



Jimma University
Jimma Institute of Technology
Faculty of Computing and Informatics

**Reducing the impact of network latency in cloud computing
environment using energy consumption load balance
scheduling mechanism**

BY
Ashenafi Hailu
Advisor: Dr. Ismael Kedir
Co-Advisor: Mr. Bekan Kitaw (M.Sc.)

**Thesis Submitted to Faculty of Computing and Informatics
Of Jimma University in Partial Fulfillment of The Requirement for The Degree of
Masters of Science in Computer Networking**

Jimma, Ethiopia
December, 2022

Approval sheet

This Independent Research entitled “Reducing the impact of network latency on cloud computing environment using energy consumption load balance scheduling mechanism” has been read and approved as meeting the preliminary research requirements of the School of Computing in partial fulfillment for the award of the degree of Master in Computer Networking, Jimma University, Jimma, Ethiopia.

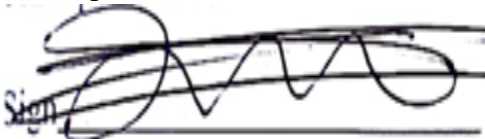
Principal Investigator: Mr. Ashenafi Hailu

Sign:

 _____

Dr. Esmael Kedir
Principal Advisor

Mr. Bekan Kitaw (MSc.)
Co-Advisor

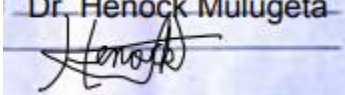
Sign:  _____

Sign:  _____

External Examiner

Name: Dr. Henock Mulugeta

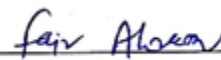
Sign:

 _____

Internal Examiner

Name: Dr. Faiz Akram

Sign:

 _____

Chair Person

Name: Kasech Tsegaye

Sign:

 _____

Abstract

Cloud Computing is one of the modern technologies in which the research Society has shown their interest. In this thesis, we have discussed about Latencies and what are the impacts of latencies on the Cloud Computing environment. A new design is proposed. The proposed design reduced the impact disturbing the cloud computing service. So that it improved the Communication speed in cloud-based service, and minimize the processing delay.

To conduct this thesis, we followed the methodology which includes determining design strategy for the study including a proposed design, defining the deliverables. The requirements to this study were: developing the load balancing mechanism using energy consumption load scheduling method and developing evaluation tool to measure the performance of the resulting the proposed design. Thus, the instruments used to develop the knowledge base to the study requirements were review of relevant literature including similar studies and implementing the proposed design using cloudsim simulator.

The thesis work is discussed and presented in table and graph format because it is important to understand the communication delay. so that results are compared between the existing cloud computing and the proposed one to show the latency difference

The time taken to process a request in every server of the proposed cloud was less than every server of the existing cloud. This indicated that its response time is low and resulting less communication delay comparing with the existing cloud. In the proposed cloud, Server1 consumed less energy than of Server2, 3 and 4 based energy consumption calculation. Since server 4 needs high energy, the load given to it would move to server 1 compared to the others which is presented in the table.

The proposed cloud computing used energy consumption load scheduling mechanism which helped to distribute the loads based on energy consumption of the servers. So, the mechanism helped proposed cloud computing to have less communication delay. Therefore, both having less response time and distributing the loads using servers' energy consumption lead to have reduced latency.

Keywords— Cloud Computing, Latency, Energy consumption, Response time

Acknowledgment

First, I would like to thank the almighty God, who helped me in any aspect of my life. I am extremely thankful to my main advisor Dr. Esmeal Kedir and co-advisor Mr. Bekan Kitaw, for their sincerity and encouragement. They have been an inspiration as I hurdled through the path of this Master's degree. This thesis would not have been possible without the guidance of both my advisors and my friends. I am thankful for their extraordinary experiences that arranged for me and for providing opportunities to grow professionally. I want to thank my wife for constant love and supports keep me motivated and confident. My accomplishments and success are because of her believed in me. Finally, I give my deepest gratitude for all supporting me in my thesis. I am forever thankful for the unconditional love and support throughout the entire thesis process and every day.

List of Figure

Fig 2. 1 An overview of Cloud Computing	5
Fig 2. 2 Architecture of Cloud Computing	6
Fig 2. 3 Layers of Latency	11
Fig 4. 1 load balancing diagram.....	22
Fig 4. 2 Architecture of the proposed design.....	24
Fig 4. 3 Hierarchy of the cloud components.....	27
Fig 4. 4 Flowchart of Our proposed design	29
Fig 5. 1 The response time of the existing cloud.....	34
Fig 5. 2 The Response time of the proposed cloud.....	36
Fig 5. 3 Comparison of latency between a cloud with load balancer and a cloud with load balancer working on energy consumption	37

List of Table

Table 2. 1 Summery of related works	17
Table 5. 1 Applications description and Simulated Parameters.....	32
Table 5. 2 The cloud devices with their parameters	33
Table 5. 3 The java file (the existing cloud computing).....	34
Table 5. 4 The java file (The remaining energy and the status of the energy consumption of each server)	35
Table 5. 5 The java file (the proposed cloud computing).....	35
Table 5. 6 Latency of the existing cloud.....	36
Table 5. 7 Latency of the proposed cloud.....	36
Table 5. 8 Comparison of latency between the existing cloud and the proposed cloud.....	37

Acronyms

API- Application programming interface

AWS- Amazon Web Services

BS-Base Station

CPU- Central Processing unit

CSV- Comma-Separated Value

DB-Database

DC-Data Center

FCFS-Fist Come First Served

FIFO-First in First Out

GAE-Google App Engine

GA-Genetic Algorithm

GPRS- General Packet Radio Service

GUI-Graphical User Interface

I/O- Input Out

IP-Internet Protocol.

IT- Information Technology

JDK- Java Development Kit

KVM- Kernel-based Virtual Machine

LTS- Long Term Support

MIPs-Million instructions per second

PDA- Personal Digital Assistant

PEs- Processing Elements

PM-Physical Machine

PSID-Provider Service Identifier

QoS-Quality of Service

RAM-Random Access Memory

SDN- Software Defined Network

SLA-Service Level Agreement

SQL-Structured Query Language

TCL-Tool Command Language

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

VM-Virtual Machine

XML: extensible markup language

Table of Contents

List of Figure.....	III
List of Table.....	IV
Acronyms.....	V
CHAPTER ONE.....	1
1 INTRODUCTION.....	1
1.1 Background.....	1
1.2 Statement of the Problem.....	2
1.3 Research Question.....	3
1.4 Objective of the study.....	3
1.4.1 General objective.....	3
1.4.2 Specific objectives.....	3
1.5 Scope and Limitation of the study.....	4
1.6 Thesis Organization.....	4
CHAPTER TWO.....	5
2 LITERATURE REVIEW.....	5
2.1 Basic Overview of Cloud computing Latency.....	5
2.2 Cloud Properties.....	6
2.3 Cloud deployment models.....	7
2.4 Cloud service models.....	7
2.5 Service Level Agreements.....	8
2.6 Why Cloud Computing?.....	8
2.7 Network latency.....	9
2.8 Sources of latency.....	10
2.9 Layers of Latency.....	11
2.10 Network Evolution.....	12
2.10.1 Connectivity and Bandwidth.....	12
2.10.2 Latency vs bandwidth vs throughput.....	12
2.11 Related Work.....	13
2.11.1 The impact of latency in different organizations/companies.....	13
CHAPTER THREE.....	18

3	Methodology.....	19
3.1	Study design.....	19
3.2	Method.....	20
3.2.1	A systematic literature reviews.....	20
3.2.2	Design and Implementation.....	20
3.3	Virtualization.....	21
	CHAPTER Four.....	22
4	DESIGN OF PROPOSED SOLUTION.....	22
4.1	Overview.....	22
4.2	The Architecture of the Proposed Solution.....	23
4.3	Load balancing model background.....	26
4.3.1	Load balancing based round Robin.....	29
	CHAPTER FIVE.....	32
5	IMPLEMENTATIONS AND RESULT EVALUATION.....	32
5.1	Overview.....	32
5.2	Load balancing metrics.....	33
5.3	Data Processing Tools.....	34
5.4	Summary.....	37
	CHAPTER SIX.....	39
6	CONCLUSION, CONTRIBUTION, AND FUTURE WORKS.....	39
6.1	Conclusion.....	39
6.2	Contribution.....	39
6.3	Future Work.....	40
7	References.....	41
	Appendix A.....	44
	Appendix B.....	45

CHAPTER ONE

1 INTRODUCTION

1.1 Background

Before coming the cloud computing, there was Client/Server computing which is essentially a centralized storage during which all the software applications, all the information and every one the controls are resided on the server side that means if a single user wants to access specific data or run a program, the user needs to connect to the server and then gain appropriate access, and then user can do his/her business. Then after, Cloud computing came into picture, where all the computers are networked together and share their resources when needed. On the premise of above computing, there was emerged of cloud computing concepts that later implemented.

The cloud service enables users to access the hardware and software managed by the third parties at remote locations. It has brought substantial transformation in the way the information is stored and accessed. The service availability is guaranteed along with unlimited scalability, cost reduction (pay per use) and much better performance than the traditional computing techniques [1] [2] [3][5].

Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [3]. It is centralized virtual machine that manages the interior and therefore the external framework for the storage, provides hardware, software platform to manage it. This technology reduces the utilization of local hardware, software for storing, sharing and retrieving data. Its important feature provides a distant access to share data. It uses different online access resources like client-server, applications of storage. Cloud computing turns into one of the most significant platforms for suppliers of cloud service to provide service requests in a virtual manner via the internet to customers [4].

Cloud computing acts as a backbone for several companies today for the reason of its greater elasticity according [5][6], user-friendliness, and volume compared to outdated online computation and storage techniques. Cloud customers across the world interchange their information with a large form of computing resources distributed through various service providers that are handling diverse categories of datasets which require to be accessible to numerous customers with dissimilar access privileges. Hence this demand for scalable computing capabilities will usually increase among the cloud consumers. As a result, single cloud server possibly will unable to discover and associate with a wide variety of capability to the application during execution time. Therefore, the scientists are in necessity to construct a virtual atmosphere for communicating various cloud servers [8].

Cloud computing is a model that allows for access everywhere, convenient to use, when there is demand for network access can be shared with the configuration of the computing resources (such as network, servers, storage, application, and service) that can be quickly established and released management with minimal effort or service provider interaction [2][3]. Cloud infrastructure refers to the hardware and software components which are servers, virtualization, storage, network, management, security, and backup & recovery [4]. Outsourcing of data center functionality and availability of desktop application online via network connection is what we term cloud computing [20]. Companies are moving to cloud computing to cut down the I.T cost having the security with less I.T staff.

The expense has been abate & traffic of network has increased to double times. The massive concern for organizations depends on the provision of knowledge, quality of the network and their performance. Primary motive behind more organization moving to cloud is that the reduction in cost and dynamic resource allocation. Since the infrastructure is hosted by the cloud providers, software enterprises don't must worry about their maintenance. Also, characteristics like Scalability, elasticity, Multitenancy, pay-per use make cloud computing the foremost wanted technology today.

Cloud Computing is still not quite widely adopted because of many factors, but mostly concerned with moving business data to be handled by third party [6], including loss of control on data, security and privacy of data, data quality and assurance, and data stewardship etc. Cloud consumers can also suffer from operational and regulatory challenges, as organization transfers their data to third parties for storage and processing [7]. It may be difficult for the consumers to check the data handling practices of the cloud provider.

1.2 Statement of the Problem

The growth of high-speed networks and computing power is making possible to process different cloud services or applications in microseconds undreamt of within the past. These rapid explosions of technology are changing the services. Cloud computing attracts users with its great elasticity and scalability of resources with a horny gag line 'pay-as-you-use' at relatively low prices. Compared to the development of their own infrastructures, customers are able to impede on expenditure significantly by migrating computation, storage and hosting onto the cloud.

Cloud computing has its own benefits; nevertheless, it's numerous issues and challenges but delay of processing information meaning latency is challenging fact as far as cloud computing got complexity. Latency is employed not only in cloud computing but in every network space since it is the delay between requesting data and receiving the information requested. Data can't move faster than the speed of sunshine, meaning that the greater the gap between two points, the longer it'll regard data to induce there.

Latency is defined, within the networking space, because the time it takes a packet to travel from source to destination (although it's often measured because the time takes from the request for information there to information being received). Meaning that latency can have a large impact on cloud computing. Therefore, it should be reduced so as to provide effective cloud computing services.

In the field of cloud computing load balancing plays a vital role as it allocate and balance the load of various resources among the distinct components and nodes on the network. The main point is to avoid the key load on the any single server. Load balancer is the main component which balances the load over the servers. The work of load balancer is to forward request to the server at backend, which again replies back to load balancer.

However, energy consumption was not considered while a load balancer distributes requests to servers. It can be highly difficult to define the proper activity that could be examined for the efficiency of energy. Energy consumption can be defined as the amount of energy used by the servers for a given service or level of activity [14]. So, a load balancer distributes requests/activities without considering their energy consumption ability. Therefore, distributing requests based on servers' energy consumption is a major concern in the large-scale data centers in order to reduce the response time.

1.3 Research Question

The research questions which are going to be answered in this thesis are:

- ✚ What is the potential impact of latency on Cloud Computing environment?
- ✚ How to design and develop the concept of load balancing using energy consumption load scheduling mechanism for the cloud computing environment?
- ✚ How to evaluate the performance of the proposed scheme using analytical studies?
- ✚ What are the future research directions for further investigation on Low latency cloud services?

1.4 Objective of the study

1.4.1 General objective

The general objective of this study is to reduce the impact of network latency on cloud computing environment

1.4.2 Specific objectives

The specific objectives of the study are:

- ❖ To explore the potential impact of latency on Cloud Computing.
- ❖ To design and develop the load balancing model using energy consumption load scheduling method for the cloud computing.
- ❖ To evaluate the performance of the proposed scheme using analytical studies
- ❖ To identify and recommend future research directions for further investigation on the factors contributing to increase network latency in cloud computing environment.

1.5 Scope and Limitation of the study

The main intent of the study is to reduce the impact of network latency on cloud computing focusing on the concept of load balancing using energy consumption load scheduling method for the cloud computing to minimize the impact of latency on cloud computing service.

