LOAD BALANCING IN CLOUD COMPUTING

A Dissertation submitted in partial fulfillment of the requirement for the

Award of degree of

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By

DEEPTI

(2K12/ISY/08)

Under the esteemed guidance of

Dr. N. S. RAGHAVA

Associate Professor

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042

2012-2014

CERTIFICATE

This is to certify that the thesis entitled **“Load Balancing in Cloud Computing ”** submitted by **Deepti (2K12/ISY/08)** 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 Information Technology
Delhi Technological University, Delhi.

ACKNOWLEDGEMENT

First I would like to express my gratitude towards my supervisor **Dr. N. S. Raghava**, *Associate Professor, Department of Information Technology* 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.

Deepti
Roll No.: 2K12/ISY/08
Dept. of Information Technology
Delhi Technological University

ABSTRACT

The advent of cloud computing in recent years has sparked an interest from different organizations, institutions and users to take advantage of services and applications. 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 promises scalability and on-demand availability of resources. As the number of users on the internet goes on increasing, it becomes difficult to handle the millions of user requests on it. Many a 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. Though having many good features such issues restrict a user to use the services. Users may not find it that much reliable.

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.**

CERTIFICATE ……………………………………………………………………………………(ii)

ACKNOWLEDGEMENT………………………………………………………………………(iii)

ABSTRACT ……………………………………………………………………………………(iv)

LIST OF FIGURES ……………………………………………………………………………(xi)

SCREEN SHOTS… ……………………………………………………………………………(xii)

LIST OF TABLES ……………………………………………………………………………(xiii)

