AN IMPLEMENTATION OF AN ALGORITHM FOR LOAD BALANCING IN CLOUD

A Dissertation submitted in partial fulfillment of the requirement for the

Award of degree of

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By

JAYANT SHARMA

(2K13/ISY/09)

Under the guidance of

Dr. N. S. RAGHAVA

Associate Professor

Department of Computer Science and Engineering

Delhi Technological University

Bawana Road, Delhi-110042

2013-2015

CERTIFICATE

This is to certify that the thesis entitled **"An Algorithm for Load Balancing in Cloud Computing"** submitted by **Jayant Sharma(2K13/ISY/09)** to the Delhi Technological University, Delhi for the award of the degree of **Master of Technology** is a bona-fide record of research work carried out by her under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: DTU, Delhi
Date: _____

Dr. N.S Raghava
Associate Professor
Department of ECE
Delhi Technological University,
Delhi.

ACKNOWLEDGEMENT

First I would like to express my gratitude towards my supervisor **Dr. N. S. Raghava**, *Associate Professor, Department of ECE* for his able guidance, support and motivation throughout the time. It would not have been possible without the kind support and help of many individuals and **Delhi Technological University**. I would like to extend my sincere thanks to all of them.

I would like to express my gratitude and thanks to **Dr. O.P Verma *(Head of Dept.)*** for giving me such an opportunity to work on the project.

I would like to express my gratitude towards my **parents** & **staff** of Delhi Technological University for their kind co-operation and encouragement which helped me in completion of this project.

My thanks and appreciations also go to my **friends and colleagues** in developing the project and people who have willingly helped me out with their abilities.

Jayant Sharma
Roll No.: 2K13/ISY/09
Dept. of CSE
Delhi Technological University

ABSTRACT

Cloud computing is a responsible term for the new technological growth in the present IT market. Cloud computing has sparked a key interest to various organizations, many technical institutions and service users to take the advantage of advanced technology for their particular process execution tasks. Because of serving a very attractive package of services, cloud technology has grasped a huge attention from academia, IT industry and government organizations. Cloud computing provides a very high scale integrated development environment for executing the request response architecture between the service provider and the service users. By the help of cloud computing a user can access huge amount of computational resources by paying corresponding money to cloud service provider without worrying the infrastructure of the resources which he is used at that moment of time for his applications. As the number of users on the internet goes on increasing, it becomes difficult to handle the millions of user requests on it. Many times it happens during the peak-hours that the traffic on the network increases abruptly. In such situations it is commonly observed that the system performance degrades. This problem is solvable if the load balancing is optimal in their load assignment by the cloud service provider.

Therefore, one of the important issues which need a major consideration of the researchers is load balancing in cloud computing systems. A number of load balancing algorithm are proposed by various researchers, to solve this problem. Two kinds of load balancing algorithms are there one is static and other is dynamic. A cloud computing system is supposed to handle the request dynamically rather than doing it by a static approach which is supposed to be the less efficient approach. Also there are number of parameters like resource utilization, throughput, scalability, flexibility, response time, which are used to validate any load balancing algorithm. Any load balancing developer is supposed to maintain a good of trade-off between these parameters. In this thesis a new approach for load balancing in cloud computing is proposed. This algorithm aims at distributing the equal load on each server in the cloud network. This also improves the resource utilization. With the proposed approach a situation in which only few resources are loaded heavily while others are just sitting idle will never arise. The proposed algorithm is also compared with the existing load balancing techniques.

TABLE OF CONTENTS

TITLE **PAGE NO.**

Chapter 3: Load Balancing

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 RESEARCH BACKGROUND

Cloud computing has significantly improved its existing technologies in context of the services they provide. In previous years, Grid as well as Distributed computing was unable to provide the flexibility as compared to cloud computing. Various attractive features have resulted in the increment of the popularity of cloud computing. The model which cloud computing follows is pay-as-you-go and provides on demand provisioning of resources. The raising standards have increased the demand of cloud computing in the field of business where infrastructure setup is high. Using this technology of cloud, one can easily have the run time environments, services and infrastructure. Users from different domains can make best of the benefits of cloud computing as per their needs. The main entity is the cloud provider which enables users to use different services as per their needs. They provide the customers with many resources like network, storage capacity which can be joined with any registration process where any authenticated user can request and work with the services of cloud.

Among the various different tasks of provider of cloud services, the crucial one is to assign nodes for requests of user. This mentioned task need to be efficient. In order to perform this, total processing time should be less [1] as possible whereas in parallel various other issues of network delays and heterogeneity need to be managed. Cloud computing is so popular in the field of Information and Communication Technology (ICT). To cope up with rapid increment in cloud requests, providers need to build more capabilities in components which can be performed by increasing their count or power or both. This situation may turn into a large network consisting of cloud users, nodes, tasks and virtual machines. As the network size becomes large, workload also need to be managed to increase overall performance. This area of load balancing aims at high detection rate and best distribution of workload in network.

Major issue which earns more attention from cloud developers is the load balancing task. Various parameters are tested for algorithms of load balancing for better quality service. In turn, this parameter is analyzed to judge the concerned algorithm.

1.2 CHALLENGES AND MOTIVATION

Cloud Computing is earning more attention from users as it is scalable, elastic, easy accessible and last but not the least delivery of services. These features contribute in making it marketable. This model may lead to set a new world to cloud computing where every user can run its services as per requirements, not knowing how the internal process is working. The overall workload of the entire procedure of cloud computing is directly proportional to the number of users accessing the services. It's a challenging task to manage the resources like memory, CPU, secondary storage. It's a prerequisite of any cloud service to provide load balancing in an effective manner.

In any scenario it may happen, that the traffic may rise in exponential manner, e.g. in the occasion of New Year, people may wish each other by sending messages which may create traffic in websites. In this critical condition, they may raise the charges of sending messages. Similar traffic may be obtained while results of Board exams are declared. Another major challenge to be faced is the federation of cloud. A cloud may comprise of private, public, community or hybrid cloud. Any communication between these may result in addition of responsibilities of service provider. Also, the requirement of Internet is the prior need to provide any cloud services. Therefore any bottlenecks in network that occur frequently may result in a major issue and then resources need to be effectively used.

1.3 OBJECTIVES AND CONTRIBUTIONS

This study of thesis relates to LB challenges in cloud computing. The major objective of this thesis is to generate an effective mechanism for load balancing, which may be a benefit for both the service providers as well as cloud users. Proper distribution of the load among the existing nodes in network will be the best way to utilize the resources and to process the requests of user. This task of load balancing can be performed by scheduling of tasks into VM and then from virtual machine to nodes existing in system. An effective algorithm for LB will get high rate of fault tolerance.

The Major contribution deals with:

- A technique for load balancing with high concern on improvement of performance parameters like response time, resource utilization and throughput.
- A Comparative study and analysis of existing major load balancing algorithms. Various contributors like peak hour usage and distribution of services are studied and analyzed for algorithms.

CHAPTER 2

Cloud Computing

2.1 INTRODUCTION

Previous years have resulted into the approach to process information in larger scope, in more effective manner. With the advancement in the area of cloud computing, trauma of storing, retrieving and processing big data through internet is not seen now. The idea of computation which is now network oriented in early 1990s led to the development of Grid computing and from 2015 , the computation entered a new field of Cloud computing. From many previous years, researchers are focusing on bringing utilities to be given as service to their users. User may want to acquire software, hardware or any platform based on usage through an internet. It can be said that cloud computing is simply a path to provide utility espoused by IT agents like IBM, Amazon, Google, HP etc. The technology dealing in cloud has a tremendous rise which is displayed in the figure 2.1. The technology spread in the entire globe in a very short interval of time.

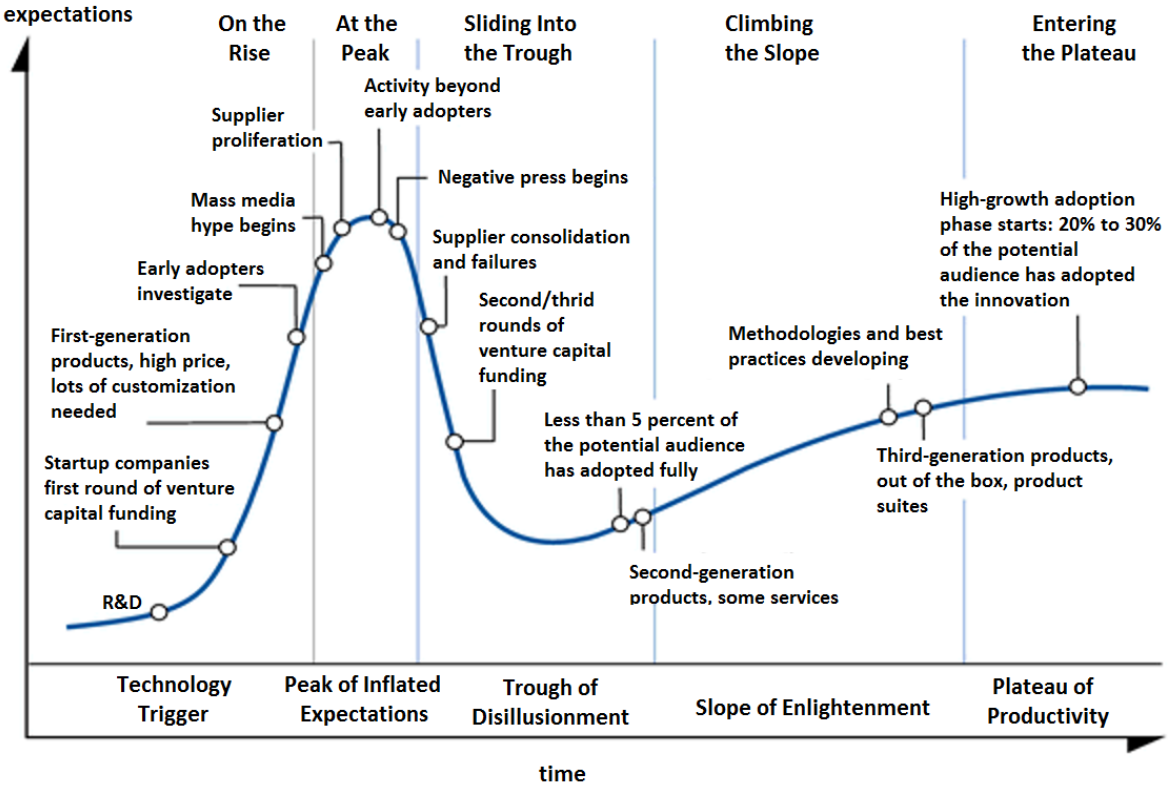

Figure2.1: Hype cycle for cloud computing

2.1.1 CLOUD COMPUTING DEFINITIONS

The word Cloud is one of the most famous and popular words since 2007 in the IT industry. Different researchers gave a different definitions for cloud computing. According to their application perspective, they gave their own interpretation of the definition to cloud computing. Among the existing ones, the two famous quoted definitions are as follows:

- **NIST:** "*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* [3]."

- **Foster:** "*A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over Internet* [4]."

In relation to this, Joe Weinmn gave a term "Cloudonomics", which states it from economical view, shown below:

1. **C**ommon Infrastructure: It states that a pool of resources should be provided to users.
2. **L**ocation-independence: It states that a user can work using any resource and it does not depend on its location.
3. **O**nline connectivity: It states that a user needs to use any services only through a connection which is consistent.
4. **U**tility Pricing: It declares that the payment should be done as per use and benefits should be provided in accordance with their demands.
5. On-**D**emand Resources: It informs that scalable and elastic resources should be managed and then provided.

2.1.2 CLOUD COMPONENTS

The components of the cloud work as an integrated unit which comprises of data centers, servers and clients as shown below:

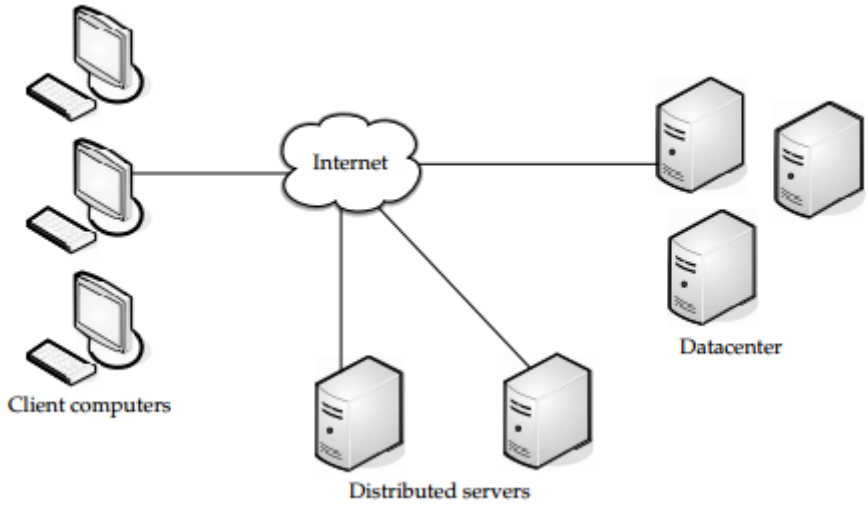

Figure 2.2: Cloud Components

Each component works as per specific role demanded by the user. They communicate in the overall network according to the requirements inform about the roles. These are explained below:

Clients

It includes the computers we generally use in our life. It may consist of desktop computer, mobile phone, PDA. These clients should be able to gain access to use the cloud interface through internet. And then they can gain the cloud services using this interface accordingly.

Data Centers

It acts as the most core element of the entire cloud process. It may comprise of many nodes in a room. Configuration of data centers is performed by service providers and they rely on service and deployment model.

Distributed Servers

Servers also considered as nodes, doesn't need to be fit in the same location as the user whereas it can be deployed anywhere irrespective of their geographic location. It seems to the user as the server is working together, but in reality they are unaware of the actual location. The use of distributed servers results in increment of fault tolerance rate.

2.2 CLOUD COMPUTING ARCHITECTURE

Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet. Such characterization includes quite different aspects: infrastructure, development platforms, application and services. It is possible to organize all the concrete realizations of cloud computing into a layered view covering the entire stack from hardware appliances to software systems. Figure 2.3 shows the layered architecture of cloud computing.

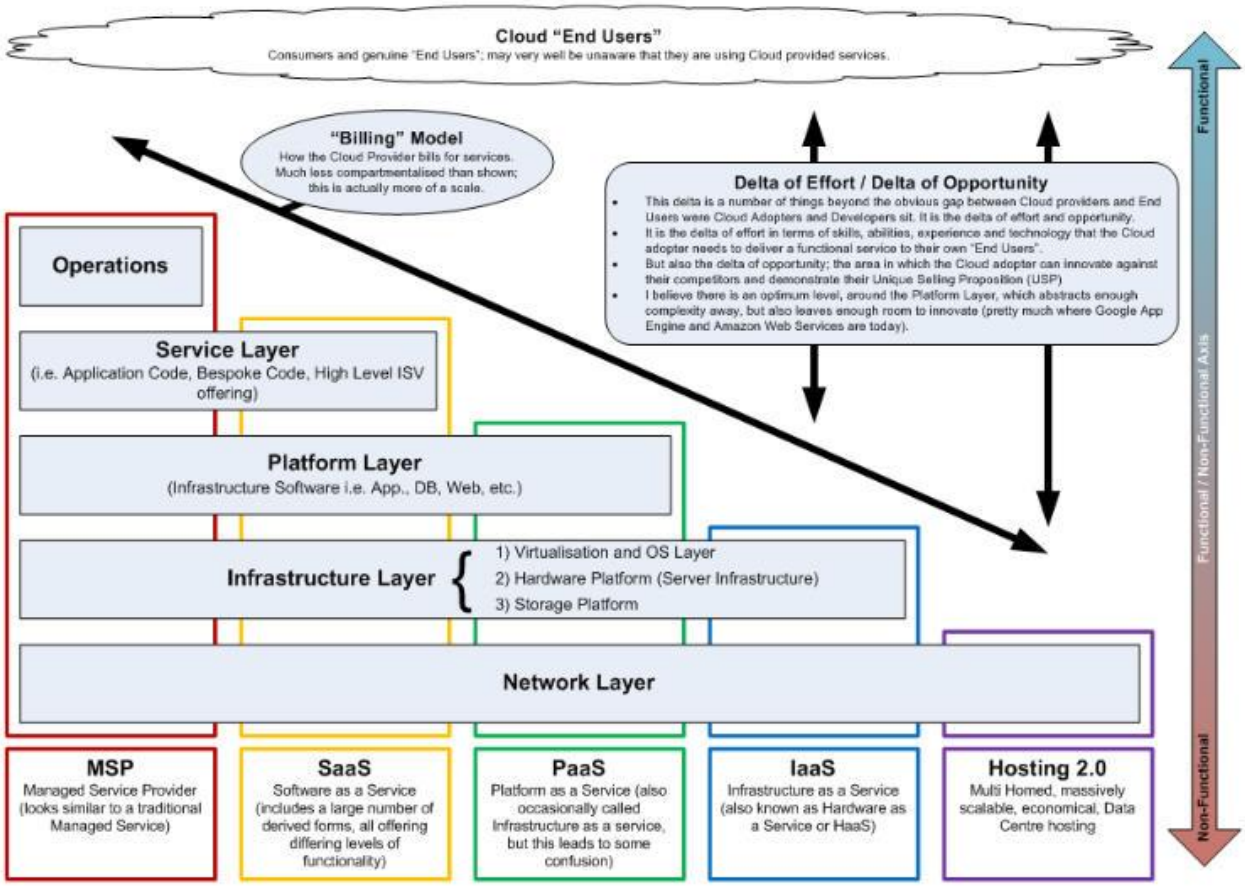

Figure2.3: Cloud Computing Architecture.

