential of this technology. These abstracted resources are enveloped in virtual machines with same functions and interfaces as that of real/actual resources. An important aspect of virtualization is that it is transparent to the cloud user. These virtual resources are offered in different sizes, performance and cost. The entire life cycle of a virtual machine is controlled a virtual machine monitor (VMM) or a hypervisor software. A physical machine or server contains numerous virtual machines and is known as a host. Virtual machines residing on a host are therefore termed as guests. Hypervisor acts as a bridge between the actual resources of a host and its multiple guests by giving a virtual operating environment to execute the guest VM tasks.

Resource management in a virtualized environment like that of cloud systems is a continuous and testing task which reflects the constant changing demand-supply graph of the virtual resources. The existing techniques for resource management are doing their jobs as per their intended purposes, but, the wide-spread adoption of cloud and its dynamic user base requires certain stringent measures towards managing cloud's assets.

The chapter-wise summary of the research is given below in brief:

CHAPTER I

INTRODUCTION

This chapter provides an introduction to cloud computing technology and its importance as a new paradigm of computing. The evolution of cloud and its classified services are described in detail. It also highlights the important attributes of cloud computing, its strengths and its weak points. A number of widely accepted and most cited

definitions of cloud computing are also given. Cloud computing is a special type of distributed computing in which any type of resource, physical or virtual, can be made available to the users, worldwide, by means of powerful technology called virtualization of resources. Virtualization is given due importance with discussion on its types and utilities. The issue of resource management techniques in cloud computing environment is discussed with an outline on the primary objectives of the research work.

CHAPTER II

REVIEW OF LITERARURE

This chapter gives details of the existing literature on cloud computing technology, available resource management techniques in cloud environment, for example job scheduling, resource auction/allocation, admission control and virtual machines consolidation issues. The rapid adoption of cloud computing by the society emphasis the need efficient and feasible resource management techniques for managing the cloud resources. An extensive study of related research papers on resource management techniques suggests that the objectives and implementation of resource management in cloud networks are very different from classical networks. For example, in classical networks resources are only physical while in cloud, resources are physical as well as virtual. Therefore, a different approach is required for cloud computing to manage resources effectively. Several reputed journals, e-books, Wikipedia, etc. are consulted for understanding the new research problems and more than 150 references are given in the thesis.

CHAPTER III

LOCATION-AWARE VM MIGRATION IN CLOUD DATACENTRES

In this chapter, a location-restricted migration mechanism of an overloaded virtual machine is presented in a clustered data centre. Server clusters are formed based on the application type hosted by VMs, and migration of VM is restricted within its home cluster. This helps in reducing the migration volume along with the total migration time to a considerable value. Selection of destination host for migration is based on the criteria of distance, which further speeds up the task at hand and also reduces energy consumption. A detailed mathematical analysis of existing live migration process is presented to point out the factors which play a crucial role in the migration performance. Sequence and class diagrams supplied help in better understanding of the proposed migration technique. The proposed migration technique is simulated on the CloudSim-3.0.3 simulator.

The content of this chapter is published in-

1. Springer series on Lecture Notes in Networks and Systems, vol. 9, pp. 119-129, 2018, ISSN 2367 – 3370.
2. Elsevier Procedia Computer Science Journal, vol. 45, pp. 823 – 831, 2015, ISSN 1877 – 0509.
3. Research Journal of Recent Sciences, vol. 3, pp. 13-20, 2014, ISSN 2277-2502.

CHAPTER IV

LOAD-AWARE VM PLACEMENT IN CLOUD DATACENTERS

In this chapter, an efficient VM placement and migration technique is proposed based on the three-tier architecture and considers load as a prime objective. We have used a multi-metrics analysis to distribute VMs evenly and to maintain a stable equilibrium

inside a data centre. We have applied Analytical Hierarchy Process (AHP) for efficient virtual machine placement using four post placement metrics which defines some of the key Service Level Agreement (SLA) parameters. The metrics considered for placement and migration are all load-centric and promise fewer migrations and SLA violations. The high points of the proposed VM placement technique are increased profits to the cloud service provider, fair and service availability to cloud users by placing a virtual machine to the best available physical machine in a dynamic fashion using the concept of clusters. It also avoids unnecessary migrations by balancing load of a cluster. Simulation results show a remarkable reduction in migrations which improves energy conservation inside the data centre. Application of AHP in balancing a data centre's load is still unexplored. The presented placement technique selects the best candidate machine for placement, hence upgrades the performance.

The content of this chapter is published in-

1. Indian Journal of Science & Technology, vol. 10(32), pp. 1-8, 2017, ISSN: 0974 - 6846.
2. BRIS Journal of Advances in Science and Technology, vol. 4(1), pp. 22-33, 2017, ISSN: 1444 - 8939.

CHAPTER V

PREFERENCE-BASED RESOURCE ALLOCATION IN CLOUD DATACENTERS

Efficient resource allocation is a major concern in utility-based systems such as cloud computing. Cloud users approach a cloud service provider to execute their tasks which require cloud resources in various measures. In return, users pay for the resources utilized by them. To cater multiple users in the same instant with varying resource

requirements, a cloud provider applies certain resource allocation techniques which must not only bridge the gap between the demand and supply but also must provide certain benefits to both the service provider and the service user. This chapter presents a resource allocation mechanism where first a market-driven auction process takes place to ensure truthfulness and profit to the service provider followed bya preferential payment process. Here, the winner of the auction is supposed to make the payment for his resource requirements by an amount which is far less than his actual bid value. The suggested resource allocation scheme is compared with the off-line VCG auction technique to register a finer performance outcome w. r. t. optimal resource distribution, fair allocation and comparatively better revenues to the service provider.

The content of this chapter is published in-

Elsevier Procedia Computer Science Journal, vol. 57, pp. 104-111, 2015, ISSN: 1877 - 0509.

CHAPTER VI

TOKEN-BASED PREDICTIVE JOB SCHEDULING IN CLOUD DATA CENTERS

Task scheduling in a distributed environment, like cloud computing, is an important function which influences the overall performance. Considerations in chapter six are a single data center where a number of users are competing for limited resources. An ideal solution must service all these concurrent jobs while maintaining quality of service parameters and ensuring optimum usage of cloud's resources. Unfortunately, dynamic user requirements and limited availability of resources often makes it difficult to satisfy every demand without compromising the quality of performance. Here, the

proposed scheduling technique ensures low turnaround time and waiting time by balancing the demand curve with allocation frequency.

The content of this chapter is published in-

Research Journal of Recent Sciences, vol. 4, pp. 29-33, 2015, ISSN: 2277-2502.

CHAPTER VII

ENERGY-EFFICIENT MULTI-CRITERIA BASED ADMISSION CONTROL IN CLOUD DATACENTERS

Job scheduling requires an intelligent selection of incoming jobs or requests so that their execution improves the efficiency of data center as a whole. Cloud data centers utilize certain admission control procedures to accept/reject a service request. Major admission control mechanisms, practiced today, consider single parameter as objective. However, there are multiple dimensions to consider while accepting or rejecting a request. This chapter presents a multi key-based admission control mechanism which ranks the incoming service requests based on various performance keys and accordingly admits the most suitable request. Simulation experiments also confirm the validity and usefulness of the scheme with respect to energy conservation in a data centre.

The content of this chapter is published in-

International Journal of Innovations & Advancement in Computer Science, vol. 5(6), pp. 110-115, 2016, ISSN: 2347 – 8616.

CHAPTER VIII

ENERGY EFFICIENCY AND OTHER SLAs

This chapter presents two aspects of modern cloud data centers- energy efficiency and cloud adoption. To make data center energy efficient, steps must be taken to limit the communication workload inside a data center and to switch off network elements when not in use. In this chapter, we put forward two models, one an energy consumption model to outline the factors which are responsible for excessive energy consumption in a data center and second an energy conservation model which tunes these responsible factors so that energy consumed is less. Proposed conservation models are applied to hierarchical and recursive architectures to verify their practicality. Simulation results mark a notable advancement in energy conservation. Also, these models are applicable to many DCN architectures.

The focus is to make cloud adoption a clean and transparent process by clearly outlining the individual rights and responsibilities of both the service provider and the cloud adopting business. In this direction, a semantic framework is introduced in this chapter which address all the risks and challenges mentioned above and provides its best possible solution. This framework establishes the trust between the two involved parties and is crucial for the success of cloud computing technology.

The content of this chapter is published in-

1. BRIS Journal of Advances in Science & Technology**,** vol. 4 (2). pp. 80-87, 2017, ISSN. 1444-8939.
2. Springer series on Advances in Intelligent Systems and Computing, vol. 434, pp. 179-187, 2016, ISSN 2194 – 5357.

CHAPTER IX

CONCLUSIONS AND FUTURE PESRPECTIVES

This chapter is devoted to the conclusions of the given research work and the future work in the area of cloud resource management. Cloud computing systems have a long way to go. Future computing world will strive on virtual resources promising seamless and continuous services. Hence, the issue of cloud resources management must be addressed in a timely and priority fashion. In the present research work, several issues of cloud resource management have been identified and independent solutions to each issue have been proposed. These are -

- VM Migration
- VM Placement
- Resource Allocation
- Job Scheduling
- Admission Control
- Energy efficiency and other SLAs

The solutions provided for each sub-problem are feasible, scalable and dynamic in nature and efficiently manage cloud resources as validated by the simulation results. As future perspectives, this work can be extended for newer DCN architectures and interoperable clouds which enable a cloud user with the flexibility of shifting his/her acquired resources between data centres and no disruption in serviceability.

CHAPTER I

Introduction

CHAPTER I

INTRODUCTION

1.1 CLOUD COMPUTING AT A GLANCE

Computing, in layman terms, is the use of computers to perform a task for its intended purpose. In the computing world, Cloud stands for the internet. Hence, one can conclude that, cloud computing is the use of server machines, connected through a network on the internet, to complete a computation task. However, technically cloud computing is way beyond this idea; it provides a way to access various computing capabilities on a demand basis without investing in hardware, software or infrastructure. In fact, cloud users are just required to pay for cloud services according to their usages.

Present scenario of computing involves services that are delivered to users irrespective of their location or delivery pattern. Utility computing, as the name suggests, is a commodity-based computing model where services are treated very much like the needed commodities as water, electricity, gas, etc. Cloud computing is one such utility-based computing model where individuals and/or businesses can access services as per requirements from any location without the knowledge of the underlying hosted infrastructure. Further cloud customers are charged for using these services for the time and quantity demanded. In return, users are free from the headache of installing and maintaining crucial services, be it hardware or software. Cloud computing delivers services which are hosted on next-generation data centers. These data centers host thousands of servers and are running on virtualized computing and storage technologies. Virtualization enables a data

centre to service multiple users by accessing applications and data from any location in the world on demand. In return, users are promised service availability at any instant. To support universal access, effective discovery and composability, cloud services need to be highly dependable, expandable and autonomic.

1.2 CHARACTERISTICS OF CLOUD COMPUTING

Cloud computing has emerged from cluster and grid computing; hence it possesses key attributes of both the computing models. However, cloud has its own distinguishing characteristics which enable it to service multiple users in a transparent manner. The key characteristics of cloud computing are highlighted below-

i. Size- a typical cloud data center consists of 100 to 1000s of servers.

ii. OS- Each node runs a hypervisor which supports multiple Operating Systems.

iii. Service Negotiation is based on Service Level Agreements.

iv. Resources Management and distribution can be centralized or de-centralized.

v. Capacity allocation must be based on resource's demand.

vi. Pricing is based on demand of the services and is usually discounted for large group of customers.

vii. Failure Management is supported by content replication and VM migration.

viii. Privacy and Security is guaranteed with support to per-file access control list.

ix. Network speed- Cloud data centers are running on high bandwidth and low latency interconnecting network.

x. Internetworking is practiced by using third party solution providers for connecting multiple clouds.

1.3 CLOUD RESOURCES

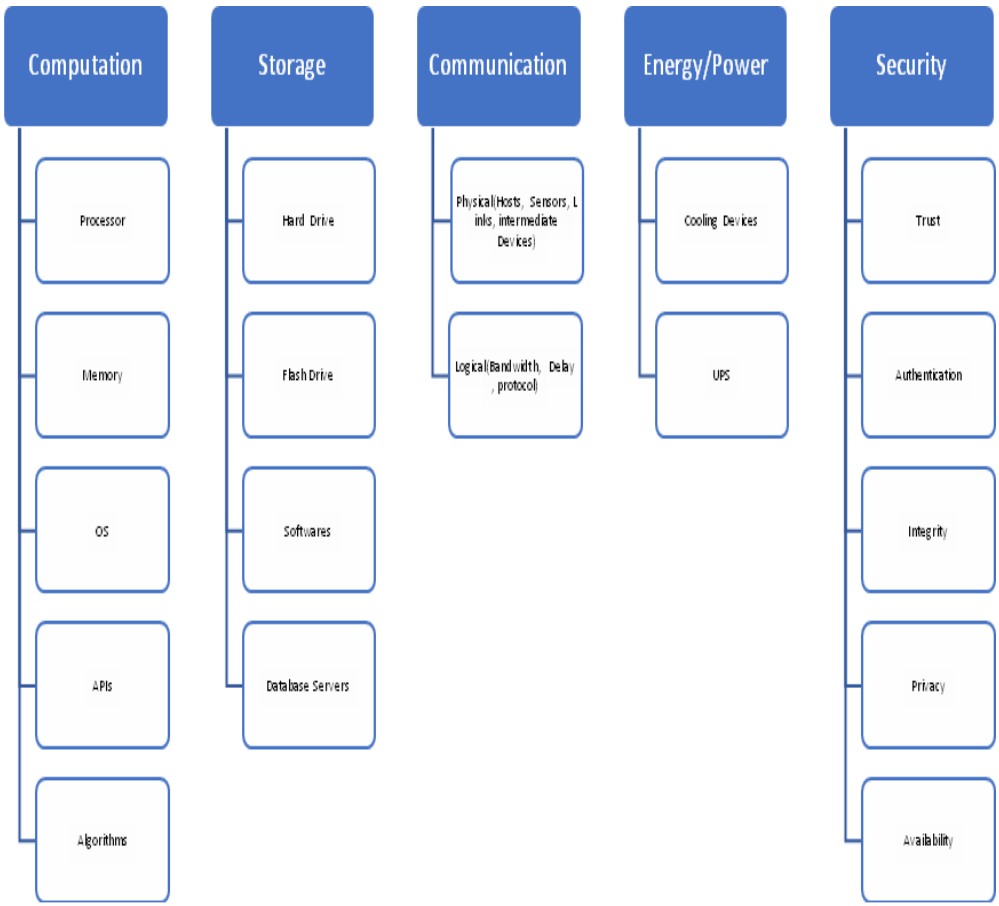

Figure 1.1 Types of Resources in a Cloud Computing System

Being a utility-based computing model, resources in cloud computing are treated as utilities. A resource in cloud computing is any entity, physical or logical, that can be used or consumed by a cloud user. A general idea of cloud computing recources is given below with its hierarchical outline in figure 1.1.

i. Computation Resources are high-speed computational utilities responsible for providing Computation as a Service. It includes memory size, processing speed, efficient algorithms, operating systems, etc.

ii. Storage Resource includes hundreds and thousands of database servers, flash drives and hard drives, usually located at a remote location. Storage as a Service feature of Cloud Computing is provided by storage utility.

iii. Communication Resource consists of hosts, sensors, physical and virtual communication links, bandwidth, protocols, intermediate devices etc. This utility provides Network as a Service feature through high internet speed.

iv. Power/Energy Resource is one of the most sought-after utility in cloud computing. It consists of UPSs and multiple cooling devices used inside a data centre. Since power consumption is extremely high in cloud systems, various energy efficient techniques are proposed and applied to bring the carbon footprints down.

v. Security is a high priority topic of concern in cloud data centres. To implement reliability, trust and safety in cloud data centres, Security as a Service is provided by this utility.

1.4 ADVANCEMENT OF CLOUD COMPUTING

1950s- Distribution of a single computing resource among multiple users on a sharing basis started way back during 1950s when mainframes were introduced. Mainframes were not very economical for one-to-one use, hence, several users were given 'dumb terminals' to access its resources.

1970s- This decade was marked by the introduction of virtualization technology, which took shared access of resources to an entirely new height. Virtualization enabled multiple computing environments to play under one physical machine in an isolated manner.

1990s- It started initially with telecommunication companies that allowed customers to use a single common infrastructure on a shared basis. It offered same service quality at a much-reduced cost. This led to various computing models as mentioned below-

a) **Grid Computing** where large computational problems were handled using parallel computing.
b) **Utility Computing** which commoditised computing resources and offered them as a metered utility.
c) **SaaS** which offered access to computer applications and software on network-based subscriptions.
d) **Cloud Computing** which offers computational resources on a utility-payment basis, servicing multiple customers with dynamic workloads.

Present- The evolution of cloud computing is in its peak form. Various companies are setting up their cloud infrastructure and providing it as a service to the

consumers. Microsoft, Amazon, IBM, Salesforce and SAP are some of the top cloud service providers today.

1.5 CLOUD COMPUTING BASICS

Cloud customers/users require cloud resources for their various applications. Cloud resources are allotted to multiple users for their requirements keeping in mind the availability. During this demand-supply cycle, certain quality standards need to be maintained by the service providers in order to sustain their operations and comply to their objectives. Cloud users specify their individual service quality standards and are negotiated and agreed upon by service providers in specific Service-Level-Agreements (SLAs). The entire functioning of cloud computing essentially includes maintaining the equilibrium of resources demand-supply chain along with honouring the specific SLAs.

The main entities in cloud computing scenario are –

Cloud Users/Brokers- They submit their application's service demands to a cloud data centre and expect allocation of resources and service processing as per their specified SLAs. In return, they need to pay to the service provider for the services and resources allocated to them. Cloud users can request services from any data centre irrespective of their own location.

Cloud Service Providers- They employ various resource management techniques to ensure service availability to various users. Their aim is to honour every aspect of the agreed SLA for each user and maintain certain QoS. They also decide and implement the pricing mechanism specifying how each service will be charged. Other than this, a cloud service provider needs to maintain the resources

accounting as how many are free, utilized, or are in demand. Load balancing a data centre is a crucial responsibility and various balancing schemes are implemented by the provider for the same. All in all, right from establishing a data centre to its maintenance along with a fair and just servicing of users' demands, everything falls under the lap of a service provider.

Virtual Machine Monitor (aka Hypervisor) - It is a piece of virtualization software that monitors the load on each VM, along with their availability and resource requirements.

Admission Control- This entity allows or blocks a particular service request from claiming resources, reasons ranging from non-availability of resources to load unevenness.

Resource Allocator- It deals with the distribution of cloud's resources among multiple service requests, keeping in mind the economic feasibility.

Job Scheduling- An optimal scheduling strategy thrives for balanced resource utilization along with the most favourable performance of the data centre. One of the main scheduling issues in cloud computing is an efficacious employment of cloud resources along with their fair distribution among users' job.

Virtual Machines- A physical server hosts multiple homogeneous or heterogeneous virtual machines to meet the growing demands of cloud users. They typically increase the user base of a data centre and provide the flexibility to combine various resources specific to service requests. Virtualization technology enables running multiple VMs under one physical machine and is the heart of a cloud data centre.

Physical Machines- A typical data centre consists of hundreds and thousands of computing servers which provide various resource utilities to satisfy the service demands.

1.6 TYPES OF SERVICES IN CLOUD COMPUTING

There can be six classes of services in cloud computing as described below-

i. **Software-as-a-Service -** Cloud applications are provided via an internet in the form of as Service. Cloud users access these application services using the interface on the web browser without any installation or download. It is a well-suited way to use the application software without any hassles. Every aspect of cloud application, from maintenance and support to storage and networking, is dealt by the cloud service provider. Examples of SaaS are Google Apps, Salesforce, Cisco WebEx, etc.

ii. **Platform-as-a-Service -** This cloud delivery model is used for cloud applications and other developments. It is a framework where cloud applications are developed and/or customized. Applications developed from PaaS model inherit all the cloud characteristics, such as, availability, scalability, failure tolerance, etc. The advantage of using PaaS is that it automates business policies and reduces the coding along with enabling application migration to other hybrid models. Some PaaS providers are Apprenda, Amazon Web Services, LongJump, among others.

iii. **Infrastructure-as-a-Service**-This model is employed to acquire a data centre's infrastructure such as storage, networking, compute, etc. It avoids the headache of purchasing any hardware, instead, offers to use it on a rental basis. With IaaS, cloud users get a hardware base on which they can further install their required

platform. Additionally, updation for new versions of these IaaS models needs to be done by the users themselves. Significant IaaS players include Microsoft Azure, Amazon Web Services, Google Compute Engine among others.

iv. **Functions-as-a-Service -** This can be an addition of a separate layer of abstraction to PaaS where instead of dealing with virtual machines and application runtimes, developers place their functional codes in the cloud to be triggered by some events. It is an instantiation of server-less computing.

v. **Integration Platform as a Service -**In order to share cloud data between SaaS and on-premise applications, certain connectors are required. iPaaS provides such connectors and enable cloud users to map, transform, and integrate work data.

vi. **Identity as a Service -** It encompasses security related attributes of cloud data centers. It uses cloud-based user profiles for authentication and facilitates access to cloud resources based on user privileges and security policies. Example of IDaaS providers are Okta, CA, Centrify, IBM, among others.

1.7 DEPLOYMENT MODELS IN CLOUD

Based on the deployment model, a cloud can be classified as-

a) **Public Cloud-** A public cloud lets a general user to access its resources, like applications, storage, etc through internet. From a cloud user point of view, public clouds offer the best possible economic scale as other than the capacity of resources used, user need not pay for hardware, application or bandwidth expenditure. However, it offers limited security and service quality features, hence may not be suitable for every organization. Major

players of public cloud are Elastic Compute Cloud (EC2) from Amazon, Blue Cloud (IBM), AppEngine (Google) among others.

b) **Private Cloud-** Cloud data centres owned and maintained by a single provider constitute a private cloud. These are costly to use as compared to public ones but offers a variety of scalability, flexibility and resource provisioning options. A tighter security and quality standards are provided by private clouds and are usually adopted by large enterprises.

c) **Hybrid Cloud-** Hybrid clouds are an amalgamation of public-private clouds. Many organizations maintain a private cloud and use public ones as and when required to tackle issues like over-loading, cost considerations, dynamicity, etc.

1.8 CLOUD ENABLING TECHNOLOGIES

i. **Virtualization-** It is the process of emulating a software or hardware environment and is completely transparent to a user. Virtualization is used to create virtual machine, virtual networks, virtual data centres and virtual storage disks. It is most commonly used in cloud computing to create virtual machines by enabling a single physical server to run multiple homogeneous or heterogeneous operating systems. A server machine running virtualization software can manage multiple applications in a concurrent fashion, even on different operating systems. Virtualization enables a data centre to dynamically share its resources among different cloud applications.

ii. **Scaling-** Cloud data centres host many physical machines or servers, connected via high speed networks. This helps in offering cloud services to

users at a relatively low cost and allows great flexibility in efficient allocation of cloud resources.

iii. **VM Migration-** This refers to the process of suspending the execution of a virtual machine on a server, transferring it to a different server and restarting its execution from the state it was suspended. VM migration is an important routine practice in cloud data centres for reasons like load-balancing, power conservation among others.

iv. **Device Power Adjustments-** Energy-smart cloud computing is the call of the hour. This concern has led equipment vendors and processors manufacturers to develop energy-smart hardware and devices. For example, Dynamic Voltage Scaling (DVS) configures a CPU to reduce its clock speed and to operate at high temperatures. This optimizes performance and reduces the necessity of costly cooling devices in a data centre.

1.9 RESOURCE MANAGEMENT IN CLOUD COMPUTING

Resource management is the task of providing cloud's resources, like storage, compute and network resources to cloud users for their applications in a manner such that it ensures optimum resource utilization and meets the stated performance objectives of both the stake holders. A cloud user's performance objectives include service availability, application performance, cost effectiveness and resources scaling to keep up with the dynamic demands of an application. From the service provider's point of view, performance includes efficient utilization of cloud's resources and profit maximization while adhering to the SLA parameters. Given below are the essential functional units for efficient resource management in cloud computing.

i. **Estimation of resource utilization-** To maintain a cloud data centre's load balance and to manage its irregular service requests, it is essential to monitor the individual and total utilization of different resource types. Server virtualization is one of the pivotal technologies used in cloud systems to increase resource utilization. It is already established that cloud resources are embodied in virtual machine. Creation of a virtual machine requires an operating system, certain number of processors, storage, RAM and is connected through network ports. Software called hypervisor, monitors the resource utilization of each resource type by all VMs active on a physical server. VM Placement is the process of selecting a host server with enough unoccupied resources to create a new VM so that a new application request can be serviced without any interruption. Thus, a VM monitor sitting on a physical server monitors and estimates the availability of resources to decide whether a new job demand can be granted resources or not. Virtualization improves the overall utilization of cloud's resources in a big way by scheduling them with fine granularity. It also ensures adherence to quality standards as each VM runs independently and individually with a proprietary resource. Hence, one can safely say that virtual machine placement is a key issue in resource management in cloud computing systems.

ii. **Resource Demands Profiling-** In a cloud computing scenario, it becomes the responsibility of the service provider to meet the SLA parameters along with ensuring a finer resource utilization. An over-provisioning of resources can lead to SLA violations, whereas an underutilization of resources may bring financial loss to the service provider. Virtual

machines guarantee flexible and efficient resource provisioning by migrating the machine state to another server. When a VM is placed on a physical server, it is given resources like memory, input-output, CPU, etc. The amount of resources allocated to each VM on a PM depends on its current demand and can be increased or decreased in the future. In any case, resource allocation must mimic the market scenario of demands and should be dynamic in nature. Virtual machine migration is a key player in allocation and/or reallocation of cloud's resources by shifting a single/group of VMs inside the data centre. It's a way to balance load and maintain an equilibrium with respect to resources, workload and energy consumption. Additionally, it also honours SLA parameters.

iii. **Profit with pricing maximization-** From a business point of view, cloud services are means of delivering computing resources on a metered basis. At present, IaaS charges VMs on a rental time-slot fashion, whereas PaaS charges VMs for both time and usage figures for bandwidth, storage and API calls. The essential business model for IaaS and PaaS providers includes offering cloud services in a usage-based, static pricing manner. However, in the recent past dynamic pricing of cloud resources is keenly adopted by players like Amazon via Amazon Spot Instances. It has been observed that resources utility can be improved by reducing their prices during light loads and vice versa. On the same lines, Cloud Users would like to obtain the lowest possible prices for the resources they want to lease, thereby, minimizing their expenditure and maximizing their profits.

iv. **Scheduling of Cloud Resources-** In a computing environment with hundreds and thousands of computing devices, each with varying degree of resources, it becomes crucial to assign resources in a just manner among incoming requests. In this regard, one looks forward to task scheduling strategies which must be capable of servicing users' jobs while guaranteeing an optimal utilization of computing resources. Cloud computing is a very dynamic computing environment. Here, the service requests vary greatly in resource requirements and quality preferences. Incoming tasks may include high priority requests with more computing resources and stringent timing requirements, whereas, there may be some tasks with flexible resource and quality demands. It, therefore, becomes the responsibility of the service provider to compare different service demands based on requirements, availability, cost, reliability, etc. The service request with the maximum weightage is given preference during resource allocation. It is observed that an efficient distribution of resources and effective load balancing measures improve the overall resource utilization to a great extent. Therefore, job scheduling in a cloud data centre to improvise overall resource management is an important research topic. Furthermore, as the service demands change, its placement and configuration modules may also change in a dynamic manner (VM placement/migration). This helps in predicting a realistic estimation of future demands. This vital functionality is realized by the Application Scaling and Provisioning functional element.

CHAPTER II

Review of Literature

CHAPTER II

REVIEW OF LITERATURE

A number of approaches have been discussed to improve live VM migrations in a cloud datacenter for various purposes like implementing energy-smart techniques, making cloud environment more fault tolerant and/or even load-balancing in an efficient manner. Timothy Wood et al. [1] present a cloud system keenly observes resource usage on all servers and decides to migrate a suitable number of VMs as and when required. Authors in [1] favor VMs swapping in combination with migration for efficient load balancing. Anja Strunk and Waltenegus Dargie [2] extend the theory of VM migrations by observing the power consumption in a data centre and the migration duration. They claim that network bandwidth as well as a VM's size effect energy consumption directly. These two factors affect the performance of migrations considerably as explained in [2].

Pre and post-copy techniques of migration are studied extensively by Diego Perez-Botero [3] for their appropriateness and operational concerns. A bundle of VM migration is presented by Samer Al-Kiswany et al. [4] called VMFlockMS which is applicable for a three-tier service. Check-pointing/recovery along with basics of trace/replay are employed to reduce total migration time by authors Hai Jin et al. [5] after which, both the old and the migrated virtual machines are brought to a steady state.

Authors Michael R. Hines and Kartik Gopalan [6] attempt to discriminate between pre and post-copy migration techniques and put forward a self-ballooning dynamic scheme to utilize unused memory. Their proposed method, however, is seen to degrade the datacenter's overall performance within the tolerable limits Alexander Stage and Thomas Setzer [7] combine VMs according to their indistinguishable workloads and propose a

resource and migration scheduling framework built on each group's topology and bandwidth requirements.

Erik Gustafsson [8] improves the existing migration technique by barring duplicate data travel from the disk in order to quickly resume a newly migrated VM before the complete memory relocation at the new host. VM migration in wide area networks is studied by authors T. Hirofuchi et al. in [9]. An increase in carbon footprints and power usage of cloud datacenters is a major area of concern nowadays. Likewise, data centre architecture, called pMapper, is proposed by A. Verma et al. [20] which places VMs on servers after carefully studying their cost and power usage requirements. A. Beloglazov et al. [11] propose VM migration in groups so that idle or poor-utilized servers can be switched off. This approach minimizes power consumption besides honoring the desired QoS requirements. An ERP (Energy-Response Time Product) metric is presented for the same. N. Bobroff et al. in [13] proposes VM migration in parallel to reduce SLA violations. To cap the frequency of VM migrations, T. Ferreto et al. [14] prioritize VMs on their steady capacities. Most of the references cited here call for the need of a live VM migration approach to lower down the service downtime and to honor the service-level agreement (SLA).

Resources consumed by virtual machines, time taken for migration and energy-consumption are some of the important factors which affect the live migration process as suggested by Voorsluys et al. in [15]. The effects of virtual machine migration on XEN machines is discussed by Diego in [16] and minimization of overheads incurred during live migration is suggested. Wei et. al in [17] propose VMFlockMS, an application-level solution, which discusses inter data-center virtual machine migrations in groups. A strategy to initiate migration at the right time is proposed by Hai et. al in [18] to make it more efficient, profitable and reliable. Pre and post copy approaches towards VM

migration are researched by Boru et. al in [19], which suggests a method to increase the migration speed by lowering down the migration volume. Replication is incorporated in cloud data-centers as a tool to reduce migration latency and improve energy efficiency as stated in [7] by Stage and Setzer. Power usage and energy consumption of servers are studied by authors in [21, 22] where VM migrations are initiated with the aim to reduce cost and carbon footprints of a cloud datacenter. Static and uneven dynamic user demands in a data-center and planning of energy conservation based on these demands is discussed by Deng et. al in [23]. Huang et. al [24] introduce dynamic grouping of servers and virtual machines for placements and migrations and registers a reduction in SLA violations. Importance is laid on VM placements in [25] by Jamshidi et. al that can eventually reduce the need for further VM migrations thereby making data-centers more efficient and stable. A detailed review of existing migration technologies, their comparison is given by authors in [26, 27] and stress is laid on the need of a strong, efficient migration technique that builds trust among users. Another technique to improve resource utility is through consolidation of servers on the basis of their RAM and CPU capacities. Same approach is considered by Hongyou et. al in [28], to lower down extra energy-consumption with the introduction of a live migration technique based on local information. VM migrations based on renewable energies using various resource distribution techniques are considered in [28].

A study of heterogeneous workloads and implementation of live VM migration techniques by consolidating these workloads is presented by Huang et. al to make energy-smart data-centers. Implementation of a live migration technique using Amazon EC2 clone, Eucalyptus, is given in [29] by Riteau et. al with the aim to upgrade the energy consumption in modern data centres. Most of the reviews discussed above attempt to make migration as effective as possible, however, they either stress on energy-reduction

or on improving migration statistics. A combined approach is lacking where not only VM migration is improved but also 'green computing' is applied.

Extensive amount of references citing live migration of virtual machine are available, analyzing the effects and behavior of migration on cloud data centers under changing scenarios. Factors affecting migration latency and resource consumption are discussed in [30] by Isci et. al and an improved migration approach is proposed based on the various workload characteristics and different hypervisor configuration. To achieve parallelism for higher efficiency rate, Kikuchi et. al in [31] call for migration of multiple virtual machines. Resources availability is a major concern for better migration results; hence, their accurate forecasting can bring a major improvement in virtual machine migration. This very fact is studied by authors in [32, 36] where an application's behavior with the availability of different types of resources is correlated.

A framework for live VM migration is given in [33] by Akoush et. al to study all the performance metrics. It is often seen that the choice of victim virtual machine for migration significantly reduces the migration as well as the service downtime. Anala et. al [34] lays out all the key aspects of live migration. Energy consumption and conservation are significant research topics in cloud computing now-a-days. A combination of energy efficiency along with efficient virtual machine migration is studied by Keijang et. al [35].

It is seen that compression and layered copying significantly lowers down the migration data volume and migration time, which results in a better cloud performance as implemented and verified using Xen VMs by authors Hai et al. in [42]. To minimize the power consumption while maximizing the utilization of cloud resources, an energy-aware heuristic is given by Zhibo and Shoubin in [37]. Failures in cloud data centers including network failure, VM failure and their effects on a data centre's performance are discussed by Xiaodong et. al [38]. A scheme to determine an appropriate PaaS provider for an

application with a specific resource requirement is presented by Gultekin and Vehbi [39]. To use cloud resources in the most appropriate way and to handle different demanding virtual machines effectively, Eric et' al [40] applies Dominant Resource Fairness Mechanisms. To combine numerous virtual machines on a multi-core computer, a layered resource management system is offered by Zenxiang et. al [41] and registers significant improvement in performance. To sustain the agreed QoS parameters values along with boosting the revenues, Garg et. al [43] put forth an admission control and scheduling mechanism which allocates cloud resources in a well-planned manner. A quadratic assignment approach, to reduce additional data centre traffic caused by live migrations, is proposed in [44] by Zeng et al. It also reduces congestion hotspots. To curb ever increasing network traffic and migration duration, an attempt has been made by presenting combined VM placement and migration algorithm by Shihong et. al in [45]. Also, for the same purpose, data likeness of replicated virtual machines is exploited in [46] by Balazs et al.

Majority of literature review in VM migration [30-46] examine live virtual machine migration from performance point of view but concentrates less on migration volume, duration, bandwidth and service downtime. These metrics not only decide the fate of a data centre's performance but also their optimization is required to study the migration statistics.

In order to improve the migration scenario, researchers need to go back and analyze when and how virtual machines are allocated to physical machines. For instance, VM allocation considered by Zaman and Grosu in [47] where a comparison has been made between the fixed-price and auction-based allocation scenarios. An important way resources can be managed inside a data centre is by switching off the servers which are under-utilized. This not only minimizes energy consumption but also maintains a healthy power-

performance trade-off as given by Khazaei et al. in [48]. VM placement has mostly been considered as a bin-packing problem by authors in [49, 50] and its feasible solutions include server consolidation as proposed by authors in [50, 51]. Using similar approach, ant colony optimization-based consolidation algorithm is proposed by Yongqiang et al. and compared with greedy based algorithm in [52]. It is the responsibility of a cloud provider to shrink the operational cost of a data centre, build profits in terms of revenues and guarantee the service level agreement parameters. To analyze VM placements, it is imperative to study VM migration in an extensive manner. Among all the different types of resources available in a cloud system, communication resources are the most crucial ones. In order to reduce communication delays and improve QoS, data replication is proposed by authors in [53, 54]. About a dozen and a half existing VM placement algorithms are compared by Kangkang et al. for varied objectives like to cut back job execution time of the incoming VM [55] using a knapsack-based placement proposal. A parallel live migration is suggested by Xiang et al. in [56] after a comparative investigation on Xen and KVM machines. One can also reduce the frequency of migrations by setting up VMs as per their steady capacities as suggested by Tiago et al. in [57].

A detailed research on cloud migration is carried out by Jamshidi et al. in [58] where efforts are being taken to reduce communication delays by routing the user's traffic which is close towards the resources in the datacenter. To serve static as well as dynamic user demands, Energy-Response time Product (ERP) is applied for evaluation of server farm management policies in [51] by Anshul Gandhi et al. and an optimized VM placement algorithm is proposed in [59] for expected and time-dependent loads by Wubin et al. A considerable improvement is registered by placing VMs and their images on the same server as it greatly reduces communication overhead during migration as suggested by

Meng et al. in [60]. Statistical multiplexing is used in [61] by Weiming et al. to consolidate multiple VMs and provision them together. A bi-level runtime reconfiguration scheme, proposed by Hing et al. in [62], is used in a vector arithmetic model which upgrades resource utilization and brings down the total migration time by implementing local adjustments and parallel migrations. Same problem is dealt by authors in [49] by considering it as a classical bin-packing problem and using greedy approach to solve it. For optimum resource utilization and reduced energy consumption, a two-stage heuristic algorithm is put forth by authors in [50] which address the issue of VM incorporation followed by its consequence on performance. A multi-objective VM placement strategy with reduced power consumption, lesser delay and improved server efficiency is detailed out in [63] by Kao et al. To guarantee improved QoS and more than expected revenues, authors Addya et al. in [64] propose a hybrid queuing method for VM placement. Nowadays, a burning issue in cloud datacenters is energy conservation strategies. Energy-smart placement techniques aim to reduce the mounting power consumption by either consolidating servers, as given by Khalilzad et al. in [65] and/or by using genetic algorithms as proposed by Liu et al. in [66]. Scheduling techniques commonly used in a cloud environment are discussed in [67] by authors Yousefian and Zadnavin with the improvisation in certain key performance factors such as response time, resource utilization, etc. Further, F Nadeem in [68] a comparative study of multiple cloud service providers is sketched out based on their merits and demerits with respect to services quality. Most of the references pertaining to an effective VM placement consider only one or two objectives, either energy efficiency or maintaining QoS as declared by authors in [51, 54]. Most of the work done in this line considers placement as a classic bin-packing problem and tries to solve it using best-fit or first-fit approach [52, 55]. It is seen that considering placement as a multi-dimensional problem may solve many

highlighted issues. A careful consideration of heterogeneity in virtual machines can improve placement issue to a great extent by reducing unnecessary migrations and power consumption while maximizing resources' utility. VM placement with the objective of load balancing in a data centre comes with its set of challenges. However, this also leads to resources being used in a disproportionate manner in a data centre, usually due to a mismatch between a VM and its host as given by Randles et al. in [69]. To prevent such scenarios, load balancing techniques have been introduced which not only maintains an even resource utilization among the PMs, it also ensures scalability, reformed response time and strength as given in [70] by Kansal and Chana. In this regard, a genetic solution is proposed by Hu et al. in [71] to map the VMs and PMs by balancing load and avoiding unnecessary migrations. It is implemented on a hierarchical structure and uses historical data to choose the best host for a VM. The downfall of this method is its complexity. An ant colony based optimized load balancing technique is presented by Wen et al. in [72] which achieves lesser migrations, better load balance and less SLA violations. However, it is unsuitable for large data centres as it operates on few servers. Moreover, only CPU resource is considered as a host's load which further reduces its advantages. An initial VM placement method proposed by Huang et al. in [73] maps VMs to PMs using probability approach and relieves hosts from over-loading. This technique provides useful results to balance the load but does not consider migration scenarios in a data centre.

A dynamic and integrated resource scheduling scheme is propositioned by Tian et al. in [74] based on queues system which selects the front VM in the queue to place on a host. DAIRS, however, does not consider communication cost in a data centre to place a VM. A hybrid placement algorithm is proposed, in [75] by Thiruvenkadam and Kamalakkannnan, to reduce the number of migrations by balancing VMs on servers. Based on 2 phases, this hybrid technique uses a heuristic approach to place VMs in the

first phase and later optimizes their placements using a meta-heuristic algorithm. With the aim of servicing as many user requests as possible, an ant colony based VM load balancing method is presented by Cho et al. in [76] which considers multiple heterogeneous VMs and allocates them dynamically in a meta-heuristic strategy. It considers rejection of VMs which is not a possible scenario in data centres today.

A distributed architecture-based resource management technique is presented by Yazir et al. in [77] that tightly couples node agents with PMs and uses a multi-criteria based placement strategy. It however, considers only usual criteria for comparing possible hosts and ignores SLA violations. Resource allocation techniques [78] have been suggested by Bennani and Menasce, for non-virtualized data centres consisting of two layers of agents, namely, local and global. Based on separate analytical models, these works use resource demands prediction methods. Authors in [78] apply queuing theory and combining it with decomposition learning approach. Allocation work presented by Das et al. in [79] introduces a commercial computing system called Unity for non-virtualized data centres based on layered architecture and utility model. Virtualization brought the resource utilization problem into the main stream with the 2-level agents getting encased into virtual machines. Research works outlined by authors in [80, 81] use centralized methods of resource distribution and are applicable in virtual data centres. Migration overhead is given due weightage by H N Van and Tran in [82] and is used as an important parameter to elect the most suitable host for VM placement.