Chapter 1: Introduction

1.1 Research Background ………………………………………………………...2

1.2 Challenges and Motivation ………………………………………………......2

1.3 Contributions and Objectives ………………………………………………....3

1.4 Organization of Thesis ……………………………………………………….4

Chapter 2: Cloud Computing

2.1 Introduction …………………………………………………………………..6

2.1.1 Cloud Computing Definitions ……………………………………………7

2.1.2 Cloud Components …………………………………………………...8

2.1.2.1 Clients ……………………………………………………………9

2.1.2.2 Datacenters…………………………………………………………...9

2.1.2.3 Distributed Servers……………………………………………………...9

2.2 Cloud Evolution ……………………………………………………………...10

2.2.1 Brief History …… ………………………………………………………..10

2.2.2 Comparison with Related Technologies………………………………….....11

2.2.2.1 Utility Computing……………………………………………...11

2.2.2.2 Grid Computing………………………………………………...11

2.2.2.3 Autonomic Computing…………………………………………12

2.3 Cloud Computing Architecture……………………..…………………....12

2.4 Cloud Service Models…………. ………………..………………………13

2.4.1 Software as Service ……………………………...............................13

2.4.2 Platform as Service …………………………………………………14

2.4.3 Infrastructure as Service ……………………………………….…14

2.4.4 Other Services………..………………………………………….…...15

2.5 Cloud Deployment Models………………...……………………….……15

2.5.1 Public Cloud………………..………………. …………………..…..16

2.5.2 Private Cloud…………..……………………….........................16

2.5.2.1 On-Premise Private Cloud………………………………….……..16

2.5.2.2 Externally Hosted Private Cloud…………………………….…….17

2.5.3 Hybrid Cloud ……………………………………………………….……17

2.5.4 Community Cloud ………………………..……………………....17

2.6 Cloud Characteristics ……………………. ………..................................18

2.6.1 Technical Characteristics………………………...............................18

2.6.1.1 Virtualization…………………………………………….……..19

2.6.1.2 Multi-tenancy…………………………………………….……..19

2.6.1.3 Security…………………………………………………….…...19

2.6.1.4 Programming Environment…………………………….……20

2.6.2 Qualitative Characteristics……………………..…………………….……20

2.6.2.1 Elasticity……………………………………………………….…..20

2.6.2.2 Availability……………………………………………….……..20

2.6.2.3 Reliability……………………………………………………….……21

2.6.2.4 Agility……………………………………………………….…..21

2.6.3 Economic Characteristics …..…………………..................................21

2.6.3.1 Pay-as-you-go……………………………………………….……..21

2.6.3.2 Operational expenditure…………………………………..……..22

2.6.3.3 Energy-Efficiency…………………………………………..…...22

2.7 Virtualization and Cloud Computing ……………..……...............................22

2.7.1 Key Characteristics of Virtualization……………………………..….23

2.7.1.1 Increased Security……………………………………………...23

2.7.1.2 Managed Execution…………………………………………….24

2.7.1.3 Portability…………………………………………………………24

2.7.2 Virtualization Techniques……...…………………………………..……24

2.7.2.1 Full Virtualization……………………………………………...24

2.7.2.2 Para Virtualization………………………………………..……..25

2.8 Issues In Cloud Computing…………………………..…………………….26

2.8.1 Load Balancing and Job Scheduling………………………………...26

2.8.2 License Management……………………………………………..…..26

2.8.3 Data Security…………………………………………………………....26

2.8.4 Scalable Storage Implementation………………………………..…27

2.8.5 Data Lock-in and Standard Protocol and API Design………………….27

2.8.6 Availability of Service……………………………………………………27

2.8.7 Cloud to Cloud Federation concerns…………………………………...27

2.8.8 Mobile Cloud…………………………………………………………..….28

2.8.9 Performance Unpredictability due to Virtualization……………….…...28

2.8.10 Bug Detection ……………………………………………………...28

2.9 Cloud Projects……………………………………..……………….………28

2.9.1 Commercial Products……………………..…………………….………29

2.9.2 Research Projects..30

2.10 Summary…………………………………………….…………….....…31

Chapter 3: Load Balancing

3.1 Introduction..………………...…………………………………………..32

3.2 Load Balancing Defined ………………………..……………………....32

3.3 Goals Of Load Balancing..……………………………….......................33

3.3.1 Technical Goals..………………………………..........................33

3.3.2 Business Goals..……...………………………………......................34

3.4. Load Balancing In Cloud Computing Environment……….......................34

3.4.1 At Host Level ………………………………………........................36

3.4.2 At VM Level ……………………….……………….......................36

3.5 Types of Load Balancing Algorithm...…….……………………………..37

3.5.1 Static Load Balancing Techniques……………….……...……………38

3.5.2 Dynamic Load Balancing Techniques………..…………………………39

3.6 Issues in Load Balancing in Cloud Computing Systems……........................41

3.6.1 Geographical Distribution of Nodes……………………...................41

3.6.2 Algorithm Complexity……………………………….........................42

3.6.3 Dynamic Vs Static Behavior of Load Balancing Algorithm...……….....42

3.6.4 Traffic Analyses Over Different Geographical Location.......………....42

3.6.5 Storage and Replication of Data in Cloud...…………………………....43

3.7 Performance Metrics Used to Evaluate Load Balancing Algorithm……….....43

3.7.1 Throughput………..…………………………….........................43

3.7.2 Overhead...………..………………………………..........................43

3.7.3 Fault Tolerance..……..………………………….........................44

3.7.4 Migration Time..……..………………………….........................44

3.7.5 Response Time..……..………………………….........................44

3.7.6 Resource Utilization....………………………….........................44

3.7.7 Scalability……..……..………………………….........................44

3.7.8 Performance…..……..………………………….........................45

3.8 Summary…………………………………………………..……..…......45

Chapter 4: Related Work………………………………………………..…….46

Chapter 5: Proposed Methodology

5.1 Introduction………………………………...50

5.2 Description…………………. ……………………..51

5.3 Process of Operation………...………………………………………………57

Chapter 6: Experimental Setup and Results

6.1 Used Technologies……………….………...…………………………..65

6.1.1 Java…………………………………………………………………...65

6.1.2 CloudSim……………………………………………………………66

6.1.3 CloudAnalyst……………………………………………………….....67

6.2 Implementation Details…....…………………………………………….…..67

6.3 Experimental Results………………… ……………………….……………71

6.3.1 Simulation Results of Existing Load Balancing Algorithm…………….71

6.3.2 Simulation Results of Proposed Load Balancing Algorithm……………78

Chapter 7: Conclusion and Future Work

7.1 Conclusion…………………..…………………………..........................87

7.2 Future Work ……………..…………………………..........................88

References………………………………………………………………..…….89

List of Publications……………………………………………………………**93**

LIST OF FIGURES

Fig. No	Title	Pg. No
1.1	Thesis organization roadmap	5
2.1	Hype Cycle for Cloud Computing. 2013	6
2.2	Components of Cloud	8
2.3	Emergence of various related technologies during different years	10
2.4	Cloud Computing Architecture	12
2.5	Cloud Service Models	15
2.6	Private, Public and Hybrid Cloud Deployment	18
2.7	Google Trends showing the searches on Cloud computing Vs Virtualization	23
2.8	Full Virtualization	25
2.9	Para Virtualization	25
3.1	Hierarchy and assignment of cloud components	35
3.2	Execution of load balancing algorithm	37
3.3	Load balancing schemes	38
3.4	Static Load Balancer	39
3.5	Dynamic Load Balancer	40
5.1	Execution flow of the proposed algorithm	56
5.2	Operation of algorithm	58
6.1	Layered architecture of CloudSim	66
6.2	CloudAnalyst component	67
6.3	Flow of program execution in CloudSim	70
6.4	Overall Response Time for all the cases	78
6.5	Comparision between the traditional approach and the proposed approach	86

SCREEN SHOTS

No.	Title	Pg. No
6.1	Simulation completed for Round Robin policy with Closest Datacenter policy	72
6.2	Simulation completed for Round Robin policy with Optimized Response time policy	73
6.3	Simulation completed for Equal Spread Current Execution policy with	74
6.4	Simulation completed for Equal Spread Current Execution policy with Optimize Response Time policy	75
6.5	Simulation completed for Throttled policy with Closest Datacenter Policy	76
6.6	Simulation completed for Throttled policy with Optimize Response Time policy	77

LIST OF TABLES

Table.No	Title	Pg.No
6.1	Registered users on Facebook as on 31-03-2012	68
6.2	User bases with parameters	68
6.3	Fixed parameters used during simulation	69

CHAPTER 1

Introduction

1.1 RESEARCH BACKGROUND

Cloud computing has made a significant improvement over previous computing technologies in terms of services they offer. Earlier, we use Grid computing and Distributed computing which are not able to provide much flexibility as provided through Cloud computing. There are number of attractive features available in cloud computing systems that made it so popular these days. Cloud computing follows pay-as-you-go model and enables on-demand provisioning of computing resources in an elastic manner. This standard has made cloud computing more demanding in the area of business applications where huge infrastructure-setup cost is required. With the advent of cloud technology users can easily rent the infrastructure, runtime environments and services. Also different users can utilize the benefits of cloud computing in various domains according to their needs. Cloud provider is the main entity, enabling various users to make use of different cloud services according to their choice. Cloud providers offer their customers the illusion of unlimited compute, network, and storage capacity often coupled with a ‘frictionless’ registration process where anyone with a valid credit card can register and start using cloud services. Some providers even offer free limited trial periods.

In Cloud Computing one of the major tasks of cloud service provider is to assign the user requests to various nodes present in the cloud. This assignment of tasks among the nodes in cloud should be as efficient as possible. For an efficient system, the total effort and the processing time for all the user requests should be as low as possible [1], while being able to manage the various affecting constraints such as heterogeneity and high network delays. These days Cloud computing has become so popular in the area of Information and Communication Technology (ICT), therefore the requirement of large and powerful data centers comes into picture. Due to rapid increase in the number of cloud users, cloud providers also have to boost the capabilities of different cloud components, it can be done by increasing their number, increasing their power or both. This kind of situation may results into a very huge network comprising cloud users, datacenters, nodes, virtual machines and user tasks. When the size of the network increases, the workload also increases. This increased workload should be managed well to have better overall performance of the system. Load balancing is one such area which only aims at the detection and efficient distribution of workload in a network.

Load balancing is one of the major issues which require greater consideration from cloud developers, so as to increase the overall performance of the system. An efficient load balancing techniques are required to improve the fault-tolerance and efficiency of the system. This will provide better quality of service to the cloud users. Various load balancing parameters are analyzed for each of the load balancing algorithm. These parameters are used to judge the quality of the algorithm. A detailed description about these parameters is given in chapter 3. A trade-off has to be maintained between all the parameters, according to the requirement of the system as well as the requirement of the users. There are number of load balancing approaches proposed by various researchers, each having different trade-off between different load balancing parameters.

1.2 CHALLENGES AND MOTIVATION

Cloud computing is attracting more number of users day by day. It has some very attractive features like scalability, elasticity, easy accessibility, better categorization of services into commodities, better delivery of services. These features make cloud computing easily marketable. A Cloud computing model comprising these features is ready to set a new era of computing, in which every cloud user can utilize its services according to their requirements, regardless of knowing about how and where these services are hosted. As the numbers of cloud users are increasing in an exponential manner, the responsibility of the cloud service provider, to balance the overall workload among the various nodes in the cloud, also increases. It is very important to manage all the resources like CPU, memory, secondary storage in a server efficiently. For this to be happen it is necessary for every cloud service provider to make its load balancer work in best possible manner.

It may happens at times when the network traffic rises exponentially for e.g. at the time of New Year, people used to send messages in huge amount at the same time instant, which results in heavy workloads on the message sending websites. In such a conditions service provider increases the messaging charges and may put restriction on free-messaging tariffs. Similar situation is observed when results of Board exams comes or at the time of festivals people used to book tickets, the server becomes heavily loaded with large amount of user requests coming at same time instants. Though, we say that clouds are scalable but at a time it becomes difficult to agree with this statement because of such situations mentioned above.