The whole architecture can be understood by studying separately each layer in the architecture. These layers are confined to work for specific tasks. The architecture covers cloud deployment models and all the service modes. These are discussed in detail in below sections.

2.3 CLOUD SERVICE MODELS

Once a cloud is developed and ready to be used by its consumers, it has to be decided that how to use the services offered by the cloud. The most common way in which it can be used is by categorizing all the services according to the commonly used business model. In cloud computing there are basically three delivery mechanisms through which a service can be delivered to the cloud users: software, platform and infrastructure. There are number of services offered by the cloud, but the primary service models which are deployed over the cloud are discussed below [5]:

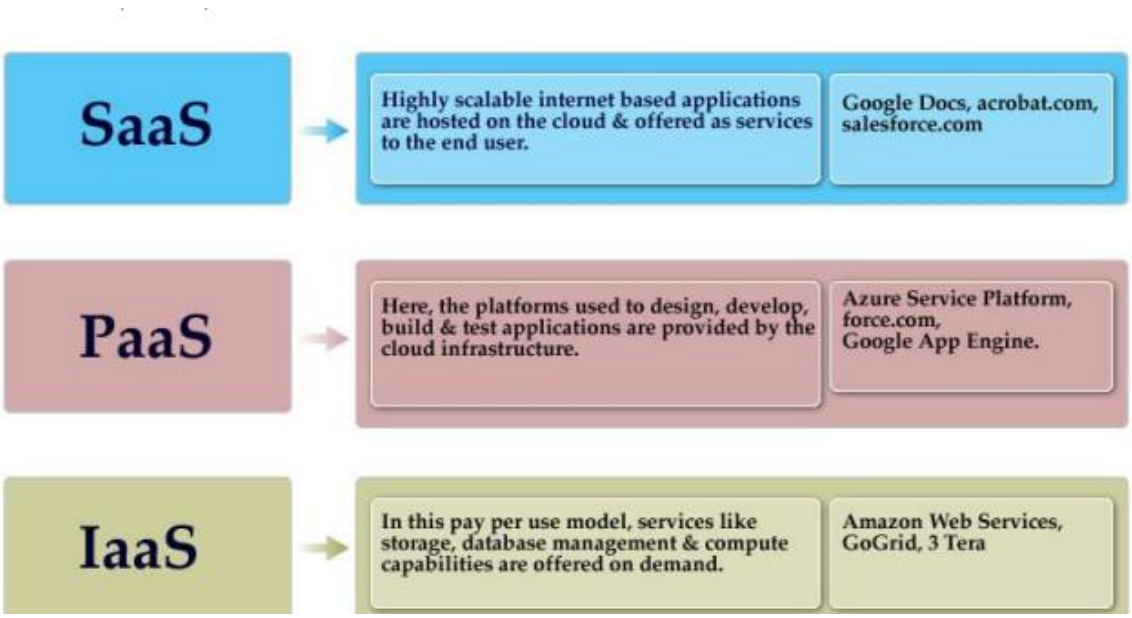

Figure2.4: Cloud Models

2.3.1 SOFTWARE AS A SERVICE

Software-as-a-Service (SaaS) is a software delivery model. In this, applications can be accessed through the Internet like a Web-based service. This delivery model facilitates the cloud users with wide variety of applications, for example social networking applications and customer relationship management applications. Users can make use of these applications by registering on the cloud service provider's website and then simply passing on the tasks to cloud service provider for completion. In this model users should not worry about the hardware and software configurations, it is the responsibility of the third party, whom they are registered with. In this scenario, customers neither need install anything on their premises nor have to pay considerable up-front costs to purchase the software and the required licenses. They simply access the application website, enter their credentials and billing details, and can instantly use the application, which, in most of the cases, can be further customized for their needs. On the provider side, the specific details and features of each customer's application are maintained in the infrastructure and made available on demand. Examples of some popular SaaS applications are: Facebook, Google Docs, NetSuit and Microsoft online.

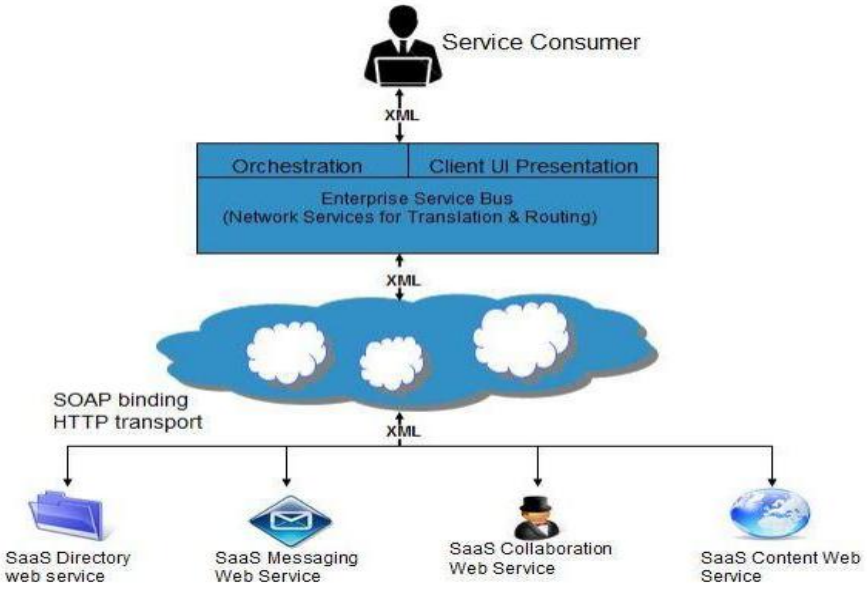

Figure 2.5: Software As A Service

2.3.2 PLATFORM AS A SERVICE

Platform as a Service (PaaS) provides facility to users to develop their own application using programming languages, libraries, services, and tools. Therefore, it can be said that the environment is integrated with various components in order to give users more number of offerings. Some PaaS providers provide a generalized development environment, while some only provides hosting-level services such as on-demand scalability and security. Management of applications and the underlying infrastructure configured by the users is done by the cloud service provider in PaaS whereas in SaaS users have to manage their applications by themselves. In PaaS, users have full control over the deployed applications and their hosting environment configuration. PaaS constitute the middleware above which applications are built by the users. Some popular examples of PaaS are Windows Azure, Engine Yard and Google App Engine.

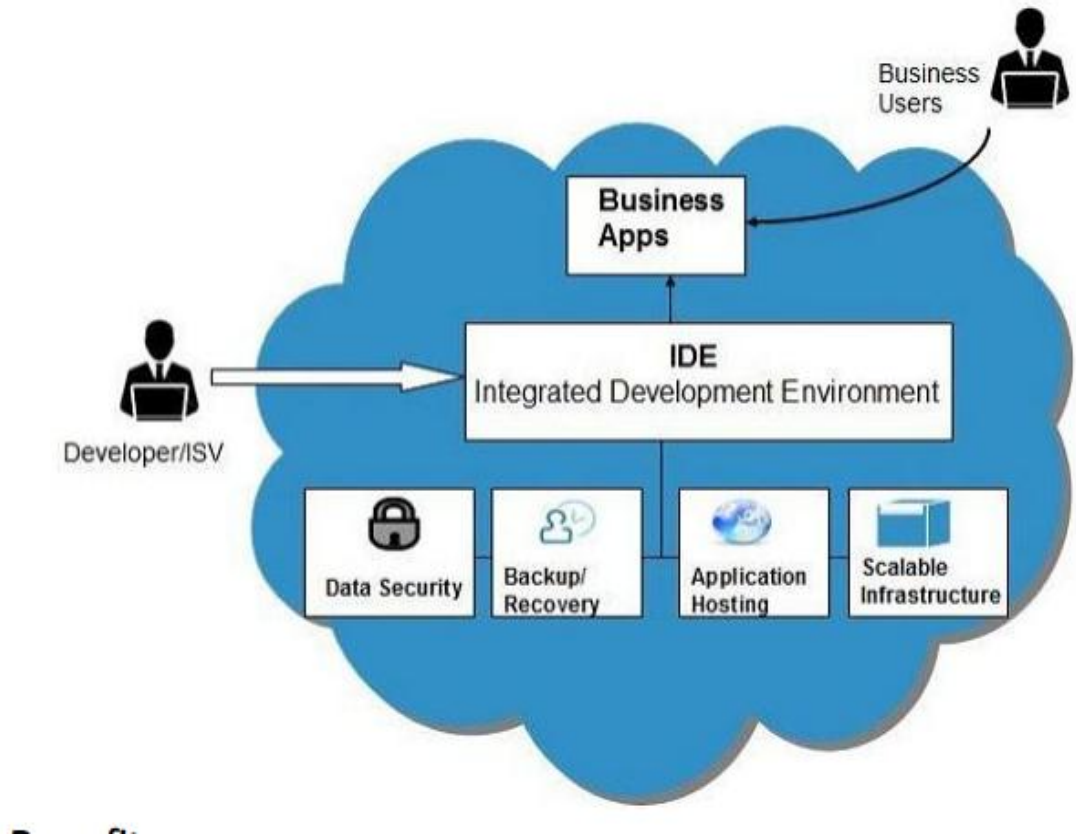

Figure2.6: Platform As A Service

2.3.3 INFRASTRUCTURE AS A SERVICE

Infrastructure- and Hardware-as-a-Service (IaaS/HaaS) offerings facilitates the user to customize the infrastructure as per their requirements. Users may demand from a single server to entire infrastructure, network devices, load balancers, databases and web servers. These services are most popular in the cloud market as these services prove to be a good option for the organizations who wants their own infrastructure with less maintenance cost. IaaS can service users with its capabilities of processing huge amount of data over a huge network, and other scalable computing resources. A user can deploy and run any software for e.g., any operating system like Linux, Solaris and other software include Matlab, Eclipse, etc. A user need not to worry about managing the cloud infrastructure, but has to manage the storage, operating systems, and deployed applications. Some popular examples of IaaS include Amazon, GoGrid and 3 Tera.

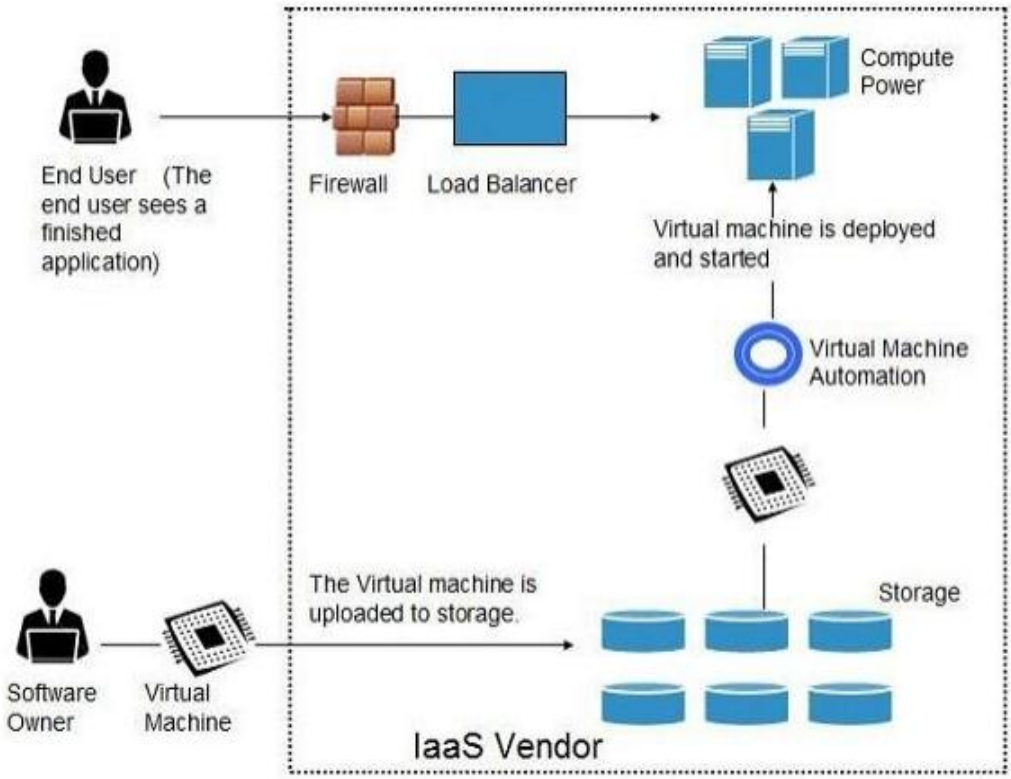

Figure2.7 Infrastructure As A Service

2.4 CLOUD DEPLOYMENT MODELS

Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid and Community. There are number of cloud consumers who want infrastructure of different sizes and each of these infrastructures requires different kind of management. Also different user groups want different types of infrastructures based on the nature and services offered by the cloud.

Clouds constitute the primary outcome of cloud computing. They are a type of parallel and distributed system harnessing physical and virtual computers presented as a unified computing resource. Cloud can be classified according to the administrative domain of the cloud. It identifies the boundaries within which cloud computing services are implemented, provides hints on the underlying infrastructure adopted to support such services, and qualifies them. There are four most important cloud deployment models.

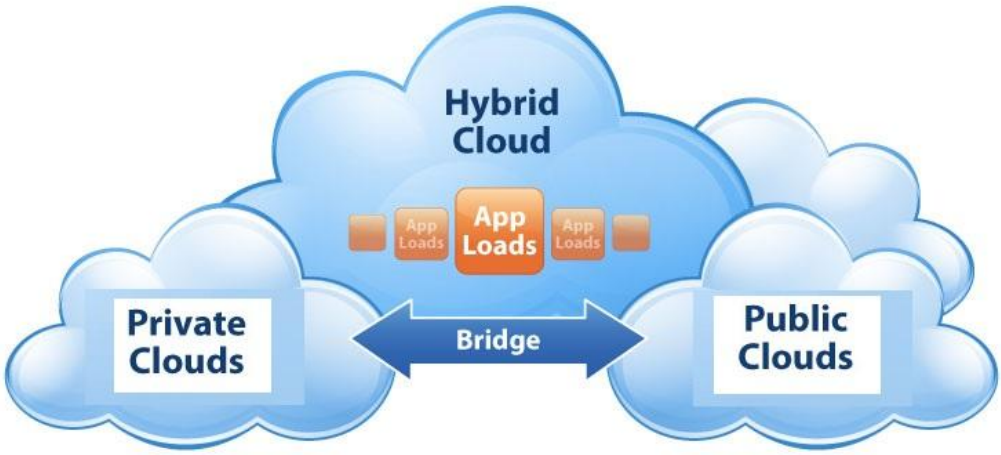

Figure2.8: Cloud Deployment Models

2.4.1 PUBLIC CLOUD

The Public Cloud allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness, e.g., e-mail. Public cloud as the name suggests are developed to serve the requirements of general public. A cloud service provider serves the service requirement like if any storage is required by the user, or if a user wants to make use of certain applications on cloud platform through internet. A public cloud may be managed, operated and owned by a government, academic, or business organization. Examples of public clouds are Amazon's Elastic Compute Cloud (EC2), Google's AppEngine, IBM's Blue Cloud, Sun Cloud and Windows Azure Services Platform.