The proposed design reduces the impact disturbing the cloud computing service. So that it will improve the latency awareness in cloud-based service, and minimize the processing delay. However, the cloud computing still required further enhancements for maintaining the quality of the services

However, this thesis did not address all factors those make latency very less rather it addressed the one factor, there are other factors therefore to be avoided. Also, the study did not address security of the cloud computing because of security by itself is a broad concept

1.6 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 is literature review, literatures are discussed in four categories which are basic overview of cloud computing latency, cloud properties, Service level agreements, network latency, source of latency, why cloud computing? and Network Evolution. Related works are also discussed so as to show their contributions and limitations.

Chapter 3 is about the methodology. this chapter explains the methods and important concepts those are used for designing the proposed design.

Chapter 4 is the proposed design. This chapter includes the overview, the Architecture of the Proposed Solution and Load balancing model background.

Chapter 5 Implementations and result evaluation; the newly proposed solution is evaluated using evaluation tools and evaluation result is also discussed.

Chapter 6 is all about conclusion, contributions and future works.

Finally, there is Reference and Appendix.

CHAPTER TWO

2 LITERATURE REVIEW

2.1 Basic Overview of Cloud computing Latency

Many papers are written regarding cloud computing, infrastructure as a service and monitoring; during this research different literatures and white papers are reviewed which are directly associated with this paper work. Cloud computing is that the delivery of hosting services that are provided to a client over the net. It's a Utility Computing that allows large-scale services without up-front investment.

The cloud could be a large set of interconnected computers which provides services anytime. Cloud services are controlled and monitored by the cloud provider through a pay-per-use business model. The cloud customer needs to pay just for the services that it's using.

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [5]. As we know, cloud computing technology is employed by both small and huge organizations to store the data in cloud and access it from anywhere at any time using the web connection. To access the cloud service, simply go surfing to the sites that provide cloud facilities and sign on and pay online if it's not free, DropBox.com, Zoho.com, Google docs.com are cloud sites.

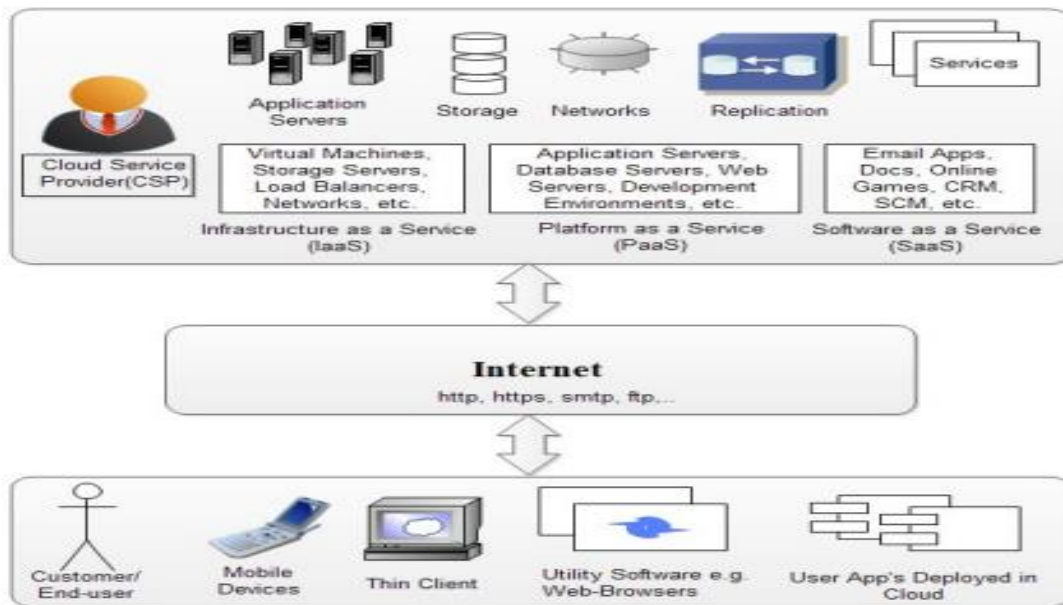


Fig 2.1 An overview of Cloud Computing, taken from [22].

Cloud computing architecture could be a combination of service-oriented architecture and event-driven architecture. Cloud computing architecture is split into the subsequent two parts –Front End and Back End

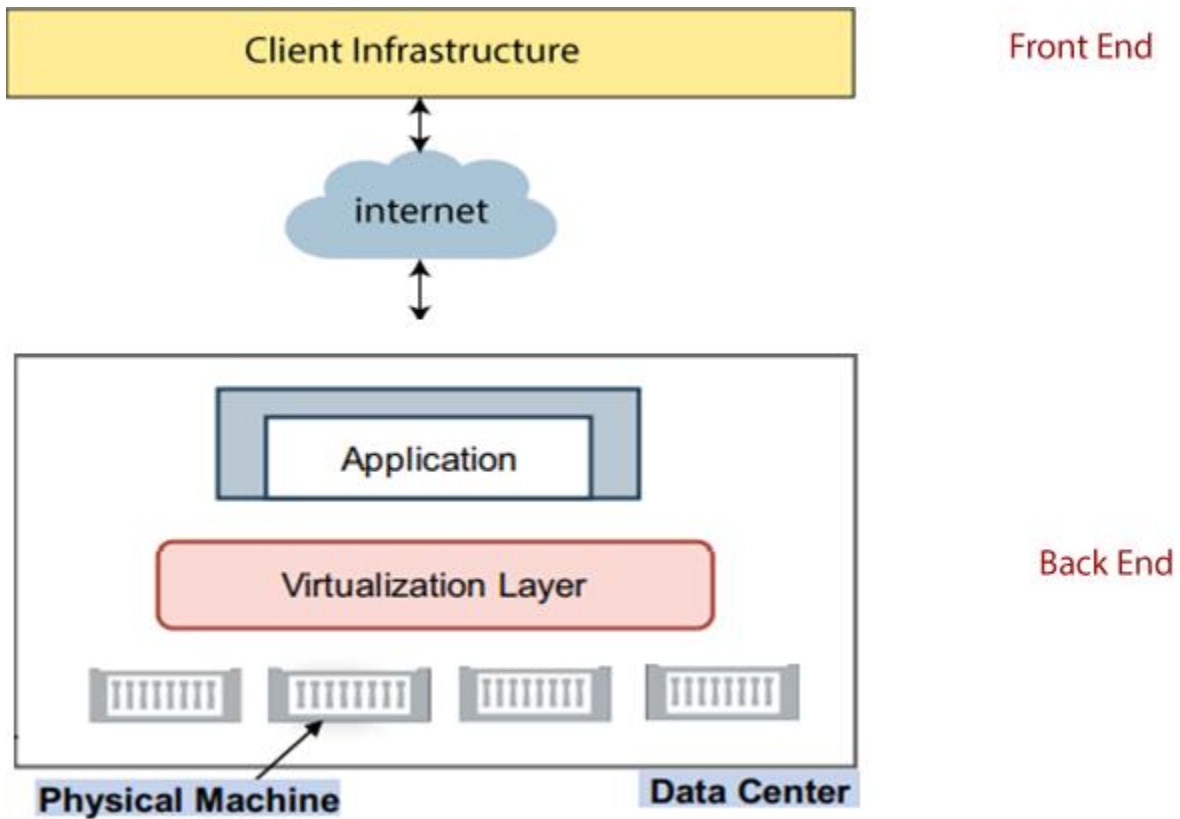


Fig 2.2 Architecture of Cloud computing, taken from [22].

Front end is employed by the client. It contains client-side interfaces and applications that are required to access the cloud computing platforms. The face includes web servers (including Chrome, Firefox, internet explorer, etc.), thin & fat clients, tablets, and mobile devices.

Back end is employed by the service provider. It manages all the resources that are required to produce cloud computing services. It includes an enormous amount of knowledge storage, security mechanism, virtual machines, deploying models, servers, control mechanisms, etc.

Note: Both side and rear are connected to others through a network, generally using the web connection.

2.2 Cloud Properties

- ❖ Resource efficiency: computing and network resources are used wisely
- ❖ Resource allocation is adapted according to user demand.
- ❖ Elasticity: computing resources can be rapidly and elastically provisioned to scale up, and released to scale down based on consumer's demand.

- ❖ Self-managing services: a consumer can provision cloud services, such as web applications, server time, processing, storage and network as needed and automatically without requiring human interaction with each service's provider.
- ❖ Accessible and highly available: cloud resources are available over the network anytime and anywhere and, are accessed through standard mechanisms that promote use by different types of platforms (e.g., mobile phones, laptops, and PDAs).

2.3 Cloud deployment models

Two major bodies play a key role in cloud deployment models: cloud consumers and cloud providers.

Cloud Consumers is a person or organization that maintains a business relationship with and uses service from cloud providers.

Cloud providers are a person, organization, or entity responsible for making a service available to cloud consumers [2]. Depending on the relationship between the provider and the consumer, a cloud can be classified into four deployment models: public, private community and hybrid cloud [16].

- ❖ **Public cloud:** is the most commonly referred to, is owned and operated by independent vendors and accessible to the general public.
- ❖ **Private cloud:** is an internal utilization of cloud technologies which is maintained in-house and solely accessible to internal users within an organization.
- ❖ **Community cloud:** supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations) and, is shared by several organizations. It may be managed by the organizations or a third party. It may exist on premise or off premise.
- ❖ **Hybrid cloud:** as the name indicates, it is a combination of two or more types of clouds (private, community, or public). For example, an organization may bridge its internally operated private cloud with other public clouds together by standardized or proprietary technology in order to satisfy business needs.

2.4 Cloud service models

Cloud computing has three main service models; these are Software as a Service, Platform as a Service and Infrastructure as a Service.

Software as a Service (SaaS): It is also known as cloud application services. Mostly, SaaS applications run directly through the web browser (i.e., we do not require to download and install these applications. The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. Applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, apart from limited user-specific application configuration settings [2].

Platform as a Service (PaaS): is also known as cloud platform services and quite similar to SaaS but the difference is that PaaS provides a platform for software creation, but using SaaS, we can access software over the internet without the need of any platform. The capability given to the consumer is to

deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application hosting environment [2][3].

Infrastructure as a Service (IaaS): is also known as cloud infrastructure services. It is responsible for managing applications data, middleware, and runtime environments. The capability given to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls) [2][4][6].

Cloud Computing is a model that empowers handy, on-interest system access to a pervasive pool of configurable processing assets (networks, storage, applications, services and servers) that might be quickly provisioned and surrendered with insignificant administration exertion or convenience supplier communication. It will be held on an agreement that is called Service Level Agreement (SLA) between cloud service provider and cloud customer. Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreement.

2.5 Service Level Agreements

Service Level Agreement (SLA) according to [12] is a mutual legal agreement signed between the end user and the service provider of cloud which define what kind of service the customer expects from the service provider. Numerous consumers want to execute diverse kinds of applications on the cloud server. So, each consumer possibly will have various Quality of Service (QoS) necessities based on their amount of work. This makes provisioning of resources a challenging task. As soon as the service providers are identified, it is essential to discover the different elements of an SLA that will be signed by agreeing metrics.

Suppliers of cloud service and the consumer are agreeing upon a certain performance related Quality of Service (QoS) properties, for instance reply time, availability a cloud provider during execution of the services, which is documented in the form of Service Level Agreements (SLAs) [12][18]. SLAs generally comprise segments to address: services definition, performance measurement, problem management, consumer responsibilities, guarantees, disaster recovery, and agreement termination [9].

2.6 Why Cloud Computing?

The best part of cloud computing is that it provides more flexibility than its previous counterparts. It has shown many benefits to enterprise IT world. The first one is its cost optimization among them, since the principle of cloud is “pay as per use”. The other benefits are increased mobility, ease of use, utmost apt utilization of resources, portability of application, etc. This means users will be able to access information from anywhere at any time easily without wasting the underlying hardware resources ideal or unused. Due to its benefit, today’s computing technology has witnessed a vast migration of

organizations from their traditional IT infrastructure to cloud. Some of the noteworthy benefits are: Cost Savings, Remote Working, Efficiency, Flexibility, Future Proofing, Morale Boosting and Resilience without Redundancy.

2.7 Network latency

We have likely heard the term latency being used before but what is latency exactly? In terms of network latency, this can be defined by the time it takes for a request to travel from the sender to the receiver and for the receiver to process that request. In other words, the round-trip time from the browser to the server. It is desired for this time to remain as close to 0 as possible.

Network latency is the term used to describe delays in communication over a network and sometimes called lag. Latency is best thought of as the amount of time it takes for a packet of data to be captured, transmitted, processed through multiple devices, then received at its destination and decoded.

Latency is the time it takes a packet to travel from source to destination in the networking space. Shortly It is the delay between data being requested and it being received. This is entirely separate to the question of bandwidth, which defines the speed at which data can flow once the connection is made.

Latency is a research issue on the Internet. Any performance in the cloud is going the same meaning of the performance of the result on the client. The latency in a cloud introduces not to be tedious. The latency is compressed back for understand how and where they're running with both smartly-written applications and an intelligently planned infrastructure. In future, cloud computing capacity and cloud-based applications are rapidly increases and latency is also increases. Cloud computing latency must be improved in the desktop PC because it creates the largest bottlenecks in the memory and storage.

In transmission, as delays are becoming small, response time will be fast and resulting low latency network(desirable) and delays are becoming longer, the response time will be high and resulting a high-latency network (not so desirable). Long delays that occur in high-latency networks create bottlenecks in communication. In the worst cases, it's like traffic on a four-lane highway trying to merge into a single lane. High latency decreases communication bandwidth, and can be temporary or permanent, depending on the source of the delays. During speed tests, latency is referred to as a ping rate. Obviously, zero to low latency in communication is what we all want. However, standard latency for a network is explained slightly differently in various contexts, and latency issues also vary from one network to another.

Cloud providers can address concerns about security, costs and control through technological advancements. But there's no technology that can transmit data faster than that of speed of light, which means latency is unavoidable with the public cloud. This is due to big public cloud providers building huge data centers in places where land and power are cheap, usually hundreds or thousands of miles away from customers' locations. For storing massive volumes of archival data at low cost, the economies of scale are fantastic. But for today's business applications, where users expect real-time response and data is being generated at many endpoints, the delays and unpredictability of the public cloud can make it hard to use for more active and performance-sensitive workloads.