Another important point of consideration is federation of cloud: a cloud may consist of various combinations of private cloud, public cloud, hybrid cloud or a community cloud. A communication between these sorts of network creates some extra responsibilities over the cloud service provider.

Additionally, we know that Internet is the prime need for using the cloud services, so an inevitable issue is that network bottlenecks frequently occur when large data is transferred over the network. In that case, it becomes difficult to manage all the available resources efficiently [2].

Based on the above analysis, load balancing is a topic worthy of research, and is a key issue in improving the overall performance of the system. This work focuses on load balancing in cloud computing environment. Performance of the services provided to cloud users in cloud computing environment is very important, so to achieve high performance and higher level of user satisfaction we need to understand the factors that can affect the performance of the system. Load Balancing is one of most important factor which affects the overall performance of system. It can provide user with better quality of services and cloud service provider can have high throughput with better resource utilization.

1.3 OBJECTIVES AND CONTRIBUTIONS

This thesis studies load balancing related issues in cloud computing. The primary objective of this thesis is to provide an efficient load balancing mechanism, which helps in benefiting both, the cloud users and the cloud service providers. Efficient distribution of the workload among all the available nodes in the network not only helps in utilizing the resources in an efficient way but also results into faster processing of the user requests. In cloud computing environment load balancing can be done by performing effective scheduling of user tasks onto virtual machine and then provisioning of virtual machine to different nodes in the system. A better load balancing technique helps the system to achieve higher fault-tolerance.

The major contributions are as follows:

- A more efficient technique for load balancing in cloud computing is introduced. The technique’s center of attention was on improving the critical performance parameters like resource utilization, average response time and throughput.
- A comparative analysis of major load balancing algorithms in cloud computing systems. Many factors like geographical distribution of nodes, peak hour usage of services over different locations, scalability, response time, and throughput are analyzed in depth for each of these algorithms.

1.4 ORGANIZATION OF THESIS

In this chapter we have discussed about the computing. Also we have discussed how the term computing sets in different domains from the older parallel and distributed computing to the newest cloud computing. The rest of this thesis is organized as follows:

Chapter 2 provides an overview of Cloud Computing and the terminologies associated with it. This chapter explains the Cloud technology in an elaborated way, its implementation details, major challenges and benefits of the technology to the IT industry. Various important aspects of the Cloud Computing are discussed here. After reading this chapter you can have a better idea about this technology.

Chapter 3 provides an overview of Load Balancing and the terminologies associated with it. It also gives information regarding various load balancing algorithms that are used in cloud computing systems, to balance the overall workload among the nodes in the system. Discussion on various performance metrics for each of the load balancing algorithm in cloud computing environment is done to have better idea about these algorithms.

Chapter 4 elaborates the research work in the area of load balancing. Various researchers have studied this topic in depth and proposed efficient methods for managing the resources in the network. Many load balancing algorithms are proposed to distribute the workload equally in the network. Each of these algorithms has unique methodology, with each having some pros and cons.

Chapter 5 describes the proposed approach that is being used for implementing the load balancing in cloud computing systems. This Chapter explains the proposed load balancing technique which should be adopted to have better performance of the system.

Chapter 6 shows the results that are obtained by implementing the proposed approach and by making certain observations. This chapter justifies the proposed approach in contrast to existing algorithms.

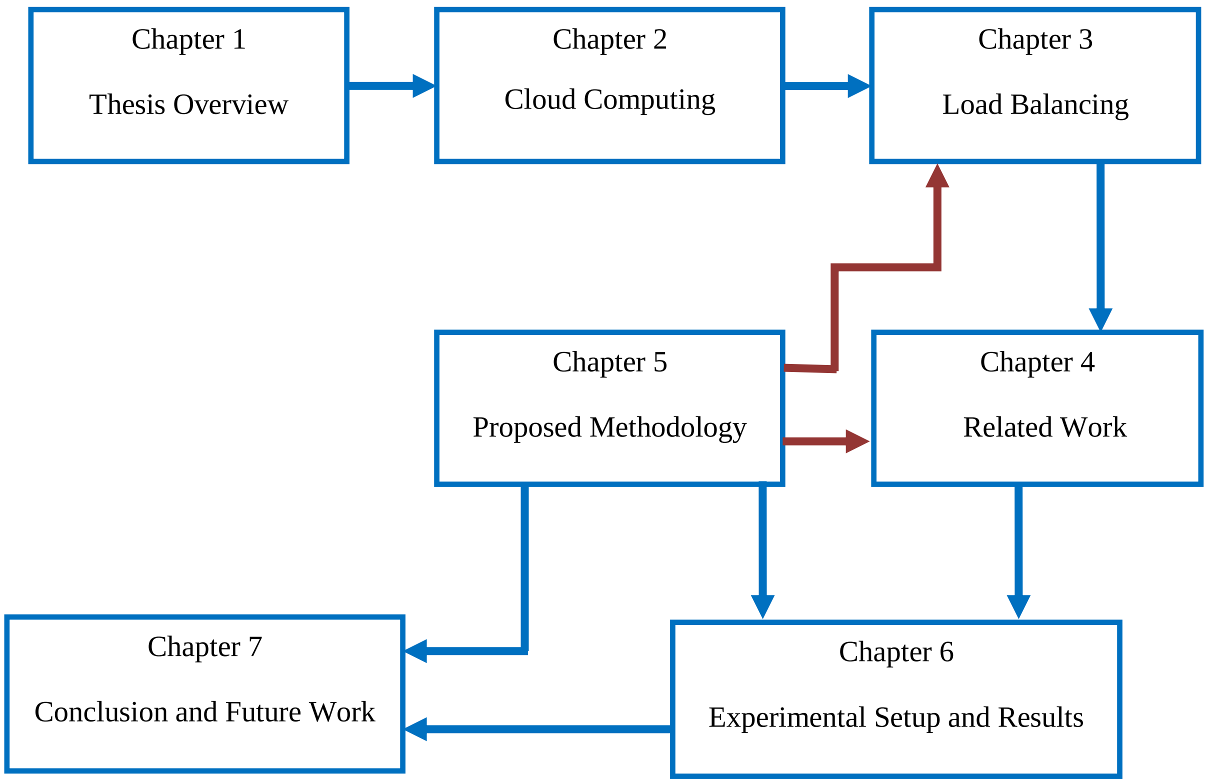


Figure 1.1: Thesis organization roadmap

Finally, **Chapter 7** concludes the thesis by discussing the overall contribution of the research in the context of related work in the area. In addition, this chapter also discusses the limitations of the approach and points to future research directions.
The structure of the thesis is illustrated in Figure 1.1.