Cloud computing services are fulfilled by renting resources on a timely fashion. Due to its various benefits like cost-efficiency, low maintenance, improved flexibility, etc, it is a lucrative offer for cloud users. Nevertheless, a cloud-based system must make certain a genuine, equitable and cost-effective distribution of its resources among users. For the same, a non pre-emptive asset evaluation and distribution scheme is given by Jain et al.

[84] for batch jobs. It also claims to guarantee economic profit to the service provider. An admission control and job scheduling procedure is presented by Garg et al. in [85] whose active players are cloud users, a SaaS provider and a public IaaS, and it lessens the cost of a service provider while enhancing user's experience. A resource assignment solution based on the Dirichlet multinomial model is set forth by Yanping et al. [86] to reduce computing cost among others. Combinatorial auctions are put forward by Haoming et al. [87] assuring cost-efficiency and veracity using price vector space. Another alternative is given in [88] by Xuelin et al. which integrates Double Auction technique. Christos et al. in [91] details out an extensive list of SLA-based resource management and allocation strategies. Resource allocation strategies detailed out by Wang et al. in [89] make use of periodic auction to go with the dynamic user requirements.

N. Ani Brown Mary et al in [90] surveyed multiple QoS parameters in cloud computing to discuss about their strengths and weaknesses. A novel job admission policy is presented by Dhok et al. in [92] which consider an overload threshold value for decision making. Resource management problems are targeted by Bo An et al. in [93] for the purpose of making revenues. Bo et al. in [93] also employ negotiation strategies between a service provider and users to further enhance the benefits of their proposed allocation mechanism. Chaisiri [94] combines Benders decomposition with sample-average approximation to address some of the reservation-based allocation issues. Further, Haiyang and Medhi [95] present a multi-time period optimization model to turn off servers for specific time durations and thus, lower down the expenditures of a service provider. Also to prevent untrue bidding behaviours of cloud users, Zhang et al. in [96] put forward a truthful resource auction which imitates the requirement-demand curve of different resources. Majority of the researchers prefer online truthful auction mechanisms by either maximizing service provider's revenue or minimizing their operational cost.

In the cloud computing environment, the mounting of cloud jobs may result in uneven and unfair utilization of cloud resources, resulting in escalating computing cost and carbon emission. To address this issue, job scheduling is seen as an effective tool to make use of cloud resources in an efficient manner.

To reduce the power consumption, a decision framework is developed by Makkes et. al. [97] which monitors the energy consumption of both local and remote sites and accordingly decides to transfer data between them. This movement of data from one site to another, however, will increase the transmission overhead; hence, the proposed idea does not give a very favorable result. Another approach presented by Chen et. al. in [98] manages cloud resources based on personalized user requirements and resources runtime behavior[2]. This approach requires full knowledge of incoming jobs which is highly unlikely in a cloud environment. An analytical model of resource management based on the patterns of user requests is proposed using cache nodes to service the requests of identifiable jobs in [99] by Sithole et. al. However, the feasibility of recognizing user patterns in a diverse environment like cloud computing still needs to be proven. Among all the processing data, research data is of prime concern and Waddington et. al [100] focus on its storage and issue concerning its preservation. Conventional approaches to scheduling are differentiated with a priority-based scheduling system in [101] by Agarwal D and Jain S, which is characterized by the dynamic property of a cloud job.

Factors effecting the energy consumption in a cloud data-center are analyzed in [102] by Liu et. al to increase energy efficiency without adversely affecting its performance and QoS parameters. Multi-objective based scheduling methods are proposed by authors in [103, 104] where authors assign job deadline as the highest priority. On the other hand, dynamic programming concepts are employed by Ghanbari et. al [105] for scheduling cloud jobs with varied execution time requirements.

A new insight dealing with virtual machine pre-emption is given by Ghanbari and Othman [105] in order to accommodate multiple jobs in a concurrent manner. This technique, unfortunately, increases the workload as it requires extensive tracing and synchronization mechanisms. One of the main objectives of task scheduling is efficient power consumption which can be realized by using genetic algorithms based on MapReduce in [106, 107]. Likewise, in [108] authors Waldspurger et al. used a genetic algorithm along with artificial neural networks to schedule tasks in an optimal fashion. References related to job scheduling aim at lowering down the energy consumption or operational cost leaving a lot of scope for further improvement in the said direction. The basic idea of scheduling, however, is to effectively reduce operational time while servicing most of the incoming requests.

Admission control is a common practice employed in data centers to filter out service requests which are not feasible to implement due to performance or availability constraints. In the context of cloud computing, resources are provided to cloud users as a service commodity as stated by authors in [111, 112]. These offered services are guided by the service level agreement (SLA) duly agreed upon by both the users and the cloud provider which defines the quality parameters expected during the service processing. Chen et al. [113] explains that incoming requests arrive at an uneven rate which makes it difficult to expect the future workload of a data center. The mechanisms of admission control are generally categorized as either request-based or session-based. The core idea behind session-based admission control is presented by Elnikety et. al in [114] which form the basis for many admission techniques originating from session-based. Yet another feature of admission constraint is given in by authors in multi-programming level which limits the number of simultaneous requests that can be handled by a data center in [115, 116].

A typical cloud service provider tries to maximize its profit by allowing as many service requests as possible which can result in SLA violations. However, a common practice is that these violations need to be proved by the user's side for penalty to be implemented. Many admission control mechanisms do rely on future estimation of service workloads based on a non-linear regression model as stated by Heiss et al. in [117]. This may work for web-based cloud applications [118] but not for all. In such cases, the probability of an incoming service request depends on its previous occurrences. A profit-oriented admission control technique is presented by Tozer et al. in [119] which work well for optical networks. Machine learning has also been attempted to use in admission control implementation [17]. For example, Tozer et. al in [119] use a regression tree to reduce error pruning. Almost all of the admission control methods discussed above look into either one kind of attribute like performance, profit, sessions (etc.).

Work presented by various authors in [130], [131], [145] have discussed various security issues in cloud computing. Risk compliance and security aspects of cloud's service delivery models are studied by Subashini and Kavitha in [129]. On same lines, Kamongi et. al. [144] developed a risk model for the cloud which failed to cope with existing compliance standards. Statistical study highlighting, the cloud service providers adapting cloud security standards, is discussed by author Dr. Jorg in [127] which raises questions on the cloud's capability of handling potential threats. Further it also addresses the practical concerns of cloud users while choosing a potential cloud service provider. The cloud computing reference architecture presented by authors in [127, 133] categorizes the potential privacy and security policies with respect to the interest of a cloud provider. The IT compliance model given by Dr. Jorg in [127] highlights network and IT infrastructure and electronic data processing. To make cloud service models transparent, security

controls and the compliance model is maintained so that consumers and end users receive a reliable data protection in the cloud.

Cloud data centers enable customers to use computing services, platform and infrastructure with high efficiency and user-friendly billing system [146]. Unfortunately, data centers suffer from high computational cost due to increasing power and energy consumption as stated by Katz in [149]. This calls for the development of certain optimization techniques to handle and reduce the increase in energy consumption without adversely affecting the reliability and efficiency of data center resources like computing, storage, bandwidth, etc. as given by Kliazovich et al. in [154]. As far as the energy consumption scenario is concerned, it is observed in [148] by Popovic and Hocenski that IT and networking equipments consume nearly 50% of the total power used in a data center. Further, Li Shang et al [150] states that approximately half of such energy consumption is due to the data traffic inside a data center. A large body of work, concerning energy efficiency in cloud data centers considers that datacenter infrastructures are underutilized [152]. Among all the solutions offered, the Dynamic Power Management (DPM) method puts idle equipments into sleep mode as given in [152] by Lin et al, whereas Dynamic Voltage and Frequency Scaling (DVFS), given by authors Horvath et al. in [153] explores the connection between power consumption *P*, supplied voltage V, and operating frequency *f*. It is seen that voltage and frequency reduction has a direct impact on the power consumed. Further, power consumption of a server is also associated directly with its CPU utilization and memory. According to the work done by Fan et al in [157], the power consumption of a server in an idle state is approximately about two-thirds of its peak state power consumption. Power consumption in switches is constant for chassis and line cards, however, energy consumed by the ports depends on the communication traffic as declared by authors in [156, 159].

Data replication models also help in optimization of data center energy as stated by various authors in [150, 151, 155 and 158]. DCell, a recursive topology, is studied by Bilal et al in [160] and it is seen that it provides better scalability and robustness as compared to fat-tree. Research studies concentrating on energy efficiency emphasize on load balancing and voltage-frequency trade-offs in switch-centric data center architecture.

CHAPTER III

Location-aware VM Migration in Cloud Environment

The content of this chapter is published in-

1. **Springer series on Lecture Notes in Networks and Systems, vol. 9, pp. 119-129, 2018, ISSN 2367 – 3370.**
2. **Elsevier Procedia Computer Science Journal, vol. 45, pp. 823 – 831, 2015, ISSN 1877 – 0509.**
3. **Research Journal of Recent Sciences, vol. 3, pp. 13-20, 2014, ISSN 2277-2502.**

CHAPTER III

LOCATION-AWARE VM MIGRATION IN CLOUD ENVIRONMENT

3.1 INTRODUCTION

The popularity of cloud services is attributed to the technique of virtualization. Virtualization technique detaches hardware from software components on a server and enables more than one heterogeneous OS instances to run on a single server. Hypervisor (or VMMonitor) supervise the virtualization procedure. As the guest operating system is not bounded by the hardware, so it is possible to shift it from one physical machine to another. This phenomenon, known as VM migration, is an important function of a cloud data center. VM Migration is a practical performance enhancing feature of a cloud system. It strengthens fault tolerance, workload-balance and saves power by shutting down under-utilized servers. A key aspect of migration is that the application or its processes are unaware of the shifting process, thus, infusing the notion of transparency. Modern data centers prefer 'live' migration over 'stop-and-copy' one, where a migrating VM is stopped to copy its contents from source to destination host. Live' migration offers a lower service downtime as compared to the other variant.

The migration technique proposed in this chapter works well in a clustered data centre and improves performance parameters like migration time and service downtime significantly.

3.2 LOCATION-AWARE VM MIGRATION

The proposed live migration technique considers clustering of physical machines (PMs) based on the 'application type'. This means that all the PMs servicing same or similar applications are grouped under one single cluster. Next, we choose a virtual machine (VM) for migration and call it as 'victim' VM; it is usually the most demanding VM on an overloaded PM. Thereafter, a destination PM is selected to host the migrated VM. It accesses the profile file and execution log of the victim VM and synchronizes its functioning at the new destination host.

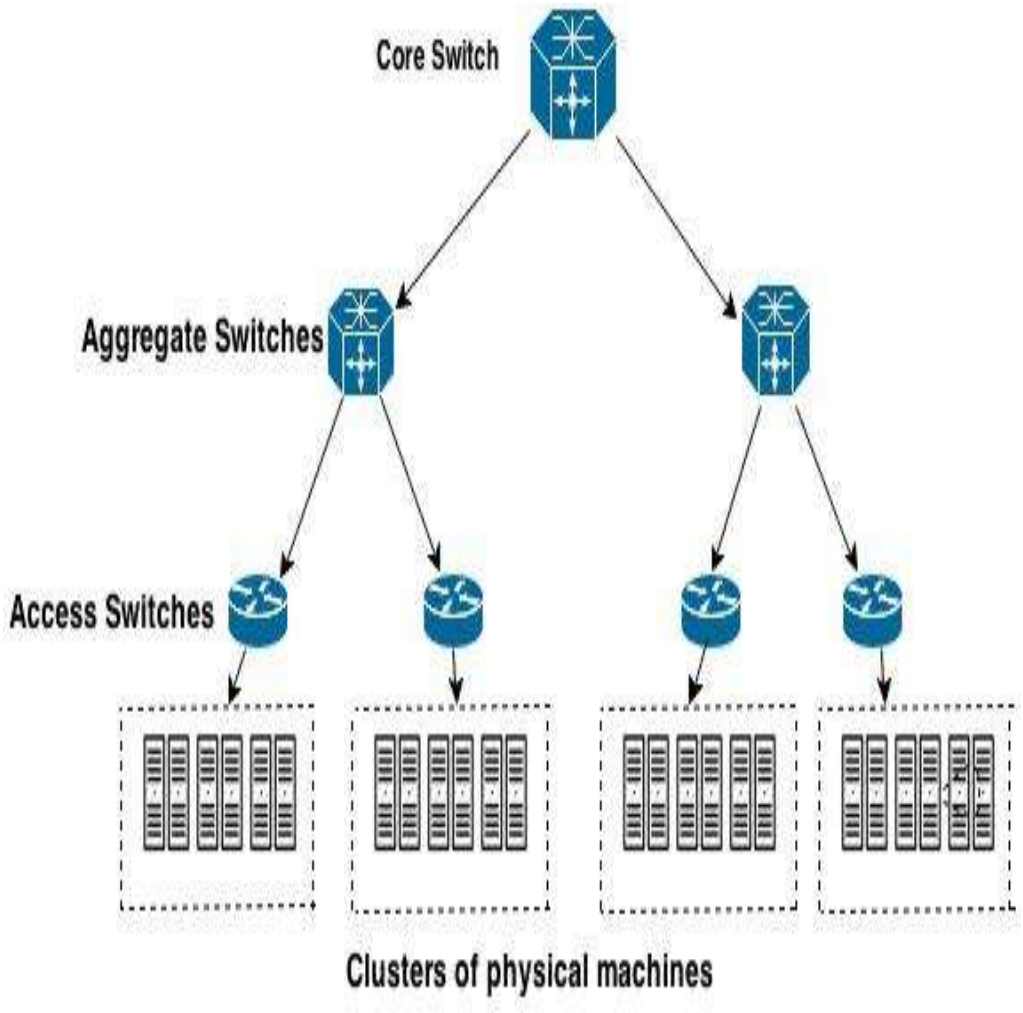

Figure 3.1 Proposed Data-center Topology

As a last step, the ongoing input-output requests are directed towards this new host. Characteristics of the suggested migration scheme are given as-

a. **Topology**- A three tier fat-tree topology as implemented in most of the modern data centres is considered to take advantage of low-cost switches, scalability and contention prevention. The adopted topology is demonstrated in figure 3.1.

b. **Clustered Environment**- Clusters are formed depending on the type of applications processed by physical machines. Virtual machines, belonging to the same or similar applications, have identical memory images, so they can be hosted on PMs which form one cluster. Further, we are restricting VM migrations within its own cluster, so as to take full advantage of identical memory images. This reduces migration volume as well as migration time. The decision to restrict VM migration inside its cluster makes it a location-aware migration. Advantages of using location-aware migration are- IP address of a migrating VM need not change after migration, network disk need not be transferred during migration and very less amount of memory image needs to be migrated, as the destination PM in the cluster also has an identical image of the migrating virtual machine. This migration approach can also be seen as location-restricted virtual machine migration.

c. **Selection of source host, destination host and victim VM**- This step selects a source host which is overloaded and needs load shedding, a destination host which will host the migrated virtual machine and a 'victim' VM which is migrated from a source host to a destination host for load-balancing.

   i. **Selections of an overloaded source host**- Physical machines in a cluster are continuously monitored for any sign of overload. This includes a sudden spike

in a VM's traffic suggesting an increase in its I/O activity in case of a heavier load application or a spike in CPU or RAM usage. We are monitoring the utilization of four resources namely CPU, disk, memory and bandwidth resources of each physical machine. Suppose a single PM is hosting 'n' active VMs, each with its maximum resource capacity/demand as given below in table 3.1-

Table 3.1 Example of VMs demands on a single server

Max Demand	CPU	Memory	Disk	B/w
VM_1	a_1	b_1	c_1	d_1
VM_2	a_2	b_2	c_2	d_2
:	:	:	:	:
VM_n	a_n	b_n	c_n	d_n

Let the total capacity of a PM w.r.t the four resources be A (CPU), B (Memory), C (Disk) and D (Bandwidth). These n VMs will be hosted by a PM iff equation 1 holds true.

$$\sum_{i=1}^{n} a_i \prec \mathrm{A} \;\&\&\; \sum_{i=1}^{n} b_i \prec \mathrm{B} \;\&\&\; \sum_{i=1}^{n} c_i \prec C \;\&\&\; \sum_{i=1}^{n} d_i \prec D \qquad \ldots (1)$$

An overloaded PM will exhibit a characteristic as shown in equation 2 below-

$$\sum_{i=1}^{n} a_i \cong \mathrm{A} \,||\, \sum_{i=1}^{n} b_i \cong \mathrm{B} \,||\, \sum_{i=1}^{n} c_i \cong C \,||\, \sum_{i=1}^{n} d_i \cong D \qquad \ldots (2)$$

Also, total load on a PM_i caused by active VMs, at any instant, will be calculated as

$$\text{Load}(PM_i) = \Theta_{CPU} + \Theta_{mem} + \Theta_{disk} + \Theta_{net} \qquad .... (3)$$

In equation 3, Θ_{CPU}, Θ_{mem}, Θ_{disk} and Θ_{net} are the total utilization ratios of CPU, memory, disk and network bandwidth respectively by all active virtual machines on any PM_i and their individual values range from 0 to 1. For example, if all the active VMs are using its host memory up to their maximum demand (i.e. $b_1, b_2, \ldots, b_n$) then Θ_{mem} is kept at 1. For cases where a particular resource is not utilized, Θ_{mem} is kept at 0. Moreover, load calculated in equation (3) range from 0 to 4. Accordingly, a threshold value of 3.5 is chosen to declare a physical machine as overloaded, i.e., if the load on a physical machine is equal to or exceeds 3.5, it is taken as 'overloaded' else not.

ii. **'Victim' VM for migration-** One of the crucial task during migration is to find out the VM(s) which can be migrated. In this chapter, we are considering the migration of a single virtual machine terming it as a 'victim' VM. An

active virtual machine, whose load on its overloaded host is the maximum, is selected for migration. To choose a victim VM, we calculate the load incurred by each VM on a PM as-

$$Load\ incurred\ by\ a\ VM\ on\ a\ PM = \frac{\text{Re}\ sources\ consumed\ by\ a\ VM}{\text{Re}\ souce\ demanded\ by\ a\ VM} \quad \ldots (4)$$

iii. Selection of a destination host- To choose a destination host, we maintain an 'availability' graph of physical machines in the cluster whose resource availabilities satisfy the migrating VM's requirement constraints. These available physical machines constitute the nodes of the graph and the edges represent the distance between them. This graph will include the source host as well. Consider the graph shown in figure 3.2 below.

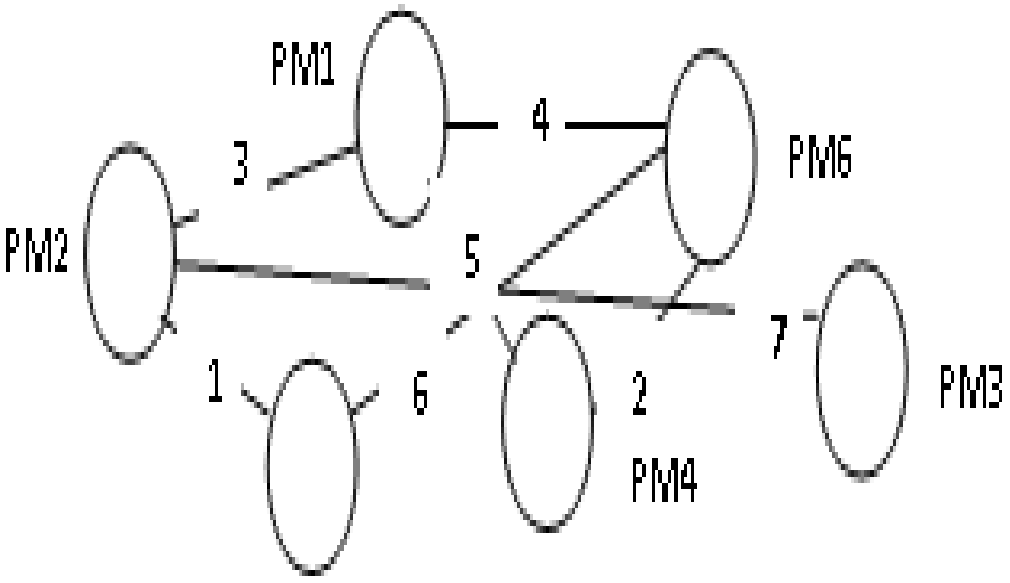

Figure 3.2 'Availability' graph example

In figure 3.2, the source host is PM3 and VM11 is the migrating VM. All the other nodes represent physical machines which satisfy the resource availability criteria of the migrating VM, i.e., all these PMs (PM1, PM2, PM4, PM5 and PM6) can host the migrating VM. In order to select the best possible destination host, we consider the distance from source host to all other available hosts. As is shown in the figure 3.2, PM3 is directly linked with PM2 and PM4 with distance 7 and 2 respectively. With the aim to reduce the migration time, we select the minimum distance of the destination host from source host PM3 and hence, PM4 is chosen as the destination physical machine. Note that all these PMs belong to a single cluster.

The destination host machine retrieves the profile file of victim virtual machine from Network Attached Storage (NAS) device. Profile files (*.nvram files) of virtual machines contain the data-blocks which are required at boot-time and application startup time. They assist in reconstructing the full image of the victim VM at the destination host machine. Next, the execution log of the victim VM, up to the latest checkpoint, is transferred from source host machine to the destination host machine. This helps in synchronizing the victim VM at the source host with the newly started destination VM. So far, the source VM is servicing the input-output requests of the cloud customer.

d. After the completion of step 5, the new migrated VM at the destination is in a position to service the input-output requests of cloud customers. At this juncture, source VM is stopped and all input-output requests are directed to the new VM at a new host. If the requested data is not available with the new migrated VM, then it is requested from the old VM at the source machine on a higher priority basis,

reflecting the pull phase of traditional migration. The diagrammatic view of the proposed procedure is shown in figure 3.3.

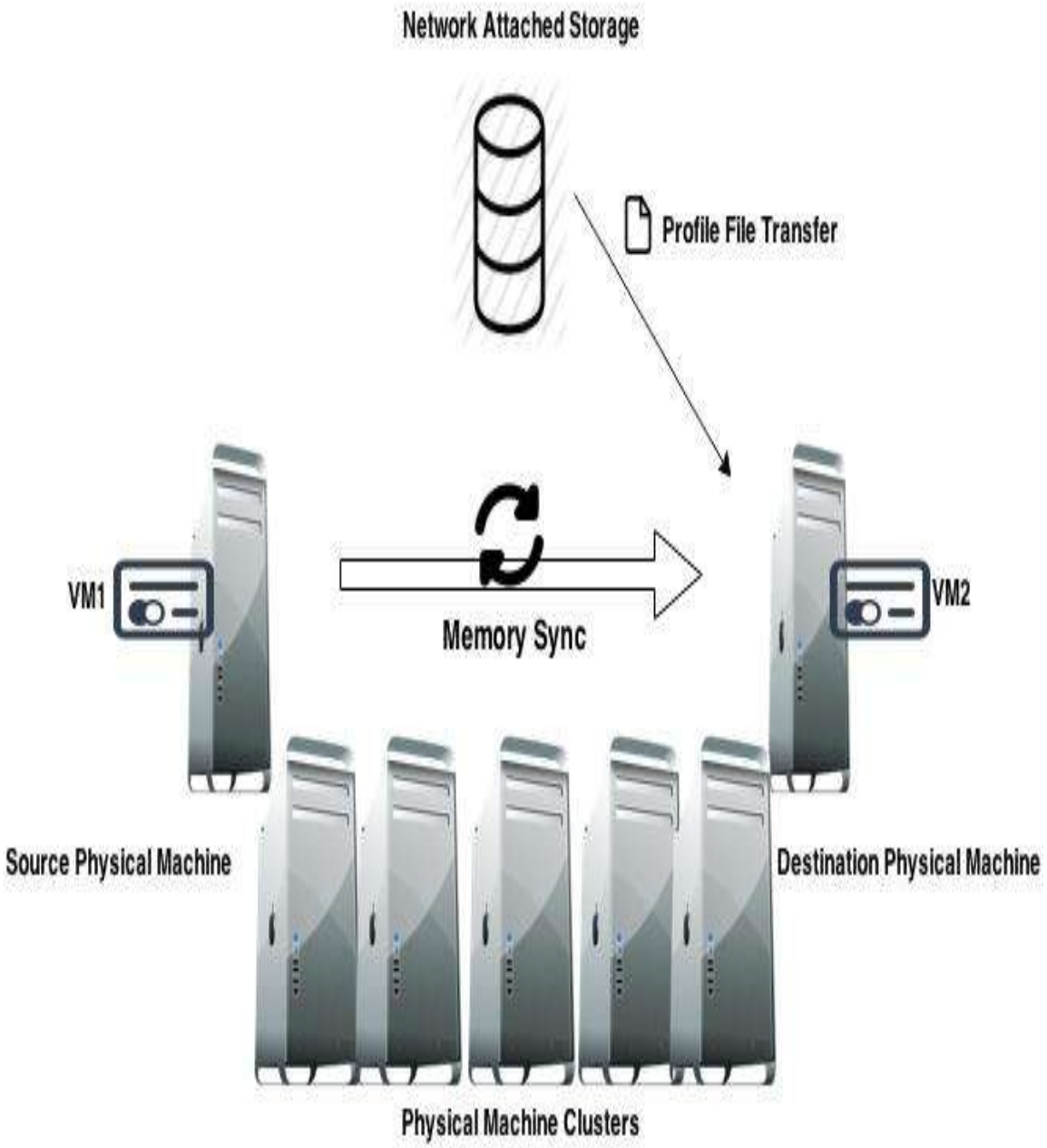

Figure 3.3 Proposed Live VM migration Technique

3.3 SIMULATION RESULTS

The proposed location-aware virtual machine migration technique is simulated using NS3 and compared it with migration in a non-clustered environment. Figure 3.4 shows the proposed live virtual machine migration in a clustered data-center consisting of 12 physical machines while figure 3.5 shows non-clustered data-center with multiple physical machines. For simplicity, figure 3.5 shows only source and destination host machines. The data-rate considered in non-clustered data-center is 1Gbps with a delay of 2ms whereas in clusters it is taken as 100 Mbps with 6560ns delay. VM image size is same in both the scenarios, taken as 1024 bytes. Both source and destination physical machines are assumed to be containing single virtual machine.

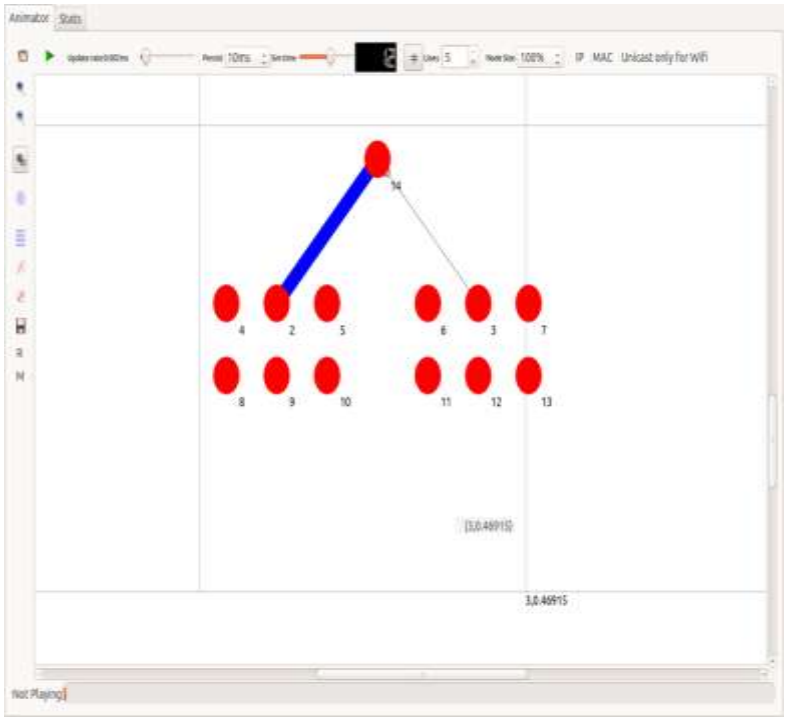

Figure 3.4 Clustered Data Center Topology

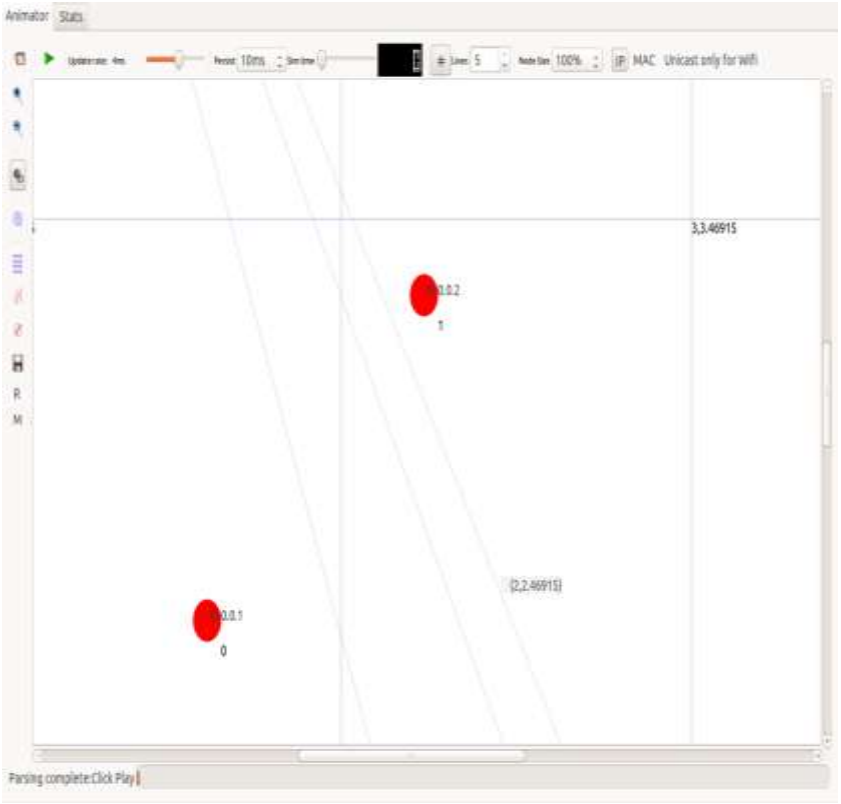

Figure 3.5 Non-Clustered Data Center Topology

Figure 3.6 compares the RTT during migrations in a clustered and non-clustered scenario. As the proposed migration is taking place within cluster, its round-trip time is lower than the non-clustered migration, thereby, making proposed migration faster. The migration technique is seen to bring down the total energy consumption of a data centre as seen in figure 3.8.

Energy consumption is calculated with respect to time in the experiment. Intra-cluster migration takes lesser time than inter-cluster migration, therefore resulting in reduced energy consumption inside a cluster. Faster migration in clusters also enables a source host to switch off early as compared to delayed VM migrations in non-clustered data-centers as,

$$EnergyConsumed \propto Time\ a\ server\ is\ switched\ on$$

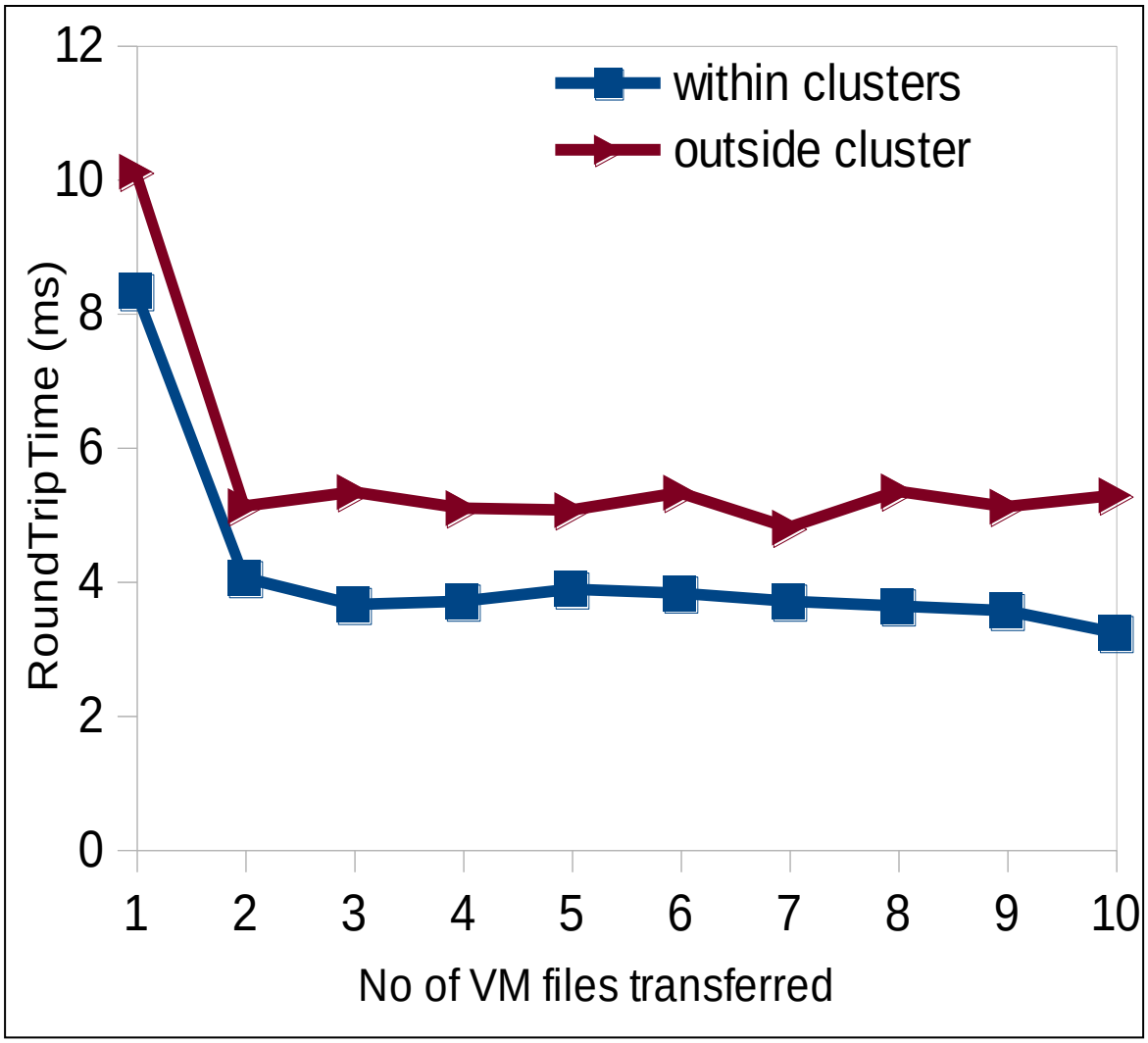

Figure 3.6 RTT comparison

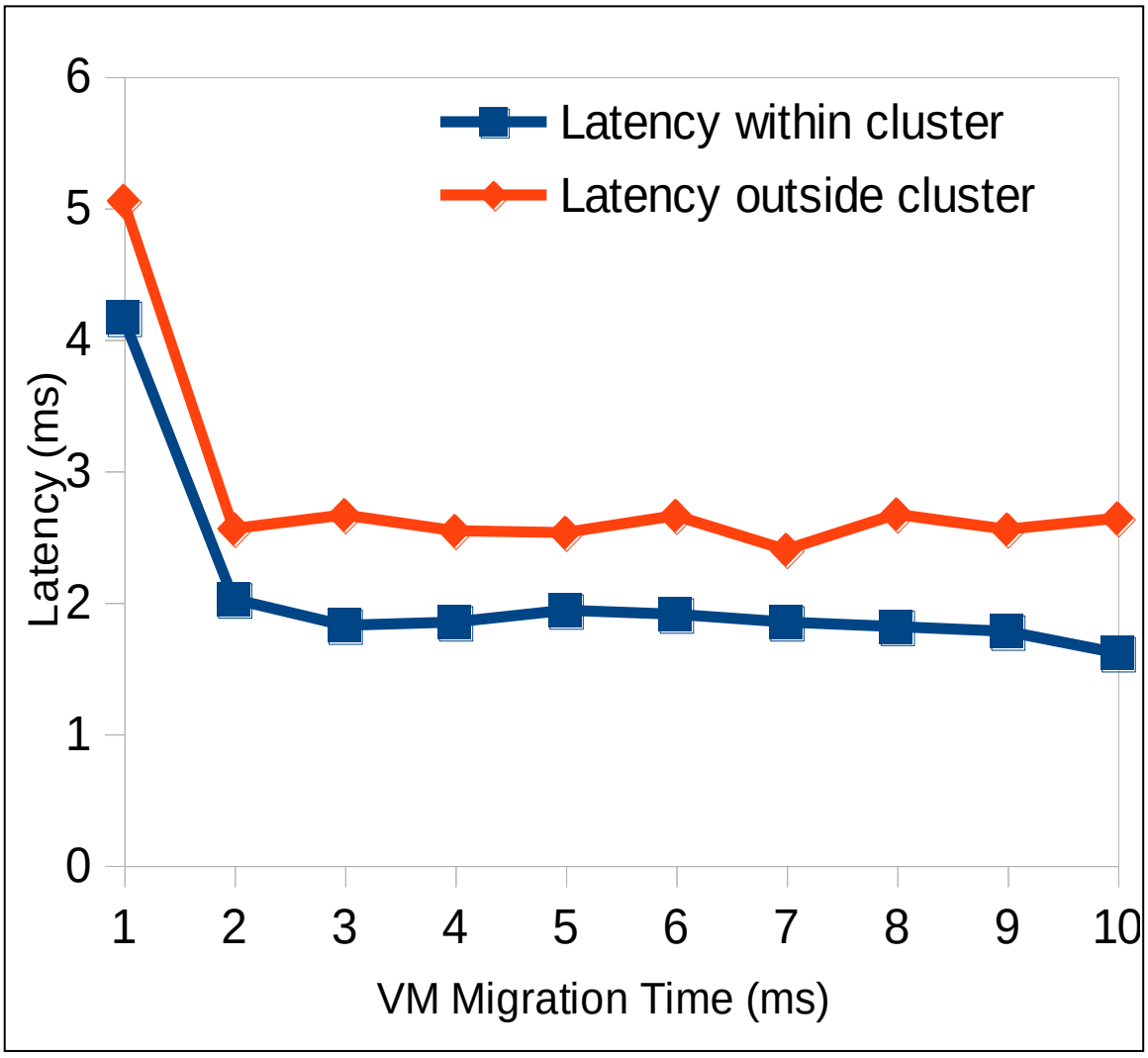

Figure 3.7 Latency comparison

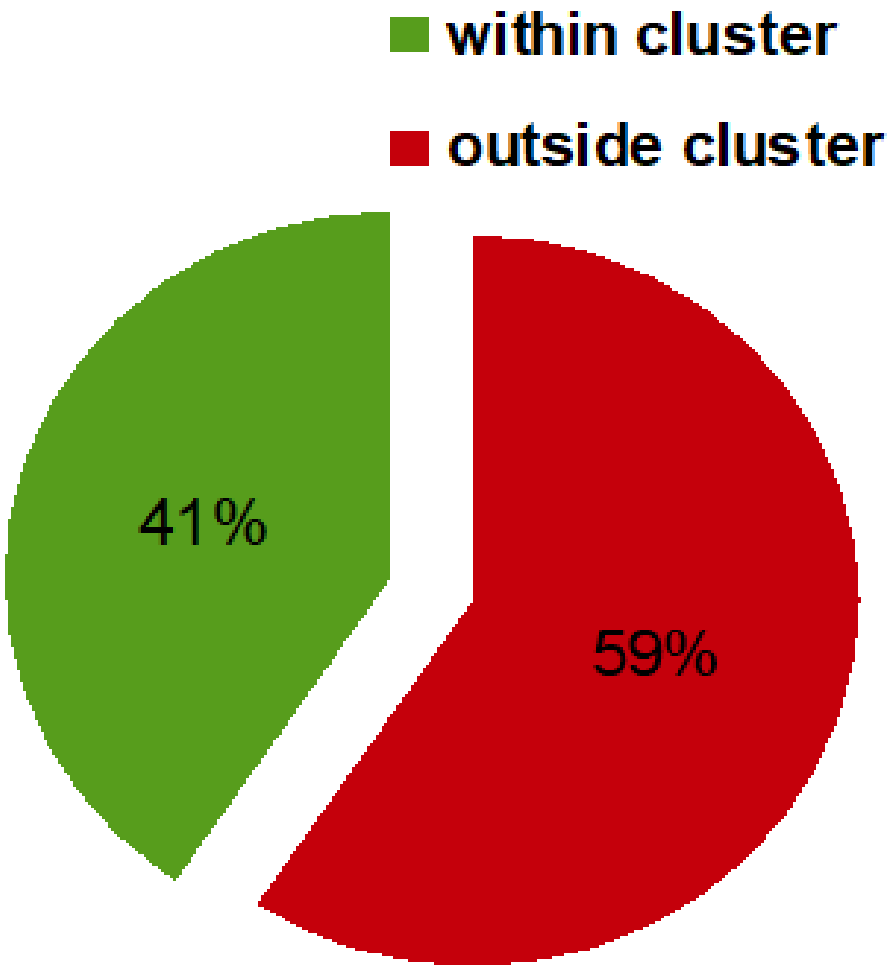

Figure 3.8 Energy consumption

Thus, the above performance claims show that the proposed location-aware migration is more advantageous than the one in a non-clustered data centre.