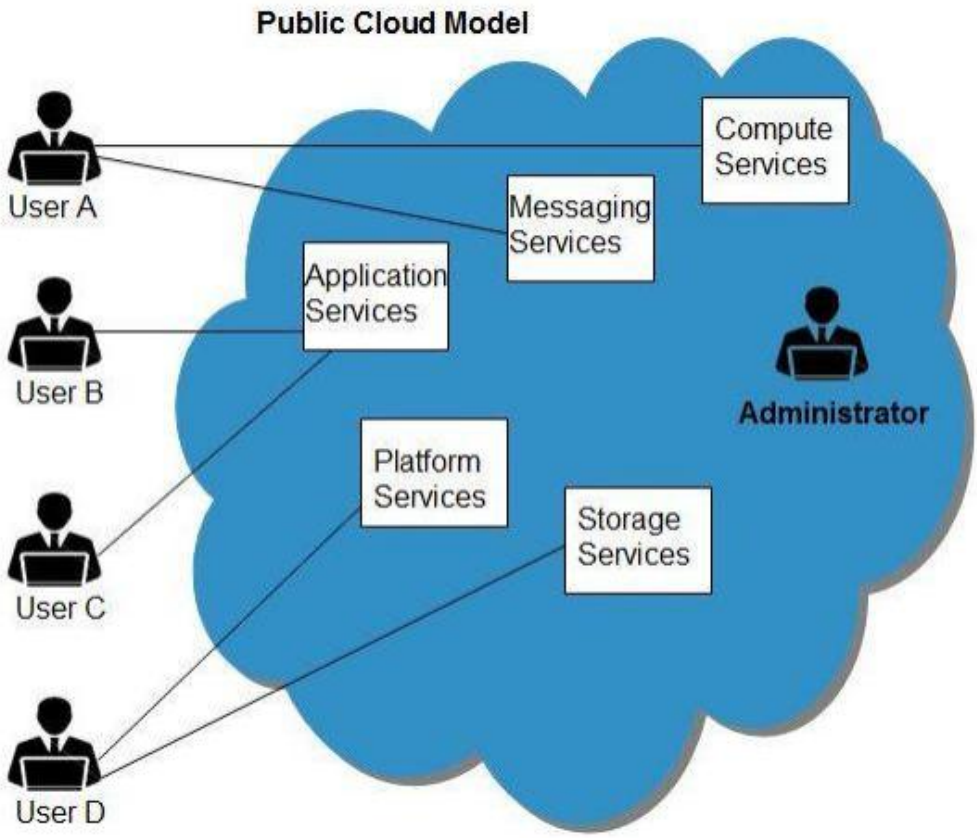

Figure2.9: Public Cloud

2.4.2 PRIVATE CLOUD

The Private Cloud allows systems and services to be accessible within an organization. It offers increased security because of its private nature. In Private cloud, the cloud infrastructure is provisioned for private use. An organization can request to a third party or it may develop its own private cloud infrastructure. Big organizations like Google, Microsoft have their own cloud infrastructure, which is supposed to be more secure than the public clouds. An organization can serve many other users of it by its own private cloud for e.g., Google is serving its users by providing them services like Gmail, Drive, GoogleApp engine and many more. Depending upon the deployment of the cloud, there are two types of private cloud:

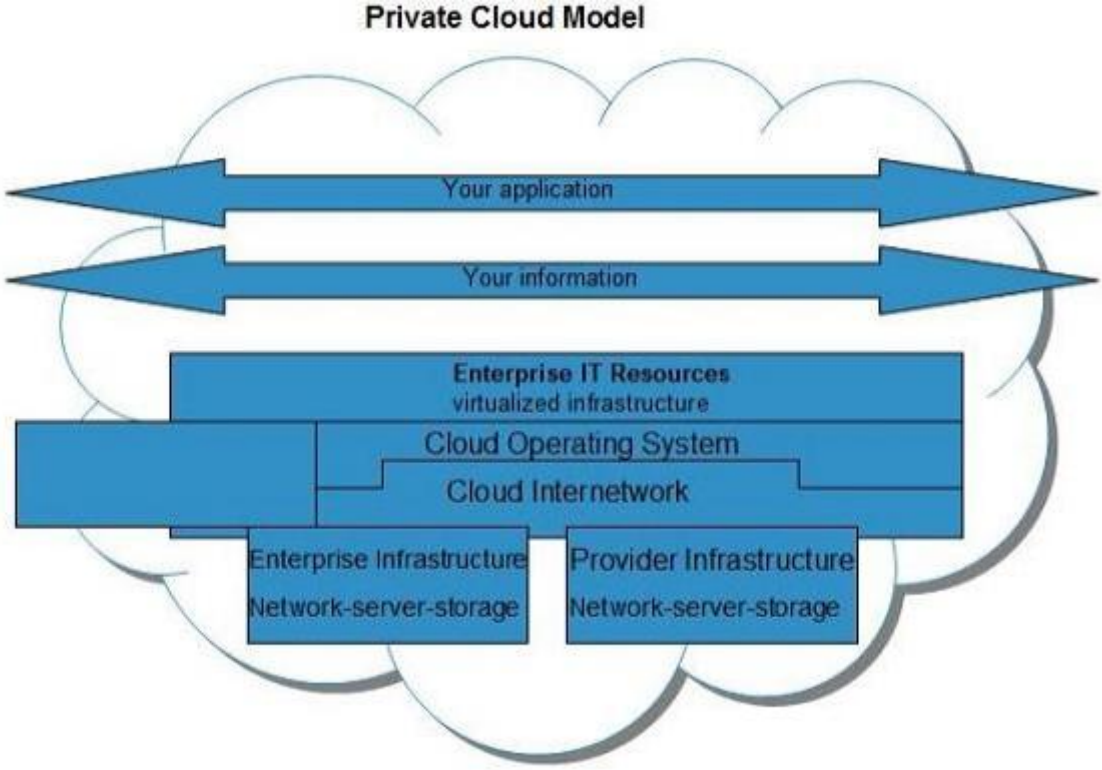

Figure 2.10: Private Cloud

- **ON-PREMISE PRIVATE CLOUD**

As the name suggests on-premise private clouds are hosted privately within its own datacenter. Therefore, these are also known as internal cloud. Benefits of this type of model includes security and standardization of processes, but still some issues related to size and scalability restricts a person from choosing this kind of model.

- **EXTERNALLY HOSTED PRIVATE CLOUD**

It is also known as external cloud. This type of model provides a special cloud environment with full privacy, as these are hosted externally with a cloud provider. Enterprises who don't want to share their physical resources in public cloud can be benefitted by using this model.

HP CloudStart and eBay are two popular providers of private cloud deployments.

2.4.3 HYBRID CLOUD

The Hybrid Cloud is mixture of public and private cloud. However, the critical activities are performed using private cloud while the non-critical activities are performed using public cloud. Hybrid cloud as the name suggests is a combination of two or more different types of cloud. It may be a combination of a private cloud, a public cloud or a community cloud. Cloud infrastructures of these clouds are entirely different. A cloud service provider needs to manage between the clouds to provide a required service to its users.

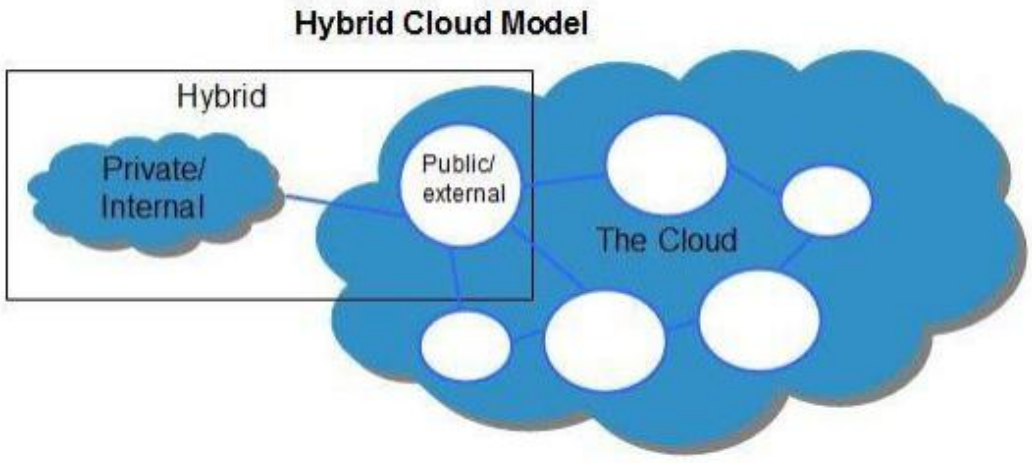

Figure2.11: Hybrid Cloud Model

2.4.4 COMMUNITY CLOUD

The Community Cloud allows systems and services to be accessible by group of organizations.

The concept of community cloud is similar to grid computing. There are certain specific communities of users from different organizations. As a day to day example let us consider a private firm in which there are two main business domain which works separately from one another, in such cases a firm can create two community clouds for the separate working of the two domains. In such cases the mission, security requirements and the other policies for the two domains are entirely different. This type of cloud can be owned by firm itself.

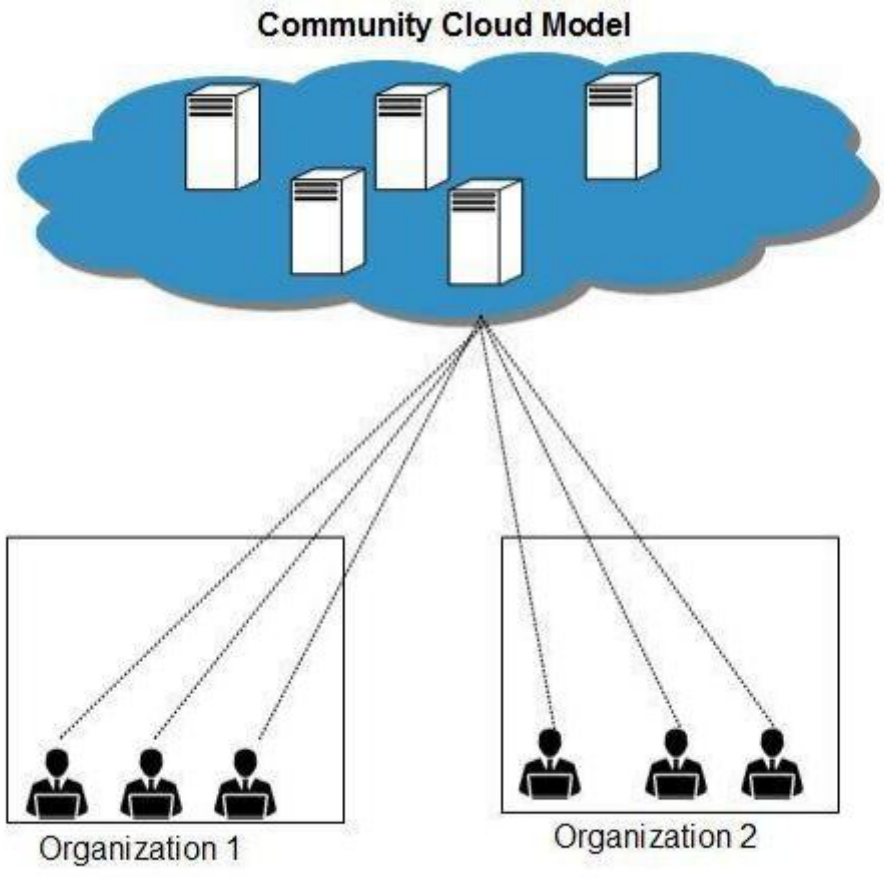

Figure2.12: Community Cloud

2.5 CLOUD COMPUTING VIRTUALIZATION

Virtualization is a technique, which allows sharing single physical instance of an application or resource among multiple organizations or tenants (customers). It does so by assigning a logical name to a physical resource and providing a pointer to that physical resource when demanded. Creating a virtual machine over existing operating system and hardware is referred as Hardware Virtualization. Virtual Machines provide an environment that is logically separated from the underlying hardware. The machine on which the virtual machine is created is known as host machine and virtual machine is referred as a guest machine. This virtual machine is managed by a software or firmware, which is known as hypervisor. Hypervisor is a firmware or low-level program that acts as a Virtual Machine Manager. There are two types of hypervisor.

Here are the three types of hardware virtualization:

2.5.1 Full Virtualization

In Full Virtualization, the underlying hardware is completely simulated. Guest software does not require any modification to run.

2.5.2 Emulation Virtualization

In Emulation, the virtual machine simulates the hardware and hence become independent of it. In this, the guest operating system does not require modification.

2.5.3 Para-Virtualization

In Para virtualization, the hardware is not simulated. The guest software runs their own isolated domains.

2.6 CLOUD COMPUTING OPERATIONS

Cloud Computing operation refers to delivering superior cloud service. Today, cloud computing operations have become very popular and widely employed by many of the organizations just because it allows performing all business operations over the Internet. These operations can be performed using a web application or mobile based applications. The operations are:

- Always employ right tools and resources to perform any function in the cloud. Things should be done at right time and at right cost.
- Selecting an appropriate resource is mandatory for operation management.
- The process should be standardized and automated to avoid repetitive tasks.
- Using efficient process will eliminate the waste and redundancy.
- One should maintain the quality of service to avoid re-work later.

2.7 CLOUD COMPUTING BENEFITS

Enterprises would need to align their applications, so as to exploit the architecture models that Cloud Computing offers. Some of the typical benefits are listed below:

- **Reduced Cost**

There are a number of reasons to attribute Cloud technology with lower costs. The billing model is pay as per usage; the infrastructure is not purchased thus lowering maintenance. Initial expense and recurring expenses are much lower than traditional computing.

- **Increased Storage**

With the massive Infrastructure that is offered by Cloud providers today, storage & maintenance of large volumes of data is a reality. Sudden workload spikes are also managed effectively & efficiently, since the cloud can scale dynamically.

- **Flexibility**

This is an extremely important characteristic. With enterprises having to adapt, even more rapidly, to changing business conditions, speed to deliver is critical. Cloud computing stresses on getting applications to market very quickly, by using the most appropriate building blocks necessary for deployment.

2.8 CLOUD COMPUTING CHALLENGES

Despite its growing influence, concerns regarding cloud computing still remain. In our opinion, the benefits outweigh the drawbacks and the model is worth exploring. Some common challenges are:

- **Data Protection**

Data Security is a crucial element that warrants scrutiny. Enterprises are reluctant to buy an assurance of business data security from vendors. They fear losing data to competition and the data confidentiality of consumers. In many instances, the actual storage location is not disclosed, adding onto the security concerns of enterprises. In the existing models, firewalls across data centers (owned by enterprises) protect this sensitive information. In the cloud model, Service providers are responsible for maintaining data security and enterprises would have to rely on them.

- **Data Recovery and Availability**

All business applications have Service level agreements that are stringently followed. Operational teams play a key role in management of service level agreements and runtime governance of applications. In production environments, operational teams support appropriate clustering and Fail over Data Replication System monitoring (Transactions monitoring, logs monitoring and others) Maintenance (Runtime Governance) Disaster recovery Capacity and performance management If, any of the above mentioned services is under-served by a cloud provider, the damage & impact could be severe.

- **Management Capabilities**

Despite there being multiple cloud providers, the management of platform and infrastructure is still in its infancy. Features like „Auto-scaling" for example, is a crucial requirement for many enterprises. There is huge potential to improve on the scalability and load balancing features provided today**.**

- **Regulatory and Compliance Restrictions**

In many countries, Government regulations do not allow customer's personal information and other sensitive information to be physically located outside the state or country. In order to meet such requirements, cloud providers need to setup a data center or a storage site exclusively within the country to comply with regulations. Having such an infrastructure may not always be feasible and is a big challenge for cloud providers.

CHAPTER 3

Load Balancing

Cloud computing naming origin is unclear. It is a powerful emerging standard which is close to turn utility computing into reality. In computing cloud is used to describe a set of resources whose details are not known by end user in a given context. Utility computing means "Pay and Use", in context of computing resources. Cloud computing relies on shared network resources rather than using local resources to store or process huge data without burden of backend implementations or hardware configurations. Load balancing is critical for cloud provider to improve performance and distribute load on all network nodes effectively to maximize throughput per network processing cycle. Outline of load balancing which is the theme of this thesis will be explained in this chapter.