2.8 Sources of latency

In addition to the advances in data processing and database technology, there is no number one escaping data's public enemy like that of latency, the time delay before a response is generated and returned. Gartner stated that latency can never actually be zero in his definition of a zero-latency because computers need time to "think". While we may never truly achieve zero latency, the goal is always to deliver information in the shortest amount of time possible, so ensuring predictable, low latency processing is key when building a real-time application. Before, during, and after computing the response, there are number of areas that can add unwanted latency. Below are some common sources of latency.

- ❖ Network I/O - Most applications use the network in some manner, whether between the client application and the server or between server-side processes and applications. The important thing to know here is that distance matters. The closer your client is to the server, the lower the network latency.
- ❖ Desk I/O- Many real-time applications are data intensive, requiring some sort of database to service the real-time request. Databases make data durable by storing it to persistent storage, but for high-velocity real-time applications, making data durable can add significant unwanted latency, and disk I/O, like network I/O, is costly. Accessing memory can be upwards of 10,000 times faster than a single disk seeks.
- ❖ The operating environment - The operating environment by which we run our real-time application on shared hardware, in containers, in virtual machines, or in the cloud can significantly impact latency.
- ❖ The Code- there are some common core functionalities that can pose barriers to speed when it comes to coding.
- ❖ Combating the enemy - Building real-time applications requires that the application developer not only write efficient code, but to also understand the operating environment and hardware constraints of the systems on which our application will be deployed. Our real-time latency requirements won't singularly solved by provisioning the fastest networking equipment and the fastest CPUs.

Thoughtful application architecture, efficient software algorithms and optimal hardware operating environment are all key considerations for fighting latency. Generally, Latency is unavoidable due to the way networks communicate with each other and depends on several aspects of a network. It can vary if any of them are changed. There are four main components that affect network latency. These are:

- **Transmission medium:** is the physical path between the start points and the end point. The type of medium can impact latency. For example, old copper cable-based networks have a higher latency than modern optic fibers.
- **Propagation:** The two apart nodes are the more latency. Latency is dependent on the distance between the two communicating nodes. Theoretically, latency of a packet going on a round trip across the world is 133ms. In actuality, such a round trip takes longer, though latency is decreased when direct connections through network backbones are achieved.

- **Routers:** The efficiency in which routers process incoming data has a direct impact on latency. Router to router hops can increase latency.
- **Storage delays:** Accessing stored data can increase latency as the storage network may take time to process and return information.

Ajith Singh and Hemalatha [10] conducted a survey on how latency occurs in different geographical location and also revealed an analysis work of how different browsers provide different latency. A test conducted to show effect of bandwidth reveals that when one tries to access cloud-based Google docs in cybercafé or GPRS connection it took 20 sec while when tried to open at the campus of university which provides 5.4 mbps it opens in 2 sec. The problem of latency in the cloud network will be solved with the faster adaption of 3g and 4g in the coming years.

Ankush Veer Reddy [24] proposed a security model for cloud-based applications by implementing a firewall using two applications i.e., web-based application and database application to simulate and test the efficiency of the model.

Sonia and Satinderpal Singh in [18] reviewed academic research published in the field of energy efficient cloud environment and aimed to provide an overview of analyzing the energy consumption in different types of networks with downloading/uploading speed and computing the performance of networks.

2.9 Layers of Latency

One of the factors that have the biggest impact on the speed of a system is the latency. Latency is a measure of the delay experienced by the components of a system during their processing of a request. Latency exists at every stage of the cloud computing chain.

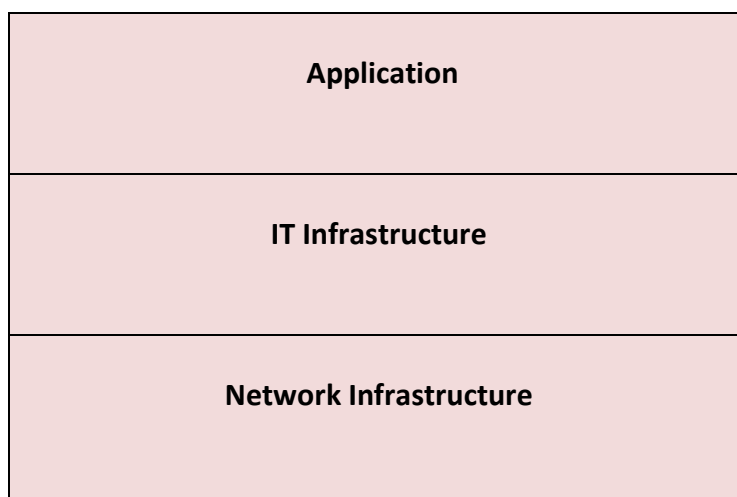


Fig 2.3 Layers of Latency

Latency in the network layer is inherent. The type of the communication media, the network architecture and the network protocols used are other factors which contribute to the network latency. Choosing the most efficient network equipment and the appropriate communication protocols does help with reducing the network latency, but in most cases, delays cannot be completely eliminated.

The IT infrastructure layer covers everything which is not related to network or the trading applications. This includes the hardware platform (the physical machine with all its devices), the operation system and all auxiliary software (virtual machines, messaging middleware, and databases). Along with the application layer, this is the area where latency can be reduced the most if proper solutions are employed. System architects and developers make every effort to identify the components which generate the highest delays, since optimizing those components will have the most impact on overall system latency. Latency is also not constant and can be influenced by many factors external to the trading infrastructure.

2.10 Network Evolution

2.10.1 Connectivity and Bandwidth

The network evaluation is divided into three main phases: calculating the latency between the previous defined groups and the edge nodes; defining a connectivity score; and a bandwidth score.

2.10.2 Latency Vs bandwidth Vs throughput

Latency, bandwidth and throughput are all equal contributors to the quality of communications. While these three factors work together, they have different meanings. To understand this, you can imagine that data packets flow through a pipe:

Bandwidth is the width of the pipe. Therefore, the narrower the pipe, the less data allowed to travel back and forth through it. The wider the communication band, the more data that can flow through it simultaneously.

Latency is the speed that data packets inside the pipe travel from client to server and back. Packet latency is dependent on the physical distance that data must travel through cords, networks and the like to reach its destination.

Throughput is the amount of data that can be transferred over a specified time period. Low latency and low bandwidth mean that throughput will also be low. This means that while data packets should technically be delivered without delay, a low bandwidth means there can still be considerable congestion. But with high bandwidth, low latency, then throughput will be greater and the connection much more efficient.

Throughput is the average amount of data that actually passes through over a given period of time. Throughput is not necessarily mean to bandwidth, because it is affected by latency and other factors. Latency is a measurement of time rather than how much data is downloaded over time. Latency, bandwidth, and throughput are all interrelated, but they all measure different things. Bandwidth is the maximum amount of data that can pass through the network at any given time.

2.11 Related Work

2.11.1 The impact of latency in different organizations/companies

The cloud has enabled enterprises to dramatically improve how they operate their businesses, bringing information and applications to each corner of the world and freeing up cupboard space as big data grows in popularity and volume. Users can access applications from literally anywhere within the cloud in the world, requiring only a web connection, and applications may be housed across multiple data centers sprinkled round the globe. Due to the flexibility and availability that the cloud offers, quite 30 percent of enterprises worldwide use a minimum of one cloud-based solution.

Cloud revenue is anticipated to grow 500 percent from 2010 to 2020 as cloud applications and firms multiply and expand. In addition to the cloud opening such a large number of possibilities, it's not always able to deliver on performance demands, sometimes resulting in subpar user experiences. For example, Mike Belshe (Google Chrome executive) found that 20 milliseconds of network latency may end up during a 15 percent decrease in page load time. Other studies from Amazon and Google found that a half-second delay causes a 20 percent call in traffic on Google, and a one-tenth of a second delay can lower Amazon's sales by 1 percent. Clearly, latency isn't only a nuisance, but also a heavy problem for enterprises that house their applications within the cloud.

It is important to know what causes it to mitigate the growing effects of latency, in addition as how enterprises can reduce it. Latency is a way more complicated than one might suspect with both the net and cloud computing laying a task in how we share and access applications. Latency was defined just by the quantity of router hops required for data to travel from origin to destination in prior to the arrival of the web. For the foremost part, Enterprises owned their network and everyone its components. So, packets would travel the space between two computers or servers, leading to more latency for transfers that involved more hops.

Today, most networks are lessened into hundreds, if not thousands of components that are each owned, operated and managed by different entities. Enterprises, therefore often don't have insight into the performance of their network, in addition to the flexibility to optimize its performance or reduce latency. Often labeled as distributed computing, implies that if even one server out of hundreds is experiencing latency, cloud application users will see slower load times and halted performance. Because these widespread and complicated networks are increasingly common in today's world, many connectivity providers now provide service level agreements, or SLAs, that outline a minimum level of service and guaranteed network performance. Service providers of all sorts, whether telecom or cloud, work very hard to uphold the minimums outlined within their SLA. However, when it involves cloud transactions conducted over the net, service providers often don't establish SLAs.

Cloud latency is such a brand-new phenomenon and connectivity providers are still understanding how they'll ensure strong uptime for cloud applications, to not mention what levels to line. It's clear that these three factors: - intricate networks, virtualization and a scarcity of SLA standards create extremely unpredictable and unregulated service levels. However, the matter isn't necessarily latency, but the unpredictability of it. To overcome this unpredictability, enterprises must establish a baseline for

performance and so keep as many cloud applications as possible performing thereto level. Only then can they work to cut back it.

Many have found that establishing an instantaneous connect with a public cloud is a method to assist reduction of the cloud's unpredictability. These connections are offered by many leading cloud companies, including Amazon Web Services (AWS), to enable a connection between an enterprise's network and therefore the public cloud without involving many other servers or virtual machines. This essentially implies that an enterprise can founded its own lane within which cloud applications travel back and forth to the house network.

As a result, cloud traffic is not any longer subject to the unpredictability of the overall Internet and performance becomes way more calculable. Performance metrics for these services follow strict quality of service guarantees as these cloud onramps" are seen as a key offering within cloud providers' solutions. Outages and latency continually remind us that the cloud isn't perfect. Despite the high-performance, capabilities of the cloud, we must detain mind that this is often very new technology and both users and providers are still trying to figure out the kinks to ascertain steady service levels.

However, the primary step to achieving a low-latency, high-performing cloud is identifying the causes of performance degradation. From there, we are able to only move onwards and upwards into the cloud. Latency can occur in networks connecting cloud to the tip user and at the user end.

A. Intra Cloud Latency: - In cloud, latency can arise when two vm's co-located on the same server communicate with each other. This problem is limited by introducing Nahanni Memcached, a port of the well-known Memcached that uses inter-VM shared memory instead of a virtual network for cache reads [11]. Facebook, for example employs Memcached as one of several caching layers.

B. Network Latency: - Network latency cause applications to spend amount of time waiting for responses from a distant data center, then the bandwidth may not be fully utilized and performance will suffer [19]. Network latency is comprised of Propagation delay, Node delay and Congestion delay.

Good network design can minimize node delay and congestion delay but not propagation delay [20][21][22]. Network delay illustrates how much time it takes for a bit of data to move across network from one node to another node.

- ❖ Propagation Delay- The amount of time taken for head of signal to travel from sender to the receiver and it can be defined as the ratio between link length and the propagation speed over the specific medium
- ❖ Congestion Delay- Network congestion occurs when a link or node is carrying so much data that its quality of service deteriorates. Typical effects of congestion delay include Packet loss, Blocking of new connections and Queuing Delay.

C. Processing Delay: - The processing delay is the time which routers take to process the data. It is the important component in the network delay.

Furthermore, many web applications are backed by a data store, such as an SQL database or a cloud-scale NoSQL storage engine. For example, Amazon offers Simple DB, and GAE includes a component called Data store that is backed [8]. Because data store access can be expensive, contributing significantly to web application latency, it is common to use an in-memory cache.

In IaaS environments, application-level caching is usually the responsibility of the application developer, who may choose to deploy a caching service such as memcached [13] using the virtual machines in which their web application is deployed.

In PaaS environments, a caching mechanism may be provided in the API (e.g., the memcache API in GAE and AppScale). This API could be backed by memcached (as it is in AppScale) or a similar system. They presented a modified version of the memcached server that uses inter-VM shared memory to store cached data, while maintaining compatibility with standard memcached clients.

Adam Wolfe Gardon and Paul lu (Sept. 16, 2011) present a memcached client library for C, Java, and Python that can take advantage of the shared memory design. They had demonstrated that their implementation can reduce the latency of cache reads by up to 86%, and given reasonable cache hit rates can reduce the total latency of datastore reads by up to 45% compared to using standard Memcached, depending on the workload. They also showed that their shared-memory-based technique can be faster than sophisticated Para virtualized network devices, such as virtio and vhost in Linux KVM [19].

There are two performance benefits to using memcached: reduced latency when data can be found in the cache instead of being re-generated, and improved database scalability due to reduced request volume. Memcached is preferable to purely application-level caching because it can increase the total capacity of the cache by distributing the cache across multiple servers, and it allows multiple web servers to access the same cache.

In their solution, they assumed that edge nodes are connected to each other through a network and that users connect to the closest BS. In many cases, however, the service may not be scheduled in the closest edge node to its user. For example, the edge node might not have the type of resources required by the service or it might be overloaded. They, therefore, need to take into account the latency between groups and the edge nodes. This method is valid since network latency can be estimated using many different techniques, which range from active monitoring by sending probes between servers and then measuring the latency, to passive monitoring that capture information from the network device about network paths.

The evaluation of the delay, executed for every path connecting a user a utility function previously defined by the provider. One of the main factors affecting the overall service quality is the available bandwidth of the paths between users and the VM.

Adam Wolfe and Paul Lu [4] proposed a memcached named Nahanni Memcached which can reduce the communication overhead between Virtual machines (VM) located in same server and used it with VDE

networking to improve the total read latency for a workload by up to 45% (i.e., read latest workload) compared to standard memcached.

Ajith Singh and Hemalatha [5] conducted a survey on how latency occurs in different geographical location and also revealed an analysis work of how different browsers provide different latency. A test conducted to show effect of bandwidth reveals that when one tries to access cloud-based Google docs in cybercafé or GPRS connection, it took 20 sec while when tried to open at the campus of university which provides 5.4 mbps it opens in 2 sec. The problem of latency in the cloud network will be solved with the faster adaption of 3g and 4g in the coming years.