CHAPTER 2

Cloud Computing

2.1 INTRODUCTION

Previous decades have reinforced the idea of processing information at larger scale, in a more efficient manner. With the advent of cloud computing headache of storing, processing and accessing data through internet at larger scale disappeared now. The network-oriented computing idea led to the evolution of *Grid computing* in the early 1990s and since 2005, to *utility computing* which ultimately led to the development of *cloud computing.* From a very long time researchers are trying to provide utilities as a service to its users. Users may demand software, platform or hardware from a provider through an internet and charged on the usage basis. Therefore cloud computing is nothing but, a path to utility computing espoused by IT giants such as Microsoft, IBM, HP, Amazon, Google, etc. The Cloud technology has a sensational rise in the industry, this can be seen in Figure 2.1. Within very short span of time this technology has spread over the globe.

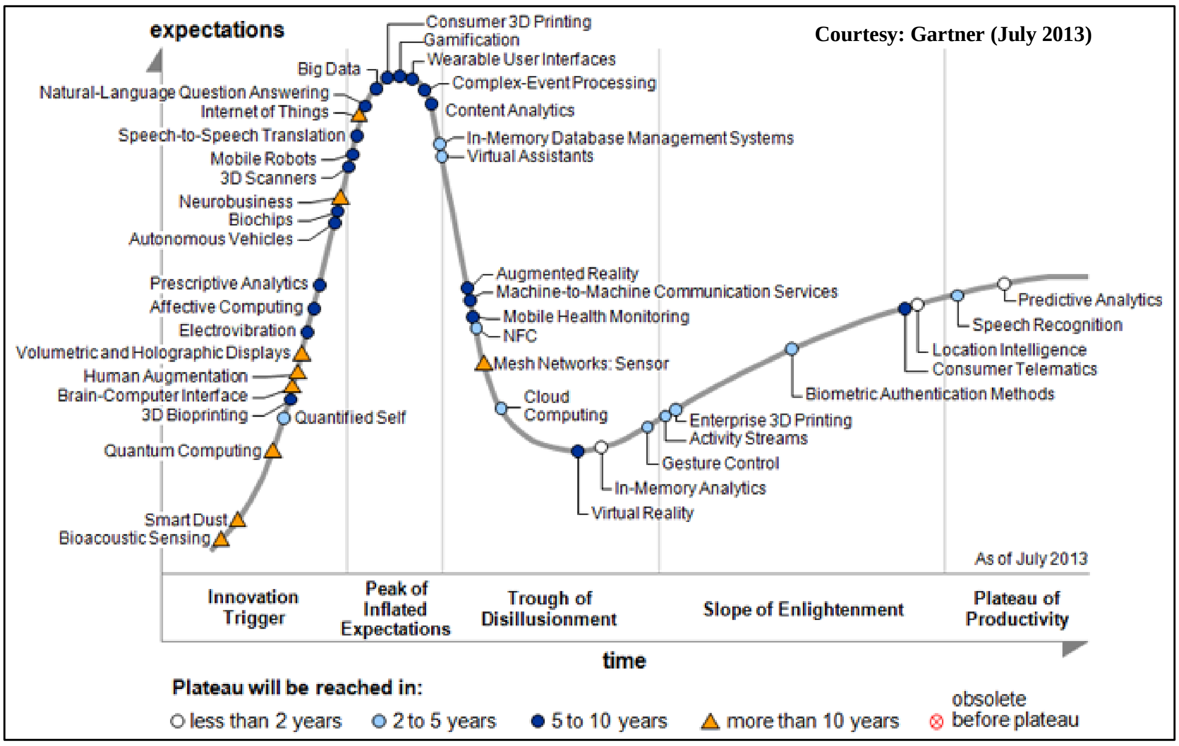


Figure.2.1: Hype Cycle for Cloud Computing. 2013

2.1.1 CLOUD COMPUTING DEFINITIONS

Since 2007, the term Cloud has become one of the most popular words in the IT industry. There are various researchers who have given a vast number of definitions for cloud computing. Everyone has defined it according to different application perspective but there is no common definition of cloud computing. Among the many definitions, we choose two widely quoted definitions of cloud computing 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]."

U.S. National Institute of Standards and Technology provides a very specific and goal-oriented definition of cloud computing. This not only defines cloud concept overall, but also specifies the essential characteristics of cloud computing with its delivery and deployment models. Foster's definition is little bit different as an educational representative he focuses on several methodological features that differentiate cloud computing from other distributed computing paradigm. For example, computing entities are virtualized and delivered as services, and these services are dynamically driven by economies of scale.

Joe Weinman has given a term "Cloudonomics", which define cloud computing from economical perspective, discussed below:

1. **C**ommon Infrastructure: a common and standard pool of resources is made available to all cloud users.
2. **L**ocation-independence: any user can access any resource or services from anywhere around the globe, leading to better performance of the system.
3. **O**nline connectivity: to access the services or resources over the cloud a consistent connection is maintained via internet.
4. **U**tility Pricing: pay-per-use pricing and benefits the user as per their demands increases.
5. on-**D**emand Resources: scalable, elastic resources provisioned and de-provisioned without delay or costs associated with change.

2.1.2 CLOUD COMPONENTS

From the topological aspect of cloud, there are some elements: clients or users, the datacenter and the distributed servers. These components integrate the cloud as one unit. In Figure 2.2, these components are shown:

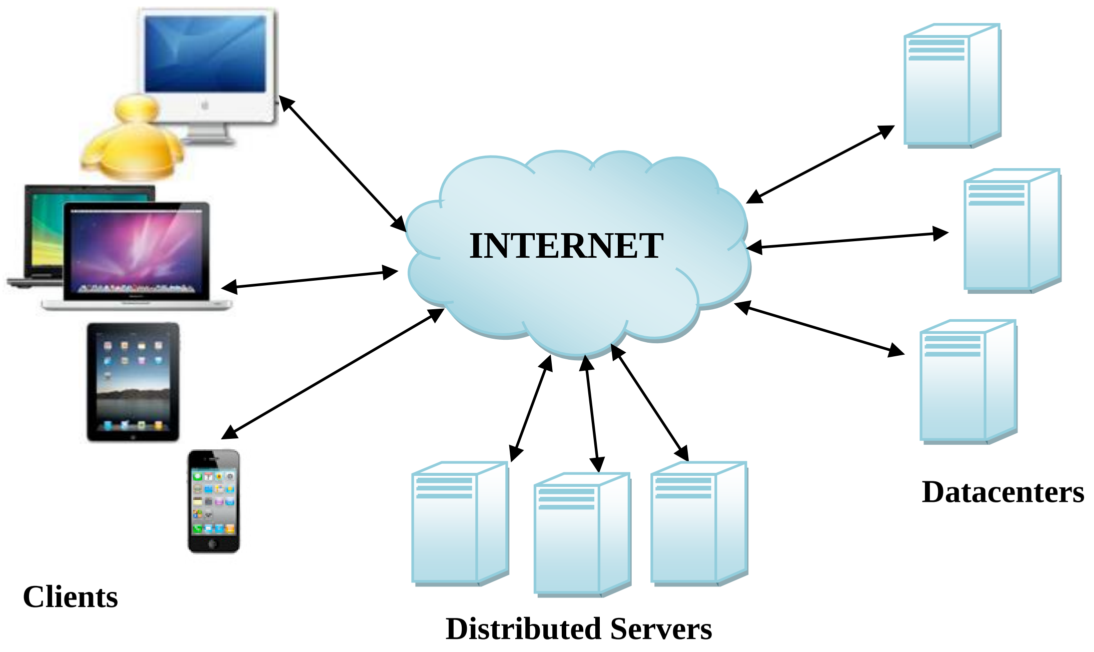


Figure.2.2: Components of Cloud

Each component has specific role in delivering services asked by the cloud users. These components communicate over the entire network, as per the requirements and configuration. These components are discussed below in detail.