3.1 INTRODUCTION

As the user load increases on cloud, the existing processing performance degrades automatically which leads to delay in serving users. There might be a case in which some nodes are overloaded with requests and some are under loaded at the same time. We need to implement load balancer to improve our load sharing between network nodes which use load balancer algorithms to improve load balancing among network nodes. A wide range of load balancing algorithms are defined to implement load balancers. Implementation of correct load balancer was important for network performance only in older distributed systems. Now load balancing includes low power consumption with respect to processing as network nodes are usually power hungry blade servers. Energy consumption is crucial in any cloud role out and load sharing helps in cost reduction. Usually cloud implementer defines a specific policy for load balancing to improve system performance and it is a field of extensive research.

3.2 LOAD BALANCING DEFINED

A network or cloud consists of various servers, network devices and storage units. Workload on resources increased abruptly as number of users' increase and resource management become difficult in the network which leads to poor performance and increased resource consumption.

System throughput and resource utilization can be improved by using load balancer which can improve system response time and performance effectively. A good load balancer's job is to equally distribute workload among all nodes present in the cloud. Ineffective utilization can be explained with example of social networking providers which has hundred millions of users or search engine provider which handles millions of queries per minutes and it is difficult to serve so many requests from a centralized system. It can lead to bad throughput or delay increasing response time. It can solved by using geographically distributed systems, which leads to a scenario where a system in a particular geographical system is overloaded with requests at a particular time for example a new product launch or multiple request for viral video but other adjacent systems are underutilized . A good load balancer can forward request to underutilized nodes which can normalized system response without increasing processing capabilities of system. Various schemes or mechanisms are used by cloud developers to tackle low throughput and data replication for fault tolerance.

3.3 GOALS OF LOAD BALANCING

Various IT and Engineering organizations have own networking systems and use customized policies for load balancing in their network. They use global policy for cloud requirement and specify technical goals for load balancers which compliment business goals.

3.3.1 TECHNICAL GOALS

All technical specs, problems and targets are covered in technical requirements and address all problems and functionalities regarding network.

Important technical requirement are listed below:

- Performance improvement of system
- Redundancy in case of system failure
- Ensure stable throughput and normalize work load among all nodes

- Scalable for future modifications, easy interfacing to with other systems and extensible
- Maximum service availability
- Response time to serve content or response to user

3.3.2 BUSINESS GOALS

Technical specs are driven by business logic of an organization and have business polices dedicated to improve user experience and minimize cost which can make it a market leader in its niche and move faster or develop more features and provide different solutions to clients or end user than its competitors to dominate market or capture more market share.

- Minimize operational cost.
- Conform to industry standard and compliances and acquire certain patents or certificates from responsible agencies.
- Prepare service level agreements and licensing agreements for clients.
- Check all legal issues and government policies.
- Business model for every geographical and political situation with other parameters like maximum usage hours of a day.

3.4 LOAD BALANCING IN CLOUD COMPUTING ENVIRONMENT

Reponses Time is the measurement for a cloud performance and load balancer should minimize it without underutilization of resources and increasing operational cost. Enhancing resource management needs an effective load balancing technique with a task scheduler with optimized mechanism to avoid any kind of resource blocking in the system. Cloud developer, service provider and user are main concerned entities in any cloud system.

Cloud developer's Key responsibility is to design and deploy system on provider's end. Provider is responsible for providing services to user and user is any business group or any other entity which is served by cloud provider and service level agreement is signed by both cloud provider and user which specify rules and minimum requirements fulfilled by cloud provider.

A cloud network consists of various nodes which consist of processing units and storage units. Every node has many virtual machines to serve requests and a task is assigned to any of the virtual machine or resource. Resource is released after completing a task and can be allocated again to any other request according to some allocation policy. CloudSim can effectively simulate load balancing or scheduling in cloud environment.

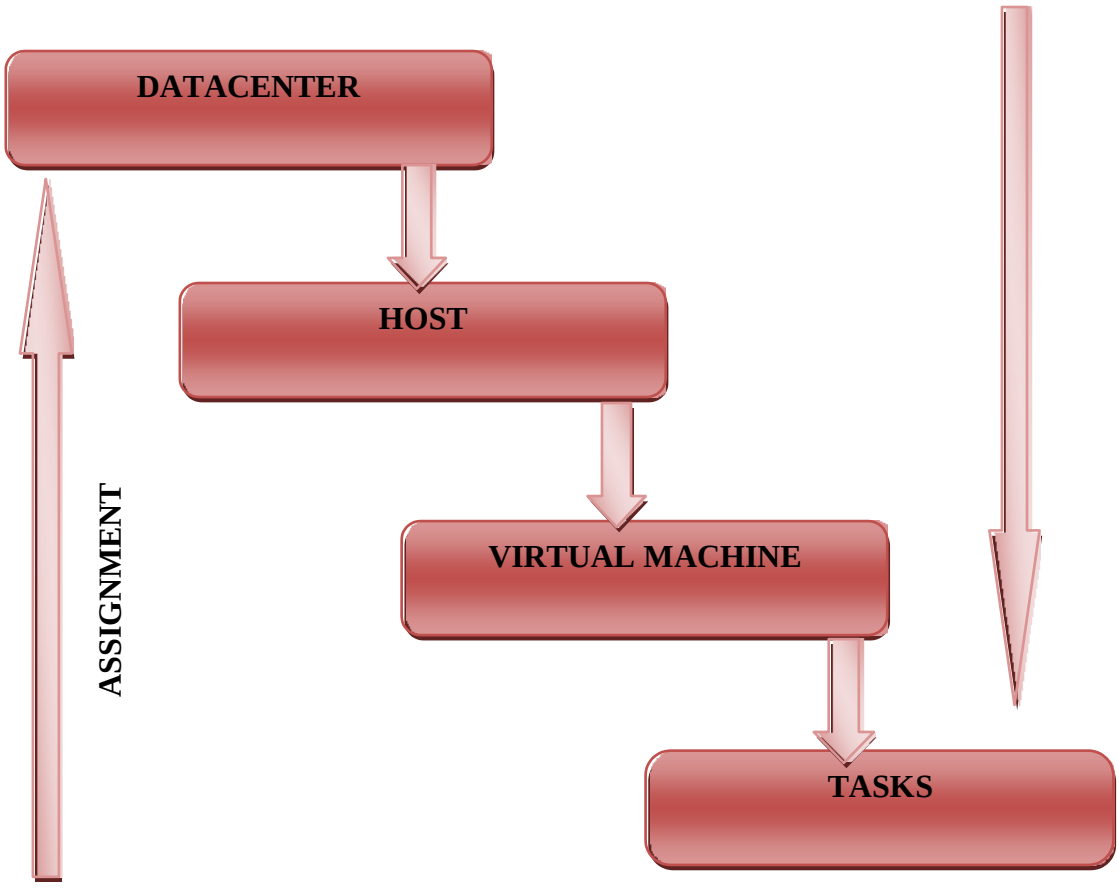

Figure 3.1: Hierarchy and assignment of cloud components

Task scheduling and resource provisioning are main mechanism which is responsible for proper load balancing in cloud. Resource provisioning defines resource mapping with tasks or any other network entity. Resource provisioning can be enabled on two different levels:

- Host level
- Virtual Machine

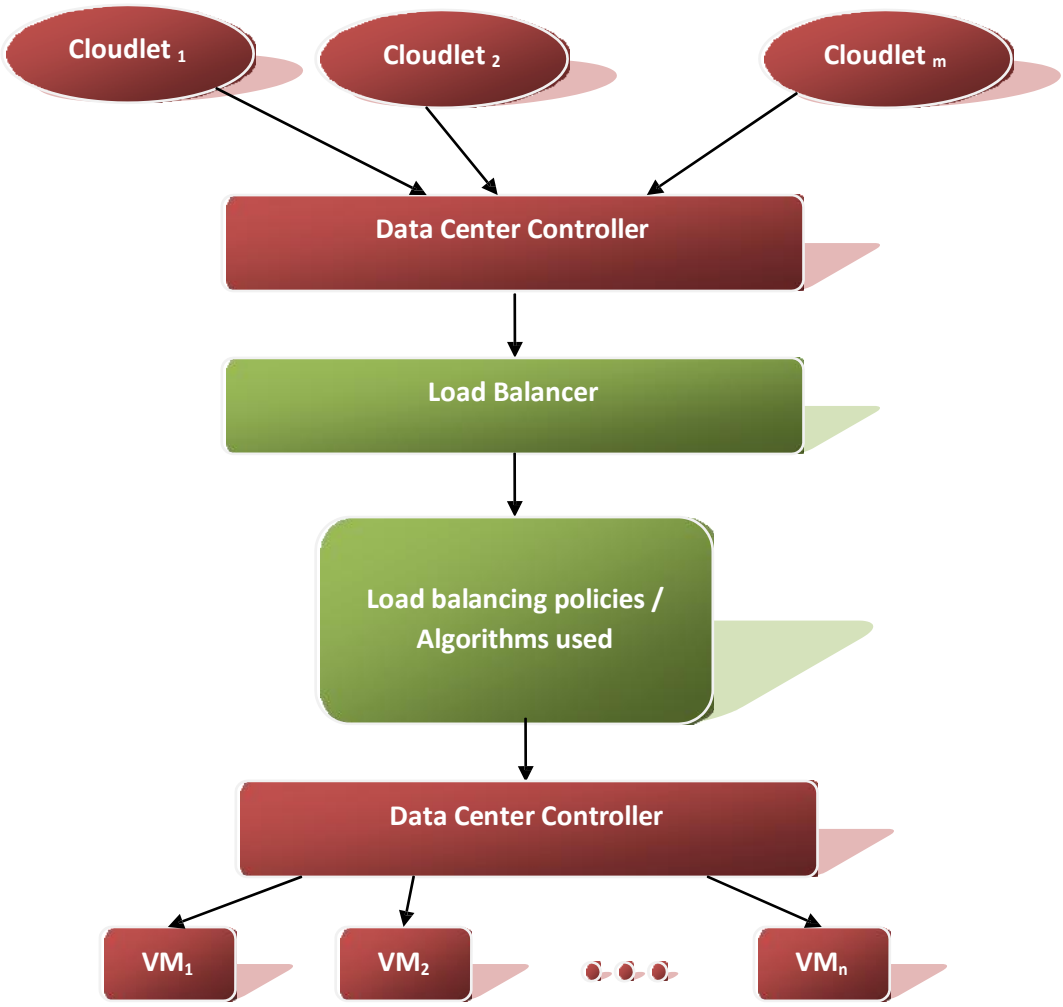

Figure 3.2: Execution of load balancing algorithm

3.4.1 AT HOST LEVEL: ALLOCATING THE HOST TO VM

There is many to one relation between physical server and virtual machines as many VM's can be hosted by a single physical server more than one virtual machine can be allocated to user as per allocation policy defined for that particular user.

3.4.2 AT VM LEVEL: ALLOCATING VM TO USER TASKS

User tasks are mapped to virtual machines .VM's and user tasks has many to many relation means a user task can be handled by various virtual machines or various tasks can be handled by a single instance of virtual machine .Virtual Machines can be mapped to users dynamically or statically to user which is also defined by allocation policy.

Various load balancing algorithms are executed as per the provisioning policy in the cloud environment. The user tasks are provisioned for virtual machines by the virtual machine manager.

3.5 TYPES OF LOAD BALANCING ALGORITHMS

Various load balancing algorithms has been developed and are in use since area of distributed computing. Load balancing is based on many factors like system configure and system topology and old techniques still can be used in cloud computing for improved resource utilization.

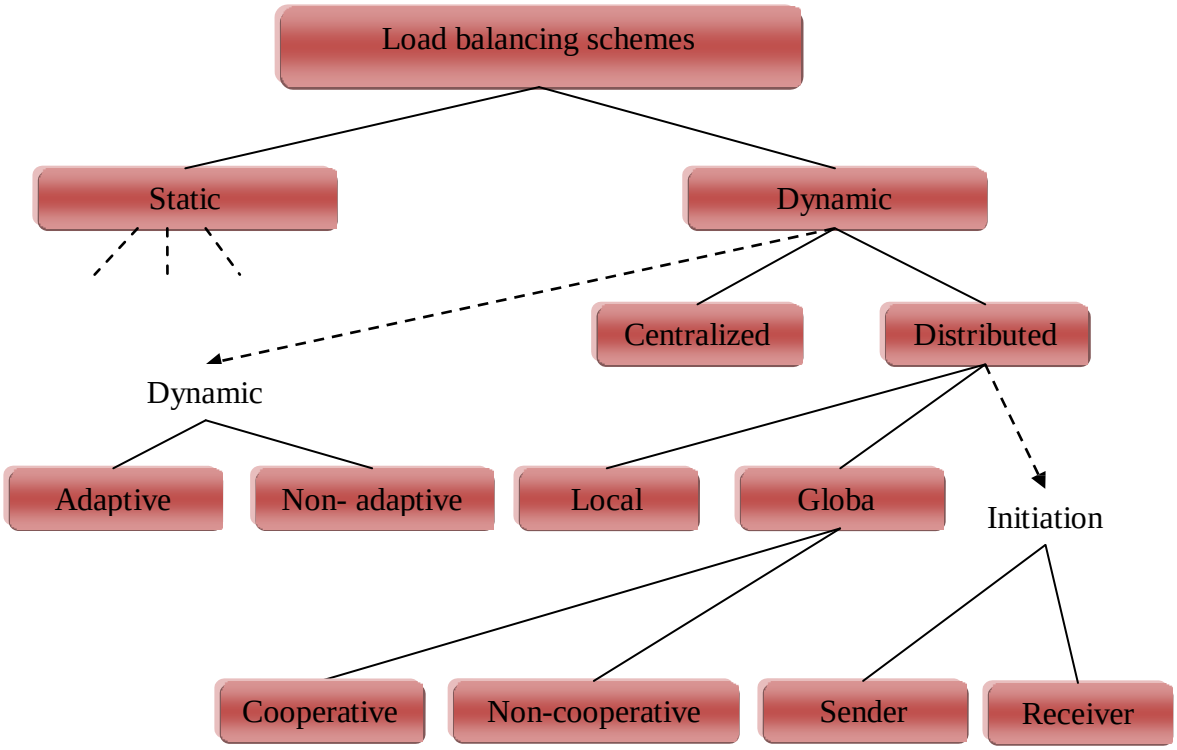

Figure 3.3: Load balancing schemes

Broadly all load balancing techniques can be classified into two categories

- Static Load Balancing
- Dynamic load Balancing

3.5.1 STATIC LOAD BALANCING TECHNIQUES

Static techniques are easy to define and predefined resource mapping with task. It is feasible for small and fixed request handling network .Static or deterministic techniques make a system less scalable and cannot handle changes in nature of tasks .It requires more and manual effort in mapping more users to system resources.

- **Round Robin**

  It handles clients request in a circular manner and every task get lock on virtual machine till it get completed and serves user tasks on FIFO basis.

- **CLBDM(Central Load Balancing Decision Model)**

  It is an improvement over round robin and also finds out time duration of session established between client and node. Time Duration can be defined as execution time taken by task till completion on given virtual machine or resource.

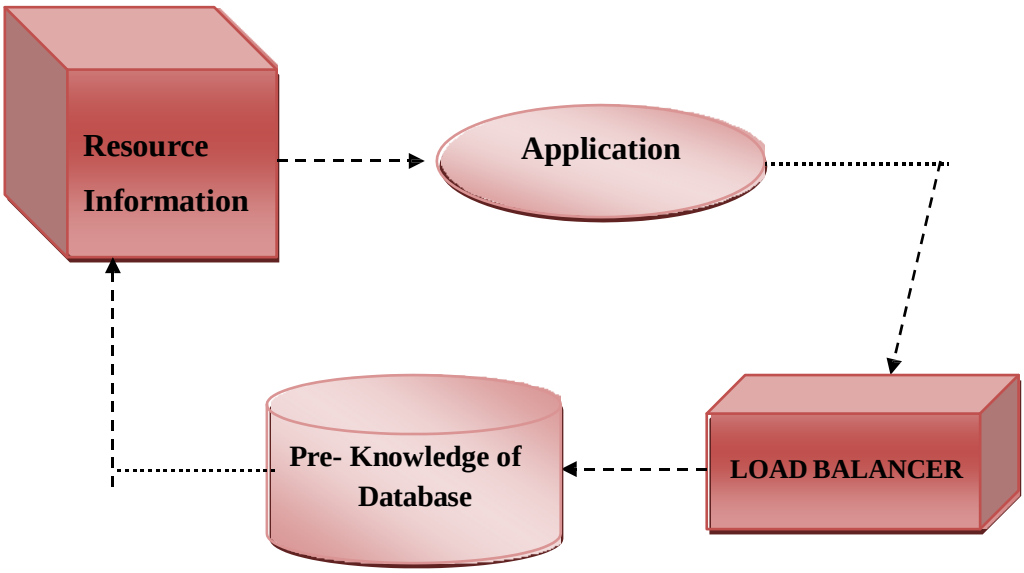

Figure.3.4: Static Load Balancer

3.5.2 DYNAMIC LOAD BALANCING TECHNIQUES

Dynamic techniques can also be classified in centralized or distributed topology. All decisions for system resource provisioning or scheduling is controlled by a master and assigned to a slave means slaves merely serves request forwarded by master to them and cannot take their own decisions whereas in distributed system control is disturbed in network and more than one node can handle provisioning or scheduling of resources.