Mohammad Haideri [6] tried to highlight the modelling and simulation for different kinds of computer network attacks and their impact on computer and networks. He explained applications for modelling and simulation of computer network security. It presents a comprehensive suggestion to solve the problem in modelling and simulating in the field of Information Security. He simulated the cloud network and implemented the botnet attack on one of the cloud applications i.e., FTP to analyze the effect of the attack on FTP server.

Ankush Veer Reddy [7] proposed a security model for cloud-based applications by implementing a firewall using two applications i.e., web-based application and database application to simulate and test the efficiency of the model.

Pardeep Sharma, Sandeep Sood and Sumeet Kaur [8] had proposed the benefits of cloud computing along with its flip side. This Paper also introduces various issues in Cloud Computing and suggested the possible measures to overcome them and the proposed algorithm is used to calculate and compare the net revenue by using the cloud and data center.

Sonia and Satinderpal Singh in [9] reviewed academic research published in the field of energy efficient cloud environment and aimed to provide an overview of analyzing the energy consumption in different types of networks with downloading/uploading speed and computing the performance of networks.

Raihana Abdullah, Mohd Faizal Abdullah, Zul Azri Muhamad, Mohd Zakri Mas Ud, Siti Rahayu Selamat and Robiah Yusuf in [10] had addressed the current trend of Botnet detection techniques and identifies the significant criteria in each technique. Several existing techniques are analyzed from various researchers and the capability criteria of botnet detection techniques are analyzed. The techniques have been shown on the selected detection criteria.

Ashraf Zia and Muhammad Naeem Ahmad Khan in [11] had discussed performance issues in cloud computing. A number of schemes pertaining to QoS issues are critically analyzed to point out their strengths and weaknesses. Some of the performance's parameters at the three basic layers of the cloud. IaaS, PaaS and SaaS are also discussed in this paper.

This paper also observed the key challenging areas that how resources are allocated to clients and what are the roles of cloud providers. Also investigated how the performance can be increased by improving various components in a scalable way with low cost, better performances and QoS. Some technical and functional issues in cloud that affect the performance of a cloud are also pointed out.

Nagaraju Kilari and Dr R.Sridaran in [10] had proposed various security threats in a classified model and illustrated how cloud and virtualization vulnerabilities affect the different cloud models. The classification of various security threats presented in this paper would definitely benefit the cloud users to make out proper choice and cloud service providers to handle such threats efficiently.

As more cloud-based applications keep evolving the associated security threats are also growing. Many researchers work on cloud security exist in partial forms of either specifically on cloud issues or Virtualization-related security issues.

Table 3. 1 Summery of related works

Reference	Year	Mechanism	Contribution	Limitation
[15]	2016	Task Scheduling	<ul style="list-style-type: none"> • Better scalability • Less response times • Small execution cost 	<ul style="list-style-type: none"> • Less resource utilization • Less degree of balance
[27]	2017	Resource Scheduling	<ul style="list-style-type: none"> • Good resource utilization • Enhanced throughput 	<ul style="list-style-type: none"> • response time not chosen • degree of balance not chosen
[25]	2017	Resource Scheduling	<ul style="list-style-type: none"> • Optimal LB is achieved at Nash equilibrium point. • Minimize expected response time 	<ul style="list-style-type: none"> • High task execution time
[29]	2018	VM/Task Scheduling	<ul style="list-style-type: none"> • High scalability • Less response times • High resource utilization 	<ul style="list-style-type: none"> • Low throughput • Low fault tolerance • High make span
[28]	2018	Task Scheduling	<ul style="list-style-type: none"> • Minimize make span with improved task acceptance ratio • Minimize task rejection ratio 	<ul style="list-style-type: none"> • The experimental are run on Cloudsim using space shared policy only and not time-shared policy.
[30]	2018	VM Scheduling	<ul style="list-style-type: none"> • minimize the response time • good resource utilization 	<ul style="list-style-type: none"> • less security
[41]	2019	service scheduling	<ul style="list-style-type: none"> • create a novel Edge scheduling framework for latency-sensitive services • evaluate different deployment solutions and scheduling approaches for latency-sensitive services 	<ul style="list-style-type: none"> • Unable to enhance the developed solution by decentralizing the optimization process • Unable to considering vertical and horizontal scaling
[42]	2020	Scenario measurement	<ul style="list-style-type: none"> • Perform large-scale latency measurements • Analyze the edge computing latency for end-users with and without cloud support. • Share the latency measurements at cloud-edge-latency 	<ul style="list-style-type: none"> • Inefficient resource allocation algorithms for edge platforms, • Unable to develop a brokerage service
[39]	2021	Task Scheduling	<ul style="list-style-type: none"> • high resource optimization • less response times • high performance 	<ul style="list-style-type: none"> • Low fault tolerance • Less security

In general, to overcome the specified problems of the existing prioritization scheme, all the papers mentioned above use their approaches. However, energy consumption was not considered while a load balancer distributes requests to servers. We therefore, have taken this as research gap so as to overcome the problem by using servers' energy consumption based load balancing mechanism. Energy consumption can be defined as the amount of energy used by the servers for a given service or level of activity [14]. So, a load balancer distributes requests/activities without considering their energy consumption ability. Therefore, distributing requests based on servers' energy consumption is a major concern in the large-scale data centers in order to reduce the latency.

CHAPTER THREE

3 Methodology

The main aim of this chapter was to design the methodology to carry out the thesis, which included: determining design strategy for the study including a proposed design, defining the deliverables, and articulating the methodology. The requirements to this study were: developing the load balancing mechanism using energy consumption load scheduling method and current solutions to identify the requirements; the knowledge of theory that brought to bear in a solution to design the proposed cloud computing; and developing evaluation tool to measure the performance of the resulting the proposed design. Thus, the instruments used to develop the knowledge base to the study requirements were review of relevant literature including similar studies and implementing the proposed design using cloudsim simulator.

3.1 Study design

The thesis focused to use a consensus building approach to produce the design which contains four phases discussed below.

i. Problem identification and motivation

This phase includes defining the knowledge of the state of the problem and the importance of its solution. Researchers begun by defining the specific research problem and justify the value of a solution by conducting rigorous analysis of literatures and implementing the proposed design using simulation environment. Since based on the problem definition the researchers atomize the problem conceptually to capture its complexity for developing the proposed cloud computing design that can effectively provide a solution. The value of the solution also justified to motivate the researcher and the audience of the research to pursue the solution and to accept the results and to understand the reasoning associated with the researcher's understanding of the problem.

ii. Define the objectives for a solution

This phase aims at to identify or drive the requirements for developing the proposed cloud computing design from knowledge of the state of problems and current solutions. The researchers infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible to align with the cloud computing requirements. The qualitative objectives are used, which described how a proposed design is expected to support solutions to problems not hitherto addressed. In order to develop the proposed design, researchers started with the literature review to gather relevant requirements and aspects of existing Cloud computing environment.

iii. Design and development

Conceptually, a design research artifact can be any designed object (models, frameworks or instantiations) in which a research contribution is embedded in the design. This phase includes determining the artifact's desired functionality and its architecture and then creating the actual artifact. The knowledge of theory that can be brought to bear in a solution is required to move from objective of

a solution to design and development of the proposed. Based on the required knowledge of a solution identified in phase one and two with the literature review and cloudsim simulation to develop the proposed design: After determining the framework functionality and its architecture, the proposed collaborative cloud computing environment with the implementation plan to reduce latency on the cloud could be developed.

iv. Evaluation

This phase aims to Observe and measure how well the design supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from use of the design in the demonstration. It requires knowledge of relevant metrics and analysis techniques. Depending on the nature of the problem venue and the artifact, this study took the qualitative evaluation analysis techniques which include: a comparison of the proposed design functionality with the solution objectives.

3.2 Method

To fulfill the general and specific objectives of the research, different methods are proposed:

3.2.1 A systematic literature reviews

The literatures helped to have better understanding about the area; they also guided how the research should go to achieve its objectives and how similar works are done so far.

3.2.2 Design and Implementation

In the design phase, proposed design in highway scenarios that are specified in the objectives of this thesis are designed. To design the proposed work, we have implemented the proposed design using a cloudsim simulation.

CloudSim is a simulator that allows cloud developers to test the performance of the provisioning policies in a repeatable and controllable environment, free of cost. It helps to tune the bottlenecks before real-world deployment. It doesn't run any actual software. It can be defined as 'running a model of an environment in a model of hardware', where technology-specific details are abstracted.

CloudSim is a library for the simulation of cloud scenarios. It provides essential classes for describing data centers, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning. It is easy to evaluate new strategies governing the use of clouds using these components, while considering policies, scheduling algorithms, load balancing policies, etc.

It can also be utilized to evaluate a strategy's effectiveness from other angles, including price, application execution time, etc. It can incorporate additional policies for scheduling, load balancing, and new situations, and it can be used as a building component for a simulated cloud environment. It can be used as a library that enables us to add a desired scenario by creating a Java program because it is versatile enough to do so.

Cloudsim has the following basic properties that make it immensely popular:

- Offers a repeatable, controlled environment for the creation and simulation of cloud things.
- Can be expanded to include user-defined cloud policies.
- Offer network provisioning, application provisioning, host provisioning, and VM provisioning.
- Offers federated cloud support, flexible network architecture, and modeling of infrastructure.
- Compared to actual infrastructure, CloudSim utilizes a relatively little amount of overhead. It may simply be expanded to accommodate new policies and entities because it is built on Java as a Maven project.

The cloudsim version used in this implementation is cloudsim 6.0. this release maintains backward computability with existing code from cloudsim 3.0-5.0. the codebase has undergone massive refactoring, readability, and performance improvement. Since JDK8 is now deprecated, this release requires JDK11 and above (LTS version recommended). Also, the codebase contains many "syntax sugar" from JDK 11. Future releases will be tested and validated against the latest two LTS versions (11 and 17 as of the date of this release). Feature-wise, it contains many off-spin contributions from various contributors, including container, Geo web load balancing, SDN, etc.

3.3 Virtualization

We used virtualization mechanism to make our proposed solution effective. Virtualization is a very useful concept in context of cloud systems. Virtualization isn't real giving all the facilities of a real. It is the software implementation of a computer which will execute different programs like a real machine. An end user can use different services of a cloud using virtualization, because virtualization is related to cloud. The remote datacenter will provide different services in a fully or partial virtualized manner.

It is an essential technological characteristic of clouds which hides the technological complexity from the user and enables enhanced flexibility (through aggregation, routing and translation). More concretely, virtualization supports the following features:

- Ease of use: through hiding the complexity of the infrastructure (including management, configuration etc.) virtualization can make it easier for the user to develop new applications, as well as reduces the overhead for controlling the system.
- Infrastructure independency: in principle, virtualization allows for higher interoperability by making the code platform independent.
- Flexibility and Adaptability: by exposing a virtual execution environment, the underlying infrastructure can change more flexible according to different conditions and requirements (assigning more resources, etc.).
- Location independence: services can be accessed independent of the physical location of the user and the resource.

CHAPTER FOUR

4 DESIGN OF PROPOSED SOLUTION

4.1 Overview

As we discussed in chapter two about the impact of latency, we have designed new model for reducing latency in Cloud Computing Environment using energy consumption load scheduling mechanism. Load balancing is efficiently distributing incoming network traffic across a bunch of backend servers, also called a server farm or server pool. Modern high traffic websites must serve many thousands, if not millions, of concurrent requests from users or clients and return the proper text, images, video or application data, dead a quick and reliable manner. To cheaply scale to satisfy these high volumes, modern computing best practice generally requires adding more servers.

A load balancer acts because the “traffic cop” sitting before of your servers and routing client requests servers. Across all servers capable of fulfilling those requests in an exceedingly manner that maximizes speed and capacity utilization and ensures that nobody server is overworked, which could degrade performance. If one server goes down, the load balancer redirects traffic to the remaining online When a replacement server is added to the server group, the load balancer automatically starts to send requests to that. During this, a load balancer performs the subsequent functions:

- ❖ Distributes client requests or network load efficiently across multiple servers
- ❖ Ensures high availability and reliability by sending requests only to servers that are online
- ❖ Provides the flexibleness to feature or subtract servers as demand dictates

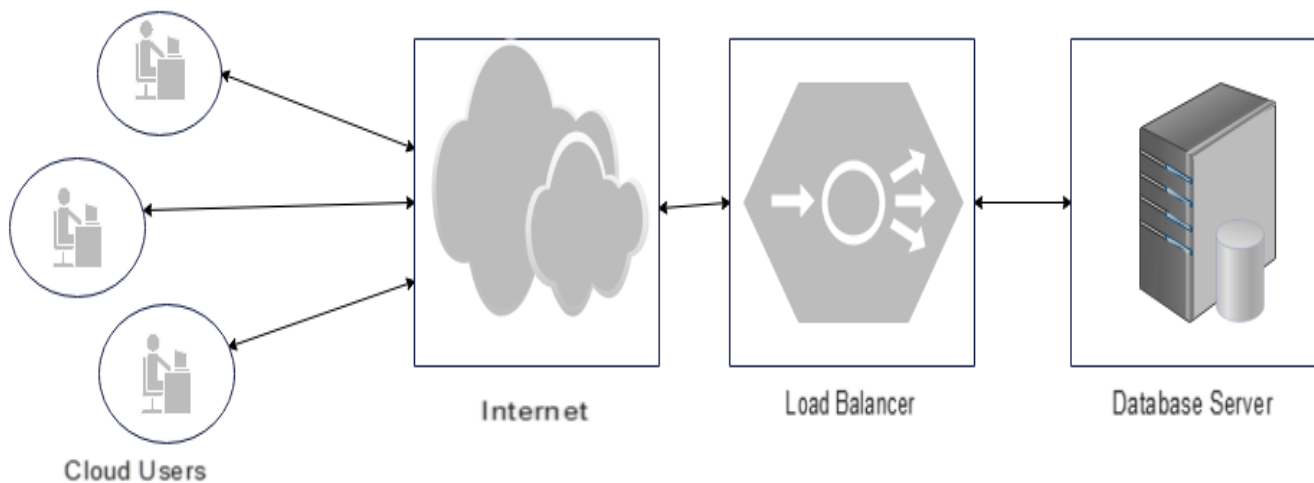


Fig 4.1 load balancing diagram

4.2 The Architecture of the Proposed Solution

In the field of cloud computing, load balancing plays a vital role as it allocates and balances the load of various resources among the distinct components and nodes on the network. Load balancer is the main component which balances the load over the nodes. The work of load balancer is to forward request to the server at backend, which again replies back to load balancer. This permits the load balancer to answer the request of the user.