2.1.2.1 CLIENTS

These are generally, the computers which are used in our day-to-day work. But it may be a laptop, a PDA, a mobile phone or a tablet computer. These all devices should be able to access the cloud computing interface via internet. With these devices end users are able to communicate to the cloud interface and afterwards can use the service or applications as per their choice.

2.1.2.2 DATACENTERS

A datacenter is one of the core elements of cloud computing systems. A datacenter consists of several nodes occupying a large room. These datacenters are configured by the cloud service providers. Thos configuration is based on various factors like cloud deployment model and the cloud service model.

2.1.2.3 DISTRIBUTED SERVERS

These servers, sometimes called as nodes, need not be deployed at the same location but may be distributed geographically as per the convenience of the service provider. From users prospect these servers seems to work together, without knowing the actual location of the server. Distributed servers increase the fault tolerance of the network.

2.2 CLOUD EVOLUTION

The idea of providing computing services on rent by investing on large distributed computing facilities is not so new. It can be seen from the past that, similar kind of technologies are always used in the IT industry with continuous modifications. In 1950 mainframe technology has given the birth for this sort of technologies. From there on, technology has evolved and been refined. This process has created a series of favorable conditions for the realization of cloud computing [12].

2.2.1 BRIEF HISTORY

Cloud computing is a steady evolution which started to emerge from 1950s, when the mainframe technology became very popular. Mainframe technology allows multiple users to access a central computer through their separate terminals. The terminals are responsible for providing the access to the mainframe. Afterwards, in 1970s the concept of virtual machine was introduced. One of the popular software used for virtualization was VMware. With the help of VMware it became possible to execute one or more operating systems simultaneously in an isolated environment. Virtualization is the key technology used in cloud computing. In 1990s, telecom companies started offering virtualized private network connections, which offer quality services at reduced costs.

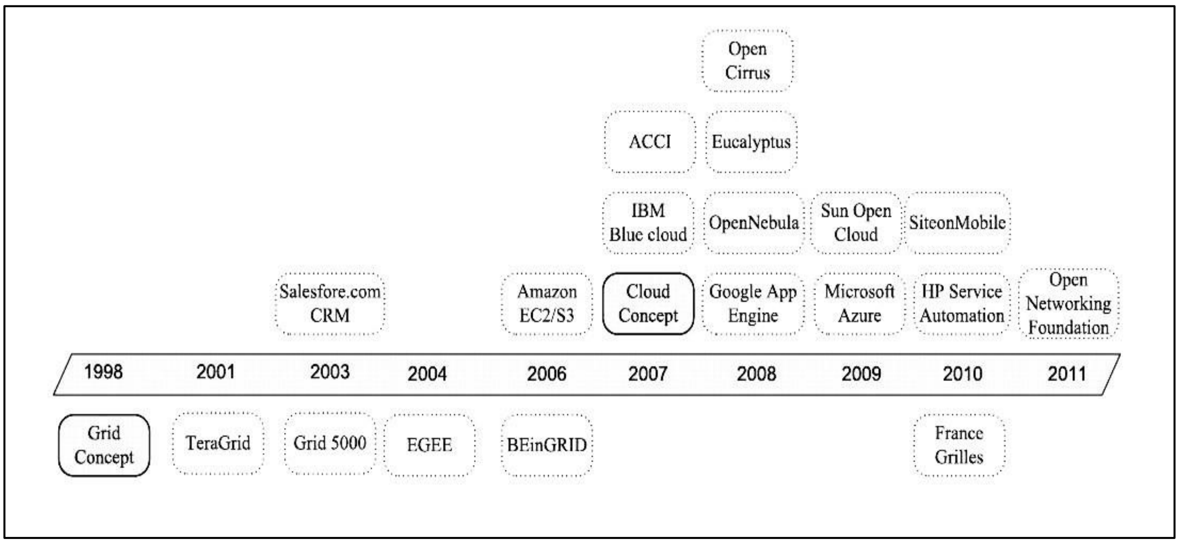


Figure.2.3: Emergence of various related technologies during different years

And then the series continued with the introduction of cluster computing afterwards grid computing. These all technologies together led to the evolution of cloud computing. In the next section a discussion will be on these technologies in relation to cloud computing, in depth. A simple history of cloud computing is shown in Figure 2.3.

2.2.2 COMPARISON WITH RELATED TECHNOLOGIES

Cloud computing comprised of service oriented architecture, virtualization, autonomic computing, utility computing and grid computing. It has emerged as a new computing paradigm which is ready to provide configurable, reliable and better class of services. Cloud computing is often confused with other related technologies like grid computing, utility computing and autonomic computing.

2.2.2.1 UTILITY COMPUTING

Utility computing is one of the very old technologies which came into picture from 1960s, when John McCarthy gave a speech in MIT. In utility computing users access services based on their requirements without regard to where the services are hosted. It is just like other metered services of water, electricity, gas and telephone.

2.2.2.2 GRID COMPUTING

This computing technology emerged in mid 90’s. Grid computing led to a new approach where a user can access large computational power with unlimited storage capacity and a facility of choosing other services too. Grids initially developed as aggregations of geographically dispersed clusters by means of Internet connections. Users can “consume” resources in the same way as they use other utilities such as power, gas, and water. Cloud computing is frequently considered as the successor of grid computing.

2.2.2.3 AUTONOMIC COMPUTING

This type of computing is first proposed by IBM in 2001. As the name suggests autonomic computing performs the task according to some adaptive policies. With increasing complexities in the computing network there occurs some limitation on future development, autonomic computing having adaptive policies used for the self-management of the computing system.

2.3 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 2.4 shows the layered architecture of cloud computing.