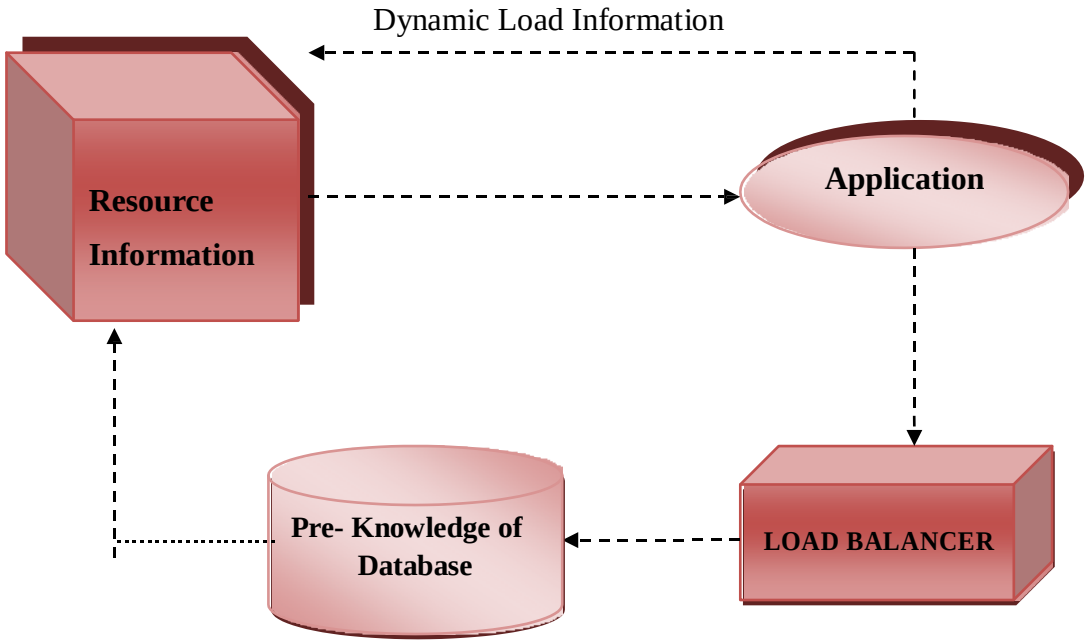

Figure.3.5: Dynamic Load Balancer

Commonly used dynamic policies are:

- **ESCE Policy**

  It equally spreads execution load to all available resources and maintain record of all resources and number of task assigned to every resource or virtual machine when any new task is received to load balancer it searches for most underutilized resource or virtual machine.

- **Throttled policy**

  Under this policy, each resource is assigned with only one task at a time and any other job can be assigned only after completion of previous task assigned to it. Load balancer or job manager maintains all resource lists.

When a new user task is received job manager check list and if any free resource is not found, it keeps task in waiting queue and poll on list to check virtual machine status.

3.6 LOAD BALANCING ISSUES

There are many open issues while taking care of load balancing in a cloud environment. Every load balancing has its own strength and weakness for example some algorithms are suited for real time systems and stress on achieving minimum RTT, or any other algorithms is designed to achieve max resource utilization or max throughput.

Some major challenges which are considered while designing a load balancing algorithm are:

3.6.1 NODES GEOGRAPHICAL DISTRIBUTION

Overall system performance is drastically effected by geographical location of nodes as it reduces number of network hops in serving any client and very significant for enterprise with large user base for example all social networking or search providers. A well geographically distributes system results in more fault tolerable and efficient system with significantly reduced response time as compared to no distributed system.

3.6.2 ALGORITHM COMPLEXITY

Load balancing algorithm complexity can also affect system performance. For example complex algorithm can increase throughput and resource utilization of a system but can increase response time due to more processing whereas a simple algorithm can reduce response time but gives poor results in fault tolerance. So an algorithm should be decided bases on system requirements.

3.6.3 STATIC VS DYNAMIC LOAD BALANCING

Load balancing algorithms are designed or chosen according to system design which leads to choose between static or dynamic implementation .Static policies are simple ,less complex and easy to implement defined earlier and hard coded in system and can lead to sudden failure of the system over an un handled change in system . Dynamic algorithms are improvement over static algorithms nut are more complex and system driven in nature. Dynamic algorithms can handle changes in system or failure of any node make system more robust and fault tolerable leading to more stabilize system operation and throughput.

3.6.4 TRAFFIC ALAYSIS OVER DIFFERENT GEOGRAPHICAL LOCATIONS

All business regions across the globe have their own time zone, own working hours and peak hours in any resource utilization, therefore load balancer should be able to handle peak hour traffic using resources of that particular location and in case of sudden burst of request, It must be able to forward it any other underutilized geographical location system maintaining throughput and handling all request effectively.

3.6.5 REPLICATION AND STORAGE IN CLOUD

Data is the most important for every organization and it should be handled carefully and replicated to handler failure of main storage unit. Geographical replication of data is used to minimize response time to serve any task.

3.7 PERFORMACE METRICS FOR LBA EVALUATION

Load balancing algorithms are measured by following metrics:

3.7.1 THROUGHPUT

Throughput of an algorithm is used to estimate number of tasks that has been executed successfully.

3.7.2 OVERHEAD

It can be defined as extra cost associated with implementation of any algorithm for example system maintenance or migration time. System Implementer should try to keep it as low as possible.

3.7.3 FAULT TOLERANCE

It measures load balancer's capability of uniform load sharing among nodes in case of any sudden node failure. Node failure in peak hours can affect whole system performance. Hence Load Balancer should be fault tolerable.

3.7.4 MIGRATION TIME

It is the time taken by load balancer to shift any job or task from one node to other node usually in case of node failure.

3.7.5 RESPONSE TIME

Time taken by system in between a task is submitted and response generated. Load balancer should try to keep it is minimum.

3.7.6 RESOURCE UTILIZATION

Uniform load sharing and optimal resource utilization. A load balancer should avoid any node from over loading as well as underutilization, avoid wastes processing cycles.

3.7.7 SCALABILITY

Load balancer should be able to incorporate changes in system infrastructure. Algorithm should be able to handle changes in node or system topology without any or minimal manual efforts.

3.8 SUMMARY

In this chapter, we studied various aspects, features and requirements of load balancing from the point its utilization in cloud network. Various technical and business requirements has been covered in this chapter which effects load balancing algorithm design decisions in cloud network implementation. Widely used load balancing policies and techniques categories were also discussed in this chapter taking in account all pro and cons of every category. It covered all current implementation standards and parameters defined to check cloud network performance.

CHAPTER 4

Related Work

Cloud computing is the current advancement in the technology field concerned to IT industry. This chapter deals with related contributions in the area of load balancing related to cloud computing.

In [16] A.Khiyaita provided a review of load balancing techniques. Classification of these algorithms is done on the basis of system topology and load. Multiple examples in the concerned paper elaborated their implementations in classical systems. Key technologies specific to cloud computing was specified. Authors also mentioned challenges faced during the implementation phase. Nidhi [17] also discussed techniques aiming to reduce the overhead, response time and the performance. It also analyzed parameters governing performance specific to techniques. Shu-Ching [18] combined the functionalities of LBMM i.e., Load Balance Min-Min and OLB i.e. Opportunistic Load Balancing in order to propose scheduling algorithm and then it gave an improvement to existing algorithm Min- Min algorithm. OLB kept every node busy not concerning the workload in particular node. OLB assigned tasks in random manner to all the available nodes whereas MCT algorithm assigned tasks to only that node which expected minimum completion time to other nodes.

Another very important and effective algorithm came into existence by Che-Lun Hung, Yu-Chen Hu and Hsiao-his Wang in [19] which was LB3M. [20] Gave the concept to get any resource of cloud by incorporating Co-operative Scheduling of power aware. It was considered as a good replacement to the challenges concerned to load balancing keeping into concern energy efficiency. It incorporated both centralized as well as distributed approach making the best use of inherent efficiency related to centralized one and energy efficiency as well as fault tolerant behavior of distributed method. In [21], percentage of node utilization in PALB is calculated and this percent decided the count of computed nodes that are kept operated when another one gets shut down completely. The technique consists of three sub areas: Balancing area gave the method to instantiate the virtual machines based on utilization percentage. Another area which is Upscale gave the method to power on the nodes. Area of Downscale was used to shut down the nodes that are idle. It gave the promise to decrease the consumption of power and also managing the resource availability in parallel.

Raul Calvo [22] proposed method to manage image collection of large amount in real world techniques. This paper creates a service and provides its use for analysis of images. Various operations on data are stated to work in distributed area for different sub-images. Now these sub-images are stored and their processing is done separately by multiple agents. It deals with the execution of method in large images in parallel way. Resource scaling and consuming power are the factors to be dealt with any load balancing method. [23] Improvements that can be measured are obtained in cloud environment Load balancing techniques should be such that to obtain measurable improvements in resource utilization and availability of cloud computing environment [23].

In [24] Alexandru Iosup studied the services of cloud and analyzed for scientific computing. They experimented on workloads of real scientific world of MTC requests of user. MTC stands for many task computing. They deal with applications that are loosely coupled which complete tasks as per the scientific goals. In [25] Srinivas proposed the algorithm for load balancing adding fuzzy logic to its computation. It made the use of speed of processor and incorporated assigned load onto overall load in cloud environment as in [26]. Here authors proposed a new logic to dynamic balancing of load in cloud environment with parameters- disk space, status of virtual machine and then stated it as FAMLB i.e. Fuzzy Active Monitoring Load Balancer. Milan E Sokile focussed his work in elaboration of load balancing techniques. They include static, shortest queue, round robin methods in client environment. The result of experimental analysis is that it states that in dynamic environment diffusive load balancing is better and efficient as compared to static load balancing. [28] Gave a difference between strategies that exists among various methods and algorithms. The efficiency in load balancing leads to benefits in cloud computing.

Parallel processing [29] can be seen in a network processor. It comprises of many on-chip processors which is used to perform operations of packet level. It assures high throughput by providing fine load balancing to the available processors. It may also have a disadvantage of high rate of out of order packers. ORR which stands for Ordered Round Robin is done for scheduling packets. The heterogeneous processor which give loads need to be handled accurately.

The processed loads from the processors are ordered perfectly. Analysis of the derived expressions and the throughput in terms of batch size, count odd processors scheduled and scheduling time. Jaspreet Kaur elaborated active virtual machine algorithm in order to get free and appropriate virtual machine in lesser time. She performed simulation to do a comparative study of round robin and to spread execution policies concerned to load balancing for lesser time and cost.

Algorithm in [31] added capacity related to dynamic balancing method of cloud. Experiments performed by Zhang Bo clearly demonstrated that it achieved better degree of load balancing and took lesser time in loading tasks. Soumya Ray [32] explored many algorithms like central queuing algorithm and many more. The analysis was carried between MIPS vs. HOST and MIPS vs. VM. The experiment demonstrated that the response time can be improved in terms of number of VMs in Datacenter. In order to handle the random selection based load distribution problem, dynamic load balancing algorithm can be implemented as the future course of work to evaluate various parameters. In [33], the authors have proposed an algorithm for load distribution of workloads among nodes of a cloud, by the use of Ant Colony Optimization (ACO). This algorithm uses the concept of ant colony optimization.

Shridhar G.Domanal and G.Ram Mohana Reddy [34] have proposed a local optimized load balancing approach for distributing incoming job request uniformly among the servers or virtual machines. Performance of proposed algorithm is analyzed using CloudAnalyst [14] simulator; further comparison is done with other existing Round Robin and Throttled algorithms. In [35], the authors have analyzed various policies utilized with different algorithm for load balancing using a tool called cloud analyst, mostly different variants of Round Robin algorithm for load balancing has been analyzed. Dynamic Round Robin algorithm is an improvement over static Round Robin algorithm [36], this paper analyzed, and the Dynamic Round Robin algorithm with varying parameters of host bandwidth, cloudlet length, VM image size and VM bandwidth. Results have been analyzed based upon the simulation carried in CloudSim simulator. In [37], the authors Ching-Chi Lin, Pangfeng Liu, Jan-Jan Wu, have proposed a new Dynamic Round Robin (DRR) algorithm for energy-aware virtual machine scheduling and consolidation.

CHAPTER 5

Proposed Methodology

5.1 INTRODUCTION

As earlier discussed, the selection of a right technique for LB for a system is one of the critical moves for developers. We are having a wide list of techniques suggested by researchers which are used for the load balancing. They are all having their respective pros and cons. There are several network oriented computing systems in which their respective applications need the computational process in parallel fashion with large quantity load while few applications only require short and quick response. For example, scientific research based activities require a very huge computational requirement for a setup while a particular simple user login job requires fast and easy evaluated processing over network. A particular range of LB techniques emphasize on effective resource utilization, some on having low execution time and few are emphasizing on setting a monitoring trade-off in between these parameters. There are two types of situations comes under LB either it would be high load or would be load below average. So after consideration of these parameters a customized LB is selected. With the respective proposed methodology our main aim is to balancing the total workload in the network architecture among different nodes either it would be server or a local host. Several metrics are discussed earlier in the chapter 3 to maintain equilibrium between theses parameters for achieving the better results. Before discussing the proposed LBA let's take a look on the steps which are needed to perform a suitable LBA:

- Starting from the very first step, user requests comes into the load balancer as an input and then load balancer checks that either load is high or low from that input.
- In next step, information is exchanged related to input parameters by load balancer among the different nodes in the datacenter.
- After that, information is exchanged related to total load by load balancer between different VMs which are deployed over the targeted nodes in DC.
- At the last step, if demand rises, load balancer focuses on the jobs of VMs migration as per LB technique.

An effective LBA can boost the performance of a cloud network. It provides the better resource utilization with increased throughput of the system. For better understanding the proposed LBA we have to cover few terminologies which are below mentioned in description.

5.2 DESCRIPTION

These are the basic recommended terminologies which are used in our proposed LBA.

- **Datacenter (DC): DC** is the main entity of the cloud. DC comprises a set of hosts either they are servers or local nodes. As per the budget and requirements of the particular organization which wants to establish a cloud infrastructure, they can also facilitate number of DCs.
- **Host:** A host is having the full server characteristics with their respective processing cores. They reside in the DC. These processing cores will further allocate to respective VMs. It depends upon the requirements that how many hosts are needed to deployed in the DC.
- **Virtual Machine (VM):** Virtual machine is the processing unit in the cloud environment. VMs are the instances which are propagated by user on the basis of his budget and requirements. These machines are allocated for the different tasks based on the configuration of cloud infrastructure.
- **Virtual Machine Capacity:** VMs are processing units for a particular cloud infrastructure. In technical terms the VMs are having the processing power to execute the task based on its processing capacity. VM capacity is commonly depends upon the certain factors. These factors are CPU utilization, memory, input-output capacity, number of cores in CPUs, CPU's clock speed, communication bandwidth between VMs and many more hardware related factors.
- **Datacenter controller (DCC):** It is a communication repository between the cloud users and the respective cloud load balancer. It is responsible for load assigning to load balancer and recipient for the requests from cloud users.
- **Total Load:** Basically the load is the total length of a task which is assigned to VM in terms of its execution time. The total load is the summation of all the requested task time which is ready to assign to VMs.

The proposed LBA is achieving the load distribution for the total load assignment from cloud users in the DC. This algorithm is focused on the load distribution at host level. VMs are responsible for load assignment in the particular infrastructure provided by the cloud service providers. VMs instances are created for the particular load request to balancing the desired need of load over the network by the help of DC.