The server power consumption depends upon the utilization of CPU. A server which is ideal consumes $2/3^{\text{rd}}$ of the highest power consumption. This energy consumption by ideal server is because server must keep the disks, modules of memory, I/O resources and other exterior operations even if not in use or no computations are performed on that. So, when the load comes on the network the power consumption is increased [26].

However, energy consumption was not considered while a load balancer distributes requests to servers. It can be highly difficult to define the proper activity that could be examined for the efficiency of energy. Energy consumption can be defined as the amount of energy used by the servers for a given service or level of activity [14]. So, a load balancer distributes requests/activities without considering their energy consumption ability. Therefore, distributing requests based on servers' energy consumption is a major concern in the large-scale data centers in order to reduce the response time.

The goal of the proposed design is to produce a high performance (reduced latency) cloud environment. This design resolves high latency within the cloud by using energy consumption of each server within the cloud environment and applying migration technique.

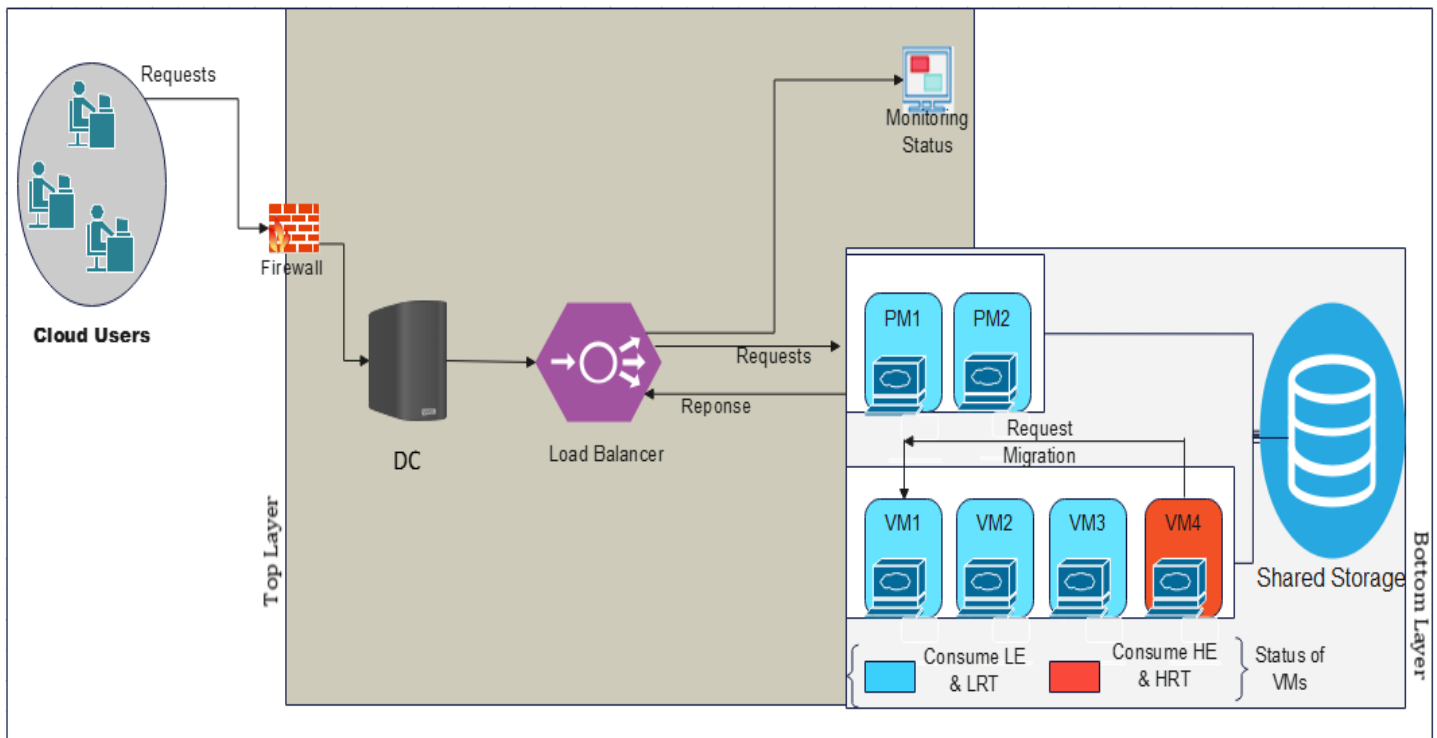


Fig 4.2 Architecture of the proposed design

In the proposed cloud, Server1 consumed less energy than of Server2, 3 and 4 based energy consumption calculation. Since server 4 needs high energy, the load given to it would move to server 1. Because Server 1 consume less energy compared to VM2 and VM3. The work of load balancer is forwarding request to the server consuming less energy and does request migration parallelly. If one server goes down, the load balancer redirects traffic to the remaining online.

When a replacement server is added to the server group, the load balancer automatically starts to send requests to that. During this, a load balancer performs the subsequent functions: distributes network load efficiently across multiple servers, and ensures high availability and reliability by sending requests only to servers that are online.

The time taken to process a request in every server of the proposed cloud was less than every server of the existing cloud. This indicated that its response time is low and resulting less communication delay comparing with the existing cloud.

The proposed cloud computing used energy consumption load scheduling mechanism which helped to distribute the loads based on energy consumption of the servers. So, the mechanism helped proposed cloud computing to have less communication delay. Therefore, both having less response time and distributing the loads using servers' energy consumption lead to have reduced latency.

In the proposed solution,

1. Cloud users; - End users interact with the clients to manage information related to the cloud.
2. Data center: - is a collection of servers hosting different applications. An end user connects to the datacenter to subscribe different applications. A datacenter may be found at a long distance from the clients. Now-a-days a concept called virtualization is used to install a software that allow multiple instances of virtual server applications.
3. Load balancer: - A load balancer receives incoming traffic and routes those requests to active targets based on a configured policy.
4. Monitoring status: - Cloud monitoring is a method of reviewing, observing, and managing the operational workflow in a cloud-based IT infrastructure. Manual or automated management techniques confirm the availability and performance of websites, servers, applications, and other cloud infrastructure.
5. Virtual machines: - The primary virtual machine resources. Virtualization decouples physical hardware from an operating system. Each virtual machine contains a set of its own virtual hardware and there are four primary resources that a virtual machine needs in order to correctly function. These are CPU, memory, network, and hard disk. The secondary hosts primarily contribute to the cluster by running virtual machines locally, monitoring their runtime states, and reporting state updates to the primary host. A primary host can also run and monitor virtual machines. Both secondary hosts and primary hosts implement the VM and Application Monitoring features.
6. We took two applications namely database application and web-based application, and two server objects are used to act as database server and webserver. The application configuration object is used to define the application and the profile configuration object is used to define the application profile.
7. We implemented a firewall at the datacenter to filter the requested data.
8. We used many processors (RAM) to improve the speed of the cloud devices.
9. The load balancer is used to distribute tasks to the servers based on the remaining energy. The load balancer also reads the remaining energy and the status of the consumption energy of those devices. Here to select/assign the incoming client request to the servers based on their energy consumption, we first check which server consumes high energy per second by sending the same sized data to them.

To show this, the following formula is applied:

$$Ec = \frac{Er1 - Er2}{Er1} 100\% \dots \dots \dots \text{Equation 4. 1}$$

Where $E_c \rightarrow$ stands for energy consumption and is described in percent

$Er1 \rightarrow$ stands for the remaining energy before data processing.

$Er2 \rightarrow$ stands for the remaining energy after data processing

One of the functions performed by the first host is to orchestrate restarts of protected virtual machines. A virtual machine is protected by a primary host after Center Server observes that the virtual machine's power state has changed from powered off to powered on in response to a user action. The first host persists the list of protected virtual machines within the cluster's data stores. A newly elected primary host uses this information to see which virtual machines to safeguard.

Top Layer: deals with requests from multiple different clients (application's users) of both mobile and desktop. Users can access the net using multiple different devices to send requests to the cloud. Devices to send requests to the cloud. In Cloud Computing, Data Center (DC) can be described as a big storage for cloud servers and data. DC receives requests and sends them to the load balancer [30]. The purpose of providing a load balancer is just in case of failure in cloud system. Failures could exist for several reasons like overloading the VMs with many requests or unnecessary migration of requests.

Bottom Layer: deals with allocation of user requests to VMs. As can be seen from the figure 4.2, VM4's status shows that it consumed high energy and its response time is high. Thus, a migration technique should be applied to transfer the requests to VM1 which consumes low energy and its response time is low. This will then turn off the load balancer and declares it as unavailable which can cause server downtime of the system in the cloud. Also load balancer can take over and continue to re-allocate requests to available VMs. The proposed design makes use of replication concepts. Therefore, VMs should have the same workload to function properly and in case of an overloading situation, a migration technique must be applied to transfer requests as can be seen in the figure 4.2.

An Optimal Migration Algorithm will be applied in the Load Balancer. The purpose of the algorithm is to balance the load in VMs with minimal data movement. It computes and applies a Migration Threshold. The threshold can be calculated based on host utilization [40] $\text{Utilization VM} \frac{1}{4} \text{Total Requested MIPS Total MIPS for that VM}$ In some cases, the active Load Balancer can make unnecessary migrations, this can be detected by training a model. The load balancer has a list of health status in 2 clusters to indicate whether it's working properly:

All VMs are balanced and the algorithm is working per configuration. Number of migrations exceeded the threshold. The model is trained to learn the optimal number of migrations. If the load balancer is in a critical state, it will then become unavailable and a Passive load balancer should be activated to handle the requests with high availability. To achieve this, a dataset must be used which can be obtained from the simulation result of the Optimal Migration Algorithm. Then following the general process of Machine Learning below to predict the limited number of migrations: After collecting the dataset, it will be pre-processed to remove any outliers and correct other mistakes. Then a Machine Learning model will be identified for the prediction of the number of migrations required, if it is beyond the limit, the status of the active Load Balancer will be critical.

4.3 Load balancing model background

The virtual machine manager and virtual machine monitor are abstracted in this model. The first level load balancing is performed at the Physical Machine (PM) level and the second level is performed at the VM level. Based on this, there are two task migration sets; Intra and Inter VM task migration. The request generator generates user requests which are user tasks that need computing resources for their execution. Data center controller is in-charge of task management. The load balancer checks which VM to assign for a given user task. The first level load balancer balances the given workload on individual

Physical Machines by distributing the workload among its respective associated Virtual Machines. The second level load balancer balances the workload across different Virtual Machines of different Physical Machines.

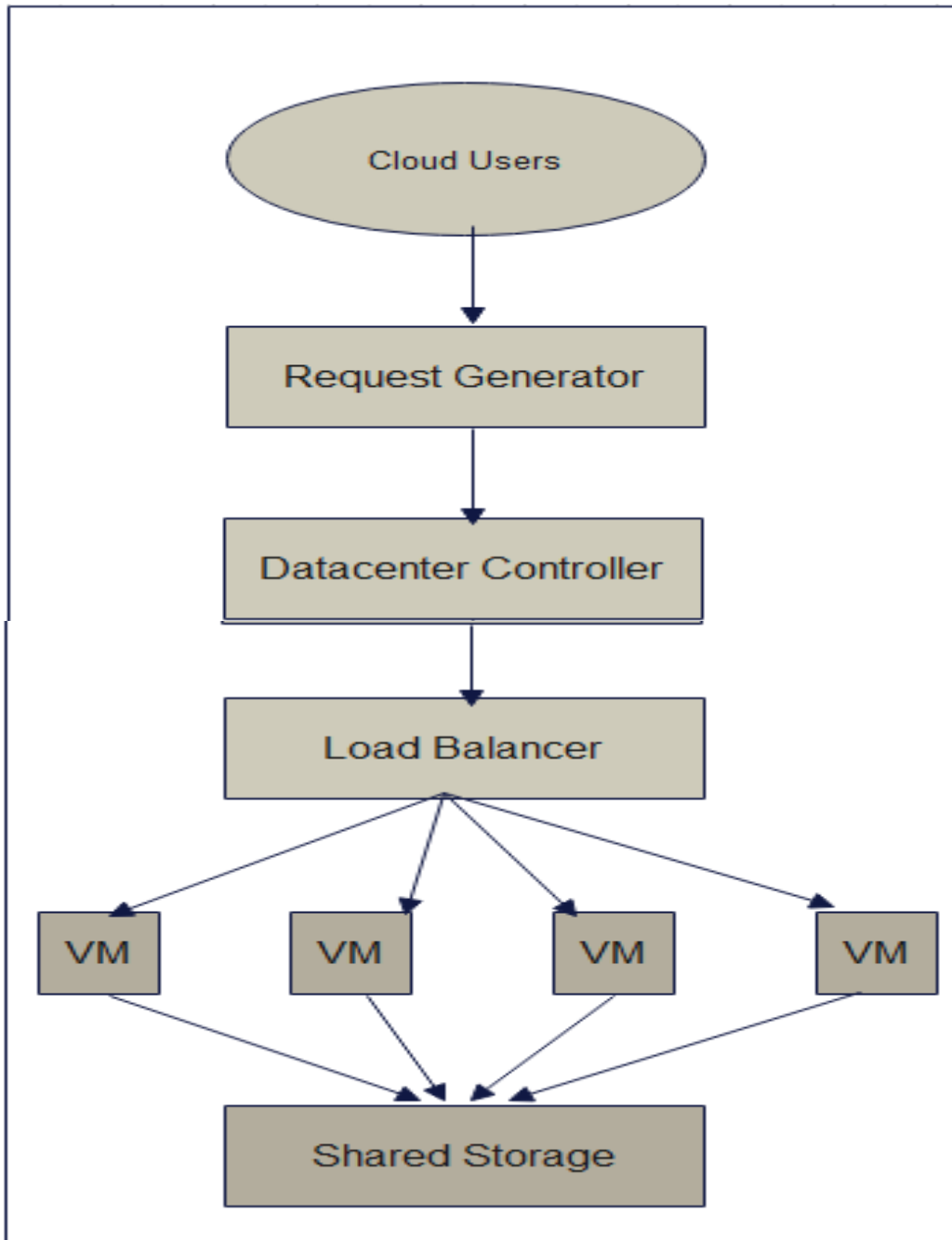


Fig 4.3 Hierarchy of the cloud components

According to the proposed solution architecture, we have showed Activities involved in load balancing Scheduling and allocating tasks to VMs based on their requirements constitute the cloud computing workload. The load balancing process involves the following activities [32].