CHAPTER IV

Energy-efficient VM Placement

The content of this chapter is published in-

1. **Indian Journal of Science & Technology, vol. 10(32), pp. 1-8, 2017, ISSN: 0974 - 6846.**
2. **BRIS Journal of Advances in Science and Technology, vol. 4(1), pp. 22-33, 2017, ISSN: 1444 - 8939.**

CHAPTER IV

ENERGY-EFFICIENT VM PLACEMENT

4.1. INTRODUCTION

VMs are software extensions of physical machines. A single physical machine can generate many virtual machines and hence, can service multiple users. These VMs exhibit fluctuating behavior in terms of resource usage and specifications leading to disproportionate resource utilization in physical machines. This load imbalance on a host degrades its performance and violates many SLA features. Hence, an effective virtual machine placement technique is required to maintain an even load in the data centre. VM placement chooses a physical server to allocate an incoming VM so that the resource demands of the VM are satisfied by availability with the server. An efficient placement avoids frequent migrations and refine the performance features in cloud computing.

This chapter presents two VM placement techniques implemented on three-tier DC architecture, namely a profitability-aware placement technique and a load-aware placement technique.

4.2. PROFITABILITY-AWARE VM PLACEMENT

The proposed placement technique ensures fair deal and higher profit to the cloud service provider, along with uninterrupted service availability to the cloud user, using the concept of clusters.

4.2.1 System Model and Problem Definition

In order to implement the proposed profitability-aware VM placement, we follow the 'divide and conquer' approach by considering a data center consisting of six queues (Q1 to Q6) and six clusters (C1 to C6). In general, for 'k' types of resources there will be 2^k queues and clusters. At present we are considering three types of resources, namely CPU (MIPS), memory (MB) and I/O (B/sec), hence, 6 queues and clusters. These queues will hold incoming VMs as per the membership criteria while clusters contain PMs. Each VM's demand vector D will enter any one of the six queues based on its resource requirements. Similarly, each physical machine in the data center will be allotted to one of the six clusters based on its resources availability. If the resource availability of a PM changes, due to a VM placement or migration, the said PM will be re-allocated to a new cluster as per the resource availability/membership criteria. Queue Q1 is considered to have all demand vectors (D), with highest processor demand and lowest I/O demand. Q2 consists of all VMs whose processor demand is the highest and memory demand is the lowest. Q3 consists of all VMs whose memory demand is the highest whereas processor demand is the lowest. Q4 consists of all VMs whose memory demand is the highest whereas I/O demand is the lowest. Q5 consists of all VMs whose I/O demand is the highest whereas processor demand is the lowest. Q6 consists of all VMs whose I/O demand is the highest whereas memory demand is the lowest. Table 4.1 describes the membership criteria for queues and clusters. The comparison of highest or lowest resource demand of a VM is done within the vector by comparing its direction cosines and is irrespective of other incoming VMs. For example, consider a demand vector D as shown below-

VM5	20	9	25	100

Here VM_{id} is VM5, processor demand (P) is 20, memory demand (M) is 9, I/O demand (I/O) is 25 and rental price (RP) to be paid is taken as 100. Direction cosines of vector D are as follows-

$$\cos\alpha = \frac{D_x}{D} = \frac{20}{33.25} = 0.601$$

$$\cos\beta = \frac{D_y}{D} = \frac{9}{33.25} = 0.270$$

$$\cos\gamma = \frac{D_z}{D} = \frac{25}{33.25} = 0.751$$

As is evident after calculating the direction cosines that vector D has highest I/O demand (0.751) and lowest memory demand (0.270), hence it will be allocated to queue Q6 (Refer table 4.1). On similar lines, all physical machines whose processor availability is the highest and I/O availability is the lowest will be clubbed together in C5. Cluster C3 contains all PMs whose processor availability is the highest and memory availability is the least. Cluster C2 contains all PMs whose memory availability is the highest and processor availability is the least. Cluster C6 contains all PMs whose memory availability is the highest and I/O availability is the least. Cluster C1 contains all PMs whose I/O availability the highest and processor availability is the least. Finally, Cluster C4 contains all PMs

whose I/O availability the highest and memory availability is the least. As an example, consider the availability vector A as

PM7	20	50	60	4

This availability vector A belongs to PM7. Its available processing capacity is 20, memory available is 50 and I/O availability is 60. It is hosting 4 VMs already. Based on this information, PM7 will be allocated to cluster C1 as PM7 has highest I/O availability and lowest processor availability (Refer table 4.1). As is evident, every VM's demand D in Q1 will be forwarded to cluster C5 for efficient placement. Similarly, VMs from Q2 will go to C3, VMs from Q3 will go to C2, from Q4 VMs are forwarded to C6, from Q5 VMs are forwarded to C1 and Q6 VMs are forwarded to C4 for placement. Forwarding a VM's demands to a designated cluster, based on resources availability, ensures efficient resource provisioning (objective (i)) and effective resource utilization (objective (v)). An important step to note here is that VMs, in each queue, are sorted as per the rental price (RP) offered, before forwarding them to their respective clusters. This step ensures that in each queue, VM which is offering the highest profit (or rental price) will be serviced first for placement. Hence, we term this placement technique as 'profit-aware' VM placement and this step ensures fulfilment of objective (iii). Figure 4.1 exhibits the system model used for the proposed placement mechanism.

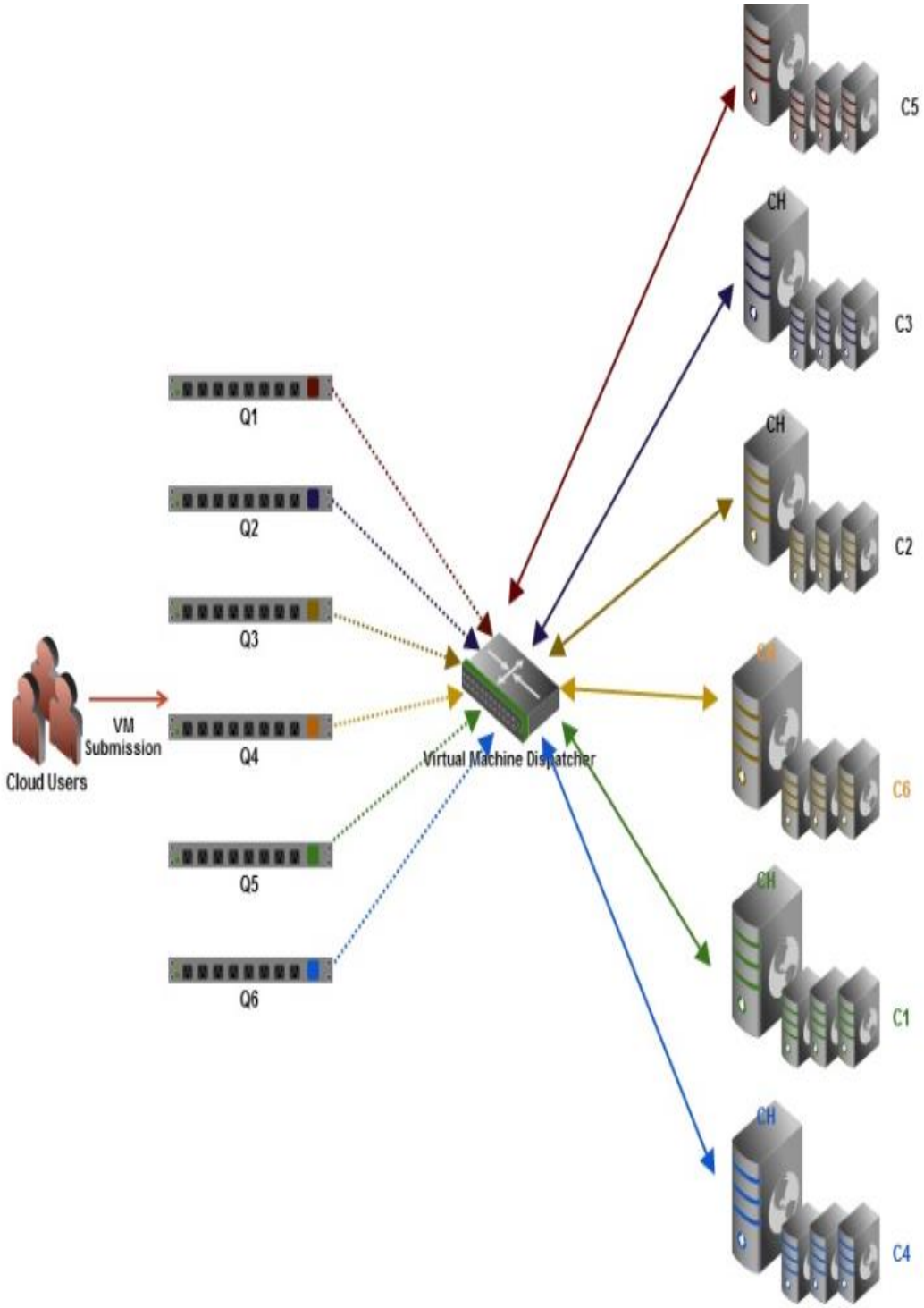

Figure 4.1 System Model of proposed VM Placement algorithm

Table 4.1 Description of Queues and clusters

Queue name	**VM's property**	**Linked Cluster**	**Cluster's property**
Q1	Highest demand- P Lowest demand- I/O	C5	Highest availability- P Lowest availability- I/O
Q2	Highest demand- P Lowest demand- M	C3	Highest availability- P Lowest availability- M
Q3	Highest demand- M Lowest demand- P	C2	Highest availability- M Lowest availability- P
Q4	Highest demand- M Lowest demand- I/O	C6	Highest availability- M Lowest availability- I/O
Q5	Highest demand- I/O Lowest demand- P	C1	Highest availability- I/O Lowest availability- P
Q6	Highest demand- I/O Lowest demand- M	C4	Highest availability- I/O Lowest availability- M

Table 4.2 (a) and 4.2 (b) describes the various parameters which are used in proposed system model, in terms of the average resources available in each cluster and average resources held in each queue respectively. VMs are dynamically allocated to queues based on the criteria explained above.

Table 4.2(a): Parameters of System Model used

Clusters	Average amount of resources available		
	P	M	I/O
C1	10	15	18
C2	7	11	8
C3	18	12	14
C4	17	17	18
C5	13	11	8
C6	10	14	9

Table 4.2(b): Parameters of System Model used

Queue	Average amount of resources held		
	P	M	I/O
Q1	8	5	2
Q2	2	2	2
Q3	2	8	4
Q4	4	9	1
Q5	6	7	8
Q6	2	1	9

Problem is defined as given a cloud datacenter with **m** heterogeneous physical machines and **n** heterogeneous virtual machines, VM placement refers to selecting a suitable PM to host an incoming VM such that the resource requirements of the VM are completely satisfied by the chosen PM. It may happen that more than one PM may meet the resources availability criteria. Hence, the proposed placement mechanism chooses a candidate PM using the multi-attribute utility theory. Demand vector D states the resources requirements (figure 4.2). As

mentioned earlier, P refers to CPU demand, M refers to memory demand, I/O refers to input-output demand and RP refers to rental price which will be paid by the cloud user once his resource demands are satisfied. On the other hand, resources in a physical machine are also represented by 2 vectors, namely, resource availability vector (A) and resource utilization vector (U) (figure 4.3 (a) and (b)). In figure 4.3 (a), PA, MA and I/OA refer to CPU availability, memory availability and I/O availability respectively. In utilization vector U shown in figure 4.3 (b), PU, MU and I/OU refer to CPU utilized, memory utilized, and I/O utilized respectively.

Objective- Our objectives, behind proposing a new VM placement technique, are-

i. **Efficient Resource Provisioning**
ii. **Dynamic Load Balancing**
iii. **Profit guarantee to the cloud provider**
iv. **Reduced VM Migrations**
v. **Effective Resource Utilization**
vi. **Faster Placement of VMs or reduced placement time**

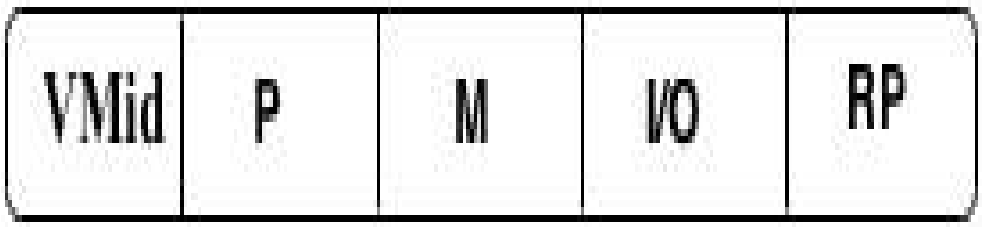

Figure 4.2 Demand Vector, D, of an incoming VM

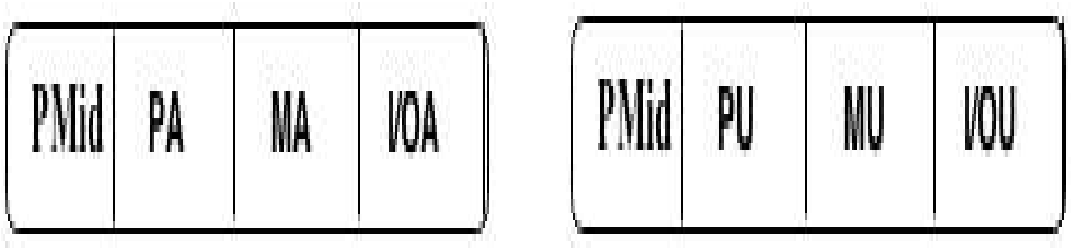

Figure 4.3 (a) Resource Availability Vector, A, 4.3 (b) Resource Utilization Vector, U

Profitability-aware VM placement starts with cloud users who submit their VMs' demands (D) in one of the designated queues dynamically. Every queue is then sorted in decreasing order of the rental price offered in the demand vector. An entity called virtual machine dispatcher (VMD) fetches the front VM's D vector from each queue and forwards it to its designated cluster. Every cluster has a nominated cluster-head (CH) which receives the incoming VM's D vector and compares it with the resource availability vectors (A) of all the physical machines in the cluster. These availability vectors are updated by PMs and submitted to the cluster-head periodically. All the PMs, whose availability vectors (A) satisfy the incoming VM's demand vector (D), are inserted in a candidate queue (CQ). CQ represents feasible PMs who can provide the service to the incoming VM comfortably. Now, the placement task is reduced to finding out the best possible PM among the chosen candidate PMs who can create the VM without unbalancing the cluster's load. For this, the average load density (ALD) of a cluster is calculated after considering that the incoming VM is placed in one of the candidate PMs. Likewise, the machine load density (MLD) of every candidate PM is also calculated after assuming that the incoming VM is placed on that PM. The method for calculating ALD and MLD is specified below.

$$ALD(of\ a\ cluster) = \frac{Total\ VMs\ Placed}{Total\ number\ of\ PMs\ in\ the\ cluster}$$

$$MLD(PM_i) = \frac{resources\ consumed\ on\ PM_i\ after\ VM\ placement}{resources\ available\ on PM_i\ after VM\ placement}$$

A candidate PM, whose calculated MLD is closest to its cluster's ALD after placement, is chosen as the host for the incoming VM. Comparing MLD of candidate PMs with the cluster's ALD ensures that placing VM on a host does not imbalance the overall load of the cluster. This step satisfies objective (ii) as mentioned above. Given below are the algorithms used in our proposed placement scheme.

Algorithm 1: VM's demand vector (D) submission mechanism in a queue

```
1. for each (incoming VM's demand vector D)
  {  do {if(max(P,M,I/O)==P&&min(P,M,I/O)== I/O)
        enqueue(Q1, D)
     else if(max(P,M,I/O)==P&&min(P,M,I/O)== M)
          enqueue(Q2, D)
     else if(max(P,M,I/O)==M&&min(P,M,I/O)== P)
          enqueue(Q3, D)
     else if(max(P,M,I/O)==M&&min(P,M,I/O)== I/O)
          enqueue(Q4, D)
     else if(max(P,M,I/O)==I/O&&min(P,M,I/O)== P)
          enqueue(Q5, D)
     else enqueue(Q6, D)
         }
```

```
    }
2. for each queue, sort in descending order of rental
   price, RP, offered in D.
```

Algorithm 2: VM forwarding mechanism performed by VMD. VMD retrieves the front VM's D from each queue and forwards it to its designated cluster as

```
1. if(dequeue(Q1,D))
    forward D to C5
    else if(dequeue(Q2,D))
          forward D to C3
    else if(dequeue(Q3,D))
      forward D to C2
    else if(dequeue(Q4,D))
          forward D to C6
    else if(dequeue(Q5,D))
        forward D to C1
           else forward D to C4
```

Algorithm 3: Selection of Candidate PMs in each cluster and subsequent VM placement, performed by CH

```
1. Calculate ALD (of each cluster) = Total Resources
   Utilized/Total Resources Available
2. foreach(incoming VM's demand D)
   do {foreach(PM's availability vector A)
   do   {if(P<=PA&&M<=MA&&I/O<=I/OA)
```

```
    enqueue (CQ, PM)
          else discard PM   }      }
3. if(CQ has more than 1 PM)
     {foreach(PM in CQ)
   do {/*Calculate MLD of PM after assuming VM
placement*/
   MLD(PM)=resources consumed/
   resources available  }
          if(|ALD-MLD(PMi)|<|ALD-MLD(PMj)|)
            VM is placed on PMi
          else          VM is placed on PMj
         /*i and j are subscripts for    candidate PMs
in CQ*/      }
      elseif(CQ has only one PM )
          place incoming VM on this PM
       else discard VM/*indicating no suitable PM for
placement in the cluster*/
```

Figure 4.4 below shows the time-line diagram of the above explained placement procedure.

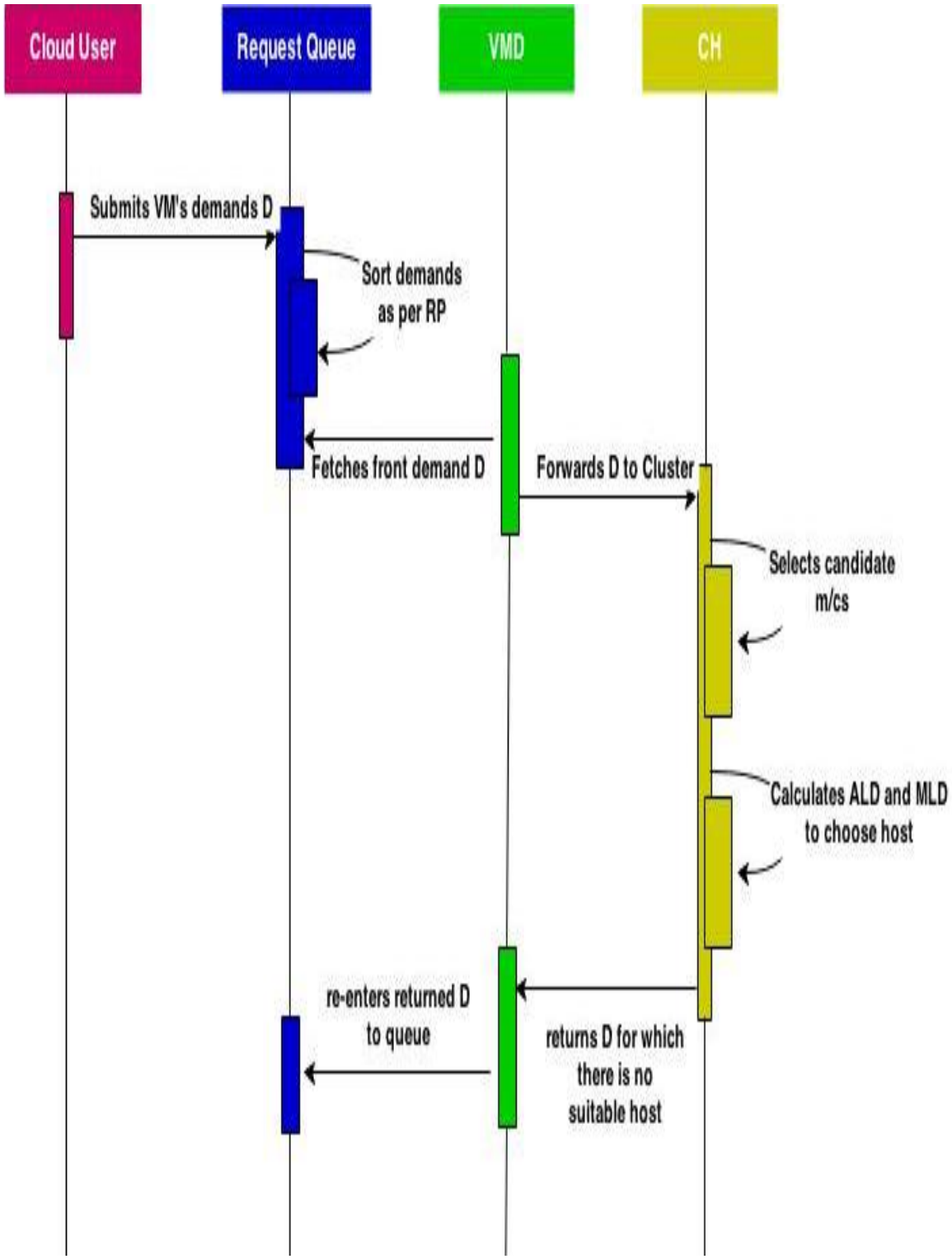

Figure 4.4 Timeline of the proposed VM placement technique

4.3. LOAD-AWARE VM PLACEMENT

Here, we propose a 2-phase load balancing technique and implemented at VM level. In the first phase, we apply Analytical Hierarchy Process (AHP) to place an incoming VM at the best possible host PM. This phase is termed as Load-Aware VM Placement. In the second phase, we use migration to ease the VM load of an over-burdened PM with the aim of energy-conservation and better user experience. Proposed technique considers heterogeneous VMs with multiple resource types and is implemented online in a dynamic fashion.

During load-aware VM placement, an incoming VM is forwarded to the lightest loaded cluster. Member PMs of this chosen cluster, whose resource availability is higher than the incoming VM's demand, are identified to form the solution space. This solution space is further reduced to contain only those PMs which are compatible with the type of incoming VM. After this reduction, multi-criteria based analytical hierarchy process (AHP) is applied to the solution space and the best PM to host an incoming VM is selected. A detailed step-by-step procedure of load-aware VM placement is given below-

a. As stated in the system architecture, all PMs are grouped into clusters based on their physical connectivity with the access level switches. An inter-cluster module present in the data center maintains a dynamic list of these clusters in non-decreasing order of their mean load as shown in figure 4.3. Mean Load of a cluster is calculated as

$$ML_{cl} = \frac{1}{m}\sum_{r=1}^{m} L(PM_r)$$

An incoming VM_g is forwarded to the first cluster in the list i.e. cluster with the minimum ML_c.

b. An intra-cluster module, present in every cluster, compares the VM_g's demands with the availability of each member PM_r and builds up the solution space as

$$SS = \forall PM_r : A(PM_r) \rangle D(VM_g)$$

Member PMs with availability less than the VM's demand are removed from the solution space. Now we consider PMs whose availability is greater than the demand.

c. Solution space is further reduced by considering PMs whose load characteristic is compatible with the VM type. For simplicity, two types of VMs are considered, namely CPU intensive and memory intensive. If the incoming VM is memory intensive (t2) then PMs with least memory load are considered for placement. Likewise, for CPU-intensive VM (t1), PMs with least CPU load are considered ideal host candidates. The directional cosines of VM_g will determine the type of VM as given below-

If $\cos \alpha > \cos \beta$ then VM_g is type t1 else type t2. Similarly, if $\cos l_1 > \cos l_2$, then PM_r has least memory load else PM_r has least CPU load.

d. Now, we apply AHP to the remaining PMs in the solution space as shown below in figure 4.5.

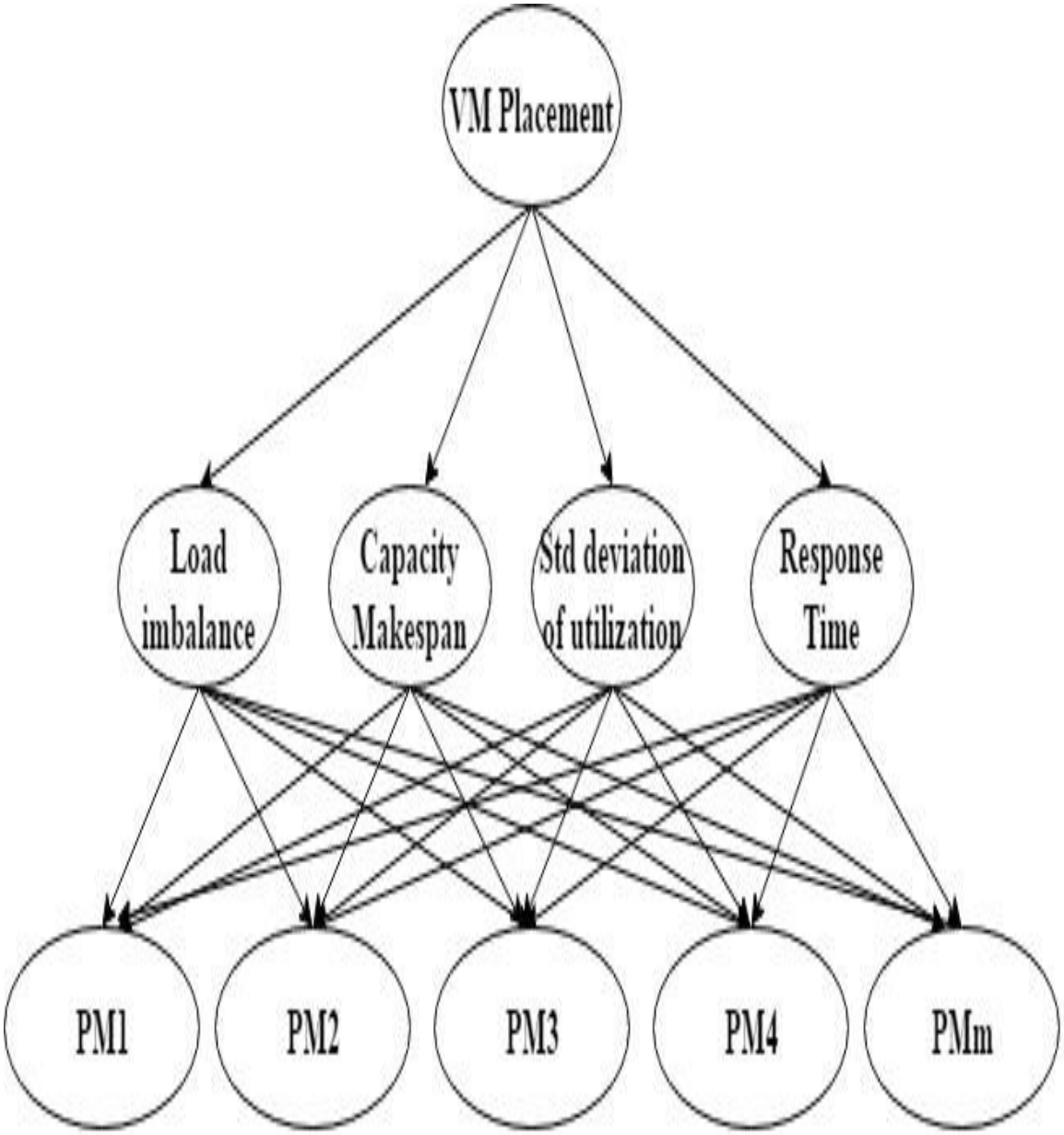

Figure 4.5 An Analytical Hierarchy VM Placement Process

The core concept of AHP involves pair-wise comparison of alternatives for each criterion and then deducing their overall rankings. A similar concept is used in the proposed technique where VM placement is the goal. We have considered four post placement metrics to select the best host PM from the solution space for incoming VM_g. These metrics are load-centric SLA parameters and the aim of the proposed work is to honor their values in order to prevent any SLA violations. Candidate PMs in the solution space left after execution of step 3 are considered as alternatives for placement. We assume there are m such hosts. The value of each post placement metric is calculated by assuming that VM_g has been placed at PM_r. The description of these four metrics is given below-

Load Imbalance – Load imbalance of a PM_r after placing VM_g is the difference between its new load (after placing VM_g) and the mean load of its cluster. If the load of a candidate PM_r before placement was

$$L(PM_r) = l_1 i + l_2 j + l_3 k$$

And the incoming VM_g resource demand is

$$D(VM_g) = \alpha i + \beta j + \gamma k$$

Then the new load of PM_r after placing VM_g is

$$L_{new}(PM_r) = (l_1 + \alpha) i + (l_2 + \beta) j + (l_3 + \gamma) k$$

And the load imbalance (LI) of PM_r will be

$$LI(PM_r) = L_{new}(PM_r) - ML_{cl}$$

Like-wise, Load imbalance of each PM present in the solution space will be calculated assuming that the incoming VM is placed on it. The minimum the load imbalance value, the better the placement.

Capacity Makespan- this metric combines the total demands of all VMs hosted on a PM with their execution times, i.e.

$$CM(PM_r) = \sum_{j=1}^{v} D(VM_j) * t(VM_j)$$

Total number of VMs hosted on a PM before placement of VM_g is represented by v. Now assume that VM_g is placed on PM_r then the new capacity makespan will be

$$CM_{new}(PM_r) = CM(PM_r) + \lfloor D(VM_g) * t(VM_g) \rfloor$$

Capacity makespan for each PM is calculated considering VM_g is placed on it. Here again, a minimum capacity makespan value is desired.

Standard Deviation of Utilization- For each PM can be calculated as the square root of its load imbalance

$$SDU = \sqrt{LI(PM_r)}$$

Standard deviation should be minimum for optimal placement.

Average Response Time – Time taken by VM_g to give its first response will vary from PM to PM depending on the already present load of that PM. Hence, this value will be calculated for each PM considering VM_g is placed on it.

$$ART(PM_r) = \frac{1}{v} \sum_{j=1}^{v} RT(VM_j)$$

The present load-aware VM placement can now be seen as-

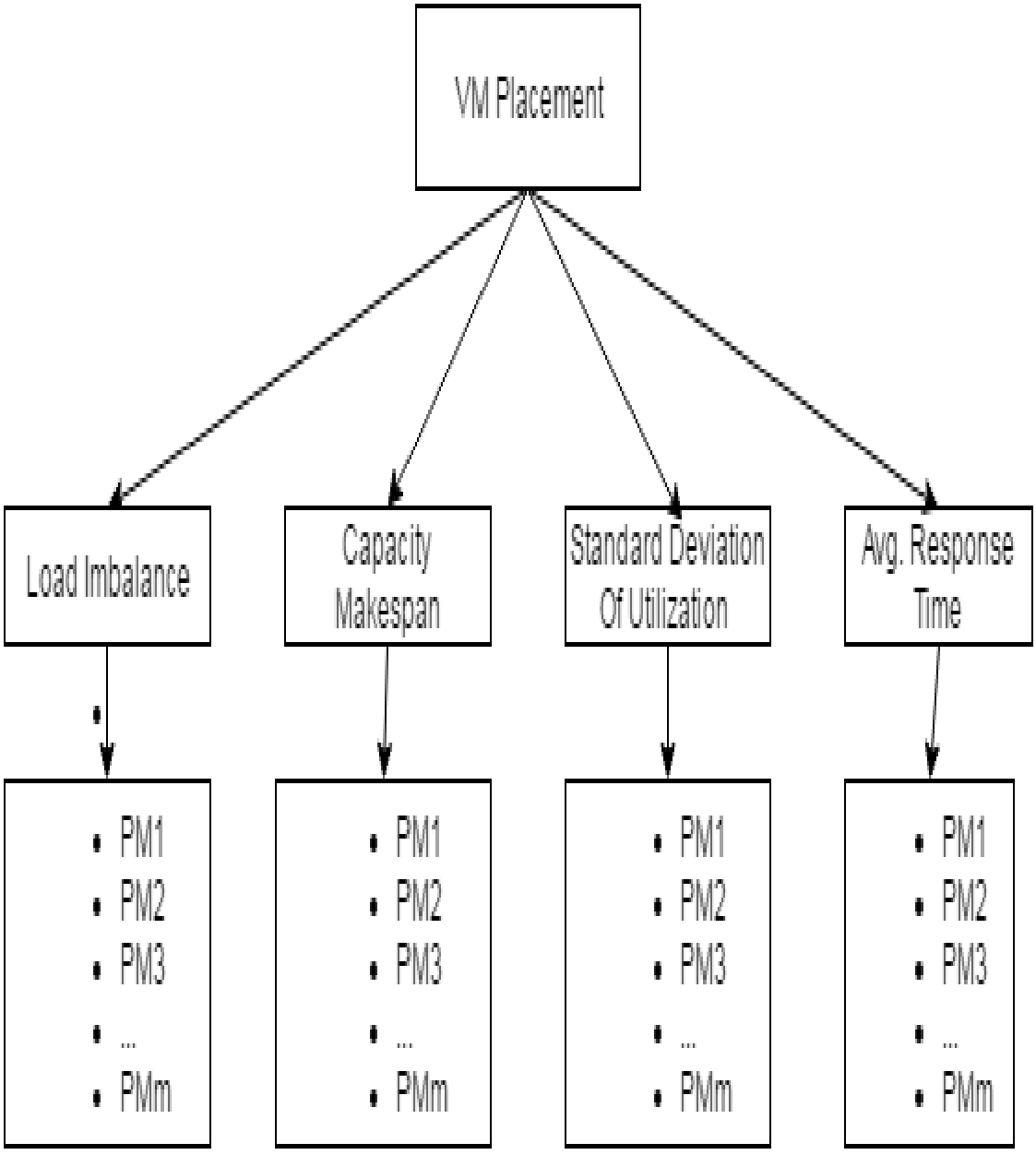

Figure 4.6 An Example of AHP based VM Placement

Let's understand the applied technique in the following steps using an example.

i. The first step is to prioritize the post placement criteria using a square matrix, called Reciprocal Matrix (RM), as shown in figure 4.7.

$$RM(k) = \begin{bmatrix} 1 & rm_{12} & \cdots & rm_{1k} \\ rm_{21} & 1 & \cdots & rm_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ rm_{k1} & rm_{k2} & \cdots & 1 \end{bmatrix}$$

Figure 4.7 Reciprocal Matrix

Values in the matrix are provided by the cloud service provider and are in the range of 1 to 9 with the following meanings-

1	3	5	7	9
Equal	Weakly favoured	Mild favoured	Strong favoured	Extreme favoured

Values 2, 4, 6 and 8 are intermediate values. If rm_{ij} value is p in the RM matrix then the value of rm_{ji} will be 1/p (refer figure 4.7). Next, we compute eigen vector of RM[4] to obtain the ranking of post placement metrics as shown below in figure 4.8.

$$\begin{array}{c} \\ LI \\ CM \\ SDU \\ RT \end{array} \begin{array}{c} \begin{array}{cccc} LI & CM & SDU & RT \end{array} \\ \begin{bmatrix} 1 & 4 & 3 & 5 \\ \frac{1}{4} & 1 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 3 & 1 & \frac{1}{2} \\ \frac{1}{5} & 3 & 2 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.8983 \\ 0.1215 \\ 0.2593 \\ 0.3306 \end{bmatrix}$$

Figure 4.8 The Reciprocal Matrix

ii. Now, we construct four square matrices each for one post placement metric and all the alternatives. In the example, we have taken 5 alternatives or candidate PMs, with ids, PM_5, PM_7, PM_{12}, PM_{15}, PM_{19} and the matrices are shown below in figure 4.9.

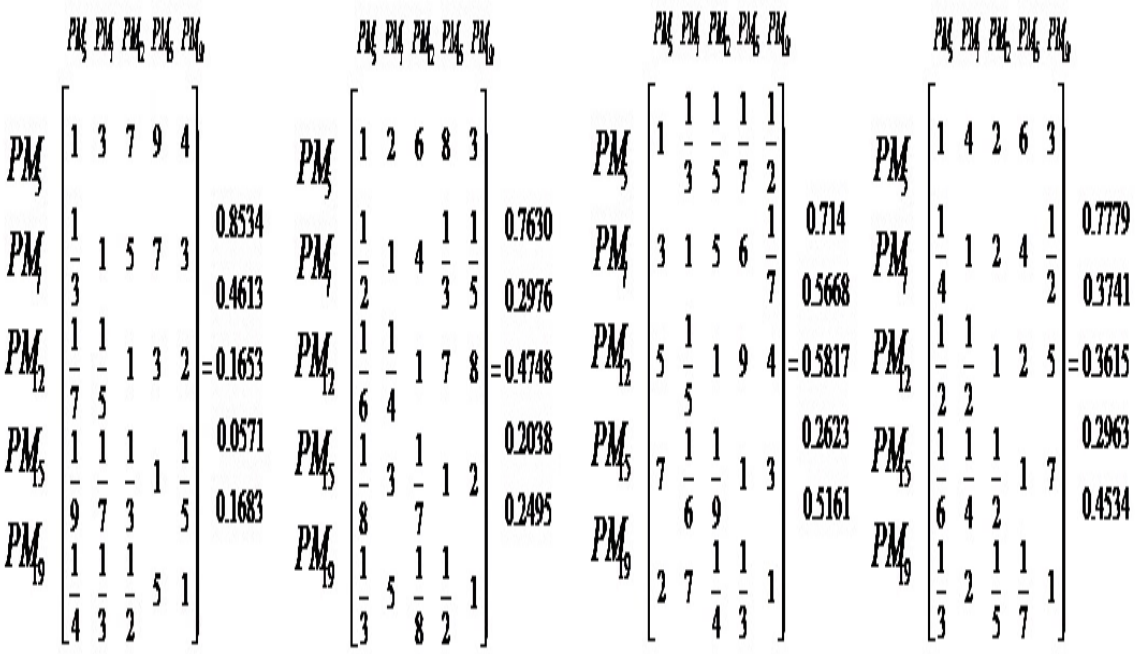

Figure 4.9 Candidate PMs rankings for each post placement metrics

Values in the load imbalance matrix are the values of post placement load imbalance LI for each candidate PM. Then we compute eigen vector of LB[5]. Similarly, we construct CM[5], SDU[5], RT[5] and compute eigen vectors of each as shown in figure 4.9. This step gives the ranking of each alternative PM for each post placement criteria.

iii. In the last step, eigen vector calculated in step i is multiplied by a square matrix of eigen vectors calculated in step (ii). Their product will give the final ranking of all the candidate/alternate PMs as shown in figure 4.10.

$$\begin{bmatrix} 0.8534 & 0.7630 & 0.7140 & 0.7779 \\ 0.4613 & 0.2976 & 0.5668 & 0.3741 \\ 0.1653 & 0.4748 & 0.5817 & 0.3615 \\ 0.0571 & 0.2038 & 0.2623 & 0.2963 \\ 0.1683 & 0.2495 & 0.5161 & 0.4534 \end{bmatrix} * \begin{bmatrix} 0.8983 \\ 0.1285 \\ 0.2593 \\ 0.3306 \end{bmatrix} = \begin{bmatrix} 1.3070 \\ 0.7233 \\ 0.4798 \\ 0.2435 \\ 0.4670 \end{bmatrix}$$

Figure 4.10 Final Ranking of candidate PMs

4.4. SIMULATION RESULTS

CloudSim 3.0.3 is used to simulate the proposed energy efficient load-aware VM placement technique. The simulation setup consists of a cloud data center with 100 PMs and 150 VMs and the performance is evaluated. In the following figures from 10 to 15, these post placement metrics are shown graphically for 5 candidate PMs namely, PM_5, PM_7, PM_{12}, PM_{15} and PM_{19}. These PMs surpassed the availability and type testing discussed in steps I to III of the proposed placement technique.