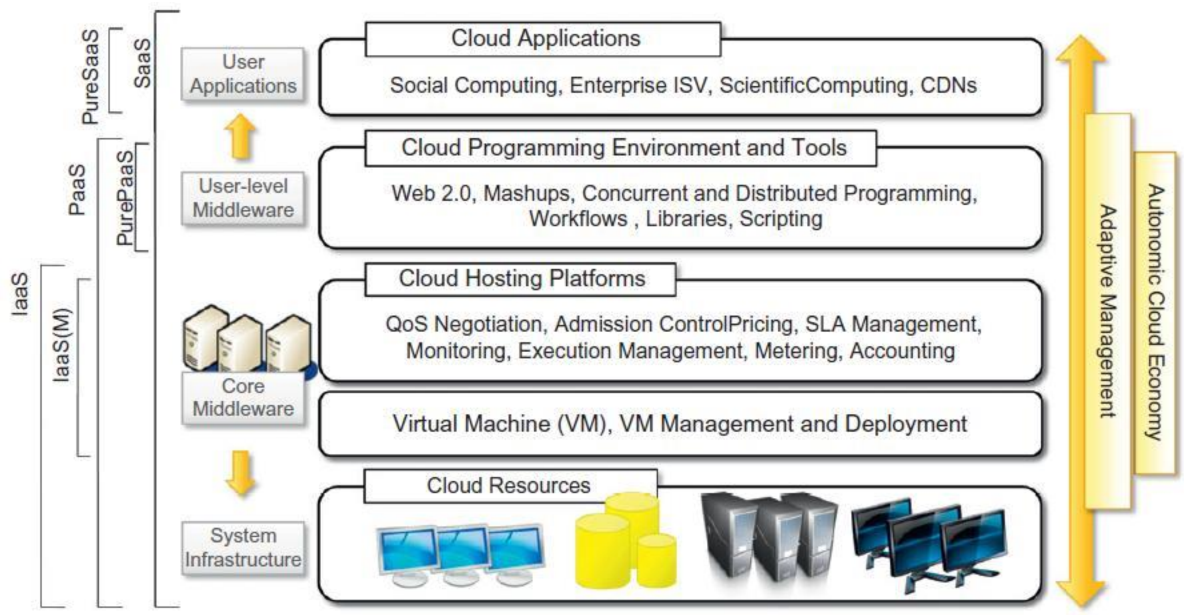

PureSaaS
SaaS
PaaS
PurePaaS
IaaS
IaaS(M)
User Applications
User-level Middleware
Core Middleware
System Infrastructure
Cloud Applications
Social Computing, Enterprise ISV, ScientificComputing, CDNs
Cloud Programming Environment and Tools
Web 2.0, Mashups, Concurrent and Distributed Programming, Workflows , Libraries, Scripting
Cloud Hosting Platforms
QoS Negotiation, Admission ControlPricing, SLA Management, Monitoring, Execution Management, Metering, Accounting
Virtual Machine (VM), VM Management and Deployment
Cloud Resources
Adaptive Management
Autonomic Cloud Economy

Figure.2.4: 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.

2.4 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]:

2.4.1 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.

2.4.2 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.

2.4.3 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.

2.3.4 OTHER SERVICES

Above discussed services are the core services which are categorized in accordance with the major market trends, user specifications and computing capabilities. Other services like security as a service, testing as a service are domain-specific services. These aims at providing particular fields of services to the users interested in that particular field for their applications.

Figure 2.5, shows the service models in cloud computing.

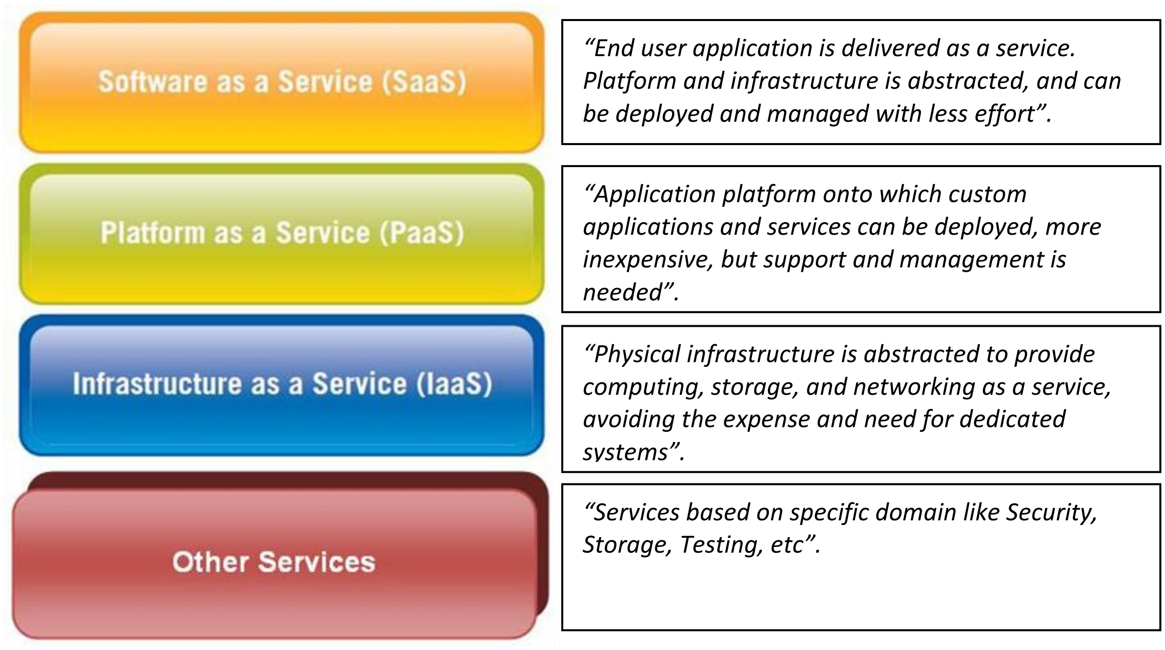

Software as a Service (SaaS)
"End user application is delivered as a service. Platform and infrastructure is abstracted, and can be deployed and managed with less effort".
Platform as a Service (PaaS)
"Application platform onto which custom applications and services can be deployed, more inexpensive, but support and management is needed".
Infrastructure as a Service (IaaS)
"Physical infrastructure is abstracted to provide computing, storage, and networking as a service, avoiding the expense and need for dedicated systems".
Other Services
"Services based on specific domain like Security, Storage, Testing, etc".

Figure.2.5: Cloud Service Models

2.5 CLOUD DEPLOYMENT MODELS

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.

2.5.1 PUBLIC CLOUD

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.

2.5.2 PRIVATE CLOUD

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:

2.5.2.1 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.

2.5.2.2 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.

2.5.3 HYBRID 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.

2.5.4 COMMUNITY CLOUD

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 or they can hire some third party for the purpose.

Figure 2.6, shows the cloud deployment model.