Identification of user task requirements: - identifies the resource requirement of the user tasks to be scheduled for execution on a VM.

Identification of resource details of a VM: - checks the status of resource details of a VM. It gives the current resource utilization of VM and the unallocated resources. The status of VM can be determined as balanced, overloaded or under-loaded with respect to a threshold based on this phase.

Task scheduling: - Once resource details of a VM are identified, the tasks are scheduled to appropriate resources on appropriate VMs by a scheduling algorithm.

Resource allocation: - The resources are allocated to scheduled tasks for execution. A resource allocation policy is being employed to accomplish this. While, scheduling is required for speeding up the execution, allocation policy is used for proper resource management and improving resource performance. The strength of the load balancing algorithm is determined by the efficacy of the scheduling algorithm and the allocation policy [17] [18][19].

Migration is an important phase in load balancing process in cloud and latter is incomplete without the former. Migration is of two kinds in cloud based on entity taken into consideration- VM migration and task migration. VM migration is the movement of a VM from one physical host to another to get rid of the overloading problem and is categorized into types as live VM migration and non-live migration. Likewise, task migration is the movement of tasks across VMs and is of two types: intra VM task migration and inter VM task migration. A large number of migration approaches has been proposed in literature. An efficient migration technique leads to an efficient load balancing. [20][21][22][23][24].

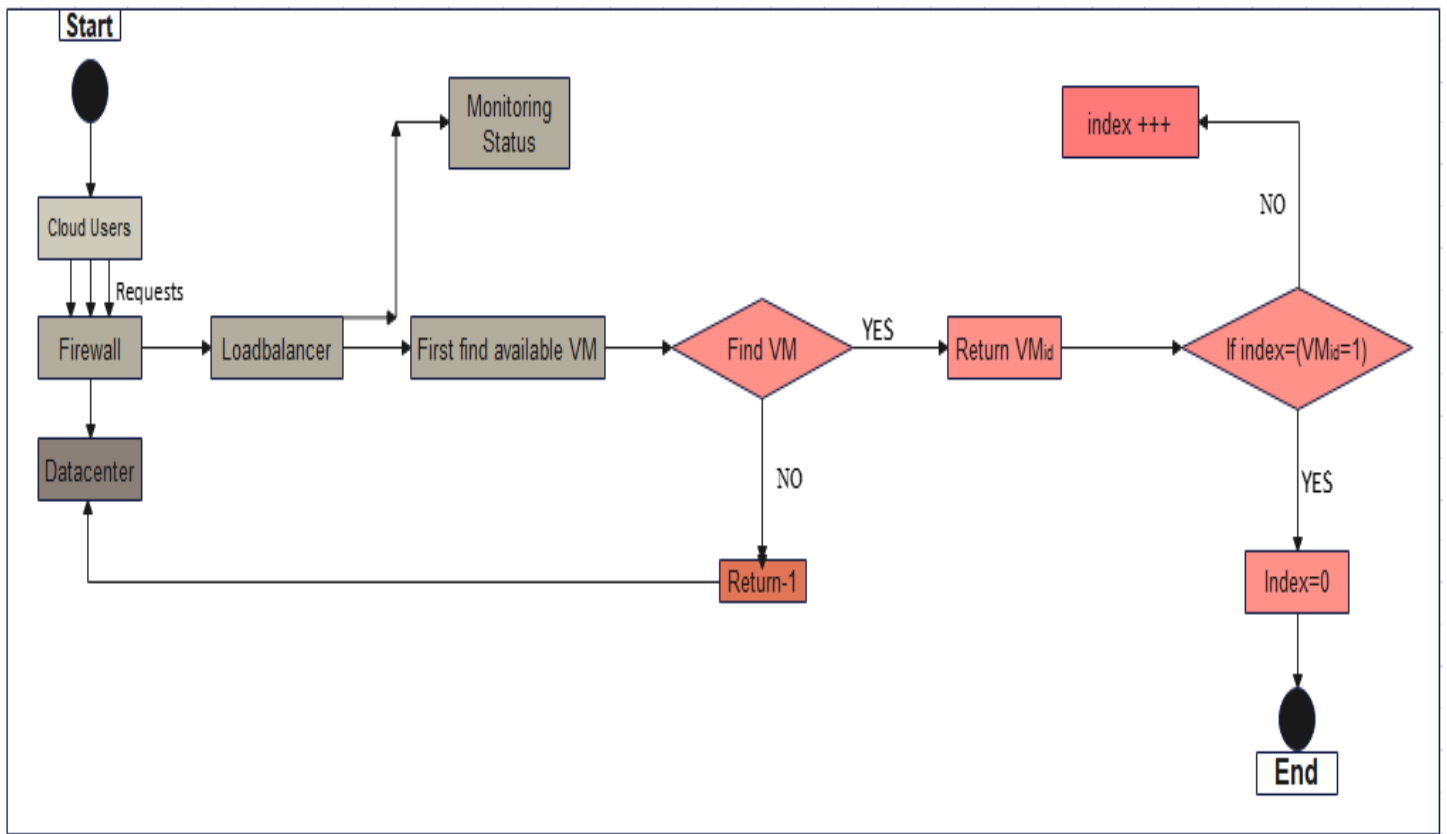


Fig 4.4 Flowchart of Our proposed design

The above figure showed the approach how the loads are balanced. We reviewed different recent approaches. The following approaches aim to improve the performance of Cloud Computing by providing efficient load balancing techniques. The algorithms are reviewed stating their strengths and weaknesses

4.3.1 Load balancing based round Robin

This subsection explains the concept of the Round Robin algorithm and how it can be used to solve Load Balancing challenges in Cloud Computing as proposed in the thesis.

Round Robin (RR) Algorithm [43] works in a circular and ordered procedure where each process is assigned a fixed time slot without any priority. This is a very common algorithm and is often used due to its simplicity in implementation. A common problem in load balancing is that after user requests, the allocation state of the VM is not saved and updated. We proposed energy consumption load scheduling algorithm to modify the existing Response Time. The algorithm allows distributing requests based on servers' energy consumption, and calculates the Response Time and waiting time for each process then decides on the scheduling process. It can reduce the Response Time; however, it did not address the problem of time quantum in RR which makes the algorithm less suitable for dynamic cloud environments.

An enhancement to RR using the Genetic Algorithm (GA) approach is presented in [43]. It aims to provide an efficient load balancing which can improve the capability of Data Centers. To resolve the issues of RR, the approach allocates requests by scanning through the hash map that efficiently contains all VMs. If a VM is available, the task is allocated otherwise, the best VM will be chosen by analysing the best-fitted tasks using GA. Results show that the algorithm reduces the Response Time of servers significantly. However, GA tends to be complex when the search space increases.

Genetic Algorithm (GA) approach is based on the evolution in nature. The algorithm works well since it does not focus on a single point and it resolves the resource deficiency issue resulting in multi-objective optimization. However, the GA algorithm tends to be complex when search space is increased which makes it time-consuming. GA consists of five main phases, the process of the below steps continues until a satisfying result is obtained and a stopped point turns to true state:

Initialization: a group of random individuals is initialized having different characteristics known as genes. They are usually depicted by binary numbers and a collection of them is known as a chromosome.

Fitness Function: an experiment can be done, for example, to select the best fit individuals. These are given fitness values which helps in the next phase, in selection.

Selection: this phase is a preparation for reproduction in the next phase, crossover. Best fit individuals are selected to pass on their genes to the next generation.

Crossover: similar to the mating process where genes are transferred between the selected individuals. The genes of the parents are split into half and recombined this is known as one-point crossover.

Mutation: after the mating process, the genes of the chosen offspring can be modified to produce the best child. This process makes sure there's diversity in the population.

By using the above concept we have proposed energy consumption load scheduling algorithm and improved the communication delay in cloud computing by considering the wasted time occurring in getting sufficient energy with in the servers. It is a problem in load balancing that occurs due to a sudden increase of users' request in cloud services.

Algorithm: - Our proposed work algorithm

1. initialization- server queue Q1, Virtual server V1, Virtual server energy E_v , client request r , request processing P_r , processing time T_p , request filtration F_r , sever allocation A_s , Energy consumption E_c , the remaining energy before data processing E_{r1} . the remaining energy after data processing E_{r2}

$$E_c = \frac{E_{r1} - E_{r2}}{E_{r1}} 100\%$$

2. **forever**
3. **While** Q1 not empty **do**// serving queue
4. allocate server;


```

5.      if filter(r)==TRUE then // request filtration
6.          process request;
7.      if Tp expires then
8.          compute process time;
9.      if Ec (v) ==TRUE then
10.         Compute server energy;
11.         adjust V1 from allocation table;
12.     if Tp expires &&filter(r1) ==TRUE then
13.         if Ec (v1) ==FALSE then//select based on their energy consumption
14.             process/select id Vk;
15.         else virtual server id Vk // select based on their energy consumption
16.         if Vk not. eq V1 then switch (Vk) // migrate to virtual server Vk
17.     endwhile
18. Endforever

```

Initialization: - is the command used to declare the activities to be done.

While Q1 not empty: - is used to check queues whether they are free or not. In this case, load balancer starts to allocate the server that is ready to process the client request.

If filter(r): - is used to filter the client request whether it is directed the available server application. This is to save energy.

If Tp expires: - is used to compute the time after the request processing is finished.

if Ec (v) is a command that is used to compare the consumption energy of the cloud devices.

If Tp expires &&filter(r): - is used to take a decision on selecting virtual server or migrate to other virtual server based on their energy consumption.

In the load balancing, as the webservers demand increases or decreases, the services are assigned dynamically to regulate the changing demands of the user. The servers are grouped under virtual servers (VS), each VS having its own virtual service queues. Each server processing a request from its queue calculates the latency. One measure of this latency can be the amount of time that the CPU spends on the processing of a request. In general, the algorithm shows the steps for processing the client requests which are directed to the available sever application as a result of filtration using firewall. Here filtration is done before processing the request. It is important to save energy of the cloud devices.

CHAPTER FIVE

5 IMPLEMENTATIONS AND RESULT EVALUATION

5.1 Overview

On the basis of the technique used, load balancing algorithms are classified as heuristics and meta-heuristics techniques, and optimization techniques. Heuristics play a crucial role in load balancing process to sort up various issues. We took two applications namely database application and web-based application, and two server objects are used to act as database server and webserver. The application configuration object is used to define the application and the profile configuration object is used to define the application profile.

Table 5.1 Applications description and Simulated Parameters

Application	Attribute	Load	Parameter	Unit
Database	Database	High Load	<ul style="list-style-type: none">• Traffic sent• Traffic received• Server DB Query load• Throughput Utilization	<ul style="list-style-type: none">• Bytes/sec• Bytes /sec• Requests/sec• Packets/sec
Web Browsing	Http	Http Heavy Browsing		

We used virtual machines. One of the functions performed by the first host is to orchestrate restarts of protected virtual machines. VM is protected by a primary host after Center Server observes that the virtual machine's power state has changed from powered off to powered on in response to a user action. the first host persists the list of protected virtual machines within the cluster's data stores.

A newly elected primary host uses this information to see which virtual machines to safeguard. Below is the virtual machine with its parameters. The collection of servers hosting different applications. An end user connects to the datacenter to send different requests. A datacenter may be found at a large distance from the clients.

Table 5.2 The cloud devices with their parameters

S.No	Parameters	Cloud Device		
		Virtual Machine	Host	Cloudlet
1.	Size	10000mb		
2.	RAM	512mb	10240mb	
3.	mips	1000		
4.	Bandwidth	1000	1000	
5.	CPU	1		1
6.	Length			1000
7.	File Size			300mb
8.	Output Size			300mb
9.	Storage		100,000mb	

5.2 Load balancing metrics

We review the metrics for load balancing in cloud computing. As discussed before, researchers have proposed several load-balancing algorithms. Literature in load balancing proposed metrics for applying load-balancing algorithms and we summarize them as follows:

- ❖ Throughput: is a metric that is used to calculate the number of processes completed per unit time.
- ❖ Response time: measures the time that the system takes to serve a submitted task.
- ❖ Make span: is a metric that is used to calculate the maximum completion time or the time when the resources are allocated to a user.
- ❖ Scalability: is the ability of an algorithm to perform uniform load balancing in the system according to the requirements upon increasing the number of nodes. The preferred algorithm is highly scalable.
- ❖ Fault tolerance: determines the capability of the algorithm to perform load balancing in the event of some failures in some nodes or links.
- ❖ Migration time: is the amount of time required to transfer a task from an overloaded node to an under-loaded one.
- ❖ Degree of imbalance: is a metric measures the imbalance among VMs.
- ❖ Performance: measures the system efficiency after performing a load-balancing algorithm.
- ❖ Energy consumption: calculates the amount of energy consumed by all nodes. Load balancing helps to avoid overheating and therefore reducing energy usage by balancing the load across all the nodes.
- ❖ Carbon emission: calculates the amount of carbon produced by all resources. Load balancing has a key role in minimizing this metric by moving loads from under loaded nodes and shutting them down.

5.3 Data Processing Tools

As the relative studies have been introduced in [40] on the data processing and analysis platforms use Cloudsim and excel. Both of them are interpreted language. This means that their code can be altered between all of the major operating system platforms and CPU architectures out there, with only small changes required for different platforms.

Table 5.3 The java run file (the existing cloud computing)

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	80	0.1	80.1
1	SUCCESS	2	1	88.89	0.1	88.99
2	SUCCESS	2	2	94.12	0.1	94.22
3	SUCCESS	2	3	106.67	0.1	106.77

Table 5.3 shows the time taken to process a request in each server independently of the existing cloud computing. Server1 took less time than of Server2, 3 and 4. This indicated that its response time is low and resulting less communication delay comparing with the others. The figure below (Fig 5. 1) represents the response time to process the requests in each server which is important to understand communication delay.