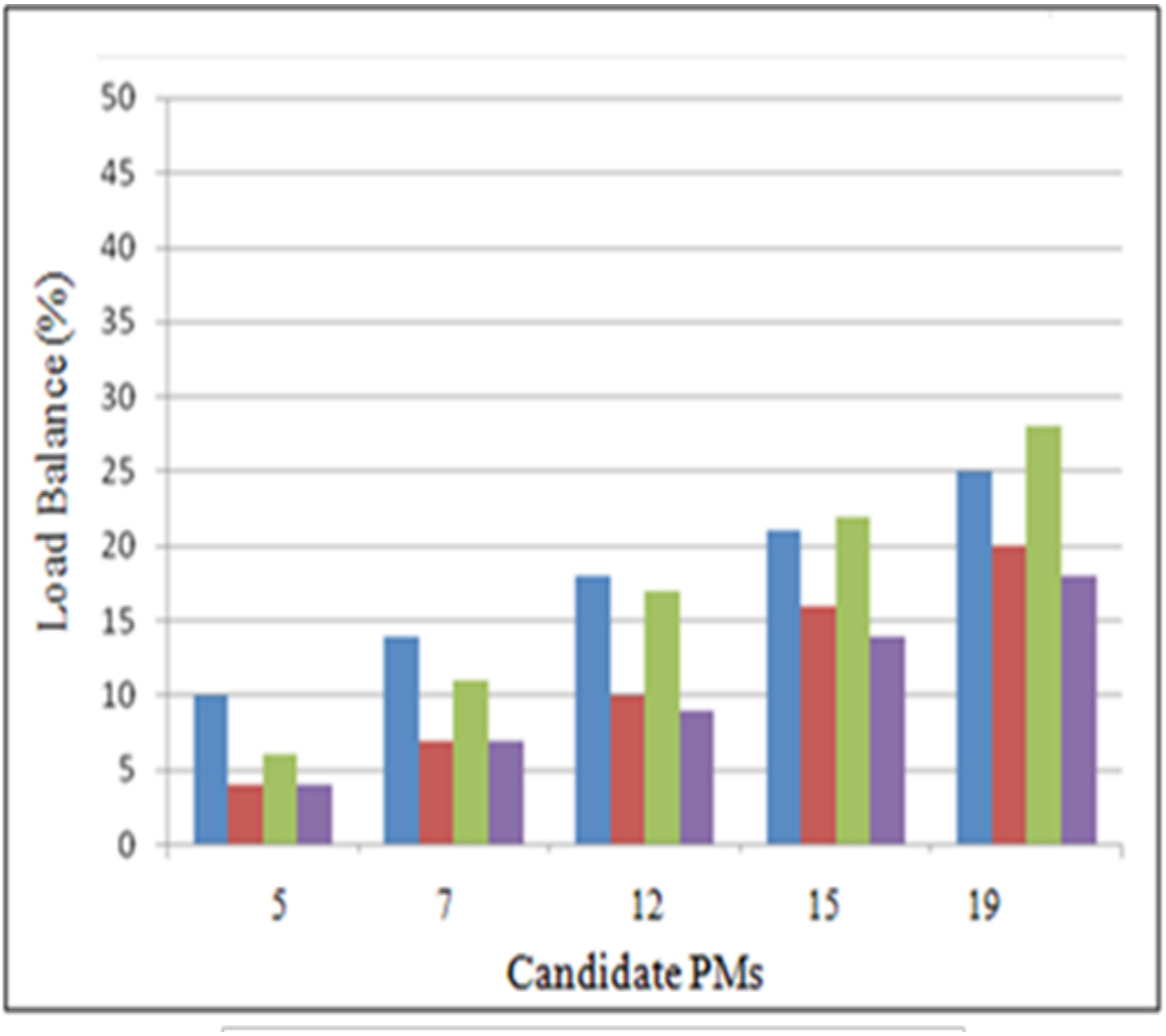

Figure 4.1 1 Load Balance of candidate PMs

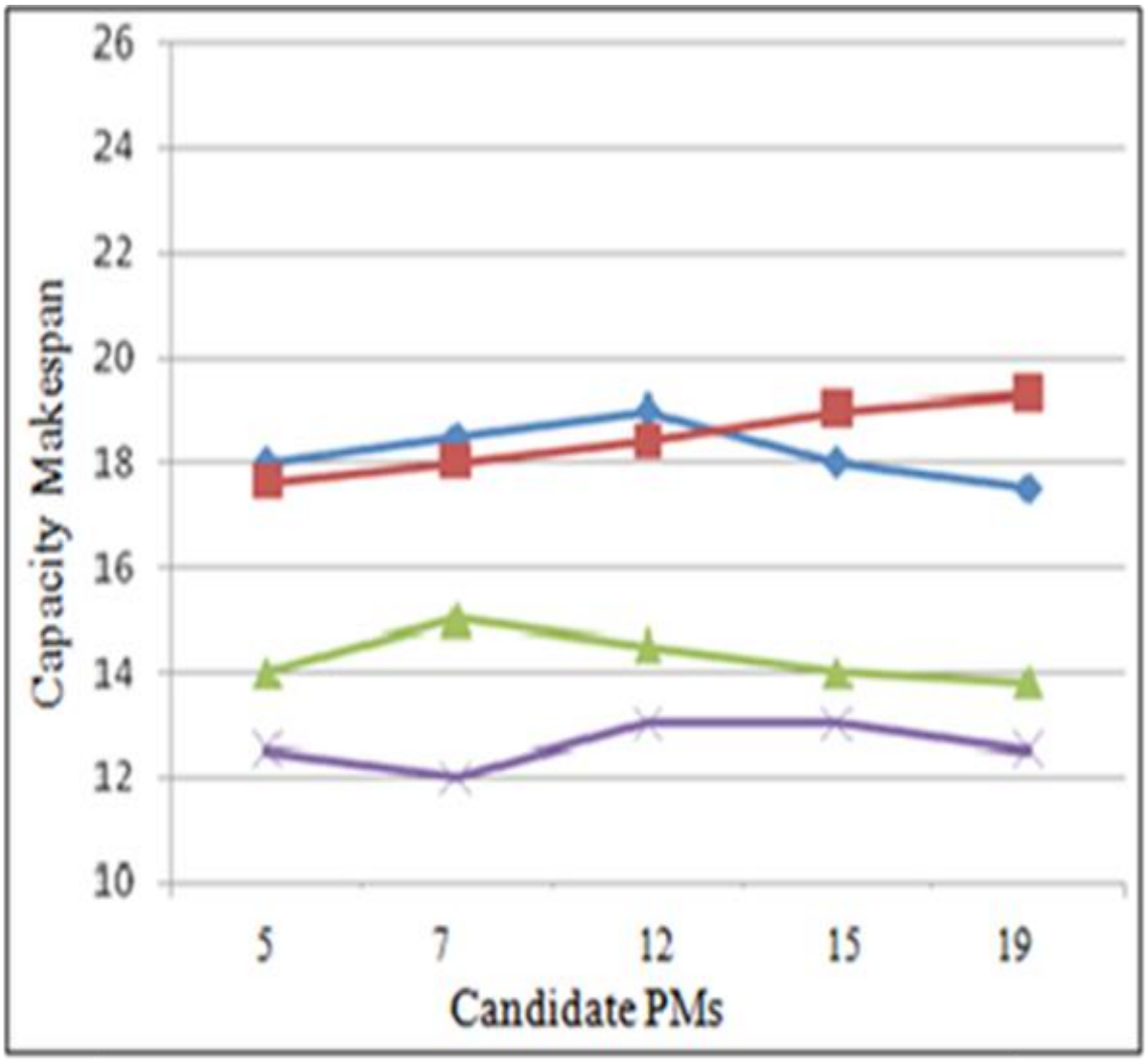

Figure 4.12 Capacity Makespan of Candidate PMs

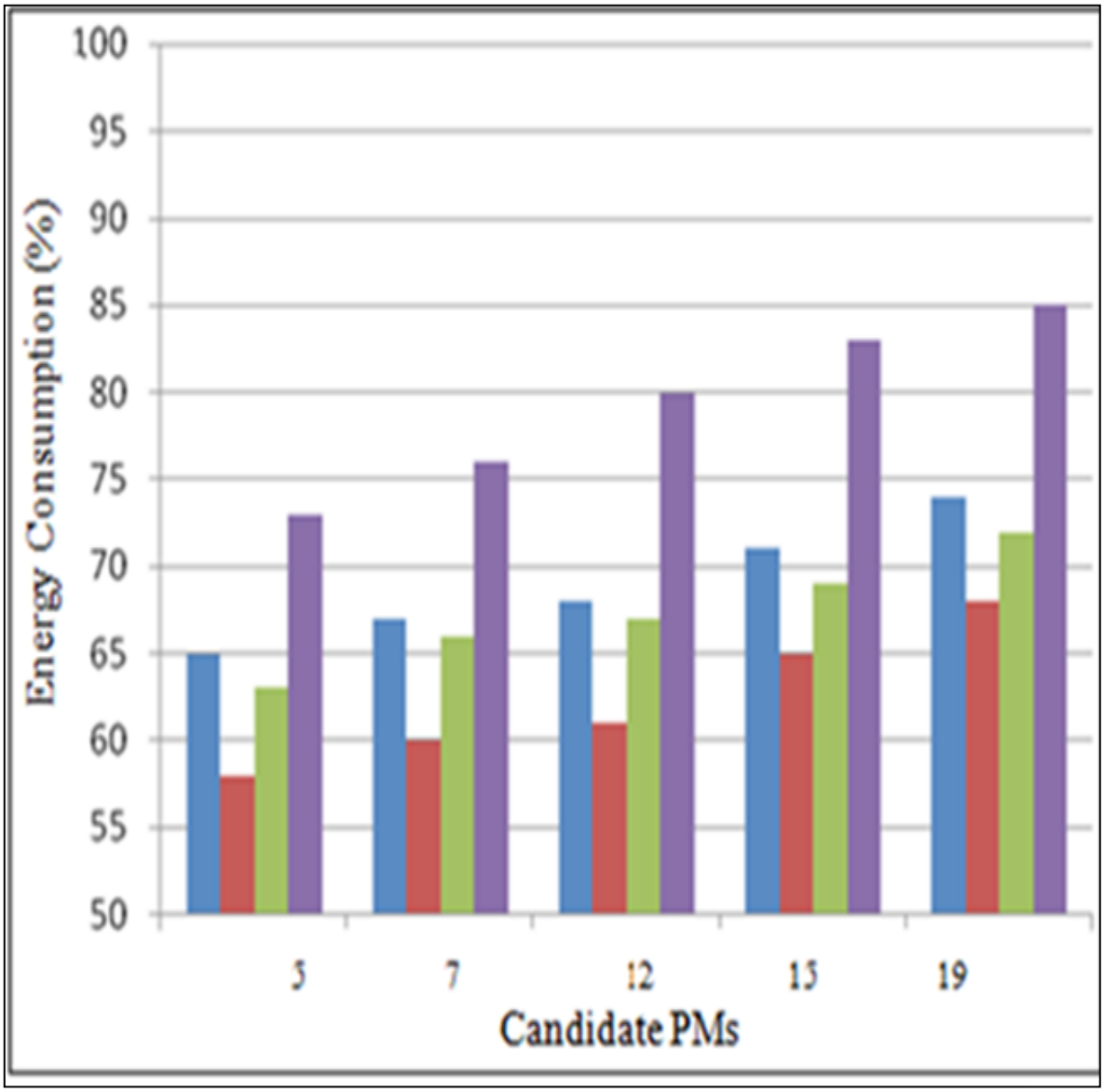

Figure 4.13 Energy consumption of Candidate PMs

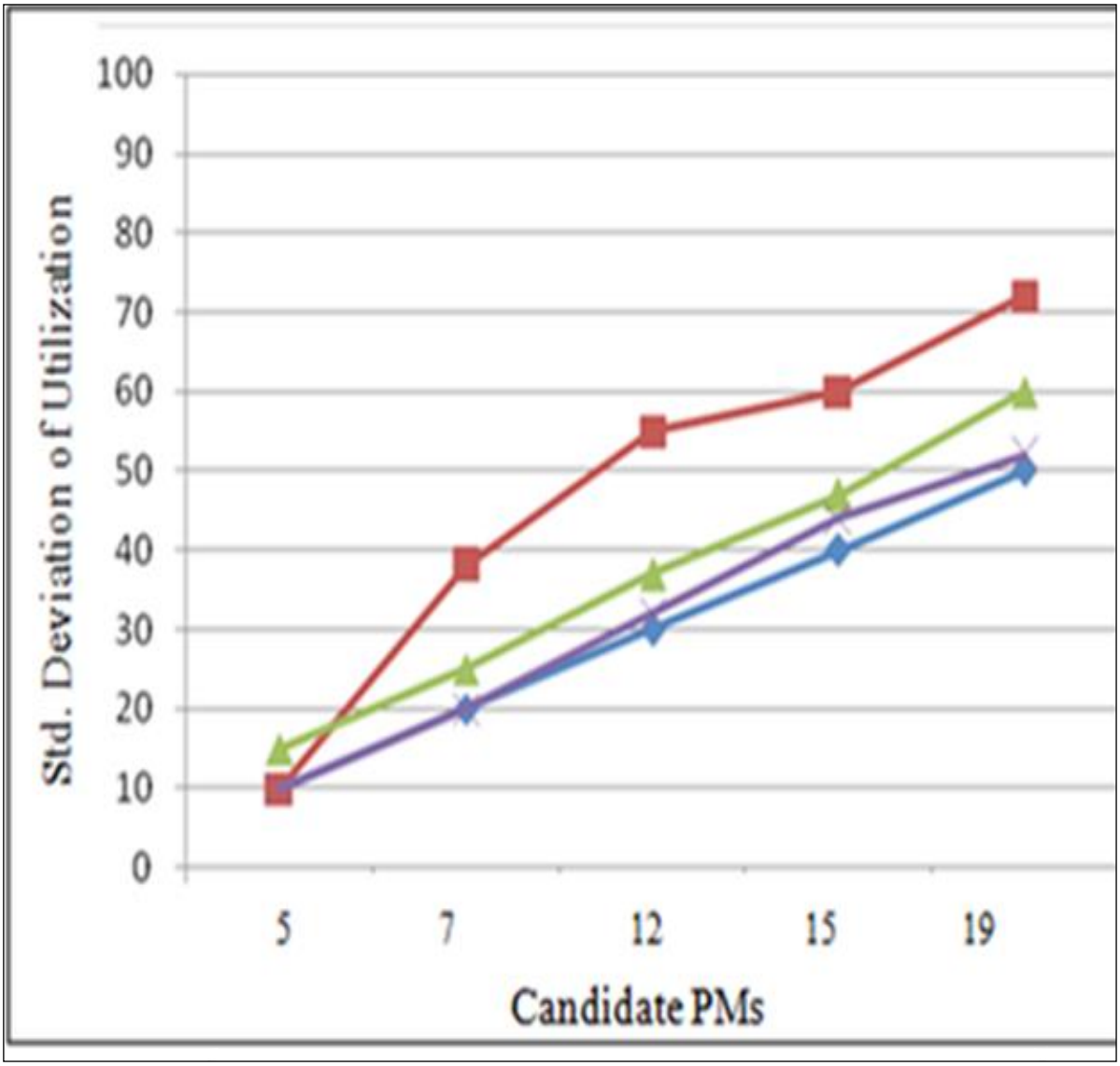

Figure 4.14 Utilization Deviation of Candidate PMs

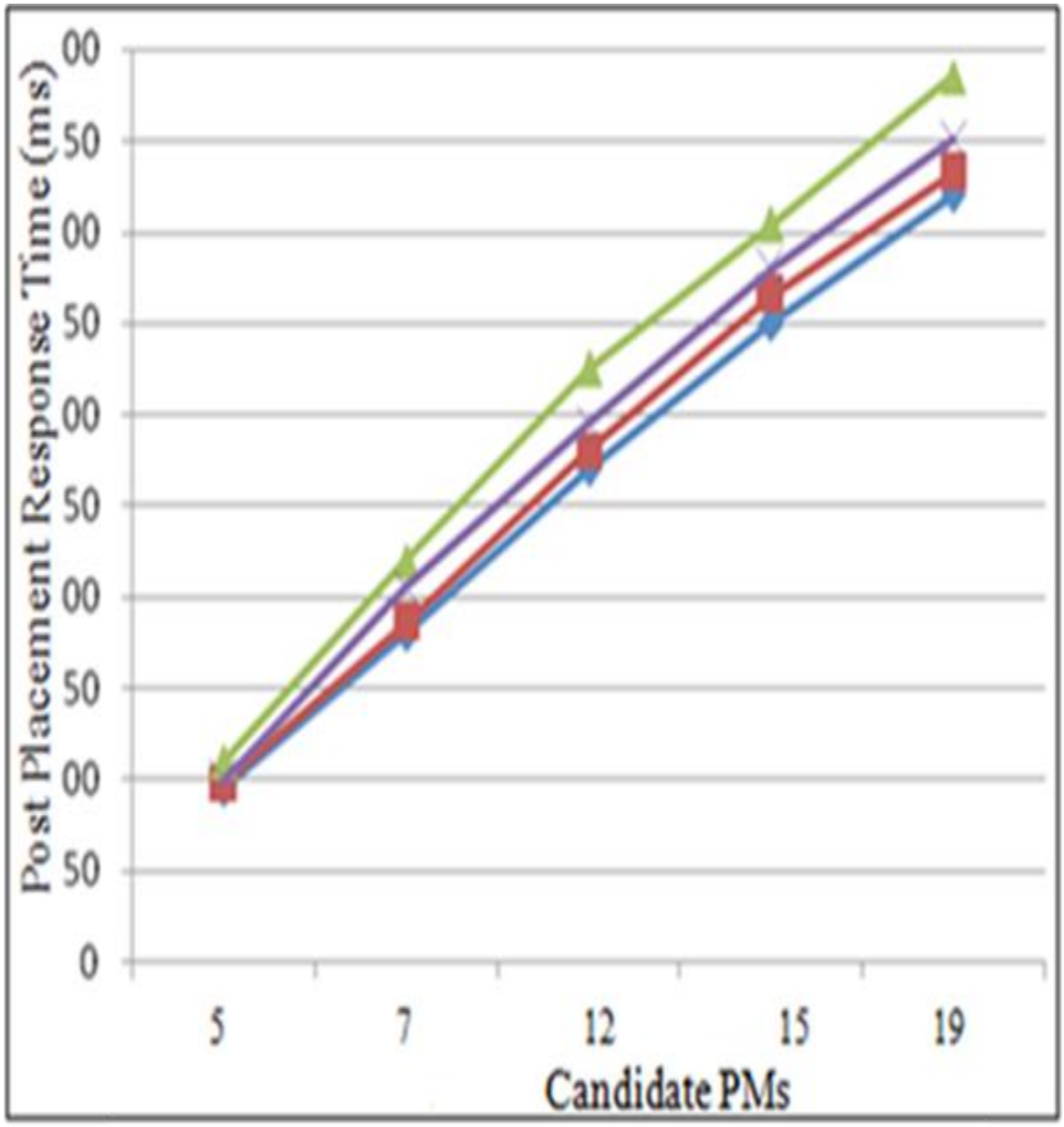

Figure 4.15 Response Time of VMs on PMs

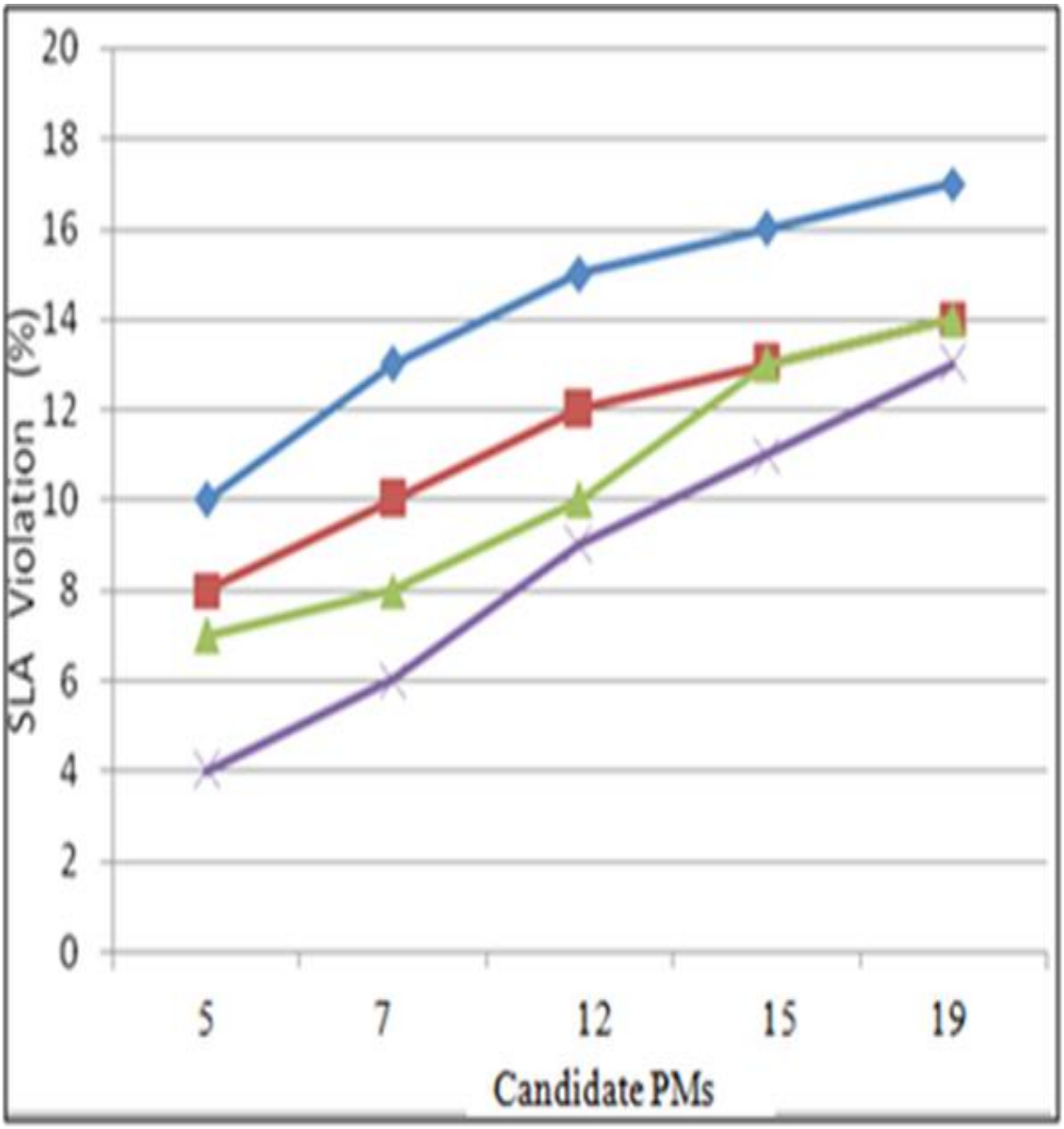

Figure 4.16 SLA Violations on Candidate PMs

It is quite clear from the results that candidate PM_5 scores the highest in terms of post-placement evaluation, hence it is chosen the host for an incoming VM. As the load in the data center reaches a stable state, instances of migration reduce and therefore energy consumption also stabilizes. This is given in figure 4.13. An optimum utilized host gives its best performance which leads to a reduction in overall SLA violations, same is shown in figure 4.16.

Simulation results indicate that our proposed placement technique is promising in terms of stable energy consumption, fewer performance violations and guarantees an optimal resource utility.

CHAPTER V

A Profitable Resource Allocation in Cloud Data Centers

The content of this chapter is published in-

Elsevier Procedia Computer Science Journal, vol. 57, pp. 104-111, 2015, ISSN: 1877 - 0509.

CHAPTER V

A PROFITABLE RESOURCE ALLOCATION IN CLOUD DATA CENTERS

5.1 INTRODUCTION

This chapter presents an auction style resource distribution algorithm in cloud computing environment which guarantees users' service preferences while allocating resources and ensures that cloud service provider reaps the maximum monetary benefit by choosing users with the highest bidding capacity.

Cloud computing offers the convenience of 'anywhere', 'anytime' acquisition of computing resources in a suitable and demand-based style. It aims for efficient sharing of resources to boost its performance. Cloud offers 'utility computing' where computing services are moulded as commercial utilities and users need to rent/pay for their requirements. Cloud resources are limited in nature; hence their management needs to be efficient and optimum. These resources are enveloped in the form of virtual machines (VMs) and catering a conglomerate of diverse and concurrent resource demands, a cloud service provider must practice an efficient and profitable resource allocation technique. Resource allocation in cloud infrastructure refers to efficient distribution of VMs among various users as per their requirements in a manner such that cloud provider's profits are maximized and also users' 'quality criteria' are fulfilled. This resources distribution is based on many factors, like availability of VMs against their requirements, prices offered for a requested VM, time span for which a VM is claimed for, the effect of VM allocation on a data centre's performance etc. Resource allocation can be online (dynamic) or offline (static). Keeping in view the wavering demands of users, online resource allocation

techniques are preferred over static ones. Here, users bid for resources as per their requirements and service provider chooses a user based on their bids so that cloud provider's profit is ensured.

5.2 CHALLENGES IN EXISTING RESOURCE ALLOCATION SCHEMES

Various cloud computing researchers have proposed many resource allocation strategies which either complement each other or are completely independent. These techniques are based on certain factors and impact one or more performance aspects of cloud computing. Some allocation techniques are reservation-based and follow prediction of users' demands. Such techniques may fail in accurate prediction and may distribute resources in an uneven fashion. Strategies based on hardware dependency, price as utility and live VM migrations limit in number and type of resources considerations. Allocation techniques based on simulated annealing algorithm and game theory do not take into account the dynamic nature of clients' requests. Outlined below are certain crucial factors which confront any resource allocation technique and must be taken into account during implementation.

i. Uneven flow of users' service requests resulting in unpredictable peak and through traffic scenarios.
ii. Heterogeneous nature of cloud's resources and their limited instances.
iii. Dynamic character of a task's resource demands.
iv. Interference among new and existing VMs.
v. Adherence to QoS and various SLA parameters.
vi. Optimum utilization of cloud's resources while maintaining certain performance standards.

5.3 SYSTEM MODEL

Here, we consider N cloud users for one cloud service provider. The cloud provider allocates VMs to users as per their payment capacities and service level negotiations. The different types of VMs considered in the allocation strategy are communication-driven (K1), processing-driven (K2) and storage-driven (K3) VMs. The cost factors for the service provider during resource allocation are virtual machine initiation cost (IC_{VMtype}), task-processing cost (TPC), data-transfer cost (DTC) and storage cost of data (DSC). Additional cost (AC) is the sum of TPC, DTC and DSC. As the auction starts, users declare their bid-prices of individual VM instances (BP_{VMtype}) and arrival time (AT). Table 5.1 lists the characters used.

Table 5.1: Summary of the characters used with their meanings

Char	Meaning	Char	Meaning
N	No. of Cloud Users shown as n_1, n_2 ….n_N	SP_{K1}	Start Bid Price of K1
K	VM types(K1, K2, K3)	SP_{K2}	Start Bid Price of K2
BP_{K1}	Best-Bid price of K1	SP_{K3}	Start Bid Price of K3
BP_{K2}	Best-Bid price of K2	R_{K1}	Number of K1 samples
BP_{K3}	Best-Bid price for K3	R_{K2}	Number of K2 samples
RAU	Resource Allocation Unit	R_{K3}	Number of K3 samples
IC_{K1}	Initiation Cost (K1)	DSC	Storage Cost of data
IC_{K2}	Initiation Cost (K2)	DTC	Transfer Cost of data
IC_{K3}	Initiation Cost (K3)	AC	Additional Cost
TPC	Processing Cost of a task	TBP	Total Bid Price
AT	Arrival Time of a task	BPP	BidPrice Payment
T	Total duration of an allocation round		

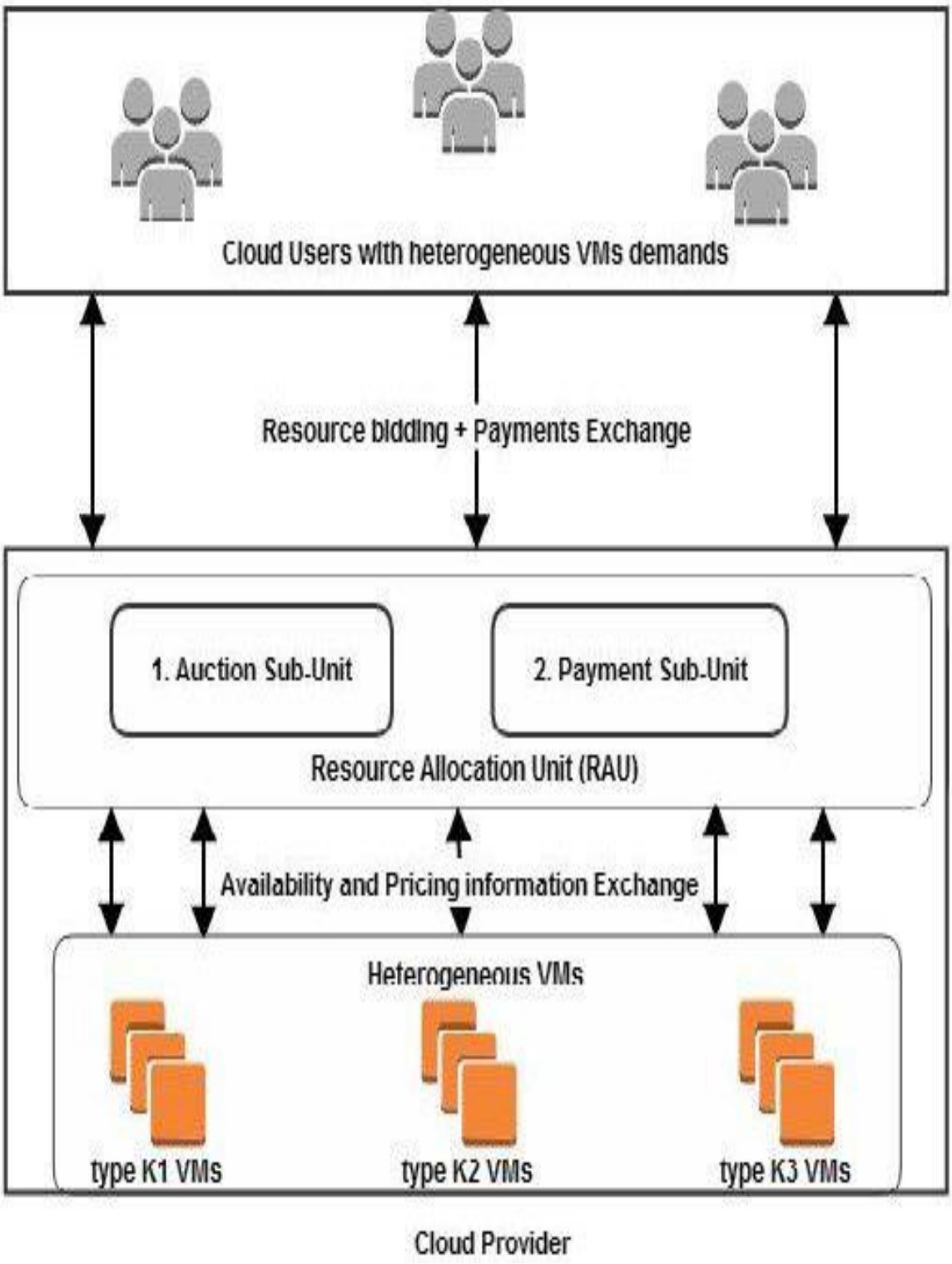

Figure 5.1 Resource Allocation System Model

5.4 RESOURCE ALLOCATION SCHEME

The proposed resource allocation technique is a two-step process, pre-auction and open auction.

5.4.1 Pre-auction- The starting bidding price of each VM type is also its reserve price and is given as

$$\left(SP_{VMtype}\right)_i = \left(MeanBP_{VMtype}\right)_{i-1} \qquad \ldots (5.1)$$

The value of $(SP_{VMtype})_i$ is either SP_{K1}, SP_{K2} or $SP_{K3}$for round i of auction. Same criteria is for (Mean $BP_{VMtype})_{i-1.}$

A resource's bid-price shows its current demand as shown in the equation (5.1). This is done to follow the market's current supply-demand scenario and it also justifies a resource's utility. The additional cost incurred by a service provider during resource allocation is given in equation (5.2)-

$$AC = DSC + DTC + TPC + IC_{VMtype} \qquad \ldots (5.2)$$

5.4.2 Market-driven Open Auction- The nest step in fair distribution of resources is an open auction process which takes place in two parts. Starting auction price of each resource is determined using the criteria

$$BP_{VMtype} \geq SP_{VMtype}$$

Bid Price of a user is his maximum payment capacity. Bidders submit resource requirements and with bid prices for VMs as shown in figure 5.2.

$$TBP = BP_{K1} + BP_{K2} + BP_{K3}$$

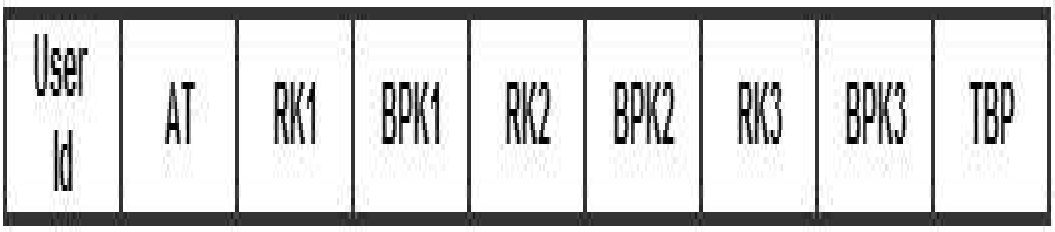

Figure 5.2 User's bid format

At auction time t = T/2, a service provider calculates the mean TBP and the meanBP for each VM type as shown in equation (5.3) and (5.4).

$$MeanTBP = \frac{\sum_{i=1}^{x} TBP_i}{x} \quad \ldots (5.3)$$

$$MeanBP = \frac{\sum_{i=1}^{x} \left(BP_{VMtype}\right)_i}{x} \quad \ldots (5.4)$$

x are the total bidders from time 0 to T/2. Winners are chosen whose TBP>= MeanTBP.

Rejected users have the choice to change their bid prices as per the criteria BP_{VMtype}>= $MeanBP_{VMtype}$ and can rebid for the second part of the auction. At t=T, service provider selects the winner again. As again, bidders with TBP>=MeanTBP are declared winners. Both the winners-lists obtained from the first and the second half of

the auction are merged together. Final list is arranged in non-increasing order of their TBPs. This increases profitability of the service provider fairness to users.

5.5 PREFERENCE-DRIVEN PAYMENT

The payments made by winners after the auction round are calculated based on their chosen preferences as shown in table 5.2.

Table 5.2: Preferences Table

Task Deadline Option	Service Time Option	VMs Possession Option
D1: Fixed	S1: Immediate	P1: Full-time
D2: Flexible	S2: Flexible	P2: Partial

Each winner is given an opportunity to choose their task's deadline option, service time choice and possession alternatives. A winner's chosen preference will determine his actual payment (AP) value, as outlined in Table 5.3.

Table 5.3: Payment Table

Preferences	Actual Payment Criteria	Actual Payment (AP) Calculation by a winner i
D1S1P1	BP of the paying winner (i) + AC	$AP_i = \left[(RK1 * BPK1)_i + (RK2 * BPK2)_i + (RK3 * BPK3)_i\right] + AC$
D1S1P2	BP of the paying winner (i)	$AP_i = \left[(RK1 * BPK1)_i + (RK2 * BPK2)_i + (RK3 * BPK3)_i\right]$
D2S1P1	BP of the next winner (p) in the list	$AP_i = \left[(RK1_i * BPK1_p) + (RK2_i * BPK2_p) + (RK3_i * BPK3_p)\right]$
D2S1P2	BP of 2nd next winner (q) in the list	$AP_i = \left[(RK1_i * BPK1_q) + (RK2_i * BPK2_q) + (RK3_i * BPK3_q)\right]$
D2S2P1	BP of last winner (r) in the list	$AP_i = \left[(RK1_i * BPK1_r) + (RK2_i * BPK2_r) + (RK3_i * BPK3_r)\right]$

Actual payment of a winning user i is compared to his bid-price payment (BPP) as shown in equation 5.5.

$$BPP_i = \left[\left(RK1 * BPK1 \right)_i + \left(RK2 * BPK2 \right)_i + \left(RK3 * BPK3 \right)_i \right] \quad \ldots (5.5)$$

Also, the resource allocation utility for a winner is given by equation 5.6-

$$Utility_i = BPP_i - AP_i \quad \ldots (5.6)$$

As is clear from the explanations above, a cloud user will always pay less than what he best-bids for resources. Two cases are worth considering-(i) when preference is D1S1P2 then the actual payment is equal to the winner's best-bid payment and (ii) when preference is D1S1P1 then the actual payment exceeds the winner's best-bid payment.

5.6 SIMULATION RESULTS

Simulation experiment is carried on CloudSim environment using 3 heterogeneous resources with multiple instances, a single service/resource provider and numerous participants. Auction winners are shown in figure 5.3 along with the variations in their actual payments. Figure 5.4 shows the revenues earned by a service provider.

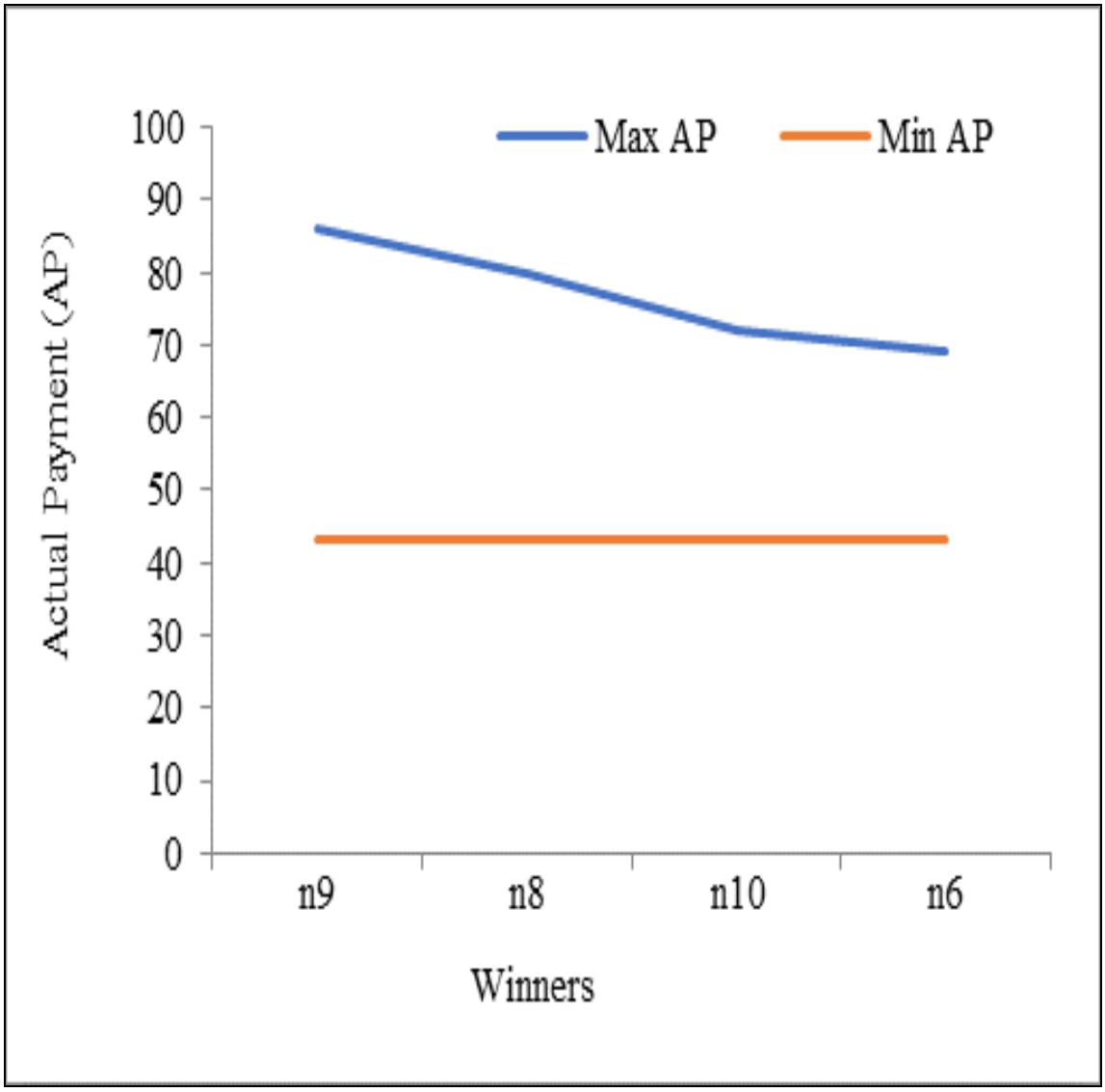

Figure 5.3 Variation in actual payments of winners

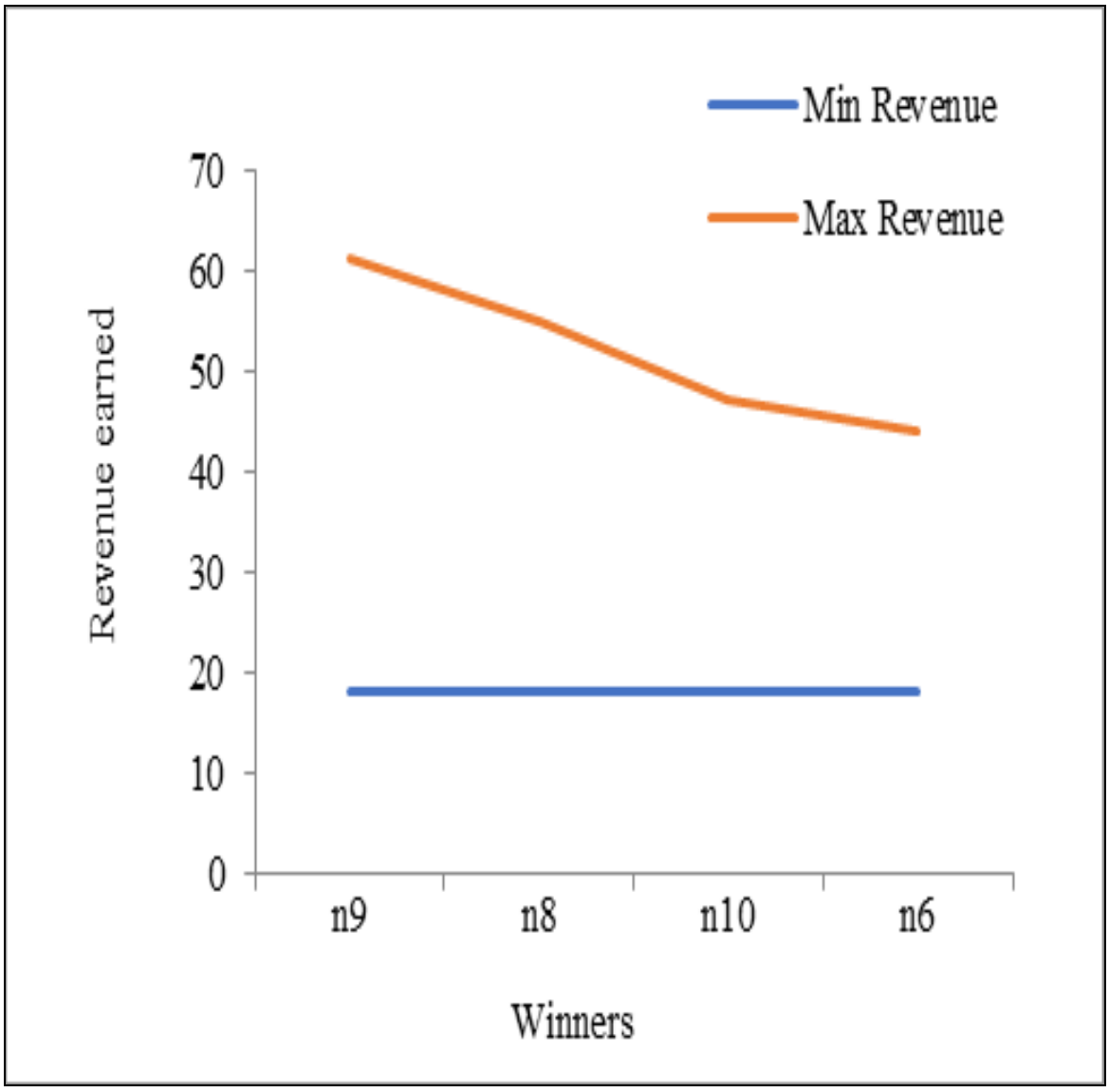

Figure 5.4 Variations in revenues earned by the service provider

A comparison between the actual payments and the best-price payments of a winning user is given in figure 5.5. As shown in the figure, the proposed allocation strategy allows a winner to pay an amount which is considerably lesser than his bid price. This payment strategy works in favour of cloud users if they quote their 'best' bid price to win the resources auction. Utility earned by a winner according to his actual payment is given in figure 5.6.