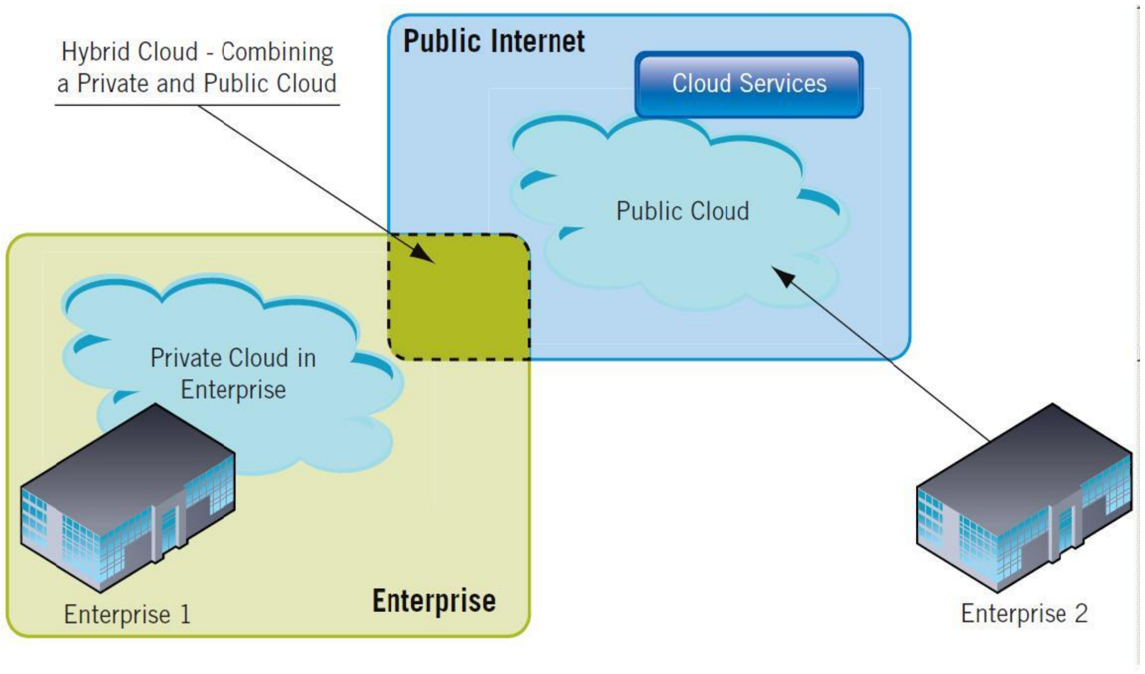

Hybrid Cloud - Combining a Private and Public Cloud
Public Internet
Cloud Services
Public Cloud
Private Cloud in Enterprise
Enterprise 1
Enterprise
Enterprise 2

Figure.2.6: Private, Public and Hybrid Cloud Deployment

2.6 CLOUD CHARACTERISTICS

Cloud computing embodies the aspects of a number of existing technologies. These existing technologies could be grid computing, service oriented architectures, utility computing, autonomic computing or internet of things. This is why it was often said that cloud computing is nothing but the identical previous concept with new label. This section describes the characteristics of cloud computing from different perspectives.

2.6.1 TECHNICAL CHARACTERISTICS

Technical characteristics serve the basis for functional and economical requirements. Generally a technology is not completely unique, but is encouraged from its predecessor technologies.

2.6.1.1 VIRTUALIZATION

It is an essential characteristic of cloud computing. Also one can say that it is the backbone of cloud computing. In cloud computing virtualization enables:

1. System security, as services can be isolated running on the same hardware.
2. Performance and reliability, as application migration is possible from one platform to another.
3. The development and administration of services offered by a provider.
4. Performance isolation.

2.6.1.2 MULTI-TENANCY

It is a mandatory concern in cloud computing. It allows multiple users to make use of resources concurrently. Also the users are separately charged based upon their usage. Taking an example from real life like in metro cities there are multi-storey buildings. The owner of the building provides the housing facility to all the tenants and the tenants pays him accordingly.

2.6.1.3 SECURITY

It is one of the major concerns in cloud computing. There is always some sensitive and private data which needs proper security in any system and cloud is no exception to it. In service level agreements the terms and condition of cloud service provider is mentioned clearly which ensures the security and trust to the users.

2.6.1.4 PROGRAMMING ENVIRONMENT

It is necessary to make use of cloud features. Cloud offers number of services to its clients. Therefore, programming environment for cloud computing should be able to address the issues like multiple administrative domains, resource heterogeneity, cloud federation, exception handling in highly dynamic environments, etc.

2.6.2 QUALITATIVE CHARACTERISTICS

Qualitative characteristics define the properties or qualities related to cloud computing. These are very much different from the technical requirements of cloud computing. Every service provider has different provisions for providing these qualitative characteristics to their users.

2.6.2.1 ELASTICITY

In cloud computing makes user free from the issues regarding peak-loads and sudden expansion of the infrastructure. Users can use the cloud services in real-time. Elasticity of resources is one of the attractive feature in cloud computing.

2.6.2.2 AVAILABILITY

It refers to a capability of the cloud to provide its users the guarantee of providing the services anytime anywhere through a dedicated internet connection. Users are made assured of having better quality of services 24x7.

2.6.2.3 RELIABILITY

In cloud computing systems means to provide cloud user constant and disrupted services. By having backups and redundant copy of data, the danger of data lost can be easily resolved. In cloud there is a large storage capability which makes it possible.

2.6.2.4 AGILITY

It is one of the most basic requirement for cloud computing. Cloud providers are supposed to give quick response to its users. User may demand for changes in resources required and various other factors, cloud service provider should be capable of doing all these changes on-line.

2.6.3 ECONOMIC CHARACTERISTICS

Form economical perspective one should focus on the cloud market trends. These features are the most attractive features which place cloud computing apart from other computing technologies. In a marketable upbringing, cloud market should also be made such as to make its service offerings compatible with the current business requirements.

2.6.3.1 PAY-AS-YOU-GO

It is the means to pay for the services a user use in cloud computing environment. Here, you have to pay only for the resources you use at a very reasonable price. Previously, a lot of infrastructure cost is required for establishing own business. Also if the present business need is not so high even then the owner has to make investment.

2.6.3.2 OPERATIONAL EXPENDITURE

It is highly reduced in cloud computing. Users can enter into the world of cloud easily, by having a credit card also some cloud service providers offer limited free subscription. They can rent the infrastructure for sporadic intensive computing tasks.

2.6.3.3 ENERGY-EFFICIENCY

It is high in demand these days in every domain. IT industry is moving towards energy-efficient technology. Various surveys show that in the upcoming time most of the cost is spent on to reduce the power consumption and making the IT green. Cloud computing aims at reducing the power consumption through the resources accompanied by the system. And as a result the system efficiency increases.

2.7 VIRTUALIZATION AND CLOUD COMPUTING

Virtualization technology is one of the fundamental components of cloud computing. In the delivery of IaaS it plays a significant role, therefore it is sometimes also known as hardware virtualization. Virtualization abstracts the underlying resources and simplifies their use, isolates users from one another, and supports replication which, in turn, increases the elasticity of the system. Though the concept of virtualization comes earlier than the cloud computing but in a very short span of time cloud computing has gained much popularity. As shown in Figure 2.7, Google search index for cloud computing increased at much faster rate.