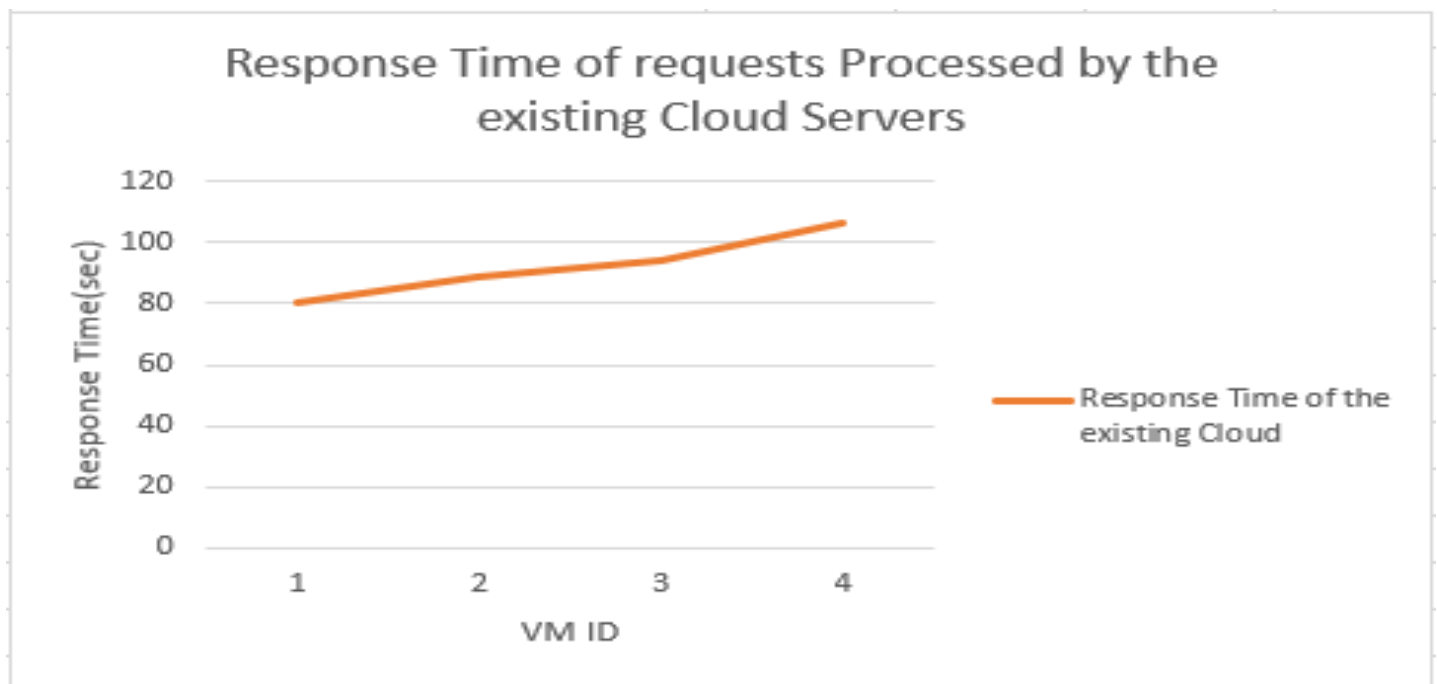


Fig 5.1 The response time of the existing cloud

Table 5.4 The java run file (The remaining energy and the status of the energy consumption of each server)

VM ID	STATUS	RE Beforsent	RE Aftersent	Energy Consumption
0	SUCCESS	75	71	5.33 %
1	SUCCESS	88	76	13.64 %
2	SUCCESS	82	68	17.07 %
3	SUCCESS	90	70	28.57 %

Table 5.5 the java run file (the proposed cloud computing)

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	22.86	0.1	22.96
1	SUCCESS	2	1	26.67	0.1	26.77
2	SUCCESS	2	2	32	0.1	32.1
3	SUCCESS	2	3	40	0.1	40.1

Table 5.5 shows the time taken to process a request in each server independently of the proposed cloud computing. Server1 took less time than of Server2, 3 and 4. This indicated that its response time is low and resulting less communication delay comparing with the others. The proposed cloud computing used energy consumption load scheduling mechanism which helped to distribute the loads based on energy consumption of the servers. So, the mechanism helped proposed cloud computing to have less communication delay. Fig 5. 2 represent the response time to process the requests in each server which is important to understand communication delay.

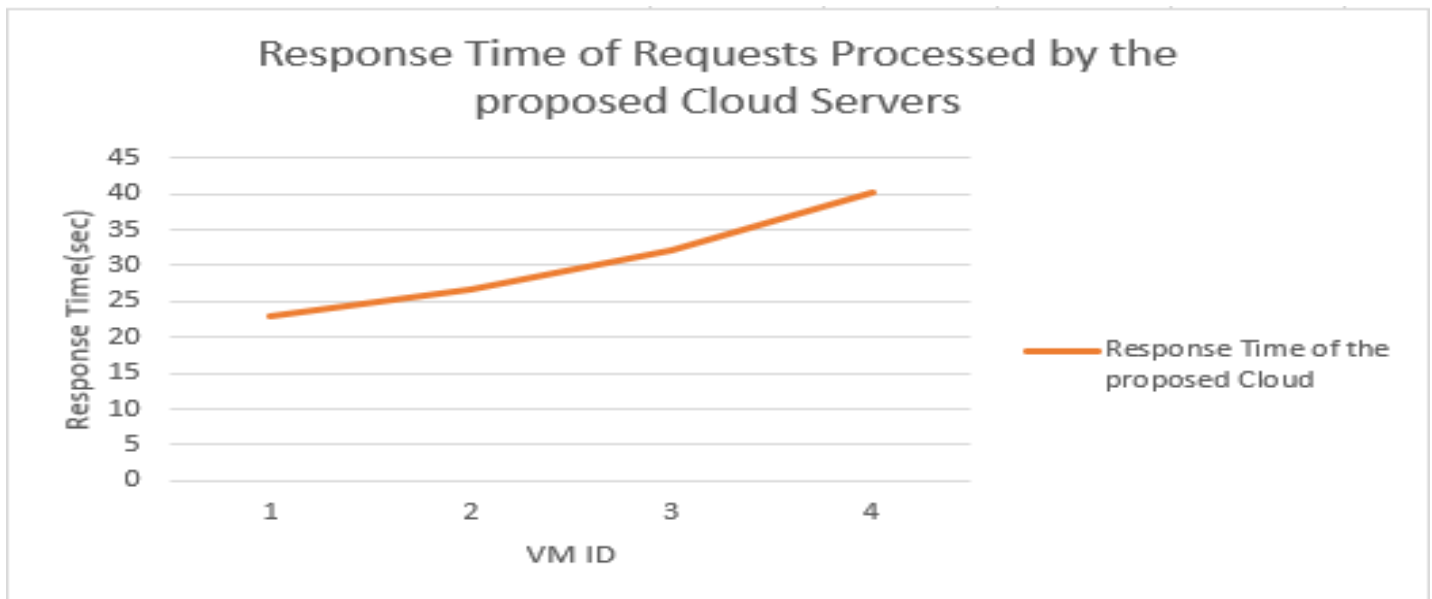


Fig 5.2 The Response time of the proposed cloud

As we introduced in the previous section design flow, cloudsim simulates our proposed work that java files (.java files) generated by cloudsim can be exported to excel. In general, we used excel in our proposed work to get the reduced latency.

Fig 5.3 represents the response time to process the requests in each server which is important to understand communication. it shows the time taken to process a request in each server independently of the proposed cloud computing. Server1 took less time than of Server2, 3 and 4. This indicated that its response time is low and resulting less communication delay comparing with the others. When we compare the response time of servers in the proposed cloud computing is less than that of servers in the existing cloud computing. The proposed cloud computing used energy consumption load scheduling mechanism which helped to distribute the loads based on energy consumption of the servers. So, the mechanism helped proposed cloud computing to have less communication delay. Therefore, both having less response time and distributing the loads using servers' energy consumption lead to have reduced latency.

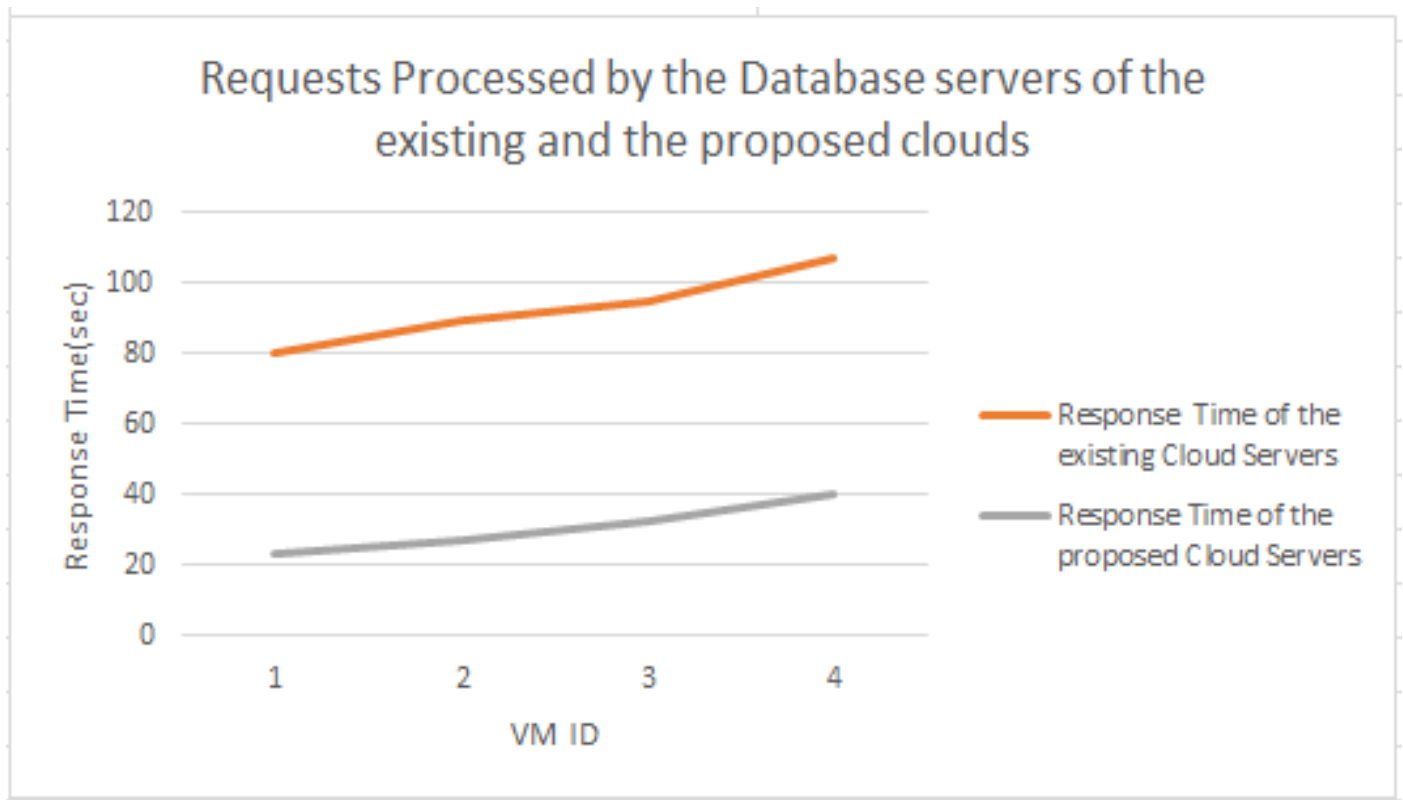


Fig 5.3 Comparison of latency between the existing cloud and the proposed cloud servers

5.4 Summary

The performance benefit to using load balancer: reduced latency when data can be found in the VM and improved database scalability due to reduced request volume. Load balancer is preferable because it can increase the total capacity of the cloud devices by distributing the data across multiple servers, and it allows multiple web servers to access the data.

Table 5.6 latency of the existing cloud

S. No	Parameters	VM1	VM2	VM3	VM4
1.	Sent data (in Mb)	75	75	75	75
2.	Response Time of the existing Cloud/A cloud with a load balancer (sec)	80	88.89	94.12	106.67

Table 5.7 latency of the proposed cloud

S. No	Parameters	VM1	VM2	VM3	VM4
1.	Sent data (in Mb)	75	75	75	75
2.	Response Time of the Proposed Cloud/A cloud with a load balancer working on energy consumption (sec)	22.86	26.67	32	40

Table 5.8 Comparison of latency between the existing cloud and the proposed cloud

S. No	Parameters	VM1	VM2	VM3	VM4
1.	Sent data (in Mb)	75	75	75	75
2.	Response Time of the existing Cloud(sec)	80.1	88.99	94.22	106.77
3.	Response Time of the Proposed Cloud(sec)	22.96	26.77	32.1	40.1
4.	Latency Reduction scored by each server of the proposed cloud (%)	71.33	69.91	65.93	62.44
5.	Average Latency Reduction scored by proposed cloud	67.40%			

The above table (Table 5.8) shows that implementation can reduce the latency of the cloud devices by 67.40% and given reasonable VM hit rates can reduce the total latency of data store compared to using standard one, depending on the workload.

In our solution, the requests came from the users are filtered by embedded firewall to avoid the requests not for concerned server. This is to reduce the consumption of the device’s energy. In the above result, database server traffic in the form of bytes/sec receives more bytes of data when we use load balancer. Therefore, Response Time is reduced with using the Load Balancing process.

CHAPTER SIX

6 CONCLUSION, CONTRIBUTION, AND FUTURE WORKS

6.1 Conclusion

With all the good news about cloud computing what are the issues? The number one issue is latency that is the time it takes for a signal to travel from one point to another in telecom networks. Businesses' increased reliance on telecom networks and improved processing speeds for computer networks and the Local Area Networks (LANs)/Wide Area Networks (WANs) that interconnect them, have combined to make the speed at which a signal travels in a telecom network more noticeable and critical.

Technologies such as cloud computing cannot be successful if the end user does not have a good experience. Applications which are most susceptible to latency are those which depend most heavily on high transaction rate processes which drive CPU per second cycles, memory and storage read/write requests, and server requests. Examples of latency sensitive applications range from multimedia streaming, video transcoding, multi-player network gaming to telesurgery and computerized trading.

In this thesis, a brief overview of cloud computing is discussed, there are many services offered by cloud platforms that has been used by many organizations. The load balancing mechanism is also explained, it can be used at a very low cost and can reduce the response time. In this thesis, we have also discussed the methods of reducing the latencies. However, this thesis did not address all factors those make latency very less rather it addressed the one factor, there are other factors therefore to be avoided. Also, the study did not address security of the cloud computing environment because of security by itself is a broad concept.