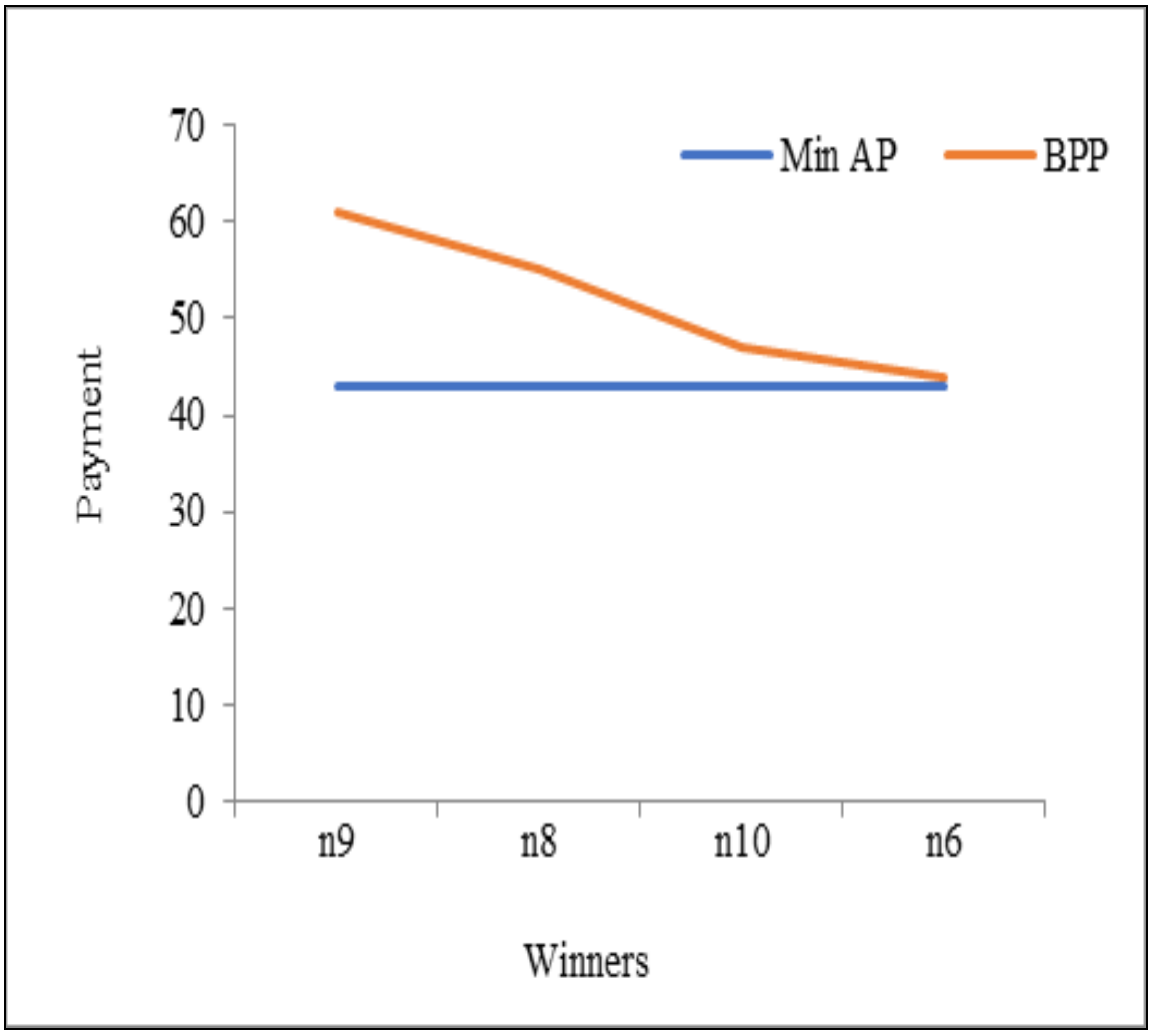

Figure 5.5 Mean AP vs BPP of winners

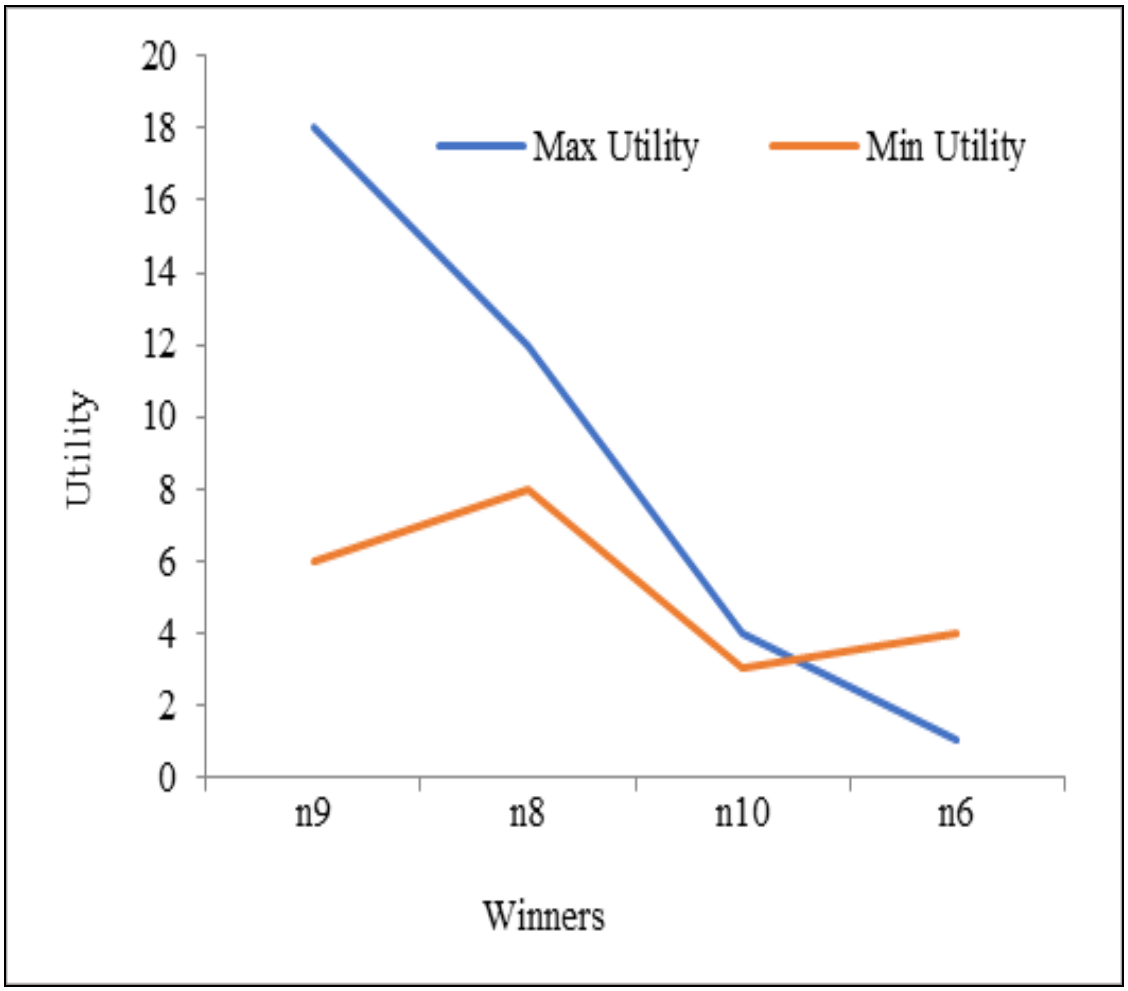

Figure 5.6 Max and Min utilities of winners

Proposed allocation strategy is compared with VCG auction mechanism and the results obtained are shown in figures 5.7 and 5.8. Payments in VCG are lesser than that in the proposed scheme as shown in figure 5.7, suggesting that the proposed allocation reaps better profits to a service provider. Additionally, cloud users enjoy less payment than their quoted amount as shown in figure 5.6. Revenues comparison of both the auction scheme is given in figure 5.8. Concludingly our given auction provides better performance results than the VCG technique.

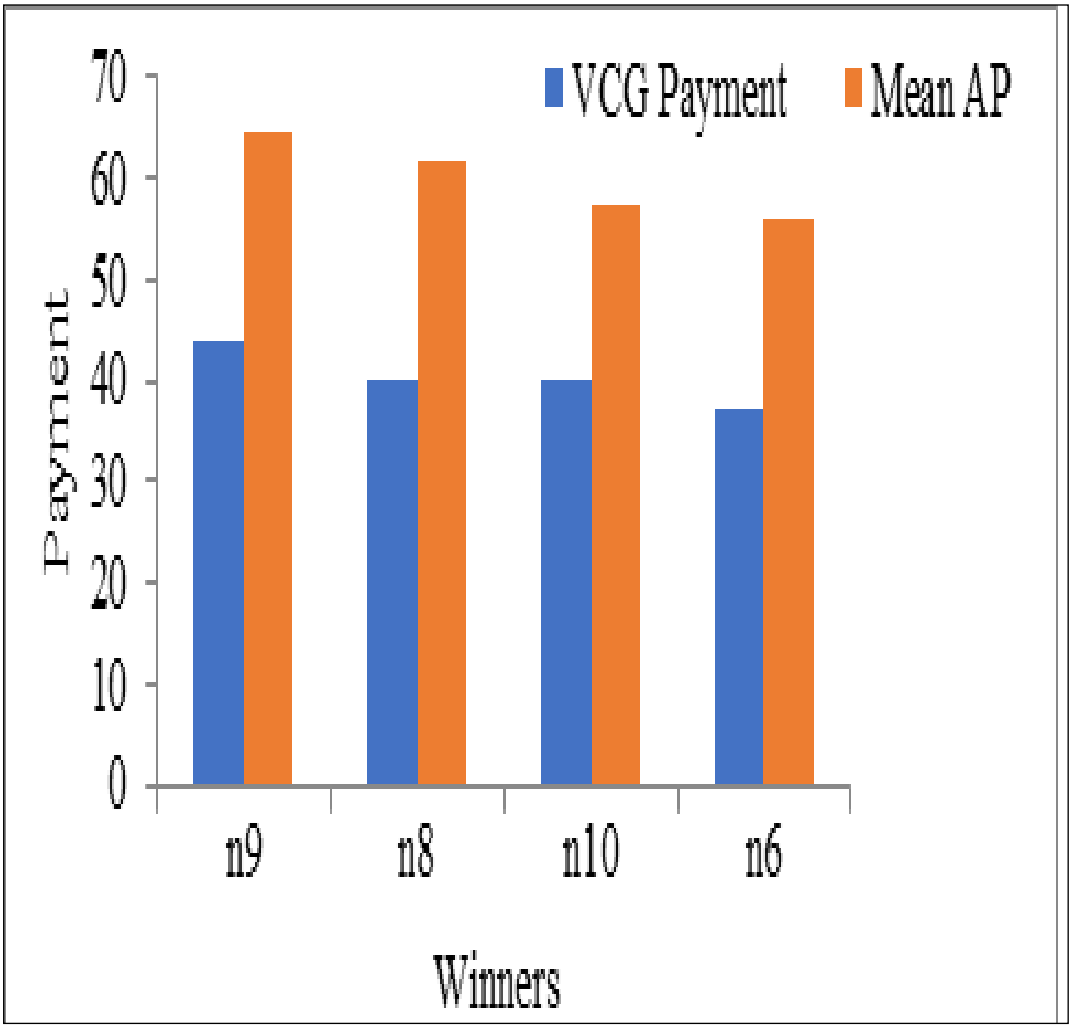

Figure 5.7 VCG payment vs proposed technique payment

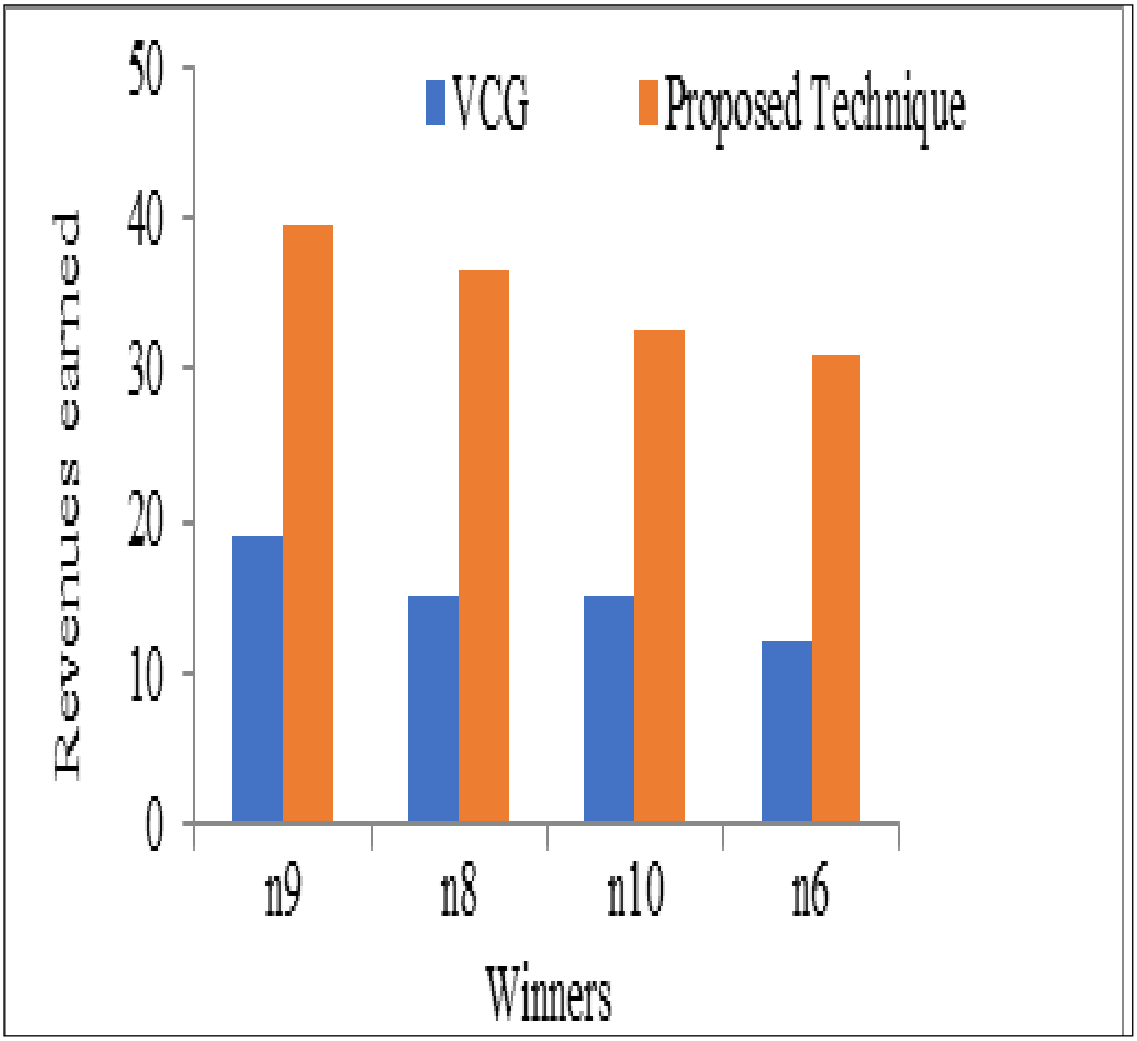

Figure 5.8 Comparison of revenues earned by the service provider

CHAPTER VI

A Token-based Job Scheduling in Cloud Data Centers

The content of this chapter is published in-

Research Journal of Recent Sciences, vol. 4, pp. 29-33, 2015, ISSN: 2277-2502.

CHAPTER VI

A TOKEN-BASED JOB SCHEDULING IN CLOUD DATA CENTERS

6.1 INTRODUCTION

Users' resource requirements fluctuate greatly in a cloud computing environment. To service uneven, heterogeneous and parallel requests, computing environment must organize users' tasks in a continuous manner. There are many scheduling algorithms which are implemented in a cloud data centre and they aim to optimize certain performance features, like, implementation cost, server load, quality of service, fault tolerance, reliability, etc. Job scheduling in cloud refers to users' tasks assignment to servers in the data centres. Different tasks have different performance requirements; hence it is crucial to assign the task to a suitable server so that all the desirable criteria are met. An optimal scheduling strategy thrives for balanced resource utilization along with the most favourable performance of the data centre. One of the main scheduling issues in cloud computing is an efficacious employment of cloud resources along with their fair distribution among users' job.

Currently job scheduling methods of cloud computing environment largely employ pre-allotment where resources are allocated to users in advance as per their job requirements. This is done primarily to guarantee the quality of service of an application. It is a very straight forward and simple scenario, however, in a dynamic cloud data centre this approach suffers from many issues. It may deliver poor usage of cloud resources as job requirements change continuously; there are few peak demand times whereas other times resource requirement is generally less. Static job

scheduling may waste resources if they are allocated to a job but are lying unused. Alternatively, some other job might be denied resources because they are already in possession with another job.

Job scheduling in a distributed system like clouds occurs in two mutually dependent steps namely- space sharing and time-sharing fashion. A user's job is submitted to the data centre along with its resources requirements. Based on its demands, the job is allocated to a single or multiple physical machines/server. This is space sharing aspect of job scheduling. Further, this job will be disintegrated into smaller components called processes and these processes will be time shared in the server to achieve metrics like fairness and response time. It is worthwhile to note that processes belonging to a single job are usually reliant on each other. It is seen that a good and complete scheduling technique involves both time sharing and space sharing aspects of a job allocation. A successful scheduling approach must consider the type of applications/jobs to be scheduled as different types have different requirements, for eg.- a real-time job requires shorter response time and for that it usually prefers space sharing over time sharing, whereas, a non-interactive batch job looks for peak throughput and hence prefers time sharing. In general, job scheduling is a classic example of making trade-offs among different performance parameters.

There is not a clear-cut definition of what is a good job scheduling algorithm. As stated earlier, for different types of applications with varied requirements, scheduling algorithms aim for different goals. In general, desirable properties of a scheduling algorithm should exhibit the following properties-

Efficiency- an efficient job scheduling technique must ensure an optimum utilization of resources, along-with satisfying other performance metrics like response time, waiting time, turnaround time, etc.

Fairness- Users applications or jobs are usually prioritized in a cloud environment based on certain factors like application type, implementation cost, quality of service demanded, etc. A fair scheduling algorithm must treat each job as per its priority and avoid starvation of a high priority job for a lesser one.

Transparency- An efficient scheduling strategy must be transparent to a cloud user. A cloud user should be unaware of other users while submitting his job and must use the resources without any interference for the time allocated.

Dynamic- Cloud computing is essentially a dynamic, on-the-go market where users keep changing their requirements. Hence, it is crucial for a scheduling technique to go as per the flow. Any static technique of job scheduling will fail the core essence, so it is recommended that a scheduling algorithm reschedules the jobs as when the demand scenario changes.

Present chapter outlines a token-based scheduling algorithm which ensures fairness by allocating user tasks to resources based on a job's token value. A user's job token essentially consists of his chosen SLA parameters, his waiting time in the task queue and the type of user job, i.e. CPU-bound, memory-bound and/or communication-bound. The proposed token-based scheduling uses prediction to match a user's demand with the resource's supply. Also, a job is delayed in execution if it's resource requirements are not currently fulfilled by any server.

Consider a cloud datacenter containing miscellaneous resources such that many user jobs are competing for these resources in varied dimensions. In an ideal computing environment, a cloud data centre must cater to every single incoming job and maintain the quality of service parameters also along with ensuring an ideal resources' utilization. However, in a practical world, dynamically changing user requirements and limited cloud resources make it difficult to service every demand without compromising the application's performance. Performance parameters considered in this chapter are turnaround time, waiting time and response time. Additionally, the suggested scheduling approach ensures fairness to cloud users by maintaining the demand curve with allocation frequency.

6.2 TOKEN-BASED SCHEDULING SCHEME

Tokens are entities indicating the resource requirements/demands of jobs. They lead to a schedule which selects a job with the highest token and so on. On the contrary, allocation interval is the opposite of a schedule sequence. It signifies the time gap between successive resources allocations to a job. For example, suppose the token values of jobs J_i and J_q are T_i and T_q respectively, such that $T_q > T_i$, this conveys two things-

i. The resource requirement of job J_q is more than J_i and

ii. Allocation interval $A_q < A_i$, i.e., job J_q will be frequently allotted the required resources as compared to J_i.

Thereafter, allocation duration for the schedule is calculated. Two schedules are considered- schedule1 has varying allocation duration defined as the division of the

highest and the lowest resource requirement's sum by the total number of coexisting requests, and schedule2 has 1 time unit as allocation duration.

As an example of the proposed schedule, consider 5 concurrent jobs J_1, J_2, J_3, J_4 and J_5 with their individual resource requirements as follows in figure 6.1-

Resources/Jobs	J_1	J_2	J_3	J_4	J_5
CPU	3	2	5	1	7
Memory	4	6	3	2	8
N/W bandwidth	5	8	9	6	1

Figure 6.1 Token scheduling example

Based on these requirements, token for each resource and its requirement is determined. In figure 6.1, consider the resource CPU wherein job J_5 has the highest requirement and J_4 has the least. Hence, J_5 token value for CPU will be the highest and J_4 token value will be the least. CPU token for a job J_i is designated as C_i, memory token as M_i and bandwidth token as B_i. Therefore, as per figure 6.1, tokens are lined up in sinking order for every resource,

$C_5 > C_3 > C_1 > C_2 > C_4$

$M_5 > M_2 > M_1 > M_3 > M_4$

$B_3 > B_2 > B_4 > B_1 > B_5$

Sequence	$J_5 \rightarrow J_2 \rightarrow J_1 \rightarrow J_3 \rightarrow J_4$
Allocation Interval	$1 < 2 < 3 < 4 < 5$
Allocation Duration	3 (schedule1) and 1 (schedule2)

Figure 6.2 Token-based sequence, allocation interval and duration

Based on the above sequence, job with the highest token value is selected. For example, among the highest tokens C_5, M_5 and B_3 it is clear that job J_5 has a majority of high tokens, hence J_5 will be put first in sequence. Similarly, among the second highest tokens C_3, M_2 and B_2, job J_2 wins the majority, so J_5 is followed by J_2 in the sequence. Based on the sequence, allocation interval for each job is fixed in such a way that the former jobs get a lesser interval than the latter ones. Allocation interval denotes the allocation frequency of a resource to a job, which is why, J_5 has the least allocation interval and J_4 has the highest. Sequence determined allocation interval for each job is shown in 2 schedules in figure 6.2. Schedule1 has an allocation duration of 3-time units calculated by dividing the sum of the highest and lowest resource requirements by the number of jobs. Schedule2 keeps fixed allocation duration of 1time units. Figure 6.3 shows both schedule1 and schedule2.

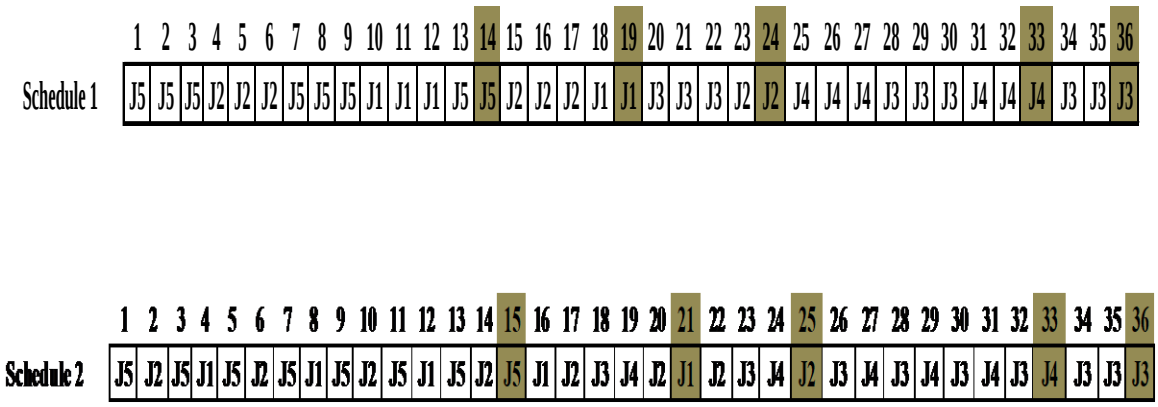

Figure 6.3 Schedule1 and Schedule2

6.3 SIMULATION RESULTS

Two execution timelines of users' jobs are shown in figure 6.3 and are entitled as Schedule1 and Schedule2. These schedules are different from each other in their allocation intervals- on one hand is schedule1 whose allocation duration is 3-time units and based on token values it has a varying allocation interval. Opposed to this in schedule2, allocation interval is same as in schedule 1 but duration is fixed at single/one-time unit. As the proposed scheduling scheme is based on tokens and is foretelling in nature, it is compared with the classic First Come First Serve scheduling algorithm in performance areas like turnaround time, response time and waiting time. Comparison results of the two schemes are shown in **figure 6.4**. Here, grey time units depict the completion of a job.

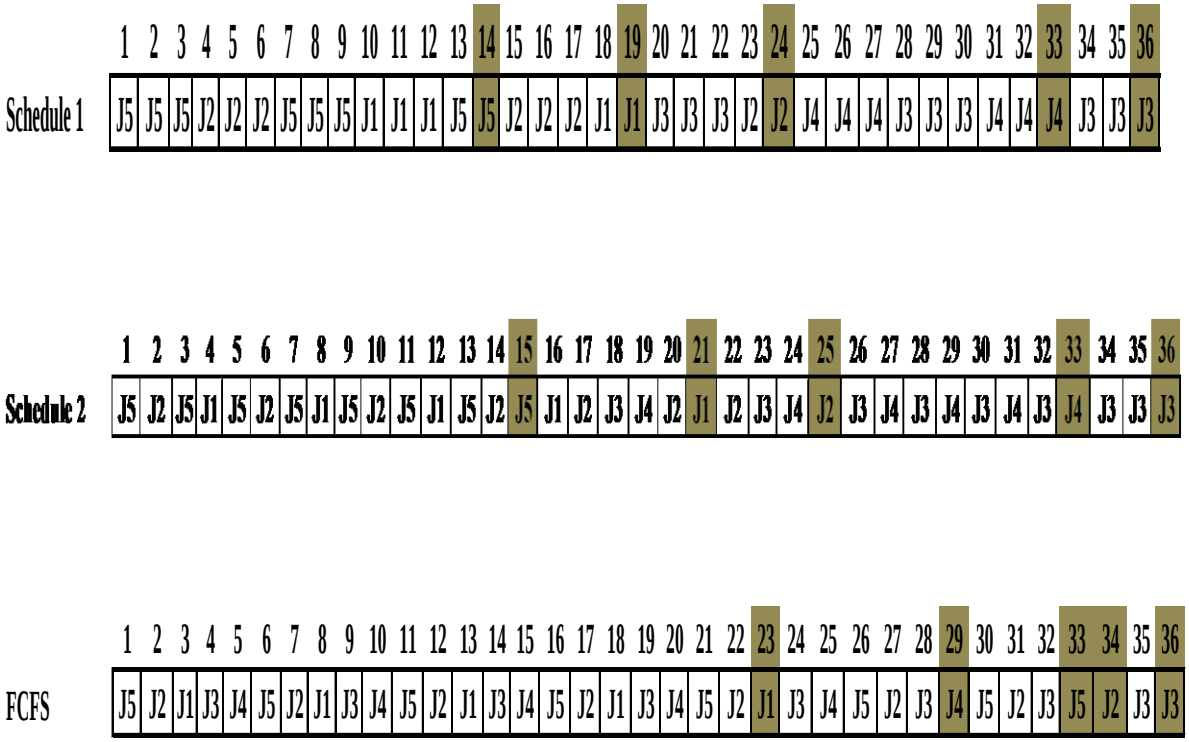

Figure 6.4 Schedule1, schedule2 of token-based scheduling and FCFS schedule

Turnaround time is one of the significant metrics to evaluate the performance of a scheduling algorithm. It is defined as the time gap between the submission of a job in cloud data centre for execution and return of its output to the cloud user. Figure 6.5 and 6.6 compare the turnaround times of schedule1, schedule2 and FCFS with five concurrent jobs J_1 to J_5 having different token values as given in the example above. Figure 6.5 compares the turnaround time between the two schedules of the proposed schedule scheme whereas figure 6.6 compares the turnaround time among both the schedules and FCFS.

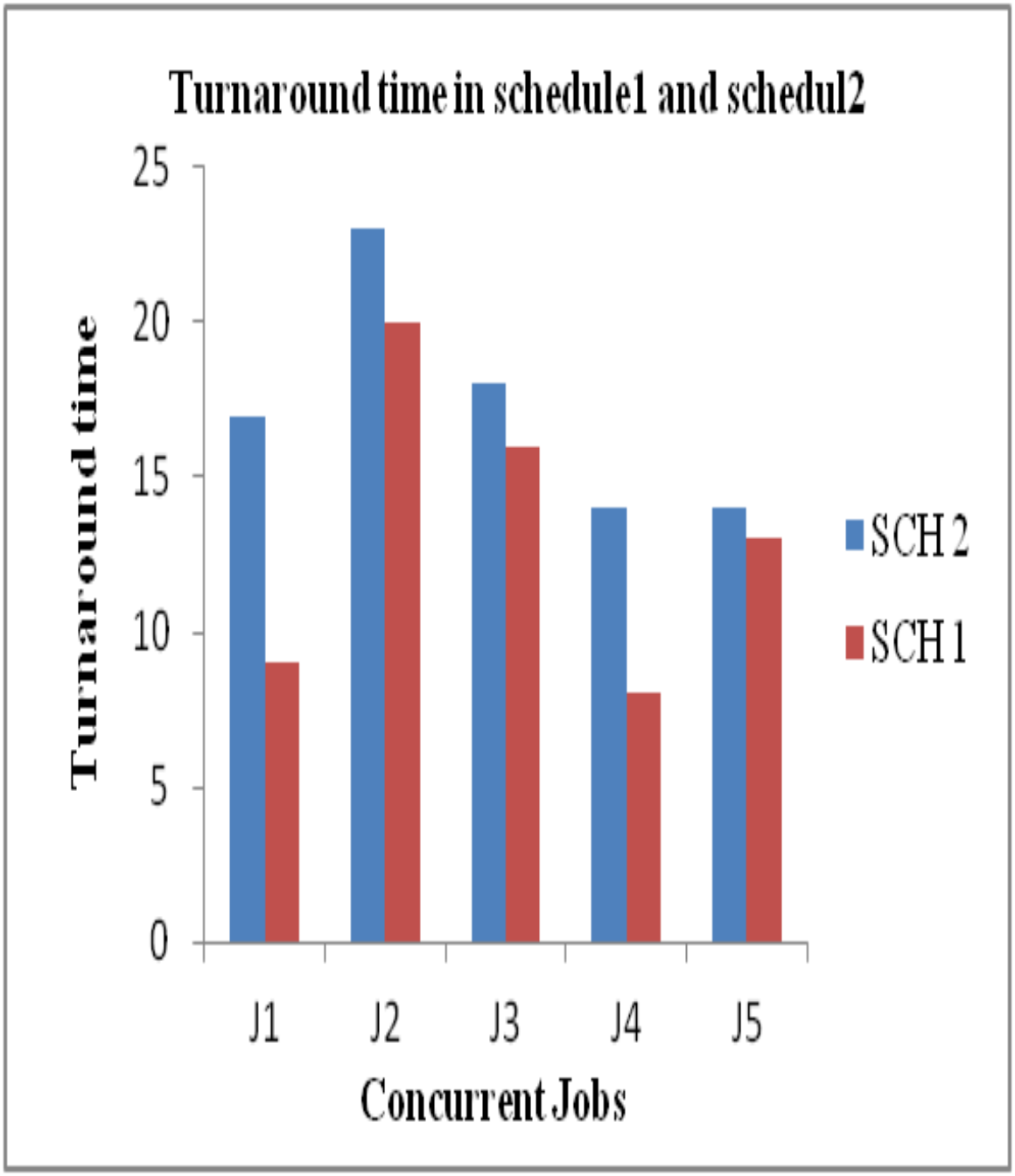

Figure 6.5 Turnaround time in Schedule1 and 2

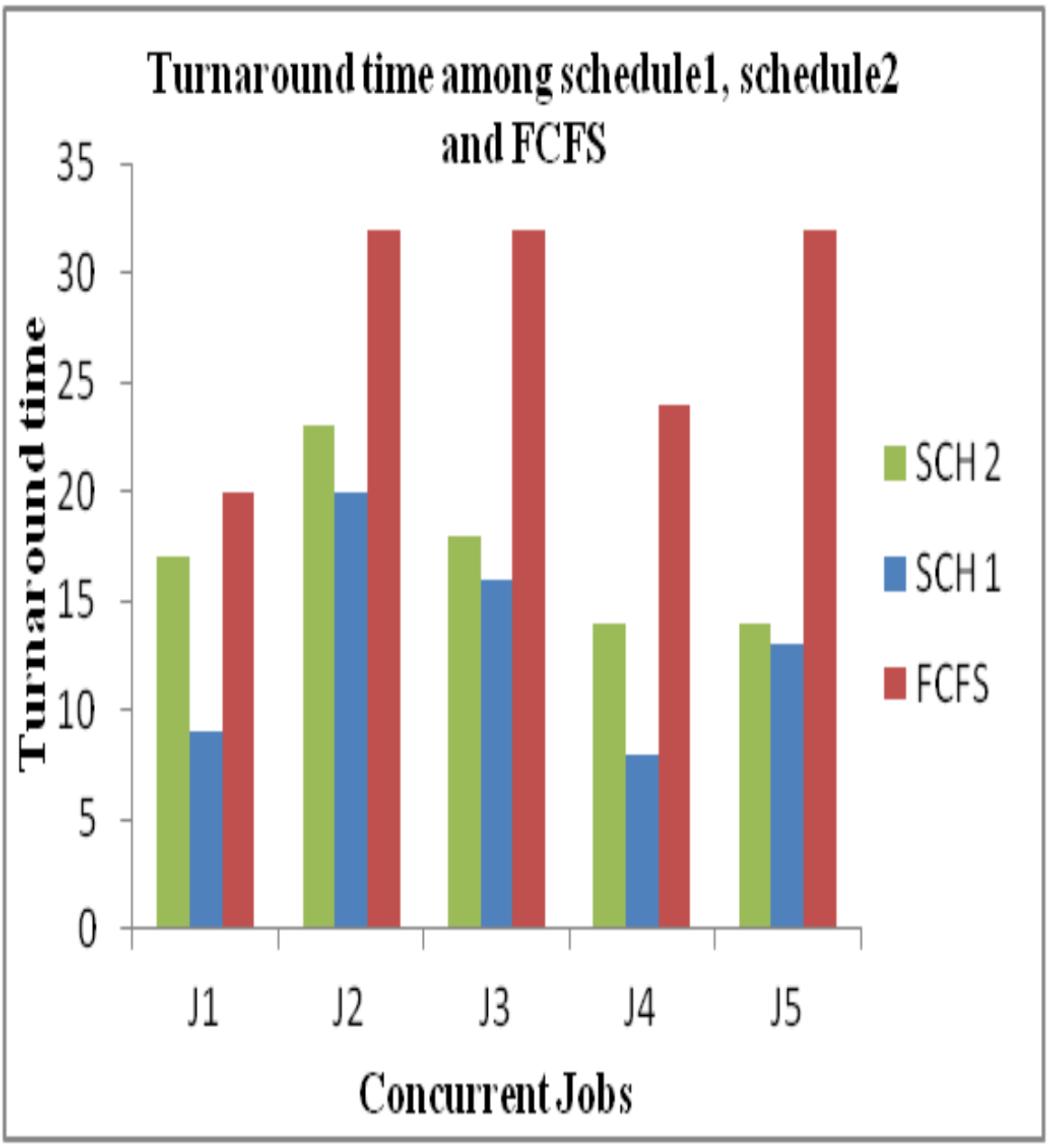

Figure 6.6 Comparison of turnaround times

Response time is another metric considered for performance evaluation of the proposed scheduling technique. In general, it is defined as the time lapse from the submission of a job to the receiving of its first response. Figure 6.7 compares the variants of token-based predictive scheduling and FCFS scheduling with respect to the response times of concurrent jobs.