5.3 MATHEMATICAL MODEL

Before describing the proposed LBA we have to go through some mathematical terms which will be used in our proposed LBA. As per the LBA's requirement we have following inputs:

1) A set of VMs < VM_1, VM_2, VM_3, ...VM_m>
2) A set of Hosts < H_1, H_2, H_3, ...H_h>
3) A set of Task time< T_1, T_2, T_3, ...T_n > which is associated to each Host.

We take the consideration that all the VMs are unrelated to each other in a parallel manner. All the tasks are Non-preemptive. Non-preemption means that when a task is in under execution phase no other task will interrupt it till its execution finished.

- **Capacity of VMs (CP_j):** To calculate the VM's capacity, we consider three factors in place of all the factors (which are discussed above) which affect the VM's capacity. The calculated VM's capacity for virtual machine j is

$$CP_j = PES_{numj} \times PES_{mipsj} + VM_{bwj}$$

Where CPj is VM capacity, PES is the processing elements, PESnumj is the total processors in VMj, PESmipsj is the MIPS (Million instructions per second) of all processors in VMj, and VMbwj is communicated bandwidth ability of VMj.

- **Total Load (TL):** Total load is the total amount of time required to do the all tasks by VMs.
- **Total Capacity(C):** Total capacity of datacenter is the summation of the capacity of all VMs.

$$\mathbf{C=\sum_{i=1}^{m} C_i}$$

For assignment of load to VMs we require an allocation policy for VMs. The proposed LBA with step-wise is described as:

Algorithm: Proposed LBA

Input:

1) VMs < VM_1, VM_2, VM_3… VM_m>
2) Hosts < H_1, H_2, H_3…… H_h >
3) Task (finishing time) < T_1,T_2,T_3 ….T_n > for each host.

Where 'm', 'h' and 'n' are total number of VMs, total number of Hosts and total number of Tasks associated with each Host for VMs allocation respectively.

Output: All Tasks < T_1, T_2, T_3 ….T_n > are occupied by VMs for their execution for each Host< H_1, H_2, H_3…… H_h >.

Step.1: Proposed LBr maintains:

- Sorted arrays of tasks with their finishing time associated with each Host,
- Index table of host id, PES column, Sorted task time array.
- A list of Processing Elements associated with Hosts.

Calculate the total VM capacity 'C' and the Total Load 'TL' which is to be assigned to VMs. For calculation of VM's capacity cloud service providers set the Threshold value (Cth) for VM capacity to each VM. Initially all the hosts are available.

Step.2: Request for LB is received by the Data Center Controller (**DCC**).

Step.3: Decision for LB.

If (**Total Load (TL) <= Total VM Capacity(C)**)
{
Load Balancing is **possible.**
Go to step 3.
}
Else
{
Load Balancing is **not possible.** Request overflow has occurred.
Go to Step 6.
}

Step.4: DCC trigger Proposed Load Balancer (LBr) for assigning the most suitable VM to that particular host.

Step.5: Proposed LBr starts the process of finding the suitable available host and follow the following path:

```
While (until all PES for each Host are not done)
    {
        Select the Host id (Hid) which has maximum number of PES.
        Assign Current Load=0, Current VM= available VM in the VM list.
        Assign capacity of current VM (CP) with the Threshold Capacity (Cth).
        While (Current Load on Current VM< Capacity of Current VM)
        {
          If (Current Load + next PES task time on current Host id (Hid) <= capacity
of current VM)
            {
                Assign the Current VM to current PES of current Host (Hid).
                Decrement the PES value.
                Decrement total number of PES on current Host (Hid).
                Update the Current Load.
                Move to next PES on that Host (Hid).
         If (Current host (Hid) has no PES)
                  {
                    Move to another Host (Hid) which has Maximum PES.
                  }
            }
        Else
            {
                Move to the Next available Virtual Machine.
                Set Current Load=0 on that VM.
            }
      }
    }
```

Step.6: DCC now exit LB system from the VM allocation process.

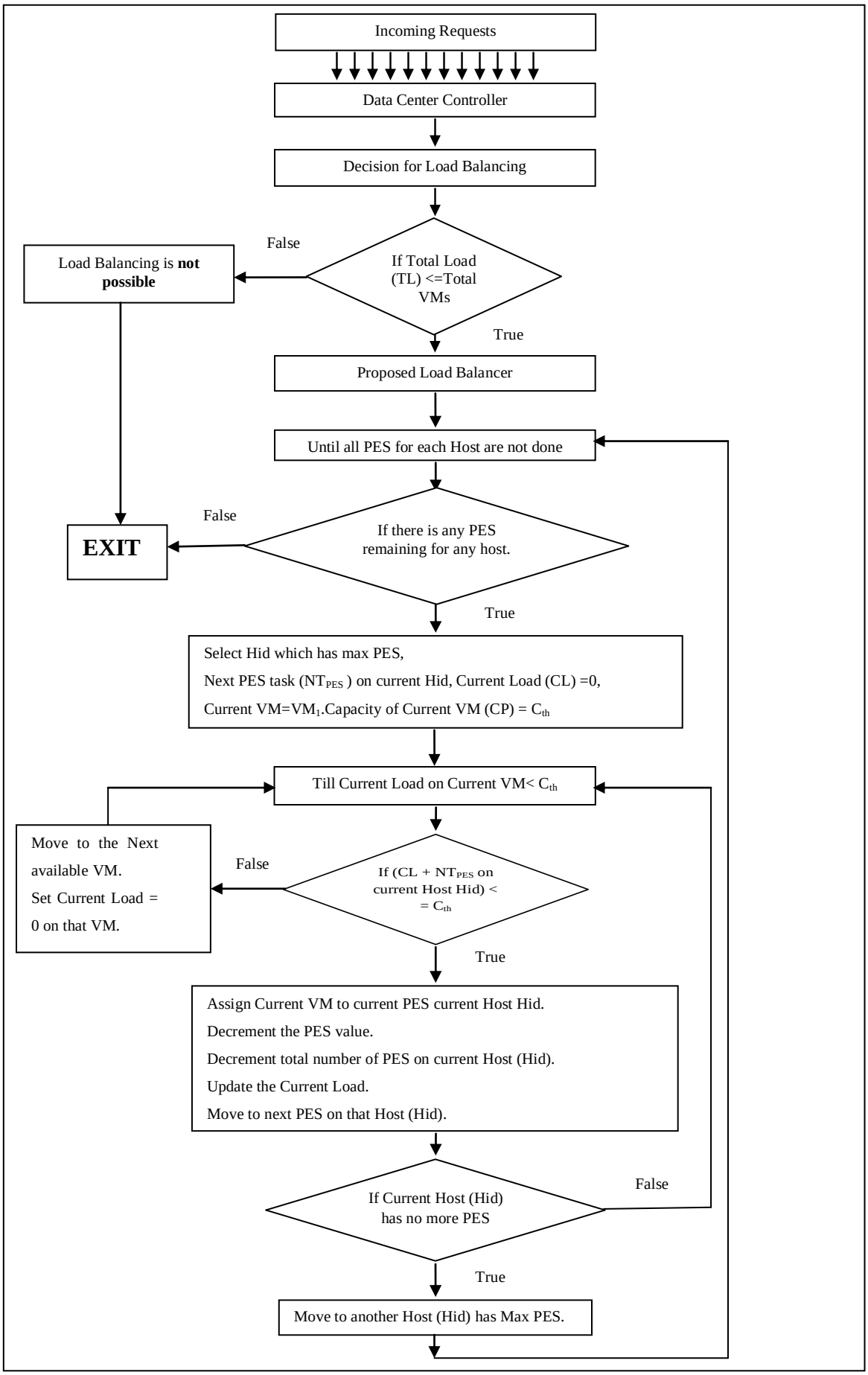

Figure 4.1: Flowchart of proposed LBA

5.4 PROCESS OF OPERATION

The proposed LBA demonstration is explained by the help of the example. In this given example, we take one DC, four Hosts, eight VMs, sorted task time array, and number of PES associated with each host for better explanation of proposed LBA. We have two tables, commonly named as Index table and VM table. Initially all the VMs are available for each host for their PES task allocation. We select the host which has maximum PES in the Index table and move to next Host with largest number of PES after done the previous Host. Suppose the threshold Capacity provided by the Cloud Service Provider is set to 40 sec. We have inputs which are as follows-

Index Table:

Serial Number	Host Id	Total number of PES (in Ascending order)	Task Time Array (Sorted)
1	H_3	6	4,6,7,8,9,18
2	H_2	5	7,8,9,16,18
3	H_1	4	10,15,17,20
4	H_4	2	12,16

Table 4.1: Index table for proposed LBA

VM Table:

VM_1	VM_2	VM_3	VM_4	VM_5	VM_6	VM_7	VM_8

Table 4.2: VM table

Host Table:

Array of requested tasks in terms of execution time associated with each Host.

$4(T_1)$	$6(T_2)$	$7(T_3)$	$8(T_4)$	$9(T_5)$	$18(T_6)$	**H_3**
$7(T_1)$	$8(T_2)$	$9(T_3)$	$16(T_4)$	$18(T_5)$	**H_2**	
$10(T_1)$	$15(T_2)$	$17(T_3)$	$20(T_4)$	**H_1**		
$12(T_1)$	$16(T_2)$	**H_4**				

Table 4.1: Host table for proposed LBA

On the basis of proposed LBA, first we have checked that LB is possible or not. For it we calculate total Capacity to VMs i.e. C.

C = Number of VMs * Threshold Capacity for a VM

= 8 * 40

= 320 sec

After that we calculate the Total Load TL by summation of all the tasks associated with the each host.

TL = 4+6+7+8+9+18+7+8+9+16+18+10+15+17+20+12+16

= 200 sec

Now check that LB is possible here or not. Since C is greater than the TL so LB is **possible** here.

Now we have to move to step 5 for selection of suitable Host. Host H3 is selected because it has maximum number of PES. Now select the PES from the Host Array which has associated tasks one by one. The process is as follows:

First PES of Host H3 is assigned to VM_1. Now current load on VM_1 is equal to the finishing time of the task T1 of Host H3 i.e. 4. After that check the condition that current load on VM_1 is less than equal to capacity of VM_1. Now this condition checks that the VM_1 has the capability to take some more tasks in parallel due to available capacity of VM_1. The PES's which get the VM_1 are as follow:

First PES of H_3 is assigned to VM_1.	4 sec	VM_1
Second PES of H_3 is assigned to VM_1.	6 sec	
Third PES of H_3 is assigned to VM_1.	7 sec	
Fourth PES of H_3 is assigned to VM_1.	8 sec	
Fifth PES of H_3 is assigned to VM_1.	9 sec	
	Total-34 sec	

Now next PES task time is 18 and we always check the condition for current load on current VM is less than equal to Threshold capacity C_{th}.

Current load on current VM = 34 sec

And condition evaluates to false when balancer checks the condition (Current Load + next PES task time on current Host id (Hid) <= capacity of current VM).

i.e. 34 +18 is not less than equals to 40. So, move to the next available VM i.e.VM_2.

So, Sixth PES of H_3 is assigned to VM_2.

Now, No PES exists in H_3 so LBr move to next Host which has Maximum PES i.e. H_2.

Sixth PES of H_3 is assigned to VM_2.	18 sec	VM_2
First PES of H_2 is assigned to VM_2.	7 sec	
Second PES of H_2 is assigned to VM_2.	8 sec	
	Total-33 sec	

Now next PES task time is 9 and we always check the condition for current load on current VM is less than equal to Threshold capacity Cth.

Current load on current VM = 33 sec

And condition evaluates to false when balancer checks the condition (Current Load + next PES task time on current Host id (Hid) <= capacity of current VM).

i.e. 33 + 9 is not less than equals to 40. So, move to the next available VM i.e. VM_3.

So,

Third PES of H_2 is assigned to VM_3.	9 sec	VM_3
Fourth PES of H_2 is assigned to VM_3.	16 sec	
	Total-25 sec	

Now next PES task time is 18 and we always check the condition for current load on current VM is less than equal to Threshold capacity C_{th}.

Current load on current VM = 25 sec

And condition evaluates to false when balancer checks the condition (Current Load + next PES task time on current Host id (Hid) <= capacity of current VM).

i.e. 25 + 18 is not less than equals to 40. So, move to the next available VM i.e. VM_4.

So, Fifth PES of H_2 is assigned to VM_4. Now, No PES exists in H_2 so LBr move to next Host which has Maximum PES i.e. H_1.

Fifth PES of H_2 is assigned to VM_4. 16 sec

First PES of H_1 is assigned to VM_4. 10 sec

Total-26 sec } VM_4

Now next PES task time is 17 and we always check the condition for current load on current VM is less than equal to Threshold capacity C_{th}.

Current load on current VM = 26 sec

And condition evaluates to false when balancer checks the condition (Current Load + next PES task time on current Host id (Hid) <= capacity of current VM).

i.e. 26 + 17 is not less than equals to 40. So, move to the next available VM i.e. VM_5.

So,

Second PES of H_1 is assigned to VM_5. 15 sec

Third PES of H_1 is assigned to VM_5. 17 sec

Total-32 sec } VM_5

Now next PES task time is 20 and we always check the condition for current load on current VM is less than equal to Threshold capacity C_{th}.

Current load on current VM = 32 sec

And condition evaluates to false when balancer checks the condition (Current Load + next PES task time on current Host id (Hid) <= capacity of current VM).

i.e. 32 + 20 is not less than equals to 40. So, move to the next available VM i.e. VM_6.

So, Fourth PES of H_1 is assigned to VM_6.

Now, No PES exists in H_1 so LBr move to next Host which has Maximum PES i.e. H_4.

So,

Fourth PES of H_1 is assigned to VM_6. 20 sec

First PES of H_4 is assigned to VM_6. 12 sec

Total-32 sec } VM_6

Now next PES task time is 16 and we always check the condition for current load on current VM is less than equal to Threshold capacity C_{th}.

Current load on current VM = 32 sec

And condition evaluates to false when balancer checks the condition (Current Load + next PES task time on current Host id (Hid) <= capacity of current VM).

i.e. 32 + 16 is not less than equals to 40. So, move to the next available VM i.e. VM_7.

So,

Second PES of H_4 is assigned to VM_7. 20 sec

Total-20 sec

VM_7

Now there is no task for which VM is not allocated anywhere. Now LBr will go to exit step of this proposed LBA.

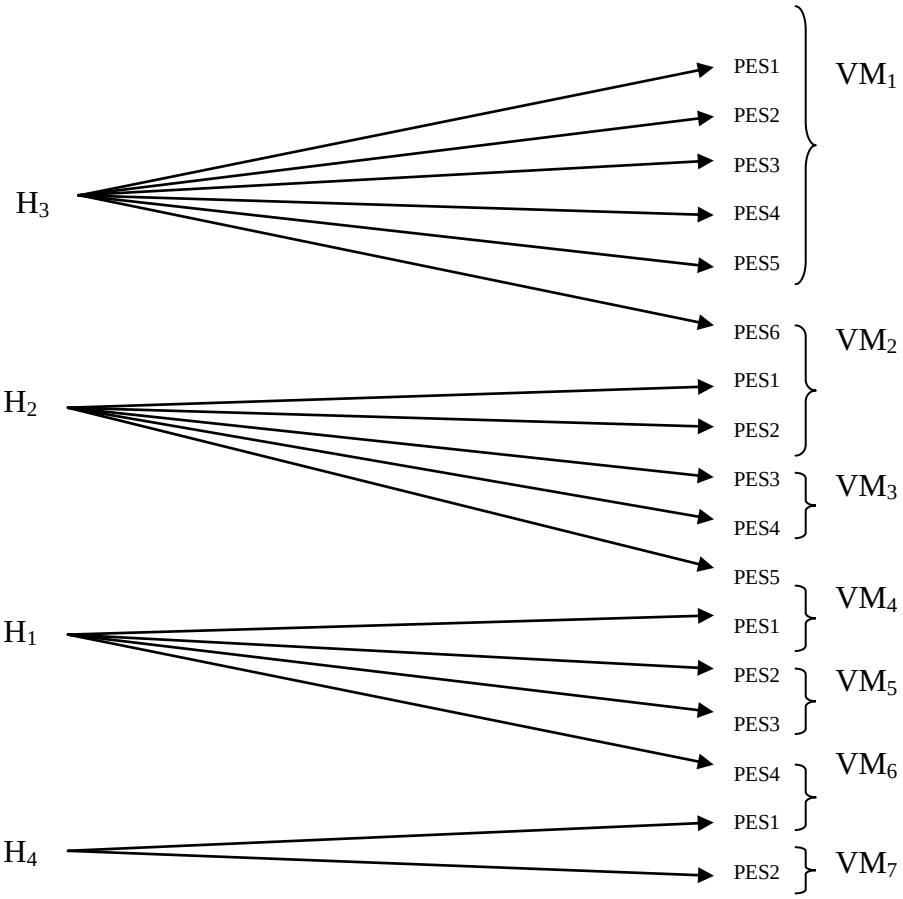

Figure 4.2: VM allocation by proposed LBA

The figure 4.2 describes the VM allocation for all the Hosts by proposed LBA. The proposed LBA efficiently distributes VMs on the network nodes in the datacenter. Since this LBA is works on the Host level so information related to Hosts, VMs, and the PES as inputs provided to LBA. A proposed LBr initially focuses on VMs, Hosts available in the DC. From above example explanation it is clear that proposed LBA is achieving its goal for balancing the load (task) in homogeneous manner. It uniquely allocates VMs to all the requested tasks efficiently. The proposed LBA is fully static and focused to the static load balancing in the cloud environment describe with the output results in next chapter.