6.2 Contribution

Our contribution is in order to analyze the consumption of energy for the servers and virtual machines. To make this, we used an efficient load balancing algorithm. response time become low; resource utilization become high. Also, the delay on load distributing is minimized because the load balancer does that the load distributing is done based on energy consumption of each server. A procedure of disseminating tasks is on the basis of energy consumption which can be analyzed as load balancing task scheduling.

Furthermore, a proposed solution addresses the high latency issue in the cloud computing environment and how to avoid it. Although, this issue has been slightly addressed by researchers in the past, however, most focus on the use of a load balancer. The proposed solution addresses on reducing latency by using load balancer based on energy consumption.

6.3 Future Work

Cloud computing networks are moving away from the typical three-layer switching topology in which access switches are connected to a large pool of aggregation or distribution switches that are then connected to the core.

The concern with the traditional model is latency that forces packets to stop at hops at every layer and doesn't provide any-to-any communication between the hundreds of servers and migrating VMs (virtual machines) necessary in a cloud environment. Since the further enhancements are still required for maintaining the quality of the services by providing less delay factor or high throughput, modules that are provided by the cloud computing.

7 References

- [1] D. S. B. Maulik Parekh, "Designing a Cloud based Framework for HealthCare System and applying Clustering techniques for Region Wise Diagnosis," 2015.
- [2] K. S. R. D. Q. J.SRINIVAS, " CLOUD COMPUTING BASICS," *international Journal of Advanced Research in Computer and Communication Engineering*, 2012.
- [3] G. T. Mell P, "The NIST definition of cloud computing."
- [4] A. W. G. a. P. lu, " Low Latency Caching for Cloud – based Web applications," Sept. 16, 2011.
- [5] A. S. a. Hemalatha, " Comparative analysis of Low latency on different bandwidth and geographical locations while using cloud based applications," Jan 2012.
- [6] M. Heidari, " The Role of Modeling and Simulation in Information Security the Lost Ring," 1989.
- [7] A. V. Reddy, " Usage of Opnet I.T tool to Simulate and test the security of cloud," *Project id-395*.
- [8] S. S. a. S. K. P. Sharma, "Cloud Computing issues and what to compute on Cloud," *International Conference on Advanced Computing Communications and Network*.
- [9] S. a. S. p. Singh, " Analysis of Energy Consumption in Different types of networks For Cloud Environment," *IJARCSSE Vol 2, Issue 2,ISSN: 2277 128X*, Feb 2012.
- [10] N. K. a. D. R.Sridaran, " A Survey on Security Threats for Cloud computing," *International journal of engineering research and technology (IJERT) Volume.1 Issue7* , September-2012.
- [11] A. Z. a. M. N. A. Khan, " Identifying key Challenges in Performance Issues in Cloud Computing," September 2012.
- [12] NIST, "The NIST Definition of Cloud Computing," *U.S.: National Institute of Standards and Technology*, 2011.
- [13] T. J. R. E. A. T.Velte, "Cloud Computing A Practical Approach," *TATA McGRAW-HILL Edition* , 2010.
- [14] S. N. a. K. Dilip, " Cluster, Grid and Cloud Computing: A Detailed Comparison," *The 6th International Conference on Computer Science & Education (ICCSE 2011)*, 2011.
- [15] Kumar M, Dubey K, Sharma SC (2018) Elastic and flexible deadline constraint load balancing algorithm for cloud computing. *Proced Comp Sci* 125:717–724
- [16] J. D. S. G. W. C. H. D. A. W. M. B. T. C. A. F. a. R. E. G. B. .. F. Chang, " A Distributed Storage System for

Structured Data," *ACM Transactions on Computer Systems*," 2008 .

- [17] H. L. a. Y. Hu, " Analysis and Research about Cloud Computing Security Protect Policy," *IEEE, 2011, pp. 214-216*, 2011.
- [18] S. a. S. p. Singh, *Analysis of Energy Consumption in Different types of networks For Cloud Environment IJARCSSE Vol 2, Issue 2,ISSN: 2277 128X.*, Feb 2012.
- [19] F. a. R. Raihana, " Revealing the Criterion on Botnet Detection Technique," *International journals of Computer science issues, vol 10,Issue 2, No 3, , March 2013*.
- [20] P. D. S. B. J. S. O. M. M. SK, " An adaptive task allocation technique for green cloud computing," *J Supercomp 405:1–16* , 2017.
- [21] F. H. M. Y. H. A. I. AH, " Resource allocation algorithm for GPUs in a private cloud," *Int J Cloud Comp 5(1–2):45–56* , 2016.
- [22] Alexandru LIXANDRU "Efficient Architectures for Low Latency and High Throughput Trading Systems on the JVM" *information economics vol. 17, no. 3/2013*.
- [23] P. D. S. B. J. S. O. M. M. SK, " An adaptive task allocation technique for green cloud computing," *J Supercomp 405:1–16*, 2017.
- [24] F. H. M. Y. H. A. Ibrahim AH, " Resource allocation algorithm for GPUs in a private cloud.,", " *Int J Cloud Comp 5(1–2):45–56*, 2016.
- [25] Rajput SS, Kushwah VS (2016, December) A genetic based improved load balanced min-min task scheduling algorithm for load balancing in cloud computing. In: 2016 8th international conference on Computational Intelligence and Communication Networks (CICN), pp 677–681
- [26] .. A. A. D. A. N. J., " Load balancing techniques in cloud computing environment," 2011.
- [27] Adhikari M, Amgoth T (2018) Heuristic-based load-balancing algorithm for IaaS cloud. *Futur Gener Comput Syst 81:156–165*
- [28] .. R. B. ., R. D. ., V. S. A. A. I. B., " Addressing Application Latency Requirements through Edge Scheduling," *J Grid Computing (2019) 17:677–698*, 2019.
- [29] .. N. N. E. J. A. M., " Load-balancing algorithms in cloud computing: A survey," *Journal of Network and Computer Applications 88 (2017) 50–71*, 2017.
- [30] M. H. A. S. N, "COMPARATIVE ANALYSIS OF LOW-LATENCY ON DIFFERENT BANDWIDTH AND GEOGRAPHICAL LOCATIONS WHILE USING CLOUD BASED APPLICATIONS," *ISSN: 2231-1963*.

- [31] D. R. S. A. LIXANDRU, " A Survey on Security Threats for Cloud Computing," *ISSN: 2278-0181*, 2012.
- [32] P. P. D. S. K. Patel, " An efficient and modified load balancing method for cloud computing," *Int. J. Innov. Res. Comput. Commun. Eng. 5 (4),8198–8205*, , 2017.
- [33] A. Zia, " Identifying Key Challenges in Performance Issues in Cloud Computing," *I.J.Modern Education and Computer Science, 2012, 10, 59-68*, 2012.
- [34] M. P. S. S. S, "Impact of Latency on the Economics of Cloud Computing," *IJCSIT) International Journal of Computer Science and Information Technologies, 2016, 1414-1416*, , 2016.
- [35] G. K. S. A, " Load balancing in cloud computing – A hierarchical taxonomical classification, Afzal and Kavitha," *Journal of Cloud Computing: Advances, Systems and Applications (2019) 8:22*, 2019.
- [36] S. K. M. SB, " Effect of Latency on Network and End User Domains in Cloud Computing (2013)," 2013.
- [37] P. L. A. WG, " Low-Latency Caching for Cloud-Based Web Applications," *ACM 978-1-4503-0652* , 2011.
- [38] S. S. a. S. K. P. Sharma, " Cloud Computing issues and what to compute on Cloud," *International Conference on Advanced Computing Communications and Network* .
- [39] A. M. Alakeel, " A Guide to Dynamic Load Balancing in Distributed Computer Systems," *IJCSNS International Journal of Computer Science and Network Security*. A. M. Alakeel, " A Guide to Dynamic Load Balancing in Distributed Computer Systems," *IJCSNS International Journal of Computer Science and Network Security*.
- [40] I. A. Vouk, "Cloud Computing Issues, Research and Implementations," *Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, 2008, June23-26* .
- [41] Atakan Aral, Rafael Brundo Uriarte , Vincenzo Scoca, "Addressing Application Latency Requirements ugh Edge Scheduling", 2019.
- [42] Batyr Charyyev, Engin Arslan, Mehmet Hadi Gunes, "Latency Comparison of Cloud Datacenters and Edge ers" IEEE Global Communications Conference,2020 Taipei, Taiwan
- [43] NS Kaurav, P Yadav, "A genetic algorism based load balancing approach for resource optimization for cloud computing environment", *int.J. computer sci.,2019-ijics.com*

Appendix A

Sample of .java file

```
Starting Cloud2...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #1
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #3 has been created in Datacenter #2, Host #1
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
0.1: Broker: Sending cloudlet 2 to VM #2
0.1: Broker: Sending cloudlet 3 to VM #3
114.38571428571429: Broker: Cloudlet 2 received
114.38571428571429: Broker: Cloudlet 1 received
114.38571428571429: Broker: Cloudlet 3 received
115.20585817060638: Broker: Cloudlet 0 received
115.20585817060638: Broker: All Cloudlets executed. Finishing...
115.20585817060638: Broker: Destroying VM #0
115.20585817060638: Broker: Destroying VM #1
115.20585817060638: Broker: Destroying VM #2
115.20585817060638: Broker: Destroying VM #3
```



```
vmList = new ArrayList<Vm>();
```

```
int vmid = 0;
```

```
int mips = 250;
```

```
long size = 10000;
```

```
int ram = 2048;
```

```
long bw = 1000;
```

```
int pesNumber = 1;
```

```
String vmm = "Xen";
```

```
vmList.add(vm1);
```

```
vmList.add(vm2);
```

```
vmList.add(vm3);
```

```
vmList.add(vm4);
```

```
broker.submitVmList(vmList);
```

```
cloudletList = new ArrayList<Cloudlet>();
```

```
int id = 0;
```

```
long length = 40000;
```

```
long fileSize = 300;
```

```
long outputSize = 300;
```

```
Cloudlet cloudlet1 = new Cloudlet(id, length, pesNumber, fileSize, outputSize,  
utilizationModel, utilizationModel, utilizationModel);
```

```
cloudlet1.setUserId(brokerId);
```

```
id++;
```



```
Cloudlet cloudlet2 = new Cloudlet(id, length, pesNumber, fileSize, outputSize,  
utilizationModel, utilizationModel, utilizationModel);
```

```
cloudlet2.setUserId(brokerId);
```

```
id++;
```

```
Cloudlet cloudlet3 = new Cloudlet(id, length, pesNumber, fileSize, outputSize,  
utilizationModel, utilizationModel, utilizationModel);
```

```
cloudlet3.setUserId(brokerId);
```

```
id++;
```

```
Cloudlet cloudlet4 = new Cloudlet(id, length, pesNumber, fileSize, outputSize,  
utilizationModel, utilizationModel, utilizationModel);
```

```
cloudlet4.setUserId(brokerId);
```

```
cloudletList.add(cloudlet1);
```

```
cloudletList.add(cloudlet2);
```

```
cloudletList.add(cloudlet3);
```

```
cloudletList.add(cloudlet4);
```

```
broker.submitCloudletList(cloudletList);
```

```
broker.bindCloudletToVm(cloudlet1.getCloudletId(),vm1.getId());
```

```
broker.bindCloudletToVm(cloudlet2.getCloudletId(),vm2.getId());
```

```
broker.bindCloudletToVm(cloudlet3.getCloudletId(),vm3.getId());
```

```
broker.bindCloudletToVm(cloudlet4.getCloudletId(),vm4.getId());
```

```
CloudSim.startSimulation();
```

```
List<Cloudlet> newList = broker.getCloudletReceivedList();
```

```

        CloudSim.stopSimulation();

printCloudletList(newList);
Log.println("Cloud2 finished!");
}
catch (Exception e) {
e.printStackTrace();
Log.println("The simulation has been terminated due to an unexpected error");
}
}

private static loadbalancer createLoadbalancer(String string) {
    for (int i = 0; i < Ec; i++){
        SelectionKey VmId;

        return null;
    }

private static Datacenter createDatacenter(String name){

    List<Host> hostList = new ArrayList<Host>();

    List<Pe> peList = new ArrayList<Pe>();

    int mips = 10000;

    peList.add(new Pe(0, new PeProvisionerSimple(mips)));

    int hostId=0;

    int ram = 2048;

    long storage = 1000000;

    int bw = 10000;

    hostList.add(

```

```

        new Host(
            hostId,
            new RamProvisionerSimple(ram),
            new BwProvisionerSimple(bw),
            storage,
            peList,
            new VmSchedulerTimeShared(peList)
        )
    );
    List<Pe> peList2 = new ArrayList<Pe>();
    peList2.add(new Pe(0, new PeProvisionerSimple(mips)));

    hostId++;

    hostList.add(
        new Host(
            hostId,
            new RamProvisionerSimple(ram),
            new BwProvisionerSimple(bw),
            storage,
            peList2,
            new VmSchedulerTimeShared(peList2)
        )
    );

    String arch = "x86";
    String os = "Windows";

```

```

String vmm = "Xen";

double time_zone = 10.0;

double cost = 3.0;

double costPerMem = 0.05;

double costPerStorage = 0.001;

double costPerBw = 0.0;

LinkedList<Storage> storageList = new LinkedList<Storage>(

DatacenterCharacteristics characteristics = new DatacenterCharacteristics(
    arch, os, vmm, hostList, time_zone, cost, costPerMem, costPerStorage,
costPerBw);

Datacenter datacenter = null;

try {

    datacenter = new Datacenter (name, characteristics, new
VmAllocationPolicySimple(hostList), storageList, 0);

} catch (Exception e) {

    e.printStackTrace();

}

return datacenter;

}

private static DatacenterBroker createBroker(){

DatacenterBroker broker = null;

try {

    broker = new DatacenterBroker("Broker");

} catch (Exception e) {

    e.printStackTrace();

```

```
        return null;
    }
    return broker;
}

private static void printCloudletList(List<Cloudlet> list) {
    int size = list.size();
    Cloudlet cloudlet;

    DecimalFormat dft = new DecimalFormat("###.##");
    for (int i = 0; i < size; i++) {
        cloudlet = list.get(i);
```