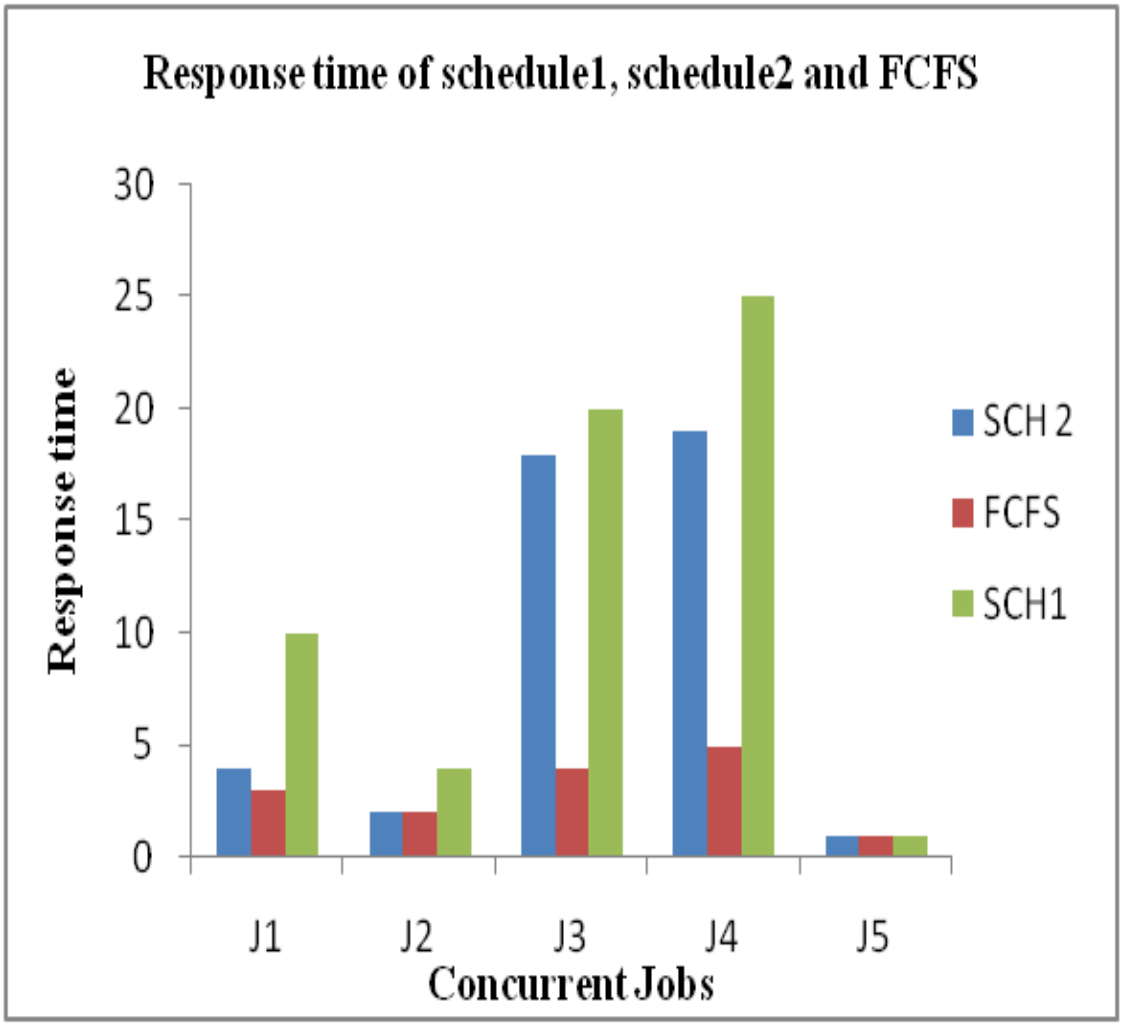

Figure 6.7 Response time comparison

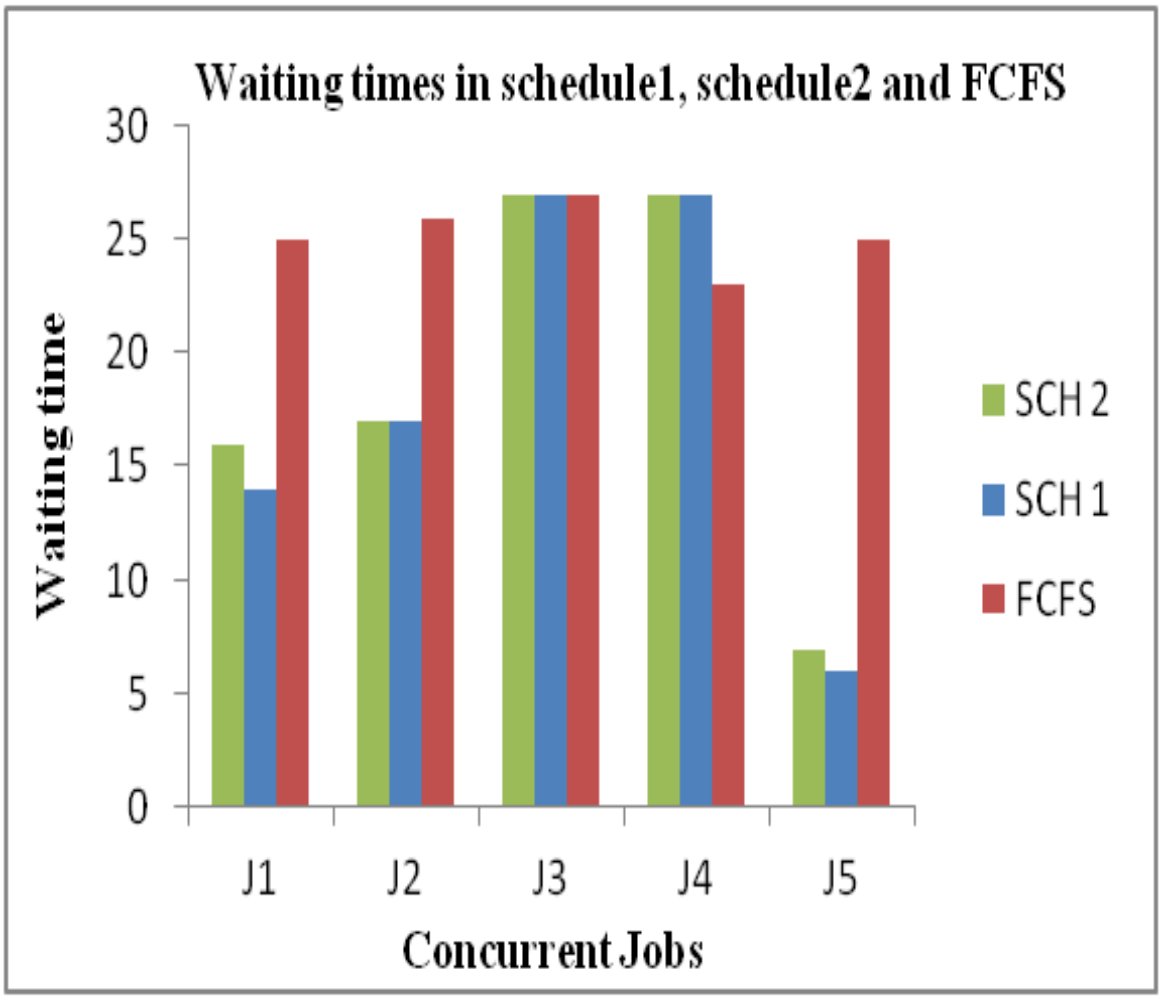

Figure 6.8 Waiting time comparison

Figure 6.8 compares the three scheduling scenarios in their waiting time. Here, waiting time is the time spends by a job in the ready queue. As the figures from 6.5 to 6.8 show, the presented scheduling scheme fares reasonably well as compared to the traditional FCFS scheme with respect to turnaround time and waiting time.

CHAPTER VII

Multi-criteria Based Admission Control

The content of this chapter is published in-

International Journal of Innovations & Advancement in Computer Science, vol. 5(6), pp. 110-115, 2016, ISSN: 2347 – 8616.

CHAPTER VII

MULTI-CRITERIA BASED ADMISSION CONTROL IN CLOUD DATA CENTERS

7.1 INTRODUCTION

Key accountabilities of a cloud service provider include scheduling incoming job requests to ensure optimum utilization of cloud resources, acceptable maintenance of QoS parameters, zero or minimal service down-gradation, among others. Reasonably, a job scheduling process must also lower down the total energy consumption of a data center. In this regard, it is seen that job scheduling, in a dynamic scenario like that of clouds, is an inherently difficult task for the following reasons-

- Server workloads are wide-ranging in terms of resource and performance scales.
- Uneven pattern of incoming service requests leading to either bursts of peak traffic and/or traffic dips.
- Interference among jobs, whether resident or incoming, inside the data center.
- Almost impossible forecasting of incoming jobs.

Hence, it can be safely said that a well-structured job scheduling technique requires an intelligent assortment of incoming cloud jobs/requests to refine the overall proficiency of a data center. In this regard, data center needs to employ certain admission control mechanisms for judicially accepting or rejecting the incoming service request. There are many factors that contribute to energy over-consumption in a data center, some of them being unstable load on servers, inadequate availability of resources along with an enormous network traffic. Besides being environmentally precarious, over-consumption of power may lower down the overall performance of a data center along

with a dip in the QoS parameters. Therefore it is imperative that to scale down the power usage, a data center must make sure that the service requests allowed by it should not create a high network traffic and/ or oversubscription of a server. Also, the QoS parameters of the accepted request must be duly honored. Admission Control is a mechanism to fortify optimum use of cloud assets along with honoring requested SLA parameters. In a dynamic environment like that of cloud computing, where various resources can be expanded or constricted as per their market-demands, an efficient admission control strategy helps in solving the following concerns-

- Can a new service request be accepted without affecting the executing requests?
- What are the feasible ways to map a service request to an available VM so that the QoS is satisfied?
- In order to cater to a new service request, what resources should be assigned to it and in which quantity?

A theoretically perfect solution to admission control should be to increase the number of accepted service requests through their efficient placement on available VMs. This chapter explains a multi-key-based admission control which ranks the incoming service requests based on various performance keys and accordingly admits the most suitable request.

7.2 THE PROPOSED ADMISSION CONTROL STRATEGY

A cloud data center encounters an uneven rush of service requests in a simultaneous manner. A data center must accept or reject these service requests after examining its own resources availability and servicing capacities. An accepted request is allotted a server to avoid oversubscription such that the request's demands are entertained while profit is also guaranteed to the service provider. At the same time, efforts are taken to

maintain the QoS with no or minimum service interruption. For this, all incoming requests are carefully examined for their resource requirements. Present work proposes a multikey-based admission control which compares and ranks all the incoming requests on more than one decision key. Decision keys, considered in this chapter, focus on energy-conserving aspects, however, they can be generalized for other performance goals as well.

7.2.1 The presented energy-smart admission control mechanism considers n decision key factors (k_1, k_2, … , k_n) and m incoming service requests (sr_1, sr_2, … , sr_m) at any instant t. These requests and key factors are arranged in a n*m matrix as shown in figure 7.1 below, where v_{ij} is the value of decision key i for service request j.

$$\begin{array}{c|cccc} & sr_1 & sr_2 & \dots & sr_m \\ k_1 & v_{11} & v_{12} & \dots & v_{1m} \\ k_2 & v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \vdots & & \vdots \\ k_n & v_{n1} & v_{n2} & \dots & v_{mn} \end{array}$$

Figure 7.1 Evaluation Matrix of keys and requests

7.2.2 In the second step, another n*n relationship matrix of decision keys is constructed exhibiting their relationship with each other as figure 7.2 depicts. Its values are $k_i = r(k_j)$ where r as 1 represents equality relationship between keys k_i and k_j.

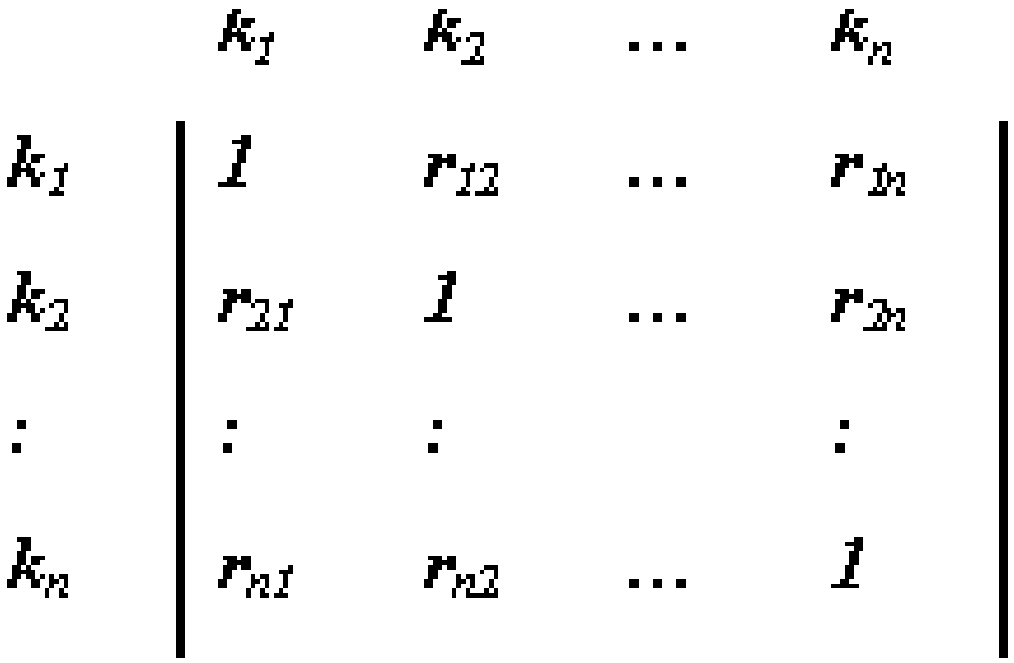

Figure 7.2 Relationship matrix of decision keys

To get a relative ranking of decision keys, Eigen vectors are calculated using this relationship matrix.

7.2.3 Similarly, m*m comparison matrices of service requests are constructed for each decision key k_i and their relative ranking is obtained by calculating Eigen vectors. An example of comparison matrix for a decision key k_i is shown in figure 7.3.

$$\begin{array}{c|cccc} & sr_1 & sr_2 & \dots & sr_m \\ sr_1 & 1/1 & r_{12} & \dots & r_{1m} \\ sr_2 & r_{21} & 1/1 & \dots & r_{2m} \\ \vdots & \vdots & \vdots & & \vdots \\ sr_m & r_{m1} & r_{m2} & \dots & 1/1 \end{array}$$

Figure 7.3 Comparison matrix for decision key k_i

7.2.4 Eigen vectors of service requests for all decision keys are arranged in n*m matrix and the product of this matrix with the decision key ranking vector gives the final ranking of service requests.

7.2.5 In the final step, a service request with the highest rank is admitted to the data center and all other requests are rejected.

7.3 SIMULATION RESULTS

Experimental testing of our proposed technique considered 4 service requests (sr_1, sr_2, sr_3, sr_4) incoming to a data center at an instant t with 3 decision keys (c_1, c_2, c_3). Here, c_1 is load balance, c_2 is availability and c_3 is throughput time. Load balance avoids oversubscription of a server resulting in less power usage. Also, a balanced server requires less or no data migration which further reduces the power consumption of network switches. Hence, decision criteria c_1 fully supports energy conservation. Figure 7.4 shows relationship matrix of decision keys and their resultant Eigen vectors and rankings.

	Load-balance	*availability*	*Throughput time*
Load-balance	*1*	*1/2*	*3/1*
availability	*2/1*	*1*	*4/1*
Throughput time	*1/3*	*1/4*	*1*

Figure 7.4 (a) Relationship Matrix

Load-balance	*0.3194*
availability	*0.5595*
Throughput time	*0.1211*

7.4 (b) Calculated Eigen vector

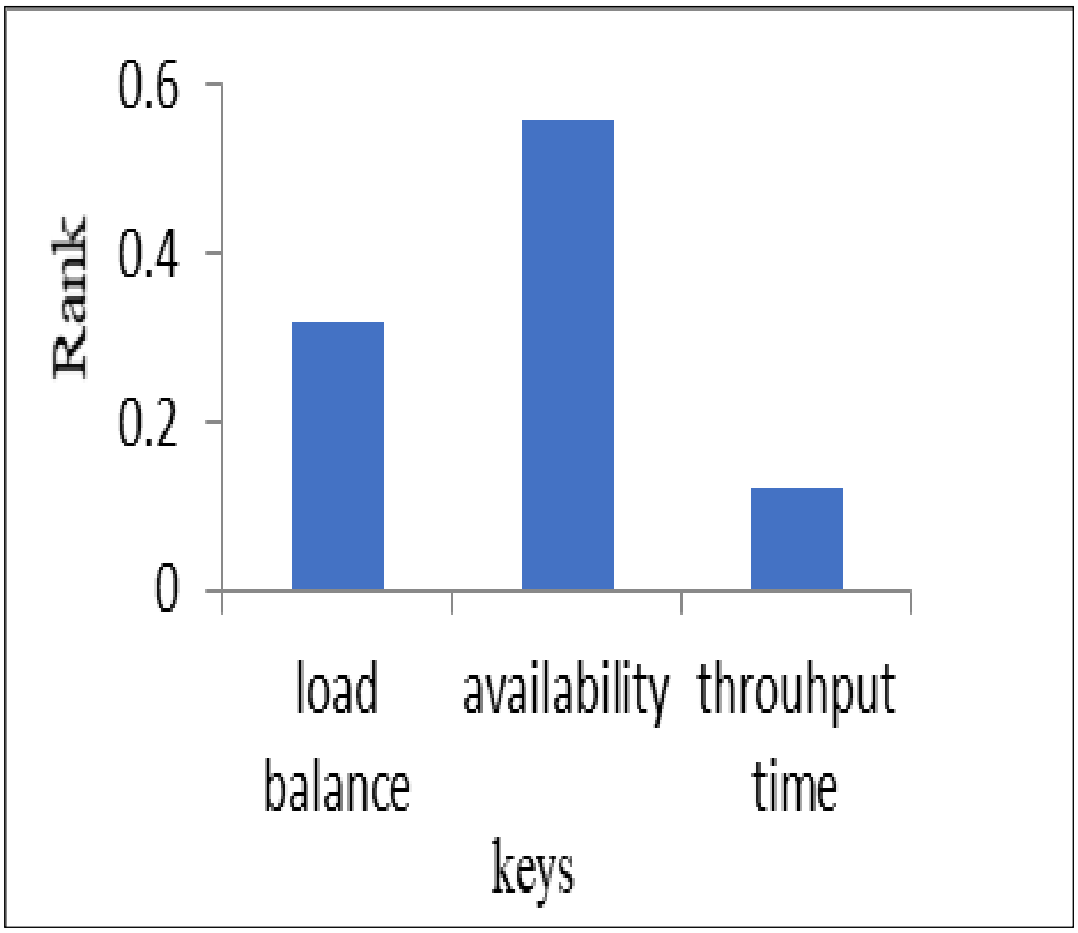

Figure 7.4 (c) Ranking of Decision keys

Figure 7.5 shows the comparison matrix of service requests w.r.t. load balance, its calculated Eigen vector and the rankings of requests for load balance.

	sr_1	sr_2	sr_3	sr_4
sr_1	1/1	1/4	4/1	1/6
sr_2	4/1	1/1	4/1	1/4
sr_3	1/4	1/4	1	1/5
sr4	6/1	4/1	5/1	1/1

Figure 7.5 (a) Comparison matrix for load balance

	Load-balance
sr_1	0.1160
sr_2	.2470
sr_3	.0600
sr4	.5770

7.5 (b) Calculated Eigen vector

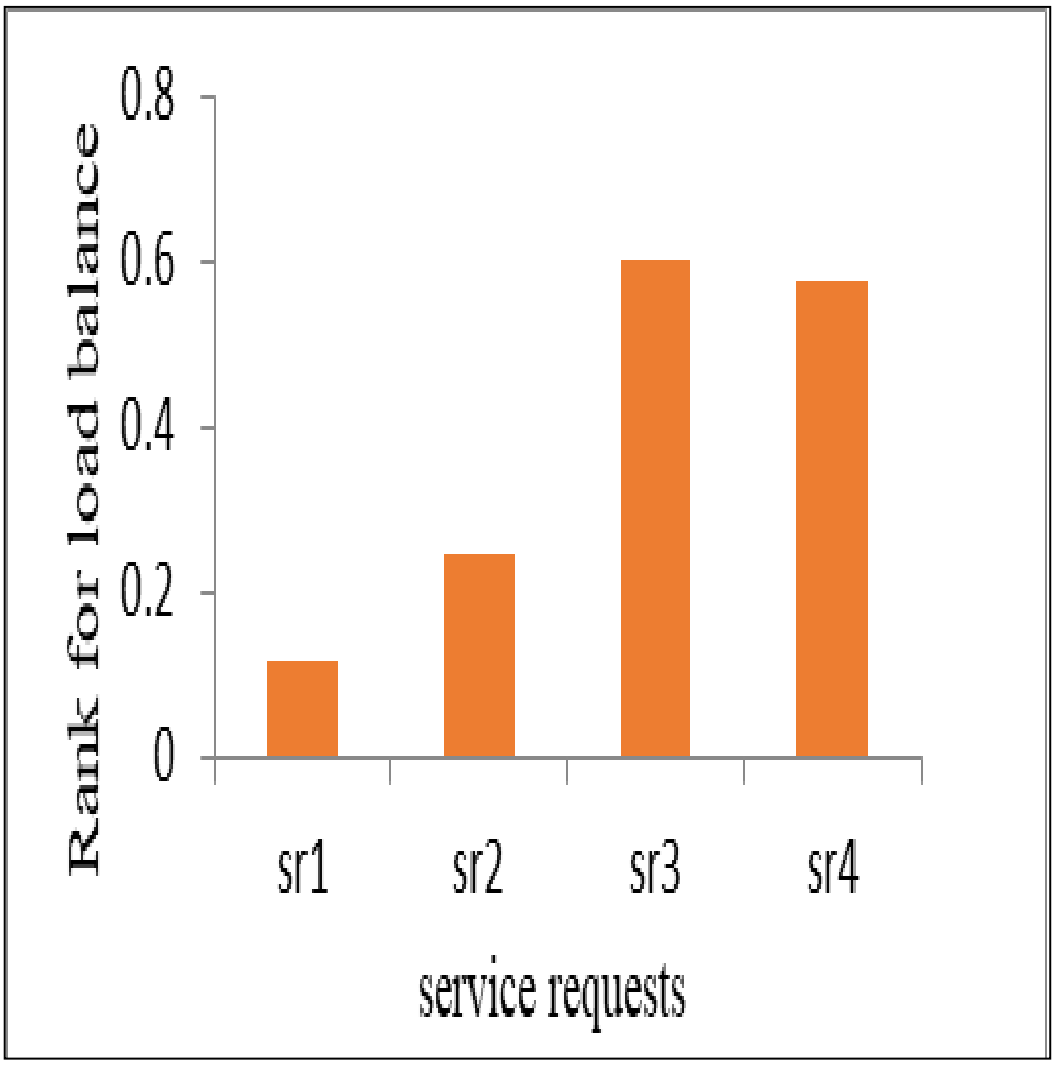

Figure 7.5 (c) Ranking of requests for load balance

Similarly, comparison matrix for availability and the corresponding rankings of decision keys is shown in figures 7.6.

	sr_1	sr_2	sr_3	sr_4
sr_1	1/1	2/1	5/1	1/1
sr_2	1/2	1/1	3/1	2/1
sr_3	1/5	1/3	1	1/4
sr4	1/1	1/2	4/1	1/1

Figure 7.6 (a) Comparison matrix for availability

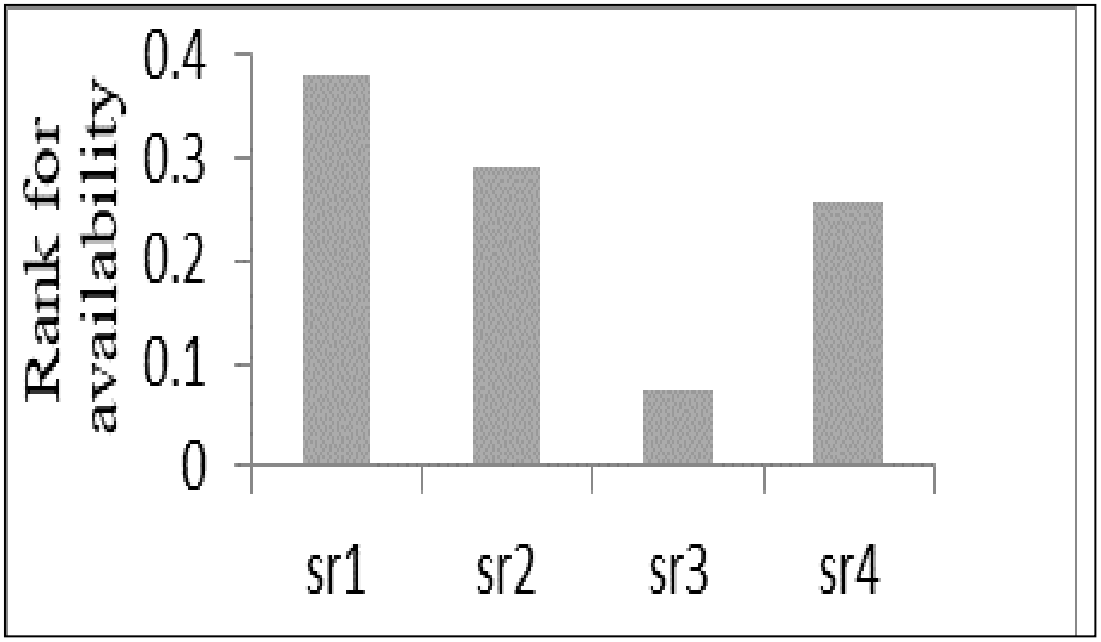

Figure 7.6 (b) Ranking of requests for availability

On similar lines, Eigen vector for throughput time and the corresponding rankings of decision keys is shown in figure 7.7.

	Throughput time
sr_1	.3010
sr_2	.2390
sr_3	.2120
sr4	.2480

Figure 7.7 (a)- Eigen vector for throughput time

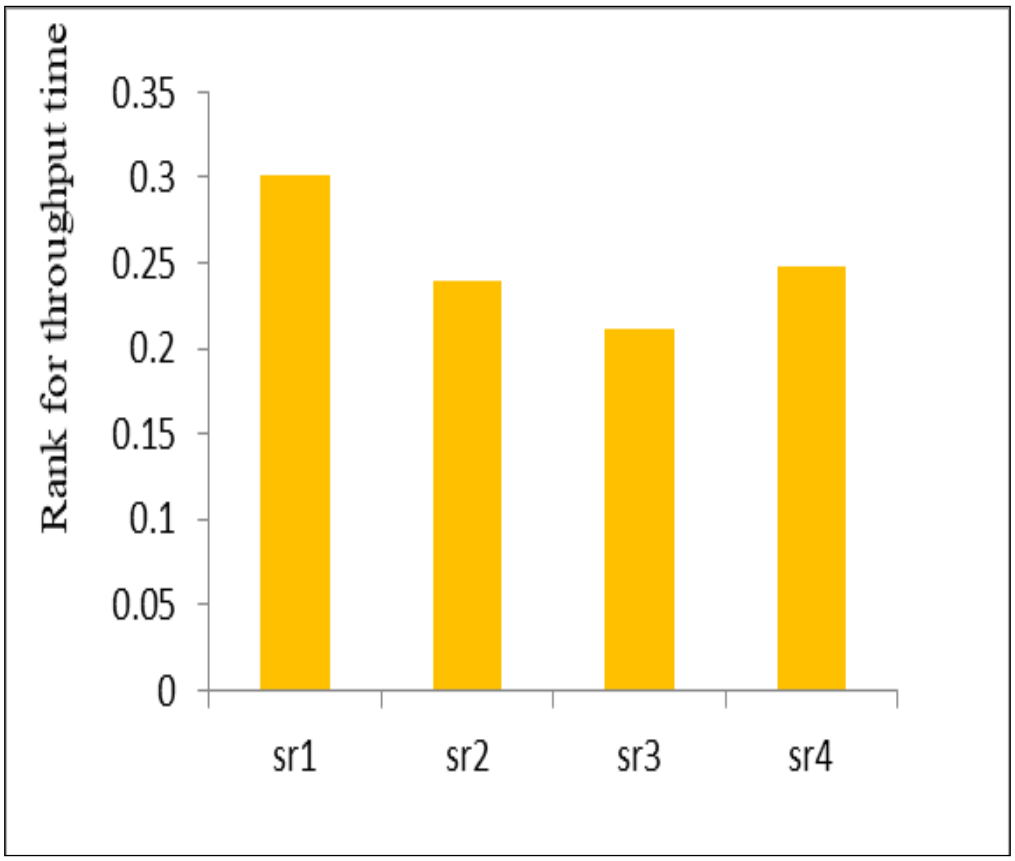

Figure 7.7 (b) Ranking of requests for throughput time

Finally, we calculate the product of the matrix obtained by respective Eigen vectors of the requests with the decision key Eigen vector as shown below-

	Load-balance	*availability*	*Throughput time*
sr_1	*0.1160*	*.3790*	*.3010*
sr_2	*.2470*	*.2900*	*.2390*
sr_3	*.0600*	*.0740*	*.2120*
sr4	*.5770*	*.2570*	*.2480*

Load-balance	*0.3194*
availability	*0.5595*
Throughput time	*0.1211*

	Final Rank
sr_1	*.3060*
sr_2	*.2720*
sr_3	*.0940*
sr4	*.3280*

As the final rank shows, service request 4 will be admitted for execution in the data center, while others will be rejected.

CHAPTER VIII

Energy Efficiency and Other SLAs

The content of this chapter is published in-

1. **BRIS Journal of Advances in Science & Technology, vol. 4 (2). pp. 80-87, 2017, ISSN. 1444-8939.**
2. **Springer series on Advances in Intelligent Systems and Computing, vol. 434, pp. 179-187, 2016, ISSN 2194 – 5357.**

CHAPTER VIII

ENERGY EFFICIENCY AND OTHER SLAs

8.1 INTRODUCTION

In the present chapter, we discuss two aspects of modern cloud data centers-

8.1.1 Energy efficiency in data center networks (DCN) - Cloud data centers enable customers to use computing services, platform and infrastructure with high efficiency and a user-friendly billing system. However, these data centers also incur high computational cost owing to increasing power and energy consumption. This calls for the development of certain optimization techniques to handle and reduce the increase in energy consumption without adversely affecting the reliability and efficiency of data center resources like computing, storage, bandwidth, etc. As far as the energy consumption scenario is concerned, it is observed that IT and networking equipments consume nearly 50% of the total energy consumption of a data center. Further, approximately half of such energy consumption is due to the data traffic inside a data center. Many energy-conserving schemes have been suggested in the past, mostly emphasizing on reducing energy consumption of servers as approximately 70% of a DCN energy consumption comes from the active and idle servers. However, proper consideration of network switches employed in a data center and their energy consumption pattern is crucial in maximizing the overall energy savings.

A large body of work, concerning energy efficiency in cloud data centers considers that datacenter infrastructures are underutilized and over provisioned. To tackle this issue, certain measures are employed like Dynamic Power Management (DPM) which puts the underutilized equipments to sleep mode. It

has been already established that power consumption of a server is linked with its CPU utilization and memory and an idle server consumes about two-thirds of its peak power expenditure. Power consumption of switches is constant for chassis and line cards; however, energy consumption rises with the communication traffic. Replication of data also helps in optimization of data center energy.

8.1.2 General Cloud Adoption Model -Cloud adoption refers to shifting business processes to clouds for benefits such as, streamlining work-load, low cost, ease of management, international visibility and improved quality practices. Reasons for cloud adoption by an organization are-

- Better decision making - Today, Cloud computing is used to analyze big data for better decision making, to apportion the data allocated to different locations by applications and to manage the expanding pool of Big data in cloud's storage for future need.
- Easy collaboration- To gain a competitive edge over other organizations in the market, cloud computing can be used to seamlessly integrate business development and operations. Also, cloud allows work to be accessed from anywhere, anytime so as to make collaboration easy and convenient.
- Support for various business needs- A wide variety of business needs such as storage, networking and data processing are supported by cloud datacenters.
- Rapid development of new products and services- Cloud computing provides realistic and spontaneous understanding of the market scenario, which helps businesses to innovate cloud products and maintain superior services.

- Visible and documented results- It is seen that organizations adopting cloud are registering advantageous results in terms of low expenditure, improved efficiency and better employee mobility.

Cloud adoption, however, comes with its share of challenges as well, such as-

- Weak Control- Without a clear demarcation of responsibilities between an organization and a cloud provider, insecurities may creep in among organizations as cloud adoption leads to tasks that are no longer handled by the organization.
- Security- Considered being a very burning issue, security leads to the fear of data theft especially in public clouds where there is a lot of vulnerability in data usage by different customers.
- Data Protection- When client's valuable data is distributed among multiple locations of datacenters for better availability and timely access, it requires measures to ensure its protection and compatibility.
- Service availability- Service downtime and poor performance become major hurdles in the wake of expansion in storage and processing capacities. Communication and computation delays can have a substantial adverse effect on performance, thereby leading to drop in sales.
- Wrong choice of provider- Cloud adoption often begins with long-term contracts and compliance with a specific architectural platform. Hence, in unfortunate cases of unsatisfactory performances by the cloud service providers, switching to another service provider becomes a difficult task.

Our aim is to make cloud adoption a clean and transparent process where the responsibilities of both the parties, i.e., cloud provider and a cloud adopting

business, are clearly documented. In this direction, a semantic framework is introduced which address all the risks and challenges mentioned above and provides its best possible solution.

8.2 ENERGY-EFFICIENT TOPOLOGICAL FRAMEWORK

An effective DCN architecture reduces the quantity of switches used for traffic transmission without adversely affecting the service performance. This chapter considers two DCN architectures, namely switch-centric and hybrid to apply the energy conservation model presented in the next section. Three-tier architecture is a type of hierarchical arrangement of network switches arranged in three layers, as shown in figure 8.1.

As evident from figure 8.1, scalability may become a major issue in such data centers as growing data traffic will bottleneck the switches and will degrade the services' performance. Moreover, as large number of switches is engaged in communication, energy consumption will be more. Steps must be taken to activate as less number of switches as possible for any workload execution.

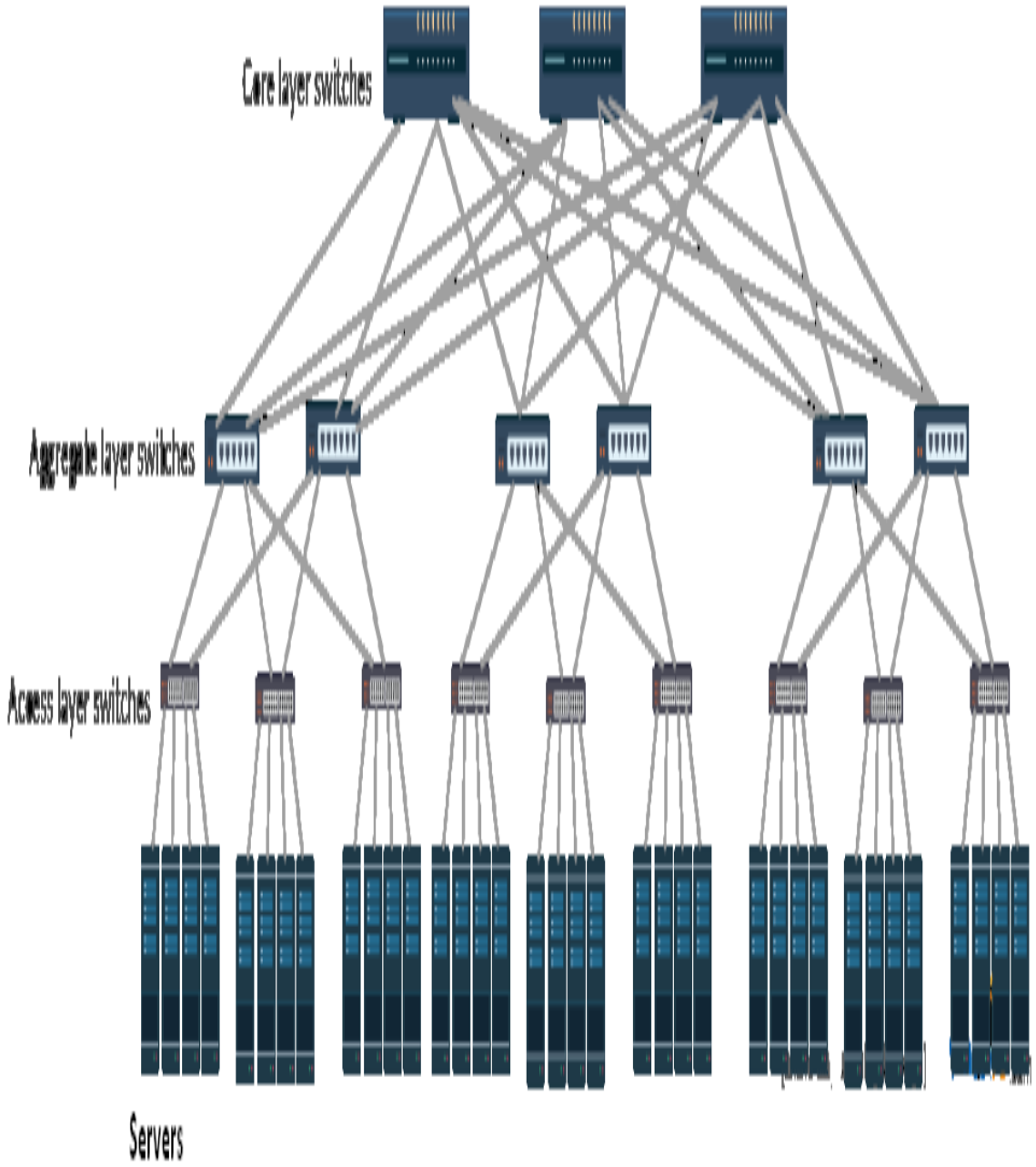

Figure 8.1 Three-tier DCN architecture

DCell is a comparatively new server-centric hybrid DCN architecture where a single cell consisting of n servers and one switch ($DCell_0$) acts as a building block of the entire DCN network. $DCell_0$ forms level 0. At level 1, n+1 $DCell_0$ are required where each $DCell_0$ is connected to other DCells at the same level. Table 8.1 shows an example of the recursive nature of a typical DCell architecture.

Table 8.1 An example statistic of DCell architecture

No. of Levels (k)	No. of DCells of lower levels (m)	No. of servers (n)
0	0	2
1	3	6
2	7	42
3	43	1806
:	:	:
k	$n_{k-1}+1$	$n_{k-1}*m$

In general, number of lower level DCells at level k are $n_{k-1}+1$ and total number of servers at a level k are $n_{k-1}*m$.

This architecture enhances scalability, robustness and removes congestion bottleneck of three-tier architecture by recursively adding cells (or pods) to a DCN. Different DCells at the same level are inter-connected via servers. Hence, DCell is hybrid architecture. Figure 8.2 shows DCell architecture at a recursive level 2.

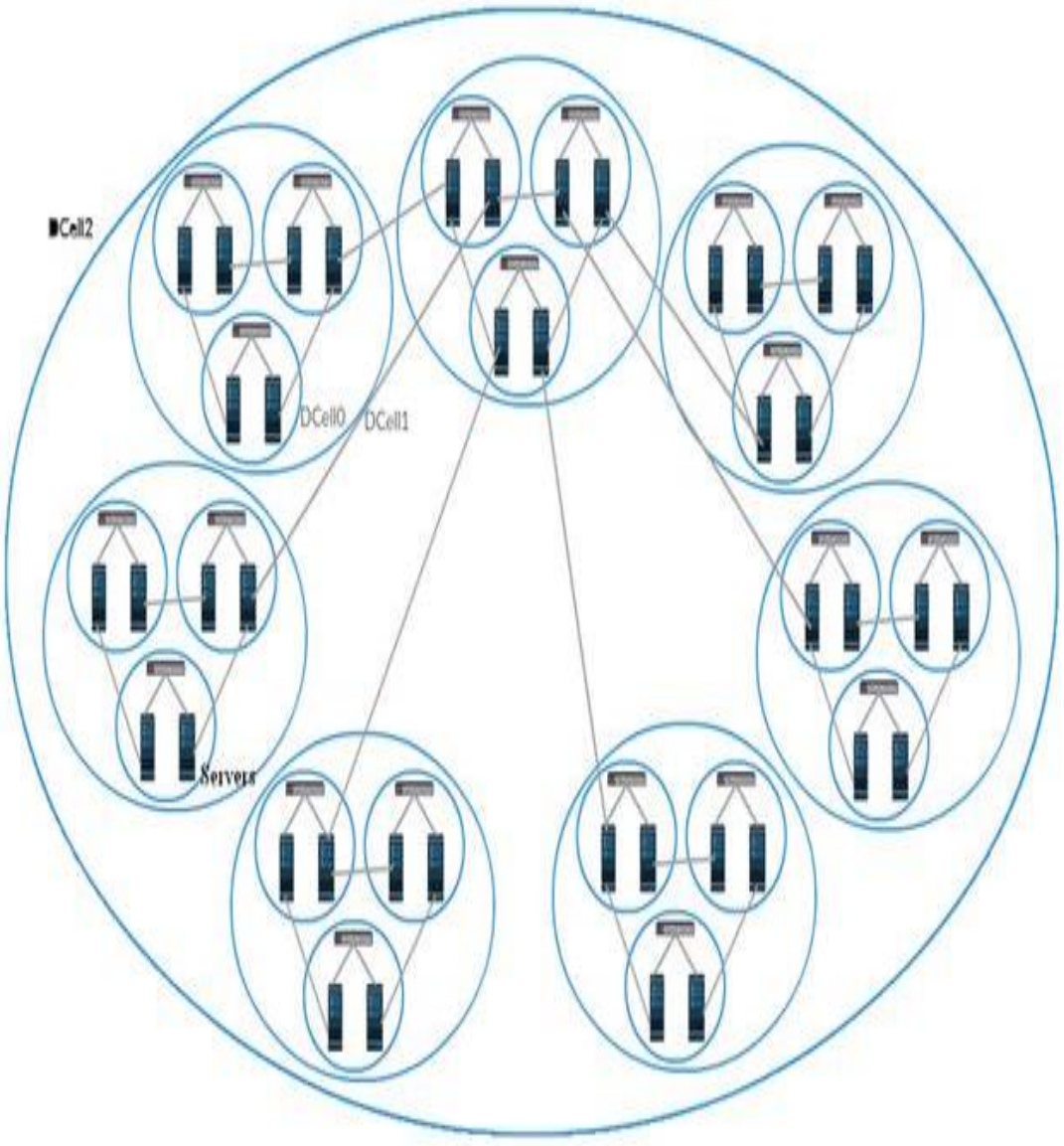

Figure 8.2 Recursive 2-level DCell Architecture Example

A distinguishing feature of DCell is that it uses fewer switches as compared to three-tier architecture. However, this also means that servers in DCell are doing additional job of switching.

Two DCN elements helpful in saving energy are-

6.2.1 Computing Servers- Data center house thousands of servers communicating with each other through the network switches. Each server consumes some fixed power even when its computing load is zero, referred here as idle power consumption or P_{idle}. Ideally, power consumption of a server depends on its load. At its maximum load, a server consumes maximum power which will be referred here as P_{max}. According to a related research study, P_{idle} is equivalent to two-third of P_{max}. Keeping this equivalence in mind, power consumption of a server and its computing load can be stated as in equation 8.1.

$$P_{cs}(wl) = P_{idle} + (P_{\max} - P_{idle})\left(\frac{wl}{sc}\right) \quad \ldots (8.1)$$

where $P_{cs}(wl)$ is the power consumed by a computing server with a workload wl and sc is the total server capacity. Usually, (wl/sc) < 1. Using substitution, one can conclude that

$$P_{cs}(wl) \cong \frac{P_{idle} * wl}{4} \quad \ldots (8.2)$$

Now, energy consumed by a computing server can be given as

$$E_{cs} = P_{cs}(wl) * T_{exec}(wl) \quad \ldots (8.3)$$

here energy consumption of a computing server is given by E_{cs} and total execution time of workload wl is $T_{exec}(wl)$. Workload's execution time is figured out as

$$T_{exec}(wl) = t_{proc} + 2 * t_{db} + t_{update} \quad \ldots (8.4)$$

Here, t_{proc} is the processing time of wl, t_{db} is the database access delay and t_{update} is the time required to update the data replicas. Processing time is proportional to the data volume, database access delay depends on the database location while data updation duration depends on the replica location.