CHAPTER 6

Experimental Setup and Results

As discussed in previous chapters, there are lots of LBAs suggested by various researchers, each having its own pros and cons. Selection of an effective LBA for a cloud system is one of the crucial tasks for the developers. With the proposed work it was tried to maintain an appropriate trade-off between different performance parameter so as to achieve the better results. In chapter 3 we have discussed about the various metrics which are used to judge the performance of any LBA which have been done. With the proposed LBA our main aim is to balance the overall workload among different nodes (server or host) in the network.

6.1 USED TECHNOLOGIES

In this section we present a brief description of the technologies which are used in our proposed LBA. These intersecting technologies are discussed below:

6.1.1 JAVA

Java is an object oriented programming language which operates by the help of objects. It supports the object oriented features which are commonly used in the present scenario. These features help developers to deploy and develop many applications and services in cloud computing environment. In cloud computing a platform is provided through which a user can interact with the cloud services and further using as per the needs. This platform must be accessible from any device whether it is a mobile, a laptop, a tablet, or any other device. Java's ability to be run on any platform makes this possible. Java's byte code made it more secure and portable for use. Many computational tasks performed in cloud network require smooth and quick networking services. Java has many networking features which are used in cloud computing network. Java's applets are one of the most popular features which have revolutionized the web and the internet technology [38]. In implementing the proposed methodology java is used intensively.

6.1.2 CLOUDSIM

CloudSim [13] works as a simulator which simulates the cloud environment in your surroundings under CloudSim toolkit. We can visualize the cloud infrastructure by the help of CloudSim. We used **CloudSim3.0** to implement our proposed LBA. For simulating demonstration of the cloud computing scenarios we follow this simulator. It implements certain scenarios as per the user requirement to test the policy or work of the particular entity in a well established manner with the effectiveness of suitable environment. For simulating demonstration of the cloud computing scenarios we follow this simulator. With the help of CloudSim we model the cloud components. These cloud components includes different datacenters, virtual machines (VMs), and various resource provisioning policies. CloudSim toolkit also offers federation of clouds i.e., internetworking of different types of clouds. Figure 6.1 shows the layered architecture of CloudSim.

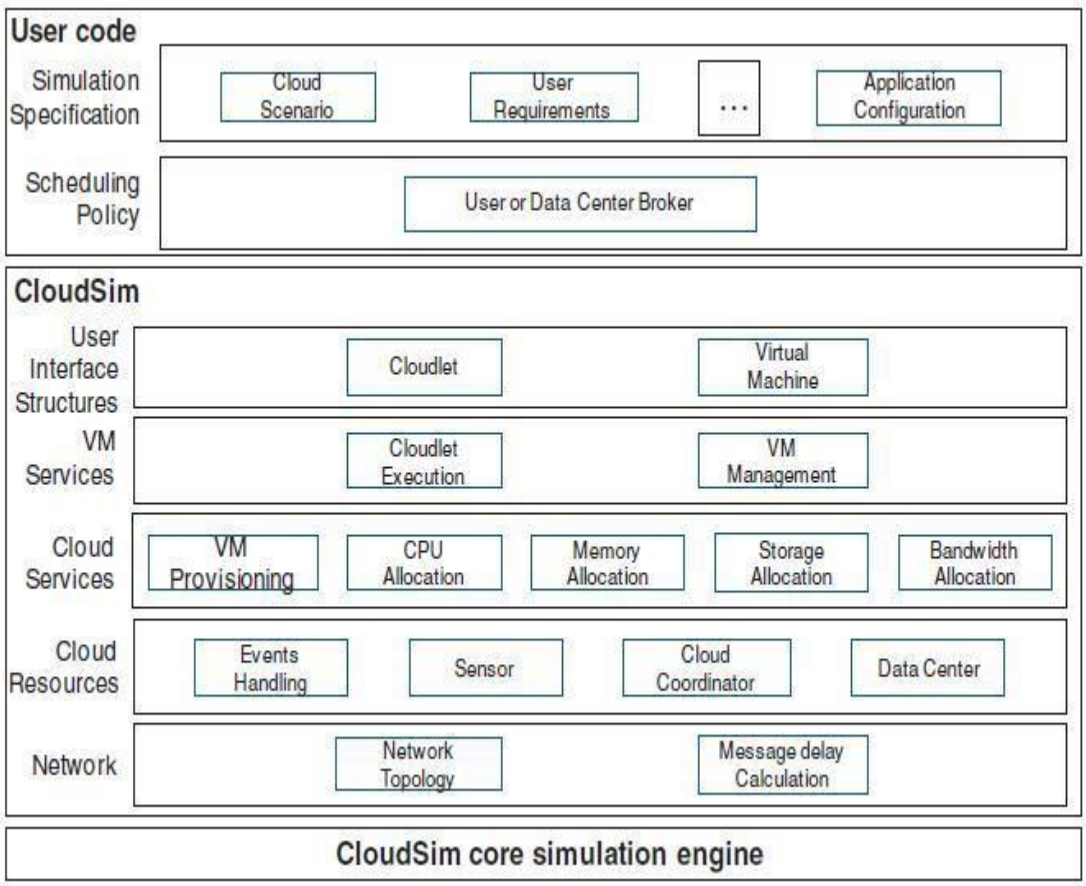

Figure 6.1: Layered architecture of CloudSim

6.1.3 CLOUDANALYST

A CloudAnalyst provides more user-friendly environment for the researchers. In CloudAnalyst a detailed report is generated for each piece of experiment. CloudAnalyst [14] is a GUI based simulator for modeling and analysis of large scaled applications. It is built on top of CloudSim toolkit, by extending CloudSim functionality with the introduction of concepts that model Internet and Internet Application behaviors. For the comparison of various load balancing policies and datacenter broker policies CloudAnalyst is used in this thesis. Figure.6.2 shows basic component upon which CloudAnalyst [14] has been build.

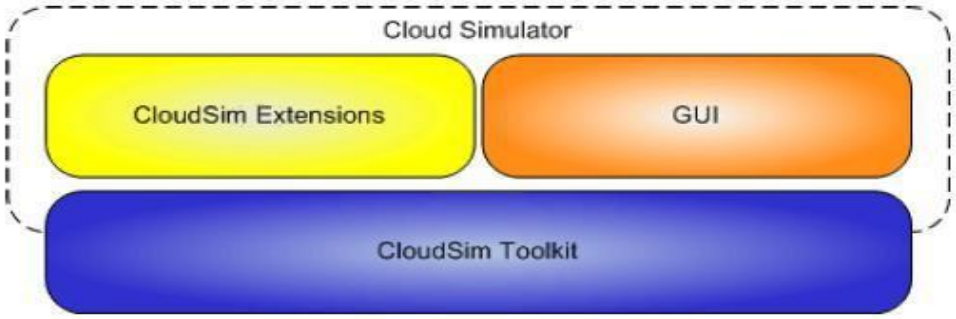

Figure 6.2: CloudAnalyst component

6.2 IMPLEMENTATION DETAILS

The implementation of our LBA is done by the help of the above technologies. The simulation work has been categorized in two sections. The first one is done using the CloudAnalyst. The second category of simulation is done using CloudSim toolkit. In this simulation of the proposed algorithm is carried out and further comparison is done with the traditional approach of load balancing. The algorithm is written in **Java** and is run on **Eclipse IDE** using CloudSim toolkit.

The program is implemented according to the flow of execution in CloudSim. Figure 6.3 shows the flow of execution of a program in CloudSim.

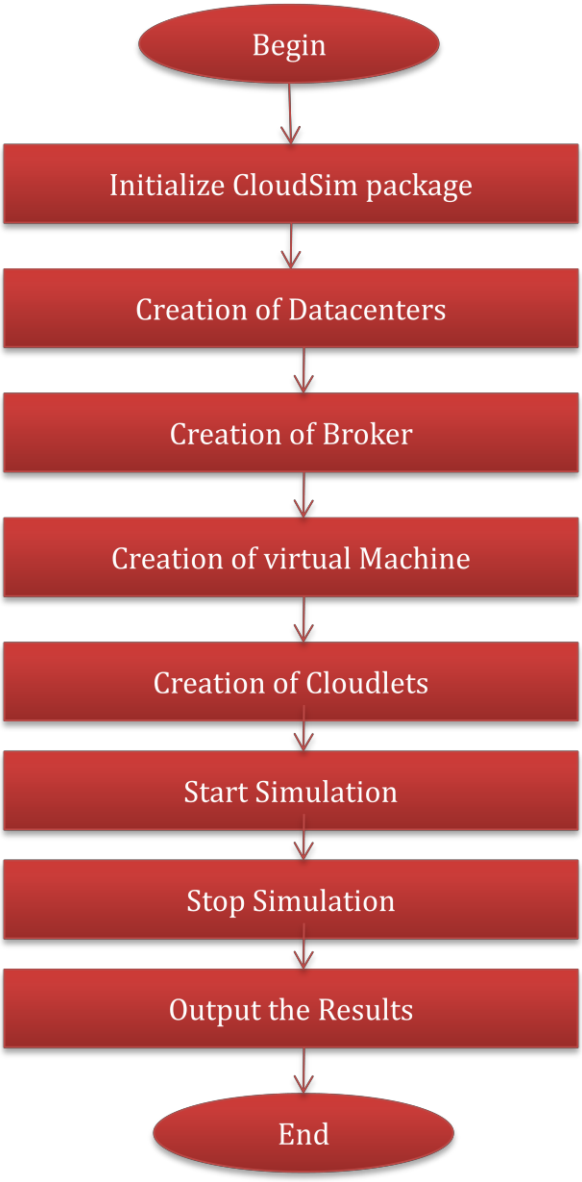

Figure 6.3: Flow of program execution in CloudSim

The CloudSim package is written in Java and it is used with the help eclipse IDE for implementation of proposed LBA. These CloudSim functional steps are shown in the output of the proposed LBA. As a functional advancement of CloudSim it is very easy to simulate the cloud environment with the all features of cloud infrastructure.

6.3 EXPERIMENTAL RESULTS

The results are produced by proposed LBA simulated in two categories. At first, simulation is done for the existing load balancing algorithms namely Round Robin, Throttled and Equal spread current execution algorithm in combination with the service broker policies like closest datacenter and the Optimized response time policy using the CloudAnalyst. Secondly, the proposed algorithm was simulated using CloudSim toolkit and then the results are compared with the traditional approach of load balancing. A graphical analysis is done in order to have clear understanding of both the approaches. Let us discuss these results in detail.

6.3.1 SIMULATION RESULTS OF EXISTING LBA

The overall response time in each case can be analyzed graphically through Figure 6.4. The maximum and average response time in each case is shown separately for each case. Generally, the average response time is considered for evaluating the performance of the overall system. Here also the average response time is considered for evaluation of the assumed scenario.

It can be clearly observed from the below graphical result that Throttled load balancing policy is best suited for our application. It gives the minimum response time compared to others. And for service broker policy, it can be concluded that Closest Datacenter policy gives the maximum profit to cloud service provider by giving the lowest cost for virtual machines and data transfer.

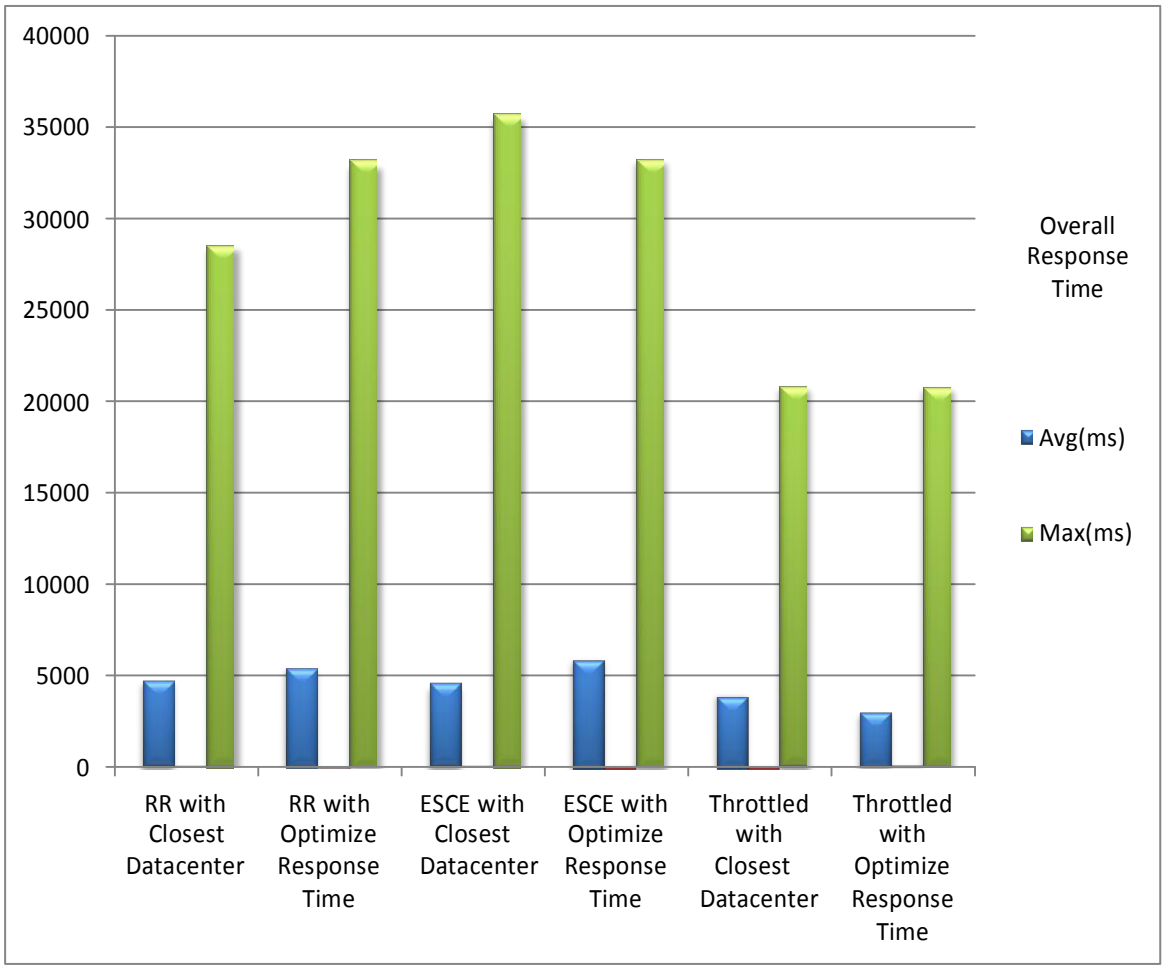

Figure 6.4: Overall Response Time for global cloud usage.

6.3.2 SIMULATION RESULTS OF PROPOSED LBA

The simulation results of the proposed LBA have been done in this section. Certain assumptions have been made regarding the experiment setup required for the verification and validation of the proposed LBA. A set of eight virtual machines and four hosts all with configurations have been modeled and configured for the experiment purpose in CloudSim. Each VM has the same configuration with equal VM capacity with the Threshold value set by the cloud service provider.