6.2.2 Communication Switches- As data communication increases inside a data center, network ports tend to be utilized to their maximum capacity. Energy consumed by a DCN switch is given in equation 8.5.

$$E_{sw} = P_{sw} * T \quad \ldots (8.5)$$

where E_{sw} is the energy consumed by a DCN switch, P_{sw} is the power consumed by a switch and T is the time a switch is active and is dependent on traffic flowing through it. Power consumed by a switch depends on the traffic passing through its ports and can be given as

$$P_{sw} = P_{const} + \sum_{q=1}^{m} U_q \qquad \ldots (8.6)$$

where P_{const} is fixed power consumed by the switch's chassis and line cards, m is the number of ports in a switch and U_q is the throughput of a link associated with a port q.

Energy Conservation Model- To reduce energy consumption in three-tier DCN architecture, we propose storage hierarchy. Traditionally a DCN accesses the central/main database located in a network cloud for every data access and update. This scenario requires data traffic to propagate all the way from a rack server to the main database and vice-versa every time, thereby, increasing traffic at all the three layers of switches. As a result, more power will be consumed at the intermediate switches, resulting in large energy consumption. To improve this scenario, we introduce rack databases at the access layer and data center database at the core layer network in addition to the original main database in the network cloud. Frequently accessed data can be stored in rack database to limit database transactions to the access layer. Likewise, lesser frequent data can be kept in DC database, to further restrict transactions within the data center. For rarely accessed data, contacting the main database will be required. This way a storage hierarchy, where databases are kept at three levels, will substantially reduce traffic at the aggregate and core level switches, resulting in decreased power usage.

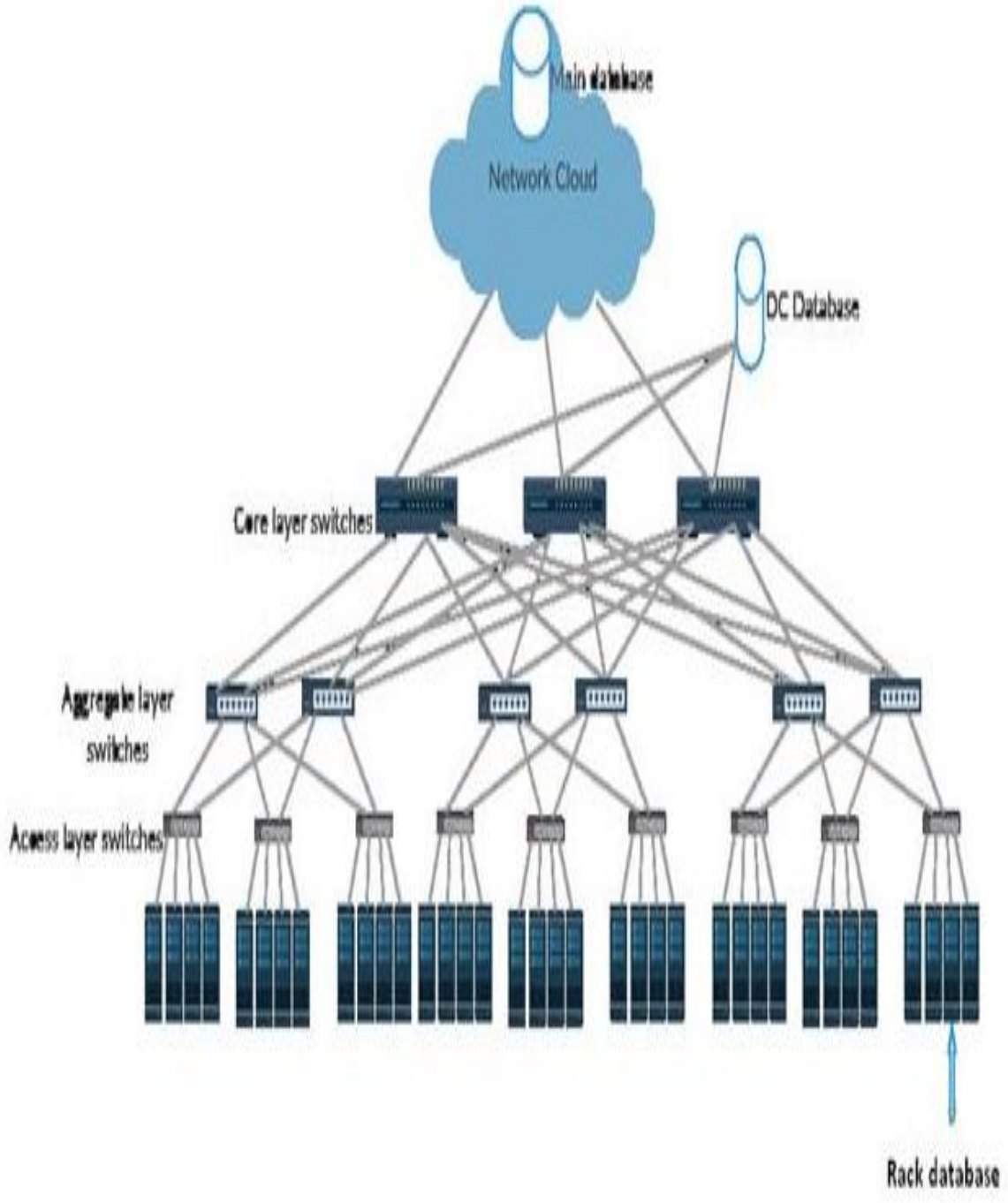

Figure 8.3 Modified Three-tier Architecture

The changes we propose in a recursive DCell architecture consist of reducing the number of links connecting DCells at a single level. Figure 8.2 shows that each $DCell_{l-1}$ is connected to every $DCell_{l-1}$ in $DCell_l$ in a recursive manner. This creates a mesh connectivity which increases redundancy and traffic interference at peak times. To cut out unnecessary communication links, a modified DCell architecture is proposed in figure 8.4. Every DCell is connected to its nearest neighbors only. Communication between two distant DCells at any level can take place via connecting DCells. By minimizing the total number of links at any level in a DCell architecture, we have reduced the routing load of each server and hence its energy will be conserved.

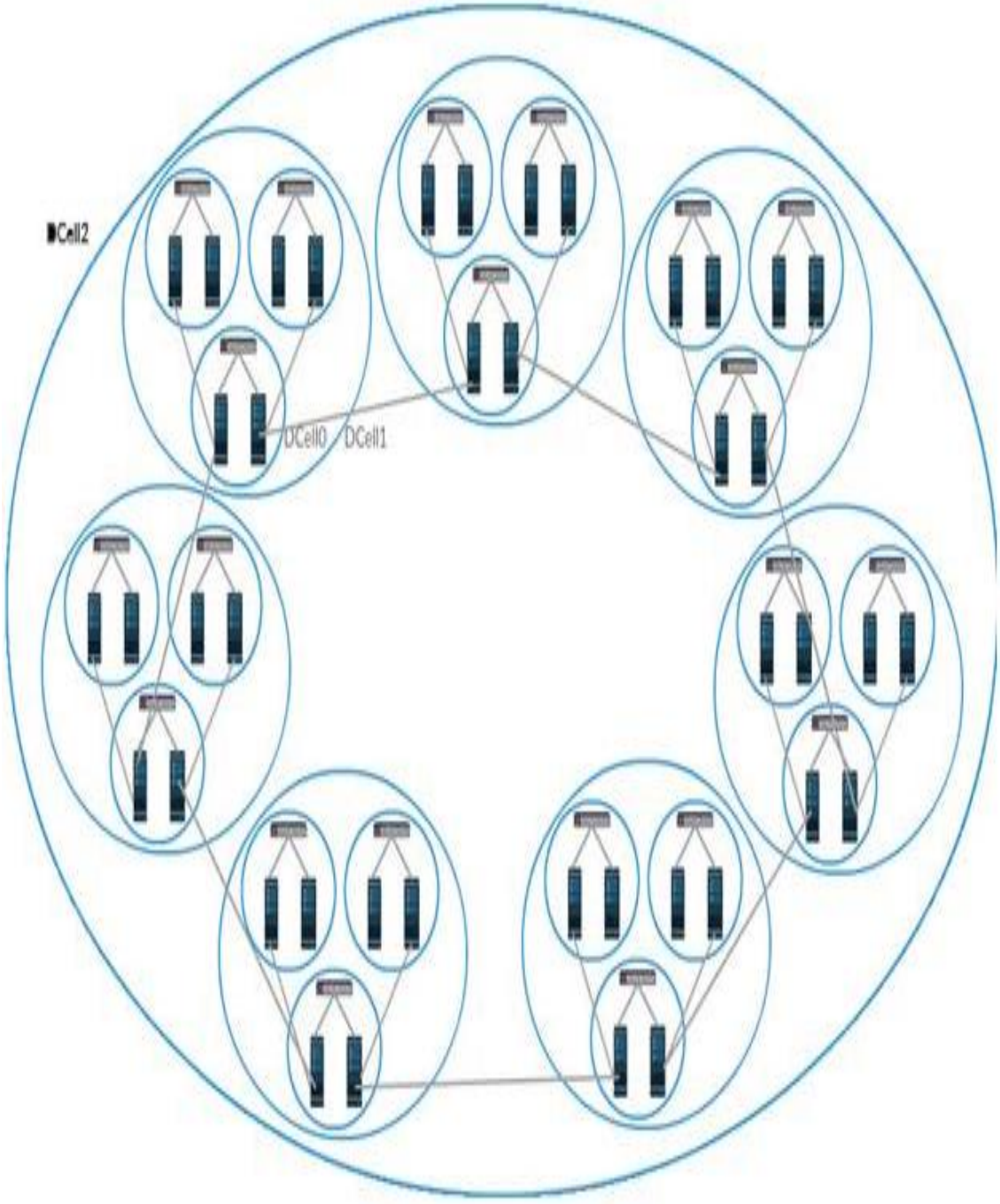

Figure 8.4 Modified DCell Architecture

Table 8.2 given below compares the number of links at level 2 of the original and modified DCell architectures. Further, we also propose 'delayed forwarding' mechanism to be implemented in servers where update data will not be immediately sent and will be delayed till non-peak traffic times, so as to reduce unnecessary traffic in the DCN during heavy traffic workload.

Table 8.2 Link minimization in DCell Architecture

Total no. of links at level 2 ($DCell_2$)	Original Architecture	Modified Architecture
	42	7

The experimental setup for three-tier architecture assumed uniform distribution of services and traffic among the servers. Parameters used for simulation are given in table 8.3.

Table 8.3: Three-tier Architecture parameters

Parameter	Value
Number of Core Switches	1
No. of aggregate switches	2
No. of access Switches	8
No. of servers	256

With the introduction of databases at three levels, energy consumption of computing servers was seen to be dipping as database access delay was greatly reduced. Figure 8.5 shows the trend of energy consumption vs. access delay in 3-tier architecture. As seen, most of the server's requests for data were successfully fulfilled by the rack database, as a result lesser trips to the main or dc databases were needed. This reduced the routing load on aggregate and core level switches, hence energy consumed was reduced.

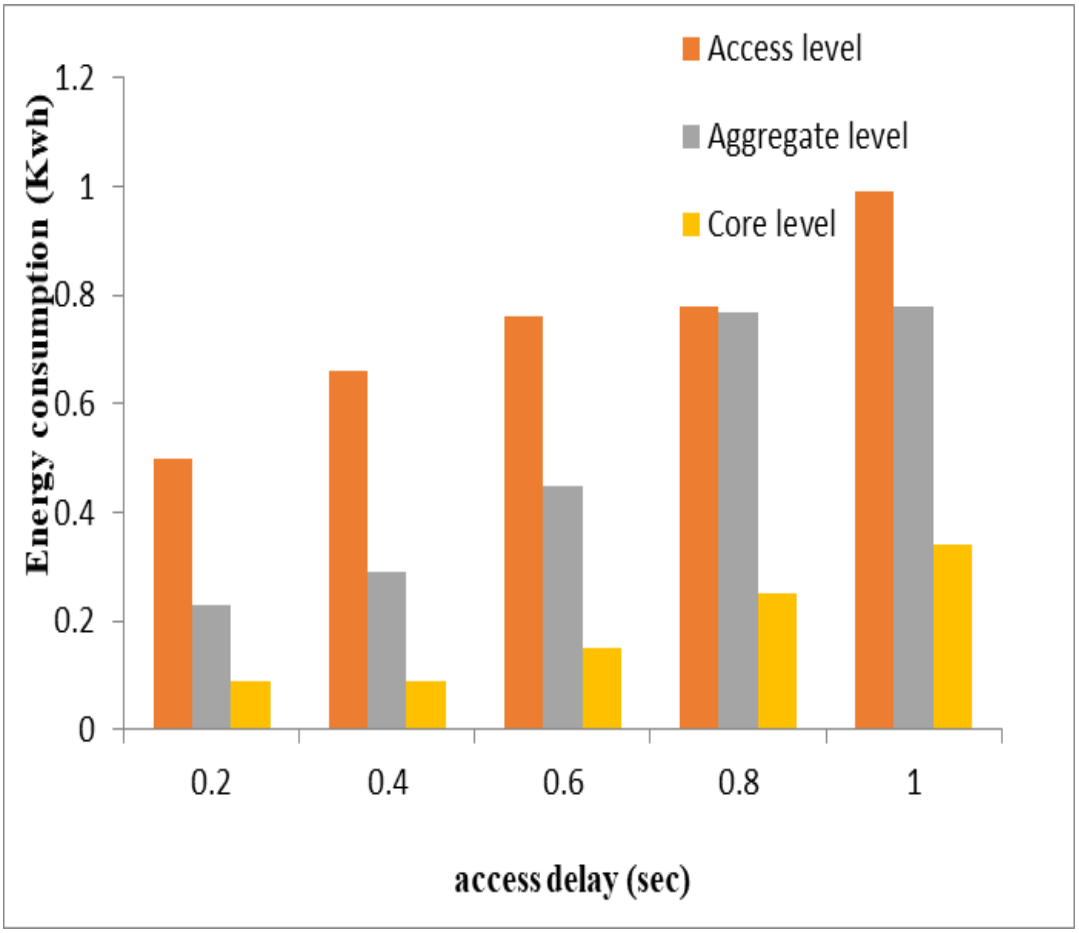

Figure 8.5 Access delay vs. energy consumption of switches in 3-tier DCN architecture

Figure 8.6 shows the access delay and energy consumed in the proposed modified DCell architecture. With the reduction in data center links and the introduction of 'delayed forwarding' mechanism, one can see that as the data access delay increases, so is the energy consumption.

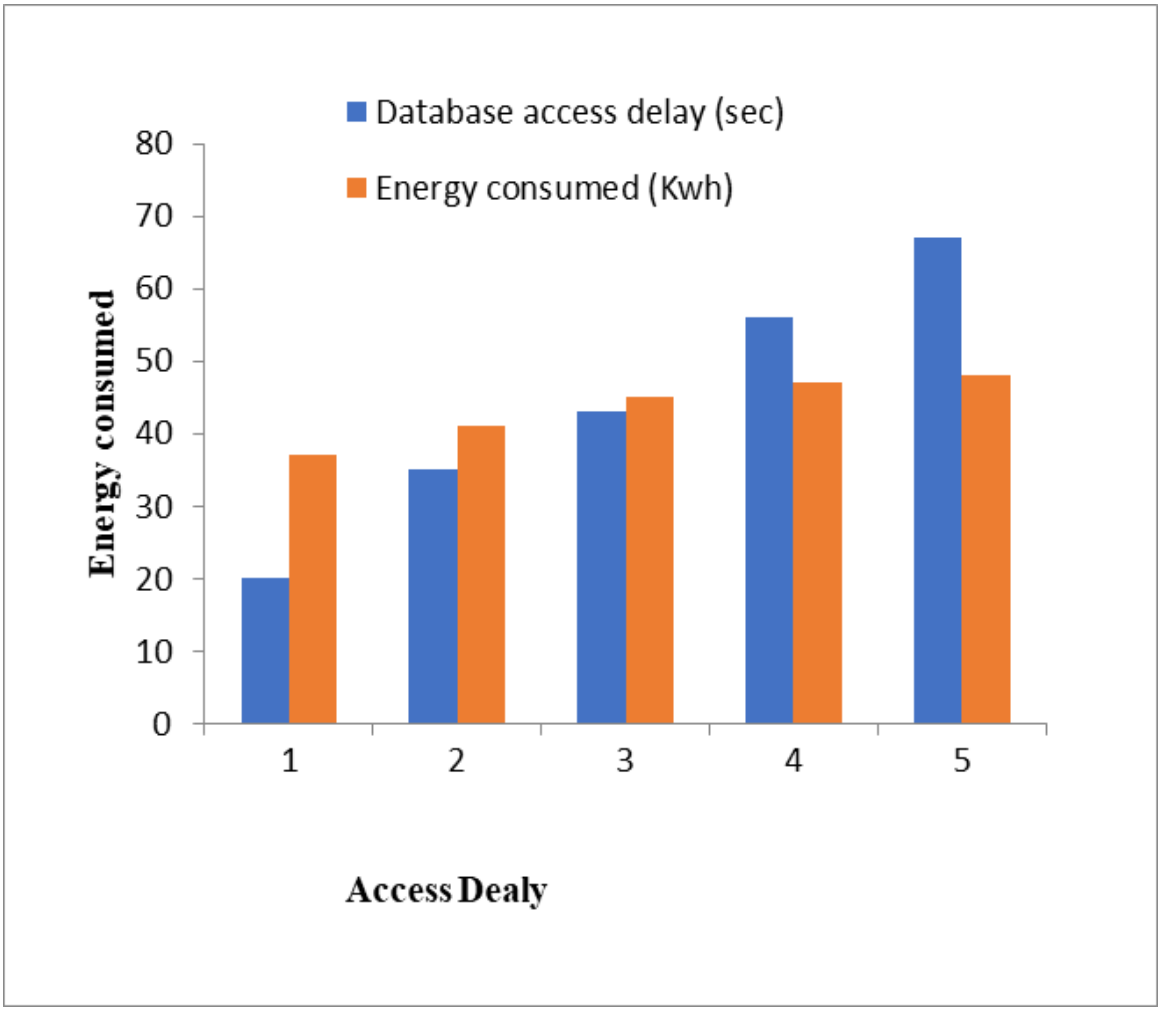

Figure 8.6 Data access delay vs. energy consumed in modified DCell

Modified DCell DCN architecture tries to reduce peak time traffic within the data center by implementing 'delayed forwarding' in routing servers. This delays the replica update operation to the non-peak traffic hours and thus, peak traffic time is dedicated to data access and processing operations. This reduces unnecessary workload on servers which are already struggling with the dual responsibilities of

computation and routing. Figure 8.7 shows a substantial reduction in energy consumption in the proposed DCell architecture.

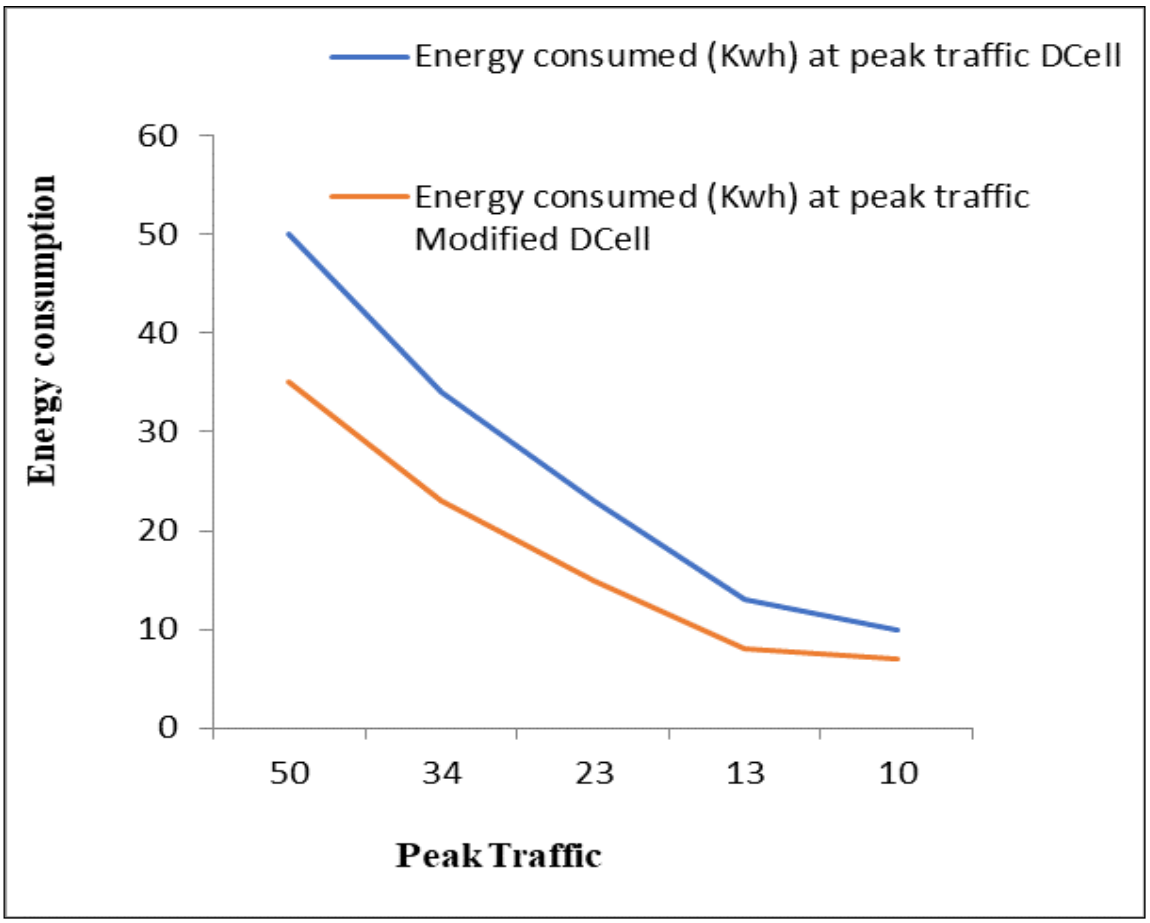

Figure 8.7 Energy consumption during peak traffic in modified DCell

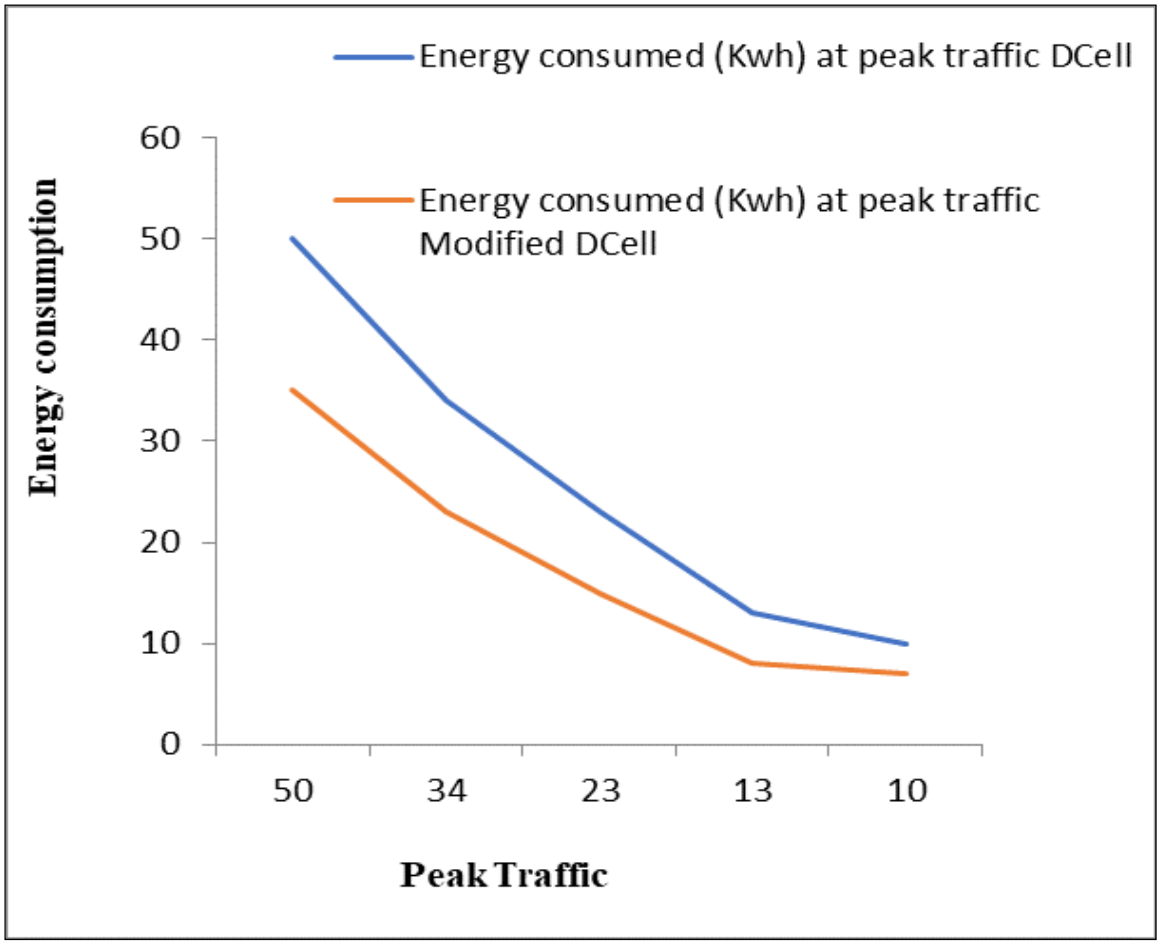

Figure 8.8 Energy consumed during packet loss in in modified DCell

Figure 8.8 shows the energy consumption during packet loss in modified DCell architecture after introducing the 'delayed forwarding' mechanisms in servers for updates and backups.

8.3 CLOUD ADOPTION FRAMEWORK

This chapter also provides a framework to standardize cloud adoption procedure for all scales of business organizations. It is applicable to public, private and hybrid clouds and is also standardized for IaaS, PaaS and SaaS service models. Figure 8.9 given below shows the cloud adoption framework with its broad categories.

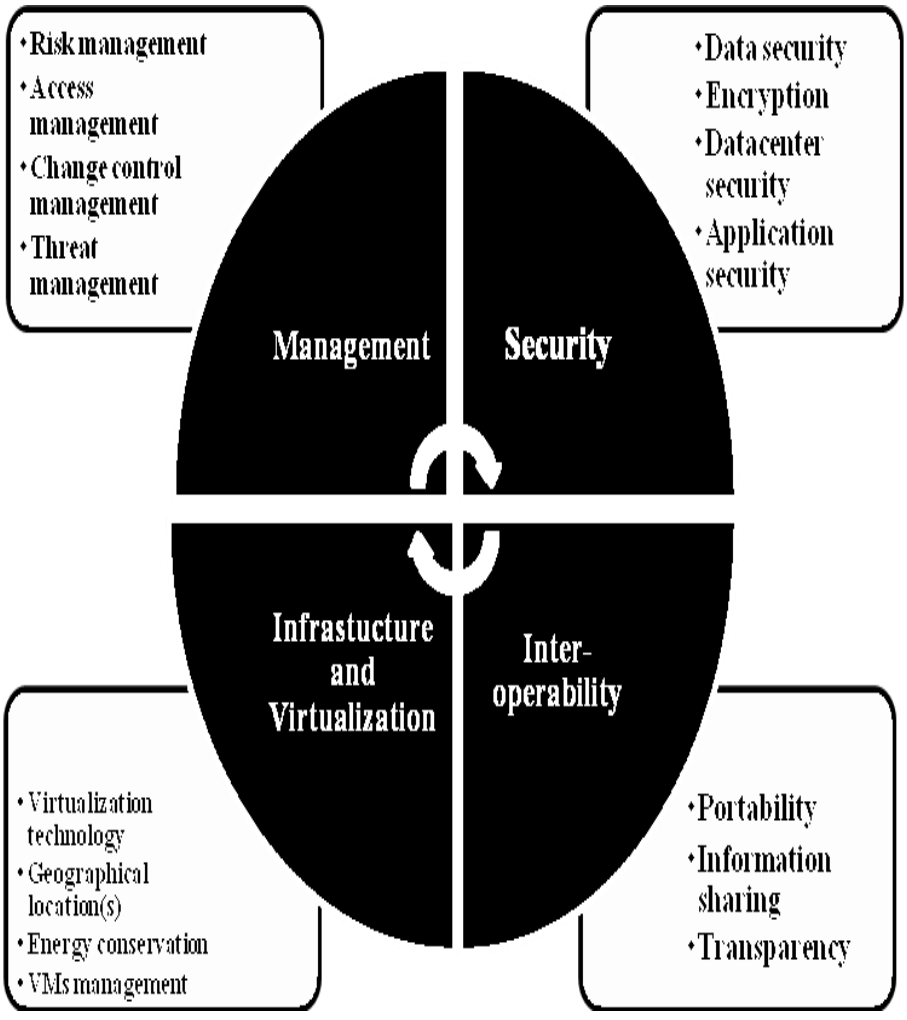

Figure 8.9 Cloud Adoption Framework- An Overview

Figure 8.9 shows our proposed framework which categorizes the cloud adoption process into four broad domains-

- Management- This domain explains of policies, procedures and measures implemented by a cloud service provider to deal with business management, risks and possible threat scenarios. It also documents change control management policies, operational maintenance and identity-access issues.

- Security- Various security features like data security, datacenter security, application and interface security, encryption, key management and communication-related security are taken care of by this domain.

- Infrastructure and Virtualization- Infrastructure encloses resources lifecycle management, equipments maintenance, cooling technologies, energy conservation techniques along with a complete knowledge of an organization's data locations. Virtualization technology employed by a cloud provider gives a clear idea to an organization about the level of performance expected. It also assesses a cloud provider's capacity for future resources requirements.

- Interoperability- For inter data centres transfers of cloud data, certain portability and architectural issues may arise, which are taken care of by this domain. Use of standard network protocols and maintaining audit logs is crucial for data integrity.

To the best of our knowledge, the above mentioned four domains and their sub-categories encompass major issues and risk factors that are crucial for selecting a cloud provider on its abilities and performance.

8.3.1 Comparison of providers based on the framework- This section presents a comparative study of major cloud providers on different factors as outlined by the cloud adoption framework. We have considered five cloud service providers, hypothetically as P1, P2, P3, P4 and P5, along with their offerings in Table 8.4.

Table 8.4: Cloud Providers and their services offerings (Sample)

Framework Domains		P1	P2	P3	P4	P5
Security	Downtime (in hours)	39.77	7.52	4.46	2.6	2.41
	Private Online Backup	No	No	Yes	No	Yes
Infrastructure & Virtualization	Total Instance Options	24	35	15	10	14
	Max Cores	40	32	49.12	16	52
Management	Resource over-provisioning	Yes	Yes	No	Yes	No
	Implementation Complexity	Yes	No	No	Yes	Yes
Inter-operability	Network Reliability (%)	28	54	67	9	33
	Cross-departmental Analysis (%)	22	78	54	45	7

Based on the example given in table 8.4, a comparative analysis of these five cloud providers is given below.

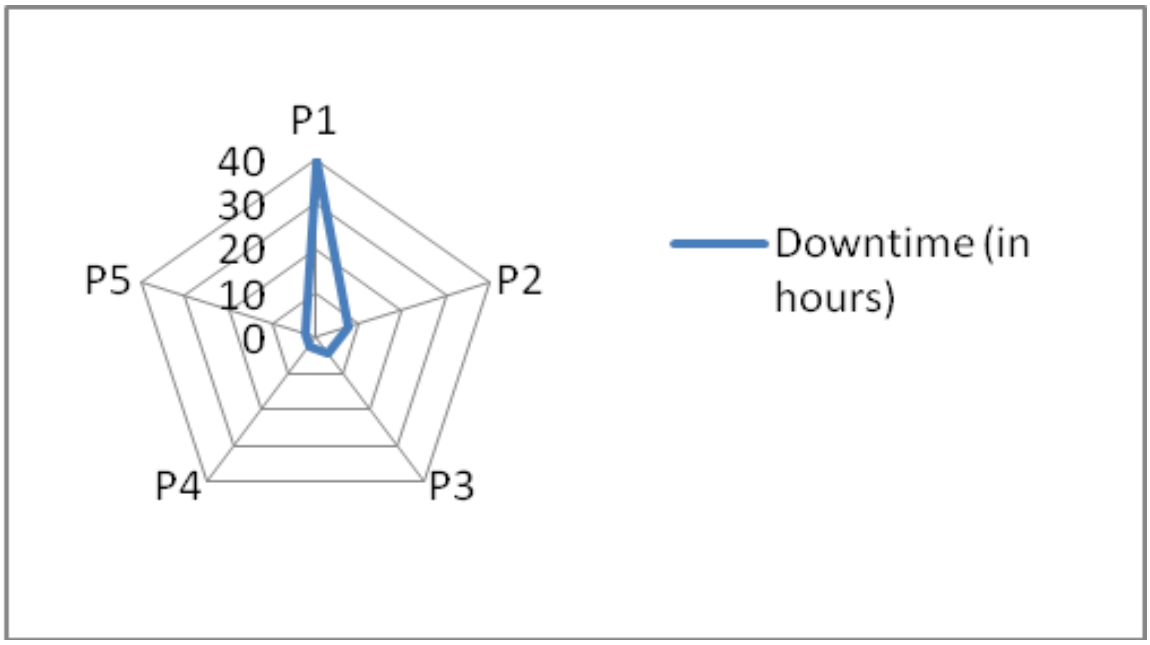

Figure 8.10 Downtime (in hours)

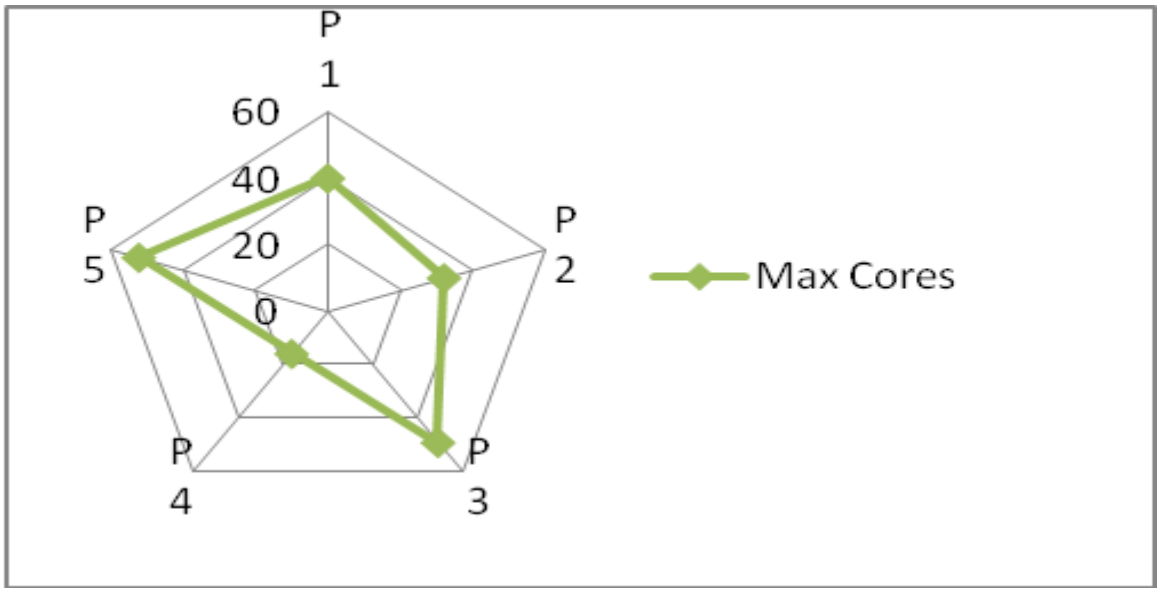

Figure 8.11 Max Cores

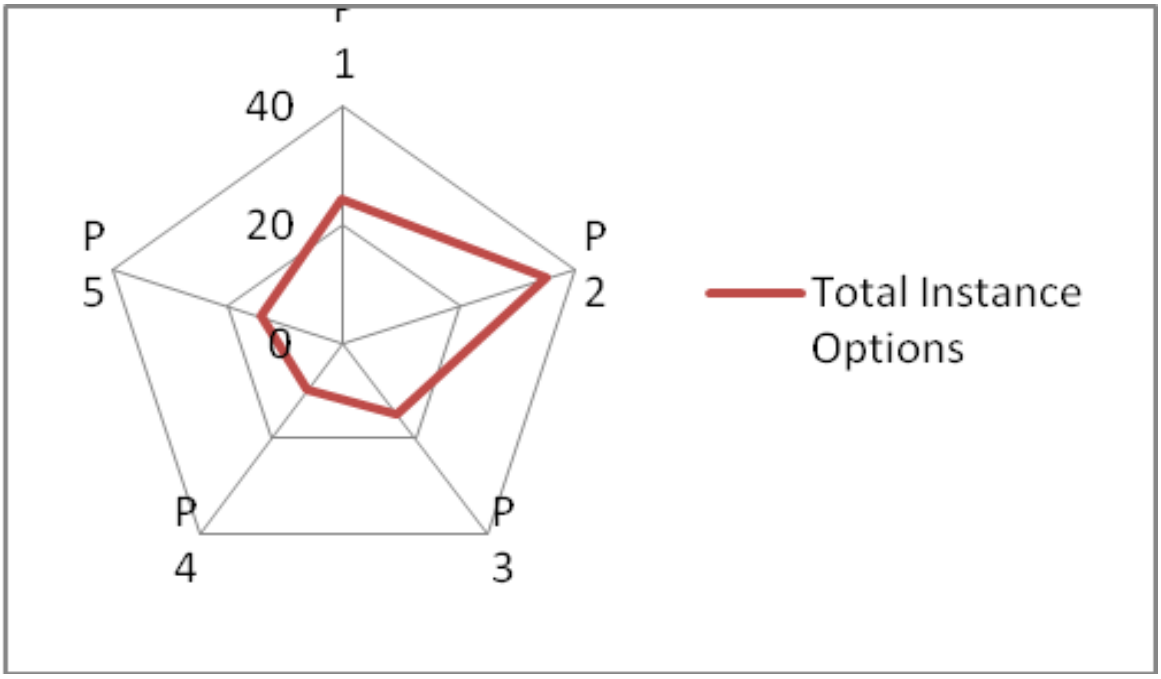

Figure 8.12 Total Instance Options

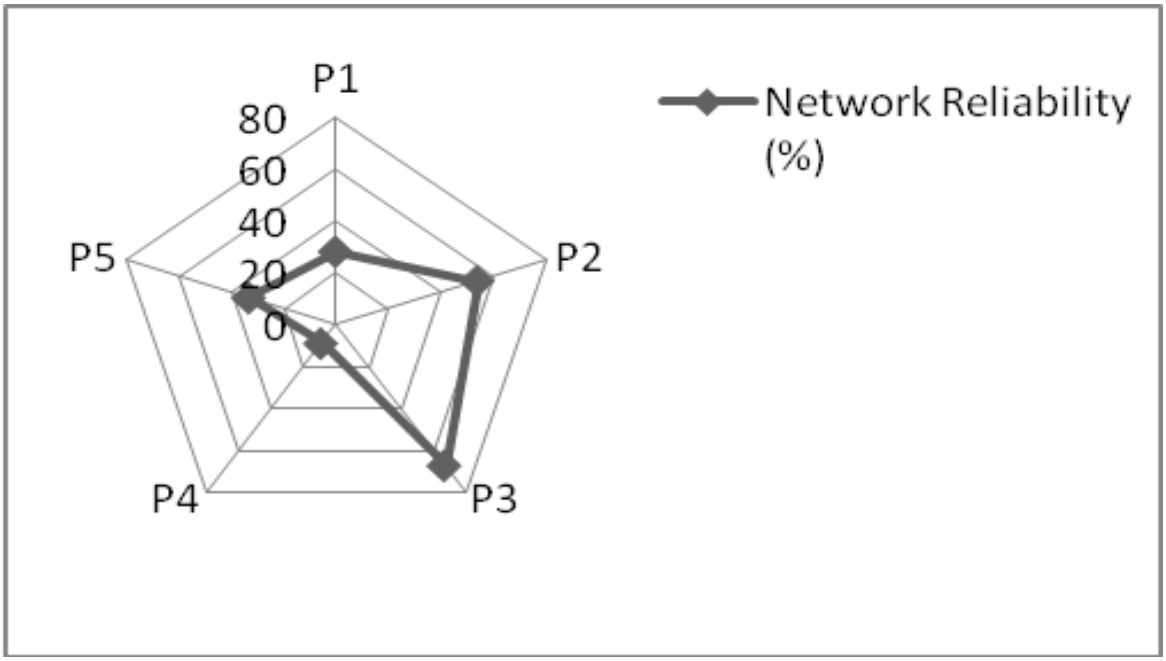

Figure 8.13 Network Reliability

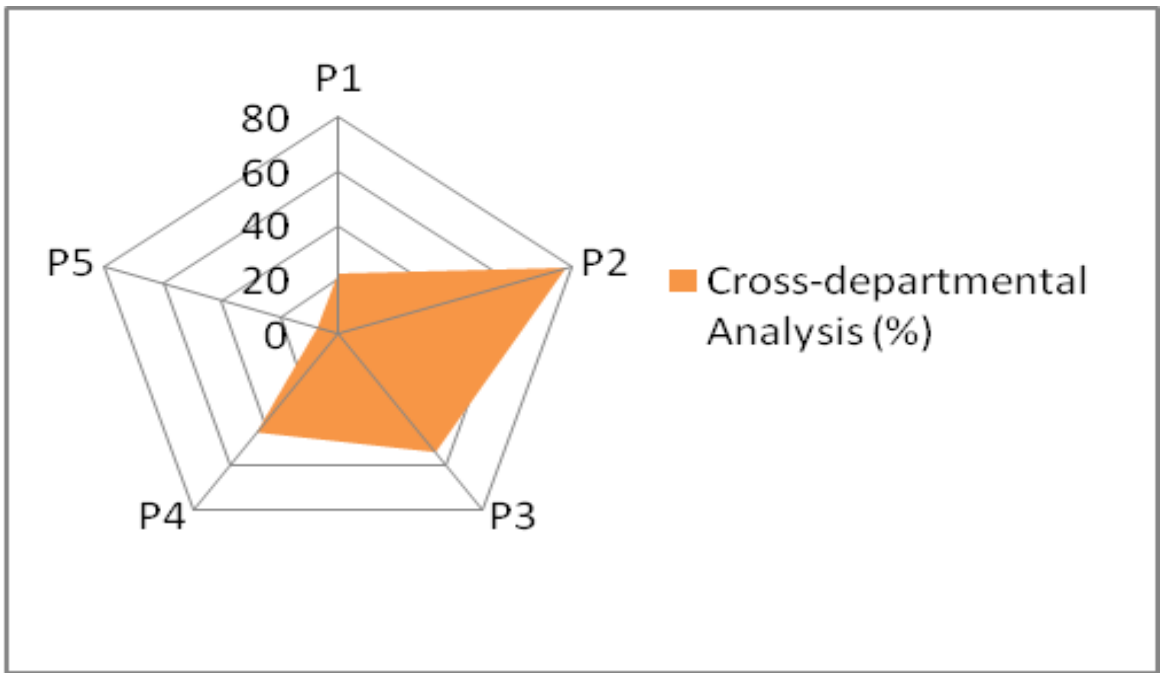

Figure 8.14 Cross-departmental Analysis

Each figure from 8.10 to 8.14 indicates a preference of cloud providers based on the framework domains, as an example, figure 8.13 shows cloud provider P3 as the preferred choice over other providers when network reliability is taken into consideration. Such analysis gives a clear insight to an organization about which cloud provider to attach its business. A detailed understanding and acceptance of the presented framework helps in better control between a cloud vendor and a business organization.