However, a proposed algorithm can be tested for a large number of scenarios, but in this thesis, for the sake of simplicity and understanding a general scenario in a cloud computing environment has been assumed. In this scenario, one datacenter is created and eight VMs are created which are to be deployed on any of the four hosts created in the datacenter based upon the LBA. Each host is having different number of processing cores i.e. PES.

Here, different numbers of processing cores are assumed in order to judge the proposed algorithm more precisely in a heterogeneous environment. Here host H1 is having four PES, similarly H2 five, H3 six, and H4 two. Also each host is same configured, each parameter like memory, bandwidth, secondary storage, number of processing cores are assigned. According to the proposed approach the virtual machines are deployed on the host with maximum number of the processing element available. It has been clearly visible in the output that the host allocation is done with the same rule of traditional approach. It is also assumed that all the virtual machine required any number of processing elements from any of the host for performing its operation.

Since, the proposed LBA is operated at host level in which hosts are allocated to the virtual machines so all the assumptions are made in accordance with this. Although the main code is written in such a way that a user can take any number of datacenters, hosts, virtual machines and cloudlets. Also different configurations can be set for the each component. Allocation for the host can be clearly seen in the below output. According to proposed LBA VMs are well managed to hold the PES requests till its capacity.

Whenever a VM hold the Various PESs of one of the host and its current load is low as compare to its VM threshold capacity, than it will also take other PES request of another host till its capacity limit exits to process the particular task. If the sum of the next PES request task time and current load of that VM exceeds the time limit beyond the capacity of VM than that PES request is allotted to another available VM. The proposed LBA provides the more continuous distribution of load assignment of VMs to various Hosts.

The output for the above stated scenario by proposed LBA is shown as:

Starting MyTest... Initializing...
Starting CloudSim version 2.0
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 #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 0.0:
Broker: Trying to Create VM #4 in Datacenter_0 0.0:
Broker: Trying to Create VM #5 in Datacenter_0 0.0:
Broker: Trying to Create VM #6 in Datacenter_0 0.0:
Broker: Trying to Create VM #7 in Datacenter_0 0.0:
Broker: Trying to Create VM #8 in Datacenter_0
0.0: Broker: VM #1 has been created in Datacenter #0, Host #3,
0.0: Broker: VM #2 has been created in Datacenter #0, Host #3, Host #2
0.0: Broker: VM #3 has been created in Datacenter #0, Host #2
0.0: Broker: VM #4 has been created in Datacenter #0, Host #2, Host #1
0.0: Broker: VM #5 has been created in Datacenter #0, Host #1
0.0: Broker: VM #6 has been created in Datacenter #0, Host #1, Host #4
0.0: Broker: VM #7 has been created in Datacenter #0, Host #4
0.0: Broker: VM #8 has been created in Datacenter #0, No Host Available, No request is pending.
0.0: Broker: Sending cloudlet 1 to VM #1
0.0: Broker: Sending cloudlet 2 to VM #2
0.0: Broker: Sending cloudlet 3 to VM #3
0.0: Broker: Sending cloudlet 4 to VM #4
0.0: Broker: Sending cloudlet 5 to VM #5
0.0: Broker: Sending cloudlet 6 to VM #6
0.0: Broker: Sending cloudlet 7 to VM #7
0.0: Broker: Sending cloudlet 8 to VM #8
80.0: Broker: Cloudlet 1 received
80.0: Broker: Cloudlet 2 received
80.0: Broker: Cloudlet 3 received
80.0: Broker: Cloudlet 4 received
80.0: Broker: Cloudlet 5 received
80.0: Broker: Cloudlet 6 received
80.0: Broker: Cloudlet 7 received
80.0: Broker: Cloudlet 8 received
160.0: Broker: All Cloudlets executed. Finishing... 160.0:
Broker: Destroying VM #1
160.0: Broker: Destroying VM #2
160.0: Broker: Destroying VM #3
160.0: Broker: Destroying VM #4
160.0: Broker: Destroying VM #5
160.0: Broker: Destroying VM #6
160.0: Broker: Destroying VM #7
160.0: Broker: Destroying VM #8
Broker is shutting down... Simulation: No more future
events
CloudInformationService: Notify all CloudSim entities
for shutting down. Datacenter_0 is shutting down...
Broker is shutting down... Simulation completed.

Cloudlet ID	STATUS	Data center ID	VM ID	Host IDs	Time	Start Time	Finish Time
1	SUCCESS	0	1	3	80	0	80
2	SUCCESS	0	2	3,2	80	0	80
3	SUCCESS	0	3	2	80	0	80
4	SUCCESS	0	4	2,1	80	0	80
5	SUCCESS	0	5	1	80	0	80
6	SUCCESS	0	6	1,4	80	0	80
7	SUCCESS	0	7	4	80	0	80
8	SUCCESS	0	8	-	-	-	-

MyTest finished!

In case of the proposed algorithm it is clearly visible from the output that each Host is allocated to a given VM as per the proposed methodology, which is comparatively **more distributed** in nature.

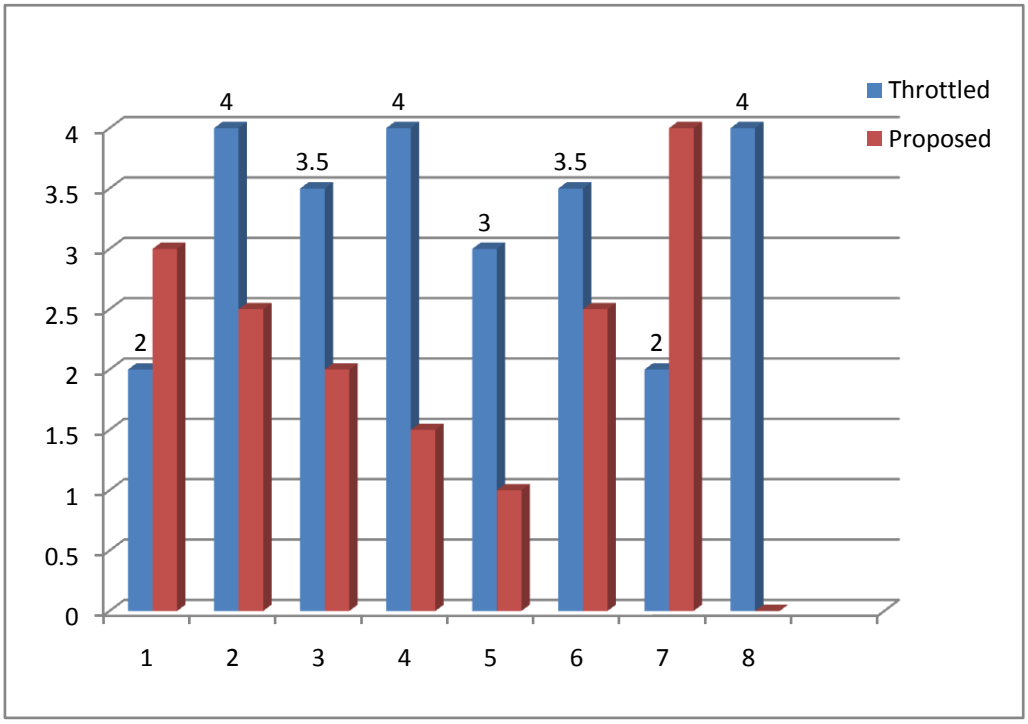

Figure 6.5: Comparison between Throttled approach and the proposed LBA approach.

The order of Host allocation is clearly visible in the above output. A level of distribution of Hosts among the virtual machines is more in case of the proposed load balancing algorithm. Similarly a number of experiments can be done to compare the proposed methodology with number of other related methodologies in any kind of scenario.

A more distributive curve will be obtained in case of the proposed methodology. A graphical analysis is done in order to clearly view the difference between the proposed methodology and the existing methodology. Figure 6.5 shows this graphical analysis. X-axis shows the number of VMs and the Y-axis shows the number of hosts. In the graph it is clearly shown that the proposed methodology is more distributive i.e., the virtual machines are more distributed over the host in case of the proposed methodology.

CHAPTER 7

Conclusion and Future Work

This chapter concludes the work presented in this thesis and also the future aspects of the proposed load balancing algorithm. Many other researchers can extend the work upon this research area and particularly on this proposed load balancing methodology. These all aspects are discussed here.

7.1 CONCLUSION

The basic purpose to apply this proposed algorithm is to achieve dynamic linking at the time of virtual machine allocation to the Hosts. Instead of taking single host for a particular virtual machine it takes the host's PES till its capacity exist. It makes cloud architecture more flexible when the number of inputs comes in bulk for the cloud service usage. We have done this at host level. The basic aim is to derive this algorithm is that when a PES has less amount of CPU execution time and virtual machine has high configuration efficiency in terms of processing speed, memory and capacity then why we cannot take the whole utilization of a virtual machine to another PES cores. In this algorithm it is the main consideration with the virtual machine capacity.

Load balancing in cloud computing can be done by provisioning the resources in the cloud computing network or by scheduling the user tasks in an efficient manner. As per the research work associated with this dissertation, resource provisioning in cloud computing is chosen as it can be more effective in distributing the overall workload to various resources in a cloud computing network. In cloud network the resource provisioning is done at two levels: VM level and host level. We have chosen resource provisioning at host level in which the hosts i.e. the physical servers are allocated to various virtual machines which are instantiated by the users as per their specification. Load balancing is one of the key challenge in the area of cloud computing, which is gaining a larger attention among all the researchers and developers over the globe. Therefore, this issue has been picked and analyzed thoroughly from different perspectives..

There are number of factors like throughput, complexity, scalability, flexibility, response time, etc which are together responsible for judging the right load balancing algorithm for a system. A developer must design a load balancing policy satisfying all these parameters which in turn increases the overall efficiency of the system. In this thesis we analyzed each of these parameters in detail. Different load balancing algorithms are compared based on these parameters. Various service broker policies are also analyzed in combination with the different load balancing algorithms. These service broker policies can play an important role in dealing the heavy real-time traffic over the network. Overall it is very crucial for any system to perform the all the related operations in a short period of time with higher efficiency. In cloud computing systems the same is expected.

7.2 FUTURE WORK

In future we can do number of modifications to have better results. Let us discuss some points which can be implemented in future:

- The proposed approach can be further modified by setting an appropriate threshold to achieve power saving while making the overall system energy efficient.
- The proposed algorithm can be integrated with the lower layer i.e. VM level at which virtual machines are allocated to the cloudlets. This will make the system more efficient and the overall response time can be reduced further.
- Also this work can be deployed in some real-time platforms to handle the real-time traffics in the network and then evaluating the performance of the system.

REFERENCES

1. Randles, M., D. Lamb and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, Australia, April 2010.
2. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
3. P. Mell and T. Grance. The NIST Definition of Cloud Computing (Draft). *National Institute of Standards and Technology*, 53:7, 2010.
4. I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Proceedings of Grid Computing Environments Workshop*, pages 1–10, 2008.
5. M. Armbrust, A. Fox, and R. Griffith. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
6. Amazon ec2. http://aws.amazon.com/ec2/.
7. Amazon s3. http://aws.amazon.com/s3/.
8. Google app engine. http://code.google.com/appengine/.
9. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
10. Xtreemos. http://www.xtreemos.eu/.
11. Opennebula. http://dev.opennebula.org/.
12. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.
13. R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling And Simulation Of Scalable Cloud Computing Environments And The Cloudsim Toolkit: Challenges And Opportunities," Proc. Of The 7th High Performance Computing and Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.
14. Bhathiya Wickremasinghe, "CloudAnalyst: A CloudSim based Tool for Modeling

and Analysis of Large Scale Cloud Computing Environments" MEDC project report, 433-659 Distributed Computing project, CSSE department., University of Melbourne, 2009.

15. Radojevic, B. & Zagar, M. (2011). *Analysis of issues with load balancing algorithms in hosted (cloud) environments*. In proceedings of 34th International Convention on MIPRO, IEEE.
16. D A Menasce, P NGO, "Understanding Cloud Computing: Experimentation and Capacity Planning", Proc. Computer Measurement Group Conf, Dallas, TX, Dec. 7-11, 2009.
17. Nidhi Jain Kansal and Inderveer Chana, "Existing Load Balancing Techniques in Cloud Computing: A systematic Review",Journal of Information Systems and Communication, 2012.
18. Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, Shun-Sheng Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", 2010 IEEE, pp. 108-113.
19. Che-Lun Hung, Hsiao-hsi Wang and Yu-Chen Hu, "Efficient Load Balancing Algorithm for Cloud Computing Network," Dept. of Computer Science & Communication Engineering, Providence University 200 Chung Chi Rd., Taichung 43301, Republic of China (Taiwan).
20. T V R Anandarajan, M A Bhagyabini, "Co-operative scheduled Energy aware load balancing technique for an efficient computational cloud", IJCSI, volume 8, issue March, 2011.
21. Jeffrey M. Galloway, Karl L. Smith, Susan S. Vrbsky, "Power Aware Load Balancing for Cloud Computing", Proceedings of the World Congress on Engineering and Computer Science 2011 Vol I WCECS 2011, October 19-21, 2011.
22. Raul´ Alonso-Calvo, Jose Crespo, Miguel Garc´ıa-Remesal, Alberto Anguita and Victor Maojo, "On distributing load in cloud computing: A real application for very-large image datasets", International Conference on Computational Science, ICCS 2010, pp.-2669- 2677, 2010.
23. Zenon Chaczko Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid, "Availability and Load Balancing in Cloud Computing", 2011

International Conference on Computer and Software Modeling IPCSIT vol.14, IACSIT Press, Singapore, 2011.

24. Alexandru Iosup, Member, IEEE, Simon Ostermann,Nezih Yigitbasi, Member, IEEE, Radu Prodan, Member, IEEE, Thomas Fahringer, Member, IEEE, and Dick Epema, Member, IEEE, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing", IEEE TPDS, MANY-TASK COMPUTING, NOVEMBER 2010.
25. Srinivas Sethi, Anupama Sahu, Suvendu Kumar Jena, "Efficient load Balancing in Cloud Computing using Fuzzy Logic", IOSRJEN July 2012.
26. Md. S. Q. Zulkar Nine, Md. Abul Kalam Azad, Saad Abdullah, Rashedur M Rahman, "Fuzzy Logic Based Dynamic Load Balancing in Virtualized Data Centers", Fuzzy Systems (FUZZ), 2013 IEEE International Conference.
27. Milan E. Soklic "Simulation of Load balancing algorithms" ACM - SIGCSE Bulletin, December, 2002.
28. Ankush P.Deshmukh and Prof. Kumarswamy Pamu "Applying Load Balancing: A Dynamic Approach" (IJARCSSE), vol. 2, issue 6, June 2012.
29. Jingnan Yao Jiani Guo ; Bhuyan, L.N. , "Ordered Round-Robin: An Efficient Sequence Preserving Packet Scheduler" IEEE transactions, vol. 57 , issue: 12, 30 May, 2008.
30. Jaspreet kaur "Comparison of Load balancing algorithms in a Cloud" International Journal of Engineering Research and Applications" (IJERA), vol. 2, issue 3, May-June 2012.
31. Zhang Bo; Gao Ji; Ai Jieqing "Cloud Loading Balance algorithm" Information Science and Engineering (ICISE), Second International Conference, 4-6 Dec. 2010.
32. Soumya Ray and Ajanta De Sarkar "Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment" (IJCCSA), vol.2, no.5, October 2012.
33. Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta and Kuwar Pratap Singh, Nitin and Ravi Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization" IEEE 2012 14th International Conference on Modelling and Simulation.

34. Shridhar G.Domanal and G.Ram Mohana Reddy "Load Balancing in Cloud Computing Using Modified Throttled Algorithm" IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), October 2013.
35. Subasish Mohapatra, Subhadarshini Mohanty, K.Smruti Rekha, "Analysis of Different Variants in Round Robin Algorithms for Load Balancing in Cloud Computing", IJCA, May 2013.
36. Ajay Gulati, Ranjeev.K.Chopra, "Dynamic Round Robin for Load Balancing in a Cloud Computing", IJCSMC, June 2013.
37. Ching-Chi Lin, Pangfeng Liu, Jan-Jan Wu, "Energy-Aware Virtual Machine Dynamic Provision and Scheduling for Cloud Computing", IEEE 4th International Conference on Cloud Computing, 2011.
38. Patrick Naughton and Herbert Schildt,Complete Reference Osborne/McGraw-Hill © 1999.
39. www.internetworldststs.com/facebook.htm