CHAPTER IX

Conclusions and Future Perspectives

CHAPTER IX

CONCLUSIONS AND FUTURE PERSPECTIVES

Cloud Computing has emerged as a new and promising era of computing paradigm where web-based services behave like commodities and are delivered to the users in a manner that is similar to other utilities like water, telephony and electricity. Research studies in the last five years, reveal that in the next decade, common users as well as business enterprises in different parts of the world, shall adopt cloud platforms for obtaining cost effective and scalable computing resources as metered services over the Internet and save themselves from the burden of owning the resources. The assurance on quality of services is given by the service provider in the form of a service level agreement signed between the consumer and the provider. Such a computing paradigm will lead us to a high-speed networked world where all the data and resources reside on the 'cloud' and accessed by 'hardware-thin' clients.

In this direction, a number of cloud service providers or cloud vendors are actively offering applications, development platforms and infrastructure resources through well-defined models of deploying cloud-based services. Regardless, the type of deployment model, the cloud vendors are backed up by huge data centers, where large scale virtualization techniques enables multiplexing of physical resources among the world-wide users. Thus, it becomes evident that the real benefits of the cloud technology can be enjoyed by cloud providers as well as the customers, only when well-structured and efficient management of resources is achieved cloud data centers.

In the present state, there is no standard mechanism or reference model for resource management, existing in the cloud industry. The common objective in all the data centers is to minimize the number of active physical machines on which multiple

virtual machines are running. The reason behind their objective is to obtain maximum utilization of the resources while minimizing the energy consumption in the data center. In this perspective different techniques for solving various resource management problems like resource scheduling, resource allocation, load balancing, virtual machine migration and consolidation, maximizing quality of services, varies from vendor to vendor. Thus, there is great scope of research work in the area of resource management in the cloud environment.

In this research work, several sub-problems of cloud resource management have been studied and different frameworks have been developed for solving them. The techniques used in the frameworks are scalable and dynamic in nature and efficiently manage cloud resources.

The first chapter gives the introduction to cloud computing and its characteristics and the second chapter provides a review of literature. In the third chapter, a virtual machine migration process is thoroughly discussed along with its analysis and a novel hybrid migration technique is presented with less migration volume and reduced service downtime. Following points are the main highlights of the third chapter.

- Two phased live virtual machine migration process is thoroughly analyzed mathematically, and it is concluded the performance driving factors are VM size, bandwidth and application's dirty rate.
- Mathematical formulations of migration time and migration volume are derived with details on iterative process.
- A location-aware hybrid VM migration is presented with benefits over traditional migration policies, like reduced energy consumption, lesser migration time and delay.

- Thus it can be safely said that an efficient and intelligent VM migration strategy can assist in optimum resource scheduling and allocation in cloud.

For virtual machine placement issue, a profitability and load-aware placement scenario is presented in the fourth chapter. Following points summarize the content of the placement problem.

- It is observed from the experiments that Analytical Hierarchy Process (AHP) can be used as a powerful tool for complex decision-making problems of select the most suitable server for an incoming virtual machine for placement.
- There are very few instances of applying AHP for VM placement in cloud data centres. Four important metrics were discussed against which a host server is selected for placement.
- Use of clustered architecture in cloud data centres and restricting placement within a cluster ensured fewer service violations, reduced energy consumption and an efficient resource utilization.
- The proposed solution is distributed in nature, which is most suitable for the cloud environment and is highly scalable.

In the fifth chapter, resource allocation problem is discussed and a profit ensuring auction mechanism is proposed for a single cloud service provider and N cloud users. The cloud provider allocates VMs in accordance with a user's payment capacity and service level preferences.

- The proposed allocation technique works in two steps, namely- preauction and market-driven open auction.
- Due consideration is given to real-time auction schemes and the proposed technique follows the supply-demand ratio, ensuring a fair and proper distribution of cloud's resources.

- In order to bring fair deal to the users, the auction is played in two rounds and rejected bidders are given an additional attempt to rebid but with an increased price. This will not only increase the profitability of the service provider but will also be fair to users.
- Simulation experiments show that the proposed allocation strategy allows a winner to pay an amount which is considerably lesser than his bid price. This payment strategy works in favour of cloud users if they quote their 'best' bid price to win the resources auction.
- The proposed resource allocation scheme works fairly well against the existing and very popular VCG auction mechanism.

In the sixth chapter, a token-based model for job scheduling is given which ensures fairness by allocating user tasks to resources based on a job's token value. It is further complemented by a predictive scheduling to match a user's demands with resource's supply to maintain a standard level of resources utilization in a cloud environment.

- The token-based predictive scheduling mechanism presented in this chapter focuses on the job's requirements, with a performance of decreased turnaround time and waiting time. It is compared to FCFS schedule on a performance scale.
- The presented technique guarantees fairness to cloud users as it balances the demand curve with allocation frequency.

  It is a well-known fact that for a scheduling technique to be efficient, it is required that a cloud data centre accepts an intelligent assortment of incoming cloud jobs/requests. For this, every data center applies some admission control mechanism to judicially accepting or rejecting the incoming service requests.

Such a multi-criteria based admission control strategy is presented in the seventh chapter.

- It is observed that an ideal admission control mechanism must expand the acceptance rate of service requests.
- Presented work introduces a multi-key based admission control mechanism by comparing and ranking the incoming requests on more than one decision key.
- Decision keys considered in this chapter are certain performance enhancing parameters like load imbalance, throughput and resources availability.

Finally, chapter eight discusses the vital factors which effect the power consumption in modern cloud data centres and presents a clear adoption framework for comparing the performance of service providers on various infrastructural and service issues.

- Two modern and widely implemented data centre architectures, fat-tree and DCell, are considered to study the factors which consume the maximum power and changes are proposed to reduce the excessive energy expenditure.
- It is observed that among all the elements inside a data centre, computing servers and communication switches offer the maximum scope to fine tune the energy conservation.
- The energy-conservation models proposed are practical, efficient and feasible solutions to the cloud's most current issue of reducing carbon footprints.
- This chapter also introduces general framework to help the cloud users in assessing the policies, practices and services offered by various cloud providers and also maintains a proper documentation eliminating any chances of doubt or concern.

- Presented framework works well for all cloud delivery models- public, private and hybrid clouds and also conforms to the standards of various service models in cloud computing, i.e., IaaS, PaaS and SaaS.

The future perspectives of the present work in cloud resource management lead to many directions.

- The present work can be extended to resource management for interoperable clouds so that users can shift their acquired resources from one cloud to another seamlessly, without any issues in providing cloud services.
- The proposed resource management framework can be applied to various data centre architectures to enhance their existing resource management system that manages the uneven workload while establishing the quality standards.
- The present research work can also aid in the effective pricing of various virtual resources to ensure maximum profits to both the cloud consumer as well as the service supplier.

Future of computing and communication will witness an explosion of data and information where everything will be considered as a resource like people, processes, data and time. Hence, in the projected domain also, the present work finds wide application in the management of new forms of resources.

References

REFERENCES

[1] Wood T, Shenoy P, Venkataramani A, Yousif M, "Black-box and Gray-box Strategies for Virtual Machine Migration", *4th USENIX Symposium on Networked Systems Design USENIX Association & Implementation, 2007.*

[2] Strunk A, Dargie W, "Does Live Migration of Virtual Machines cost Energy?", 27th*IEEE International Conference on Advanced Information Networking and Applications, 2013.*

[3] P.B. Diego, "A Brief Tutorial on Live Virtual Machine Migration from a Security Perspective", *International Workshop on Security in Cloud, 2013.*

[4] Al-Kiswany S, Subhraveti D, Sarkar P, Ripeanu M, "VMFlock: Virtual Machine Co-Migration for the Cloud", 20th*International Symposium on High Performance Distributed Computing, 2011.*

[5] Jin H, Liu H, Liao X, Hu L, Li P, "Live Migration of Virtual Machine Based on Full-System Trace and Replay", 18th*ACM international symposium on High Performance Distributed Computing, 2009.*

[6] Hines MR, Gopalan K, "Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self -Ballooning", *ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, 2009.*

[7] Stage A, Setzer T, "Network-aware migration control and scheduling of differentiated virtual machine workloads", *ICSE Workshop on Software Engineering Challenges of Cloud Computing, 2009.*

[8] Gustafsson E, "Optimizing Total Migration Time in Virtual Machine Live Migration", 2013. Online at www.diva-portal.org

[9] Hirofuchi T, Ogawa H, Nakada H, Itoh S, Sekiguchi S, "A Live Storage Migration Mechanism over Wan for Relocatable Virtual Machine Services on Clouds", *9thIEEE/ACM International Symposium on Cluster Computing and Grid*, pp. 460-465, 2009.

[10] Khosravi A, Garg S, Buyya R, "Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers", *19thInternational Conference on Parallel Processing, 2013.*

[11] Beloglazov A, Buyya R, "Energy Efficient Allocation of Virtual Machines in Cloud Data Centers", 1*0thIEEE/ACM International Symposium on Cluster Computing and Grid*, pp. 577-578, 2010.

[12] Graubner P, Schmidt M, Freisleben B, "Energy-Efficient Virtual Machine Consolidation", *IT Professional*, vol. 15(2), pp. 28-34, 2013.

[13] Bobroff N, Kochut A, Beaty K, "Dynamic Placement of Virtual Machines for Managing SLA Violations", *10thIFIP/IEEE International Symposium on Integrated Network Management*, 2007.

[14] Ferreto T, Netto M, Calheiros R, Rose CD, "Server Consolidation with Migration Control for Virtualized Data Centers", *Future Generation Computer Systems Journal*, vol. 27(8), pp. 1027-1034, 2011.

[15] Voorsluys W, Broberg J, Venugopal S, Buyya R, "Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation", 1st*International Conference on Cloud Computing, 2009.*

[16] Boru D, Kliazovich D, Granelli F, Bouvry P, Zomaya AY, "Energy-Efficient Data Replication in Cloud Computing Datacenters", *Workshop on Cloud Computing Systems, Networks, and Applications,* vol. 18(1), pp. 385-402, 2013.

[17] Deng W, Liu F, Jin H, Liao X, "Lifetime or Energy: Consolidating Servers with Reliability Control in Virtualized Cloud Datacenters", *4th IEEE conference on Cloud Computing Technology and Science*, pp. 18-25, 2012.

[18] Jin H, Liu H, Liao X, Hu L, Li P, "Live Migration of Virtual Machine Based on Full-System Trace and Replay", 18th*ACM International Symposium on High Performance Distributed Computing,* pp. 101-110, 2010.

[19] Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., Zomaya, A Y., "Energy-efficient data replication in cloud computing datacenters", *IEEE Globecom Workshops (GC Wkshps)*, pp. 446-451, 2013.

[20] Verma A, Ahuja P, Neogi A, "pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems", *ACM/IFIP/USENIX 9thInternational Middleware Conference*, pp. 243-264, 2008.

[21] Anh Vu Do, J Chen, C Wang, YC Lee, AY Zomaya, Bing Bing Zhou, "Profiling Applications for Virtual Machine Placement in Clouds", *IEEE 4th International Conference on Cloud Computing*, pp.660-667, 2011.

[22] Gandhi A, Gupta V, Harchol BM, Kozuch MA, "Optimality Analysis of Energy-Performance Trade-Off for Server Farm Management", *Journal of Performance Evaluation*, vol. 67(11), pp. 1155-1171, 2010.

[23] Deng W, Liu F, Jin H, Liao X, Liu H, "Reliability-aware server consolidation for balancing energy-lifetime tradeoff in virtualized cloud datacenters", *International Journal of Communication Systems*, vol. 27, pp. 623–642, 2014.

[24] Huang D, Yi L, Song F, Yang D, Zhang H, "A secure cost-effective migration of enterprise applications to the cloud", *International Journal of Communication Systems*, 2013.

[25] Jamshidi P, Ahmad A, Pahl C, "Cloud Migration Research: A Systematic Review", *IEEE Transactions on Cloud Computing*, vol.1(2), pp. 142-157, 2013.

[26] Mishra M, Das A, Kulkarni P, Sahoo A, "Dynamic resource management using virtual machine migrations", *IEEE Communications Magazine*, vol.50(9), pp. 34-40, 2012.

[27] Mastroianni C, Meo M, Papuzzo G, "Probabilistic Consolidation of Virtual Machines in Self-Organizing Cloud Data Centers," *IEEE Transactions on Cloud Computing*, vol. 1(2), pp.215-228, 2013.

[28] Li H, Wang J, Peng J, Wang J, Liu T, "Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centres", *China Communications*, vol. 10(12), pp.114-124, 2013.

[29] Riteau P, Morin C, Priol T, "Shrinker: efficient live migration of virtual clusters over wide area networks", *Concurrency Computation*, vol.25, pp. 541–555, 2013.

[30] Isci C, Liu J, Abali B, Kephart JO, Kouloheris J, "Improving server utilization using fast virtual machine migration", *IBM Journal of Research and Development*, vol.55(6), pp. 1-12, 2011.

[31] K. Shinji, M. Yasuhide, "Using Model Checking to Evaluate Live Migrations", *IT Professional*, vol.15(2), pp. 36-41, 2013.

[32] Yangyang W, Ming Z, "Performance Modeling of Virtual Machine Live Migration", *IEEE conference on Cloud Computing*, pp.492-499, 2011.

[33] Akoush S, Sohan R, Rice A, Moore AW, Hopper A, "Predicting the Performance of Virtual Machine Migration", *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, pp.37-46, 2010.

[34] Anala MR, Kashyap M, Shobha G, "Application performance analysis during live migration of virtual machines", *3rd IEEE International Conference on Advance Computing*, pp.366-372, 2013.

[35] Kejiang Y, Dawei H, Xiaohong J, Huajun C, Shuang W, "Virtual Machine Based Energy-Efficient Data Center Architecture for Cloud Computing: A Performance Perspective", *IEEE/ACM International Conference on Cyber, Physical and Social Computing Green Computing and Communications*, pp.171-178, 2010.

[36] Yangyang W,Ming Z, "Performance Modeling of Virtual Machine Live Migration", *IEEE 4th International Conference on Cloud Computing*, pp.492-499, 2011.

[37] Zhibo C, Shoubin D, "An energy-aware heuristic framework for virtual machine consolidation in Cloud computing", *Journal of Supercomputing*, pp.429-451, 2014.

[38] Xiaodong L, Weiqin T, Xiaoli Z, Fu Z, Liao W, "Performance analysis of cloud computing services considering resources sharing among virtual machines", *Journal of Supercomputing*, pp.357-374, 2014.

[39] Gültekin A, Vehbi CG, "Performance evaluation of cloud computing platforms using statistical methods", *Computers and Electrical Engineering Journal*, pp.1636-1649, 2014.

[40] Eric F, Ali G, Christos-Alexandros P, "Strategy proof allocation of discrete jobs on multiple machines", *15thACM conference on Economics and Computation*, pp.529-546, 2014.

[41] Zengxiang L, Xiaorong L, Long W, Wentong C, "Hierarchical resource management for enhancing performance of large-scale simulations on data centers", *2ndACM SIGSIM/PADS Conference on Principles of Advanced Discrete Simulation*, pp. 187-196, 2014.

[42] Hai J, Li D, Song W, Xuanhua S, Hanhua C, Xiaodong P, "MECOM: Live migration of virtual machines by adaptively compressing memory pages", *Future Generation Computer Systems Journal*, vol. 38, pp.23-35, 2014.

[43] Garg SK, Adel NT, Srinivasa KG, Buyya R, "SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter", *Journal of Network and Computer Applications*, vol. 45, pp. 108-120, 2014.

[44] J Zheng, Sripanidkulchai K, Z Liu, "Pacer: A Progress Management System for Live Virtual Machine Migration in Cloud Computing", *IEEE Transactions on Network and Service Management*, vol. 10(4), pp. 369-382, 2013.

[45] Shihong Z, Xitao W, Kai C, Shan H, Yan C, Yongqiang L, Yong X, Chengchen H, "VirtualKnotter: Online virtual machine shuffling for congestion resolving in virtualized datacenter", *Computer Networks Journal*, vol. 67, pp. 141-153, 2014.

[46] Balazs G, Zoltan V, Yutaka I, "Utilizing memory content similarity for improving the performance of highly available virtual machines", *Future Generation Computer Systems Journal*, vol. 29(4), pp. 1085-1095, 2013.

[47] Zaman S, Grosu D, "Combinatorial Auction-Based Allocation of Virtual Machine Instances in Clouds", *IEEE 2nd International Conference on Cloud Computing Technology and Science*, pp.127-134, 2010.

[48] Khazaei H, Misic J, Misic VB, "Performance of an IaaS cloud with live migration of virtual machines", *IEEE Global Communications Conference*, pp. 2289-2293, 2013.

[49] Jiankang D, Xing J, Hongbo W, Yangyang L, Peng Z, Shiduan C, "Energy-Saving Virtual Machine Placement in Cloud Data Centers", *13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 618-624, 2013.

[50] Jiankang D, Hongbo W, Xing J, Yangyang L, Peng Z and Shiduan C, "Virtual Machine Placement for Improving Energy Efficiency and Network Performance in IaaS Cloud", *33rd IEEE International Conference on Distributed Computing Systems Workshops*, pp. 238-243, 2014.

[51] Anshul G, Varun G, More HB, Michael AK, "Optimality analysis of energy-performance trade-off for server farm management", *Performance Evaluation Journal*, vol. 67(11), pp. 1155-1171, 2010.

[52] Yongqiang G, Haibing G, Zhengwei Q, Yang H, Liang L, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing", *Journal of Computer and System Sciences*, vol. 79(8), pp. 1230-1242, 2013.

[53] Boru D, Kliazovich D, Granelli F, Bouvry P, Zomaya AY, "Energy-efficient data replication in cloud computing datacenters", *IEEE Globecom Workshop*, pp. 446-451, 2013.

[54] Jenn-Wei L, Chien-Hung C,Chang JM, "QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems", *IEEE Transactions on Cloud Computing*, vol. 1(1), pp. 101-115, 2013.

[55] Kangkang L, Huanyang Z, Jie W, "Migration-based virtual machine placement in cloud systems", *IEEE 2nd International Conference on Cloud Networking*, pp. 83-90, 2013.

[56] Xiang S, Jicheng S, Ran L, Jian Y, Haibo C, "Parallelizing live migration of virtual machines", *9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pp. 85-96, 2013.

[57] Tiago CF, Marco AS Netto, Rodrigo N. Calheiros, César AF, De Rose, "Server consolidation with migration control for virtualized data centers", *Future Generation Computer Systems Journal*, vol. 27(8), pp. 1027-1034, 2011.

[58] Jamshidi P, Ahmad A, Pahl C, "Cloud Migration Research: A Systematic Review", *IEEE Transactions on Cloud Computing*, vol. 1(2), pp. 142-157, 2013.

[59] Wubin L, Johan T, Erik E, "Virtual machine placement for predictable and time-constrained peak loads", *8th International Conference on Economics of Grids, Clouds, Systems, and Services*, pp. 120-134, 2011.

[60] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, D. Pendarakis, "Efficient resource provisioning in compute clouds via VM multiplexing", *7th International Conference on Autonomic Computing*, pp. 11-20, 2010.

[61] Weiming S, Bo H, "Profitable Virtual Machine Placement in the Data Center", *4th IEEE International Conference on Utility and Cloud Computing*, pp. 138-145, 2011.

[62] HJ Hong, DY Chen, CY Huang, KT Chen, CH Hsu, "QoE-aware virtual machine placement for cloud games", *12thAnnual Workshop on Network and Systems Support for Games*, pp. 1-2, 2013.

[63] HY Kao, YM Yang, CH Huang, "Dynamic virtual machines placement in a cloud environment by multi-objective programming approaches", *International Conference on Intelligent Informatics and Biomedical Sciences*, pp. 364-365, 2015.

[64] SK Addya, AK Turuk, B Sahoo, M Sarkar, "A hybrid queuing model for Virtual Machine placement in cloud data center", *IEEE International Conference on Advanced Networks and Telecommunications Systems*, pp. 1-3, 2015.

[65] N Khalilzad, HR Faragardi, T Nolte, "Energy-Aware Placement of Real-Time Virtual Machines in a Cloud Data Center", *High Performance Computing and Communications Journal*, pp. 1657-1662, 2015.

[66] D Liu, X Sui, L Li, "An energy-efficient virtual machine placement algorithm in cloud data center", *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 719-723, 2016.

[67] S Yousefian, AH Zadnavin, "Scheduling Virtual Machines in Cloud Computing", MAGNT Research Report, Vol. 3(1), pp. 389- 397, 2015.

[68] F Nadeem, "Towards Comparative Evaluation of Cloud Services", MAGNT Research Report, Vol. 2(5), pp. 61-68, 2014.

[69] Randles M, Lamb D, Taleb-Bendiab A, "A comparative study into distributed load balancing algorithms for cloud computing", *IEEE 24th International Conference on Advanced Information Networking and Applications*, pp. 551–556, 2010.

[70] Kansal NJ, Chana I, "Cloud load balancing techniques: A step towards green computing", *International Journal of Computer Science*, vol. 9(1), pp. 238–246, 2012.

[71] Hu J, Gu J, Sun G, Zhao T, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment", *IEEE 3rd International Symposium on Parallel Architectures, Algorithms and Programming*, pp. 89–96, 2010.

[72] Wen WT, Wang CD, Wu DS, Xie YY, "An aco-based scheduling strategy on load balancing in cloud computing environment", *IEEE 9th International Conference on Frontier of Computer Science and Technology*, pp. 364–369, 2015.

[73] Ni J, Huang Y, Luan Z, Zhang J, Qian D, "Virtual machine mapping policy based on load balancing in private cloud environment", *IEEE International Conference on Cloud and Service Computing*, pp. 292–295, 2011.

[74] Tian W, Zhao Y, Zhong Y, Xu M, Jing C, "A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters", *IEEE International Conference on Cloud Computing and Intelligence Systems*, pp. 311–315, 2011.

[75] Thiruvenkadam T, Kamalakkannan P, "Energy efficient multi dimensional host load aware algorithm for virtual machine placement and optimization in cloud environment", *Indian Journal of Science and Technology*, vol. 8(17), 2015.

[76] Cho KM, Tsai PW, Tsai CW, Yang CS, "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing", *Neural Computing and Applications*, Vol. 26(6), pp. 1297–1309, 2015.

[77] Y. O. Yazir et al., "Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis", *IEEE 3rd International Conference on Cloud Computing*, pp. 91-98, 2010.

[78] M. N. Bennani, D. A. Menasce, "Resource Allocation for Autonomic Data Centers using Analytic Performance Models", *2nd International Conference on Automatic Computing*, pp. 229–240, 2005.

[79] R Das, J Kephart, I Whalley, P Vytas, "Towards Commercialization of Utility-based Resource Allocation", *IEEE International Conference on Autonomic Computing*, pp. 287–290, 2006.

[80] N Bobroff, A Kochut, K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations", *10th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 119–128, 2007.

[81] A. Kochut, "On Impact of Dynamic Virtual Machine Reallocation on Data Center Efficiency", *IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, pp. 1-8, 2008.

[82] HN Van, FD Tran, "Autonomic Virtual Resource Management for Service Hosting Platforms", *International Conference on Software Engineering, IEEE Computer Society*, pp. 1–8, 2009.

[83] RW Saaty, "The analytic hierarchy process—what it is and how it is used, Mathematical Modelling", vol. 9(3), pp. 161-176, 1987, http://dx.doi.org/10.1016/0270-0255(87)90473-8.

[84] Jain N, Menache I, Naor J, Yaniv J, "A Truthful Mechanism for Value-Based Scheduling in Cloud Computing", *Journal Springer Theory of Computing Systems*, Vol. 54, pp.388-406, 2014.

[85] L Wu, SK Garg, R Buyya, "SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments", *Journal of Computer and System Sciences, vol.* 78(5), pp. 1280-1299, 2012.

[86] Y Xiao, C Lin, Y Jiang, X Chu, X Shen, "Reputation-Based QoS Provisioning in Cloud Computing via Dirichlet Multinomial Model", *IEEE International Conference on Communications*, pp. 1-5, 2010.

[87] H Fu, Z Li, C Wu, X Chu, "Core-selecting Auctions for Dynamically Allocating Heterogeneous VMs in Cloud Computing", *IEEE 7th International Conference on Cloud Computing*, pp. 152-159, 2012.

[88] X Shi, K Xu, J Liu, Y Wang, "Continuous Double Auction Mechanism and Bidding Strategies in Cloud Computing Markets"**,** *IEEE Transactions on Cloud Computing*, vol. 1307, 2013.

[89] W Wang, B Liang, B Li, "Revenue maximization with dynamic auctions in IaaS cloud markets", *IEEE/ACM 21st International Symposium on Quality of Service*, pp. 1-6, 2013.

[90] N Ani Brown Mary et al, "An Extensive Survey on QoS in Cloud computing", *International Journal of Computer Science and Information Technologies*, vol. 5(1), pp. 1-5, 2013.

[91] C A Yfoulis, A Gounaris, "Honoring SLAs on cloud computing services: a control perspective", *European Control Conference*, pp. 184-189, *2009.*

[92] A Dhok, N Maheshwari, V Varma, "Learning based opportunistic admission control algorithm for MapReduce as a service", ACM 3rd*India Software Engineering Conference*, pp. 153-160, 2010.

[93] Bo An, V Lesser, D Irwin, M Zink, "Automated negotiation with decommitment for dynamic resource allocation in cloud computing", 9th*International Conference on Autonomous Agents and Multi agent Systems*, pp. 981-988, 2010.

[94] Chaisiri S, B S Lee, Niyato D, "Optimization of Resource Provisioning Cost in Cloud Computing", *IEEE Transactions on Services Computing*, vol. 5(2), pp. 164-177, 2012.

[95] H Qian, D Medhi, "Server operational cost optimization for cloud computing service providers over a time horizon", *11th USENIX conference on Hot topics*

in management of internet, cloud, and enterprise networks and services, pp. 4-4, 2011.

[96] H Zhang, Bo Li, H Jiang, F Liu, Vasilakos A.V., JLiu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands", *IEEE INFOCOM*, pp. 1510-1518, 2013.

[97] Makkes MX, Taal A, Osseyran A, Grosso P, "A decision framework for placement of applications in clouds that minimizes their carbon footprint", *Journal of Cloud Computing*, vol. *2*(1), pp. 1-13, 2013.

[98] Chen X, Zhang Y, Huang G, Zheng X, Guo W, Rong C, "Architecture-based integrated management of diverse cloud resources", *Journal of Cloud Computing*, vol. *3*(1), pp. 1-15, 2014.

[99] Sithole E, McConnell A, McClean S, Parr G, Scotney B, Moore A, Bustard D, "Cache performance models for quality of service compliance in storage clouds", *Journal of Cloud Computing*, vol. *2*(1), pp. 1-24, 2013.

[100] Waddington S, Zhang J, Knight G, Jensen J, Downing R, Ketley C, "Cloud repositories for research data–addressing the needs of researchers", *Journal of Cloud Computing*, vol. *2*(1), pp. 1-27, 2013.

[101] Agarwal A, Jain S, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", *International Journal of Computer Trends and Technology*, vol. 9, pp. 345-349, 2014.

[102] Liu J, Luo XG, Zhang XM, Zhang F, Li BN, "Job scheduling model for cloud computing based on multi-objective genetic algorithm", *IJCSI International Journal of Computer Science Issues*, vol. 10(1), pp. 134-139, 2013.

[103] Kansal NJ, Chana I, "Cloud load balancing techniques: A step towards green computing", *International Journal of Computer Science*, vol. 9(1), pp. 238–246, 2012.

[104] Gebai M, Giraldeau F, Dagenais MR, "Fine-grained preemption analysis for latency investigation across virtual machines", *Journal of Cloud Computing: Advances, Systems and Applications*, vol. *3*(1), pp. 41, 2014.

[105] Ghanbari S, Othman M, "A priority-based job scheduling algorithm in cloud computing", *Procedia Engineering*, vol. *50*, pp. 778-785, 2012.

[106] Wang X, Wang Y, Zhu H, "Energy-efficient task scheduling model based on MapReduce for cloud computing using genetic algorithm", *Journal of Computers*, vol. *7*(12), pp. 2962-2970, 2012.

[107] Maqableh M, Karajeh H, Masa'deh R E, "Job Scheduling for Cloud Computing Using Neural Networks", *Communications and Network*, vol. 6(3), pp. 191-200, 2014.

[108] Waldspurger, C. A., Weihl, W. E., "Stride scheduling: Deterministic proportional share resource management", Technical Memo MIT/LCS/TM- 5z8, *MIT Laboratory for Computer Science*, 1995**.**

[109] Waldspurger C A, Weihl W E, "Lottery scheduling: Flexible proportional-share resource management", 1st*USENIX conference on Operating Systems Design and Implementation*, 1994.

[110] Zhang Q, Cheng L, Boutaba R, "Cloud computing: state-of-the-art and research challenges", *Journal of Internet Services and Applications*, vol. 1(1), pp. 7-18, 2010.

[111] Patel P, Ranabahu A H, Sheth A P, "Service Level Agreement in Cloud Computing", Wright State University, 2009.

[112] Endo P T, Gonçalves G E, Kelner J, Sadok D, "A Survey on Open-source Cloud Computing Solutions", *8th Workshop on Clouds, Grids and Applications*, pp. 3-16, 2010.

[113] Chen Y, Iyer S, Liu X, Milojicic D, Sahai A, "SLA Decomposition: Translating Service Level Objectives to System Level Thresholds", *4thInternational Conference on Autonomic Computing*, 2007.

[114] Elnikety S, Nahum E, Tracey J, Zwaenepoel W, "A method for transparent admission control and request scheduling in e-commerce web sites", *13thinternational conference on World Wide Web*, pp. 276–286, 2004.

[115] Cherkasova L, Phaal P, "Session-Based Admission Control: A Mechanism for Peak Load Management of Commercial Web Sites", *IEEE Transactions on Computers*, vol. 51(6), pp. 669-685, 2002.

[116] Moenkeberg A, Weikum G, "Conflict-driven load control for the avoidance of data-contention thrashing", *7thInternational Conference on Data Engineering*, pp. 632-639, 1991.

[117] Heiss H, Wagne R, "Adaptive Load Control in Transaction Processing Systems", *17th International Conference on Very Large Data Bases*, pp. 47-54, 1991.

[118] Non-Linear regression model, 2014, Online at http://en.wikipedia.org/wiki/Nonlinear_regression.

[119] Tozer S, Brecht T, Aboulnaga A, "Q-Cop: Avoiding Bad Query Mixes to Minimize Client Timeouts Under Heavy Loads", *IEEE 26th International Conference on Data Engineering*, pp. 397-408, 2010.

[120] Das A, "Maximizing profit using SLA-aware provisioning", *IEEE Network Operations and Management Symposium*, pp. 393-400, 2012.

[121] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH, "The WEKA data mining software: An update", *ACM SIGKDD Explorations*, vol. 11(1), pp. 10-18, 2009.

[122] Transaction Processing Performance Council, "TPC-W Benchmark (e-commerce)", 2014. Online at http://www.tpc.org/tpcw/default.asp

[123] Haoming F, Zongpeng L, Chuan W, Xiaowen C, "Core-selecting Auctions for Dynamically Allocating Heterogeneous VMs in Cloud Computing", *IEEE 7thInternational Conference on Cloud Computing*, pp. 152-159, 2014.

[124] Ishizaka A, Philippe N, "Multi-attribute utility theory", *Multi-Criteria Decision Analysis: Methods and Software*, pp. 81-113, 2013.

[125] Xingwei Wang, Jiajia Sun, Min Huang, Chuan Wu, Xueyi Wang, "A Resource Auction Based Allocation Mechanism in the Cloud Computing Environment", *IEEE 26thInternational Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pp. 2111-2115, 2012.

[126] NIST Cloud Computing Reference Architecture, 2011.

[127] Dr. Jörg Hladjk, "Privacy and data protection", *IT compliance and IT security*, Part 1, vol. 7(4), pp. 3-4, 2016.

[128] Wei Wang, Ben Liang, Baochun Li, "Revenue maximization with dynamic auctions in IaaS cloud markets," *IEEE/ACM 21st International Symposium on Quality of Service*, pp. 1-6, 2013.

[129] S Subashini, V Kavitha, "A survey on security issues in service delivery models of cloud computing", *Journal of Network and Computer Applications*, Vol. 34(1), pp. 1–11, 2011.

[130] T Mather, S Kumarswamy, S Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", *O'Reilly Media*, 2009.

[131] Diana Kelley, "Understanding Cloud Controls Matrix v1.4.xls", CSA.

[132] Mell P, Grance T, "The NIST Definition of Cloud Computing", (Special Publication 800-145), 2011.

[133] H Xu, B Li, "Maximizing revenue with dynamic cloud pricing: The infinite horizon case", *IEEE International Conference on Communications*, pp. 2929-2933, 2012.

[134] H Zhang, Bo Li, H Jiang, F Liu, Vasilakos A V, J Liu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands", *IEEE INFOCOM*, pp. 1510-1518, 2013.

[135] Pinal Salot, "A Survey of Various Scheduling Algorithm in Cloud Computing Environment", *International Journal of Research in Engineering and Technology*, Vol. 2(2), 2013.

[136] Assunçao MD, Netto MA, Koch F, Bianchi S, "Context-aware job scheduling for cloud computing environments", *IEEE/ACM 5th International Conference on Utility and Cloud Computing*, pp. 255-262, 2012.

[137] Bilgaiyan S, Sagnika S, Das M, "An Analysis of Task Scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms", *International Journal of Computer Applications*, vol. *89*(2), pp. 11-18, 2014.

[138] Luo L, Wu W, Di D, Zhang F, Yan Y, Mao Y, "A resource scheduling algorithm of cloud computing based on energy efficient optimization methods", *International Green Computing Conference*, pp. 1-6, 2012.

[139] Chawla Y, Bhonsle M, "A Study on Scheduling Methods in Cloud Computing", *International Journal of Emerging Trends & Technology in Computer Science*, vol. 1(3), pp. 12-17, 2012.

[140] Khajeh-Hosseini A, Sommerville I, Sriram I, "Research Challenges for Enterprise Cloud Computing", *1st ACM Symposium on Cloud Computing*, 2010.

[141] Xiong P, Chi Y, Zhu S, Tatemura J, Pu C, Hacıgümüs H, "Active SLA: A Profit-Oriented Admission Control Framework for Database-as-a-Service Providers", *2nd ACM Symposium on Cloud Computing*, 2011.

[142] National Institute of Standards and Technology, Computer Security Incident Handling Guide, 2012.

[143] D. McGuinness, F.V. Harmelen, "OWL web ontology language overview", *W3C World Wide Web Consortium*, 2004.

[144] P. Kamongi, M. Gomathisankaran, K. Kavi, "Nemesis: Automated Architecture for Threat Modeling and Risk Assessment for Cloud Computing", *ASE BigData/Social Informatics/PASSAT/BioMedCom Conference*, pp. 1-10, 2014.

[145] N. Farrington, E. Rubow, A. Vahdat, "Data center switch architecture in the age of merchant silicon", *17th IEEE symposium on High Performance Interconnects*, pp. 93–102, 2009.

[146] D. Kliazovich, J.E. Pecero, A. Tchernykh, P. Bouvry, S.U. Khan, A.Y. Zomaya, "CA-DAG: Communication-Aware Directed Acyclic Graphs for Modeling Cloud Computing Applications", *IEEE International Conference on Cloud Computing (CLOUD)*, 2013.

[147] Threat Modeling and Risk Assessment for Cloud Computing, *29th IEEE/ACM Conference on Automated Software Engineering*, 2014.

[148] K. Popović, Z. Hocenski, "Cloud computing security issues and challenges", *33rd International Convention on MIPRO*, pp.344-349, 2010.

[149] R. H. Katz, "Tech Titans Building Boom", *IEEE Spectrum*, vol. 46(2), pp. 40-54, 2009.

[150] R.S. Chang, H.P. Chang, Y.T. Wang, "A dynamic weighted data replication strategy in data grids", *IEEE/ACS International Conference on Computer Systems and Applications*, pp. 414-421, 2008.

[151] Ruay-Shiung Chang, Hui-Ping Chang, Yun-Ting Wang, "A dynamic weighted data replication strategy in data grids", *IEEE/ACS International Conference on Computer Systems and Applications*, pp. 414-421, 2008.

[152] B. Lin, S. Li, X. Liao, Q. Wu, S. Yang, "eStor: Energy efficient and resilient data center storage", *International Conference on Cloud and Service Computing (CSC)*, pp. 366-371, 2011.

[153] T Horvath, T Abdelzaher, K Skadron, X Liu, "Dynamic voltage scaling in multitier web servers with end-to-end delay control", *IEEE Transactions on Computing*, vol. 56(4), pp. 444–458, 2007.

[154] D. Kliazovich, P. Bouvry, S.U. Khan, "GreenCloud: A Packet-level Simulator of Energy-aware Cloud Computing Data Centers", *Journal of Supercomputing*, vol. 62(3), pp. 1263-1283, 2012.

[155] T. Horvath, T. Abdelzaher, K. Skadron, X. Liu, "Dynamic voltage scaling in multitier web servers with end-to-end delay control", *IEEE Transactions on Computing*, vol 56(4), pp. 444–458, 2007.

[156] L. Shang, L.S. Peh, N.K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks", *9th International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 91-102, 2003.

[157] F. Ping, X. Li, C. McConnell, R. Vabbalareddy, J.H. Hwang, "Towards Optimal Data Replication Across Data Centers", *International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 66-71, 2011.

[158] M. Al-Fares, A. Loukissas, A. Vahdat, "A scalable, commodity data center network architecture", *ACM SIGCOMM*, pp. 63-74, 2008.

[159] P. Mahadevan, P. Sharma, S. Banerjee, P. Ranganathan, "A power benchmarking framework for network devices", *8th International IFIP-TC Networking Conference*, pp. 795 – 808, 2009.

[160] Bilal, Kashif, Samee Ullah Khan, Joanna Kolodziej, Limin Zhang, Khizar Hayat, Sajjad Ahmad Madani, Nasro Min-Allah, Lizhe Wang, and Dan Chen, "A Comparative Study Of Data Center Network Architectures," In *ECMS*, pp. 526-532. 2012.