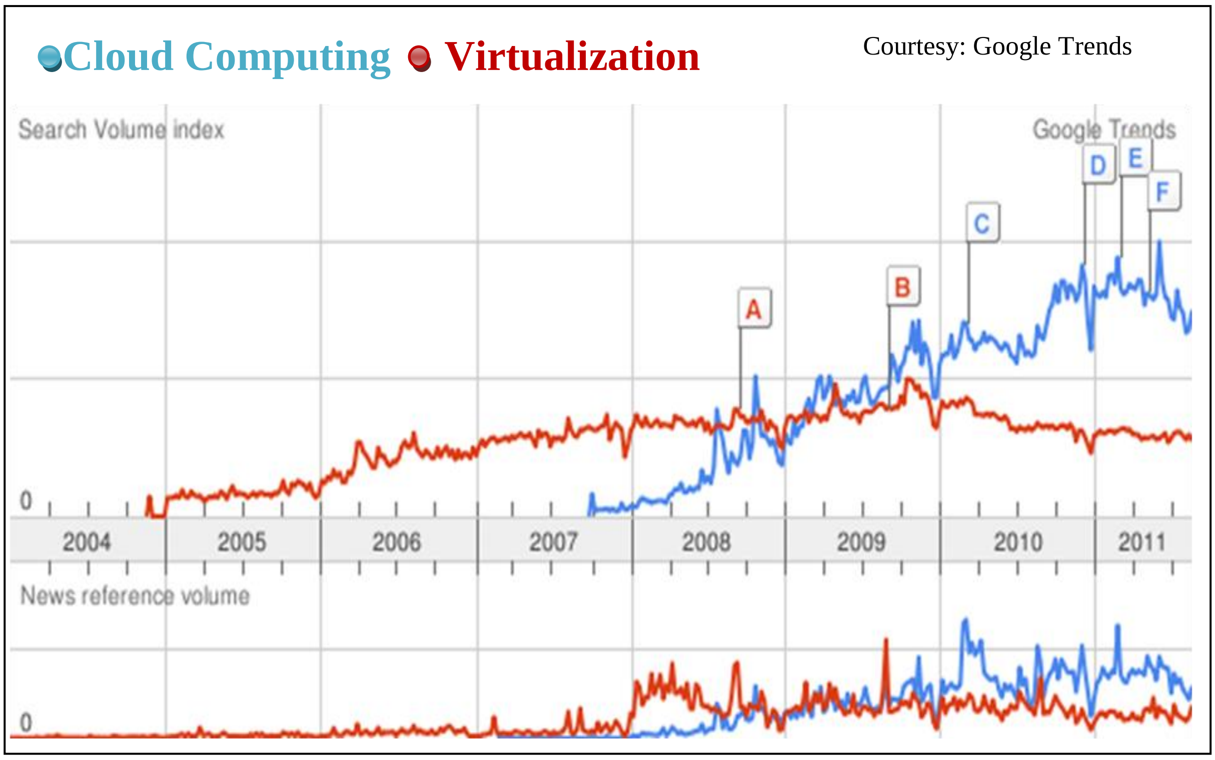


Figure.2.7: Google Trends showing the searches on Cloud computing Vs Virtualization

2.7.1 KEY CHARACTERISTICS OF VIRTUALIZATION

Virtualization technology owns some very important characteristics which makes this technology so popular. Let us discuss these characteristics in detail:

2.7.1.1 INCREASED SECURITY

By introducing a new layer of virtualization in between the guest and the host, the level of security has increased. All operations of guest are generally performed on virtual machine. Therefore, the virtual machine manager controls and filters the activity of the guest, thereby preventing some harmful operations from being performed.

2.7.1.2 MANAGED EXECUTION

A virtual machine manager is responsible for the proper management of all the tasks assigned by the guest. Each task is executed under the supervision of virtual machine manager. Sharing and aggregation among various guests is possible through virtualization.

2.7.1.3 PORTABILITY

Portability is applied differently in hardware and programming level virtualization. In hardware-level virtualization guest is packaged into a virtual image while in programming-level virtualization no recompilation is required while running different programs from various guests.

2.7.2 VIRTUALIZATION TECHNIQUES

In cloud computing there are two types of virtualization techniques:

2.7.2.1 FULL VIRTUALIZATION

In this technique of virtualization the entire system is virtualized i.e., the whole operating system is directly placed upon the virtual machine. It looks like the entire system is running on the raw hardware. In figure 2.8 the concept of full virtualization is shown.

Courtesy: [10]

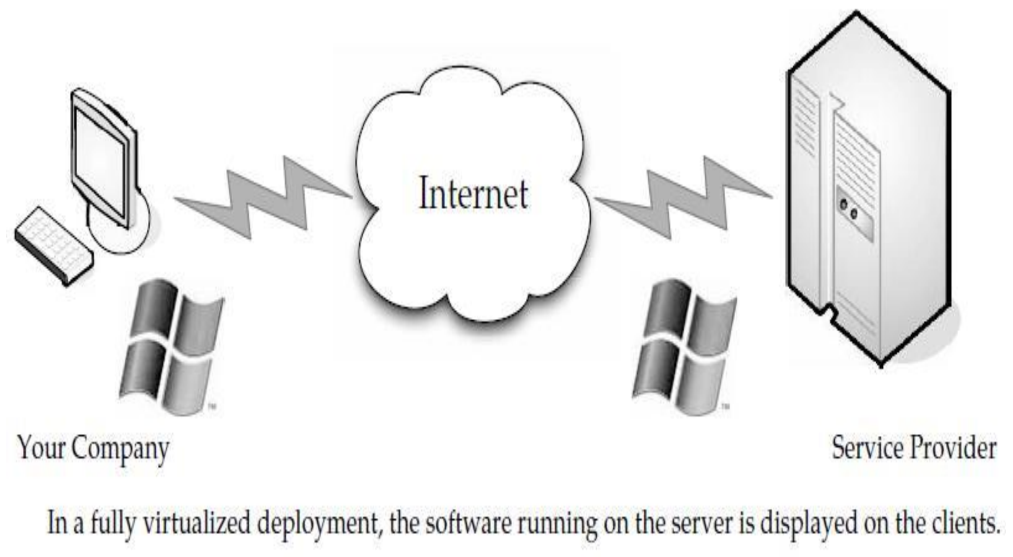

Internet
Your Company
Service Provider
In a fully virtualized deployment, the software running on the server is displayed on the clients.

Figure.2.8: Full Virtualization

2.7.2.2 PARA VIRTUALIZATION

Para virtualization provides the ability to operate performance-critical tasks directly on the host. This technique is beneficial in case of disaster recovery, migration from one system to another system and capacity management. In figure 2.9 Para virtualization is shown.

Courtesy: [10]

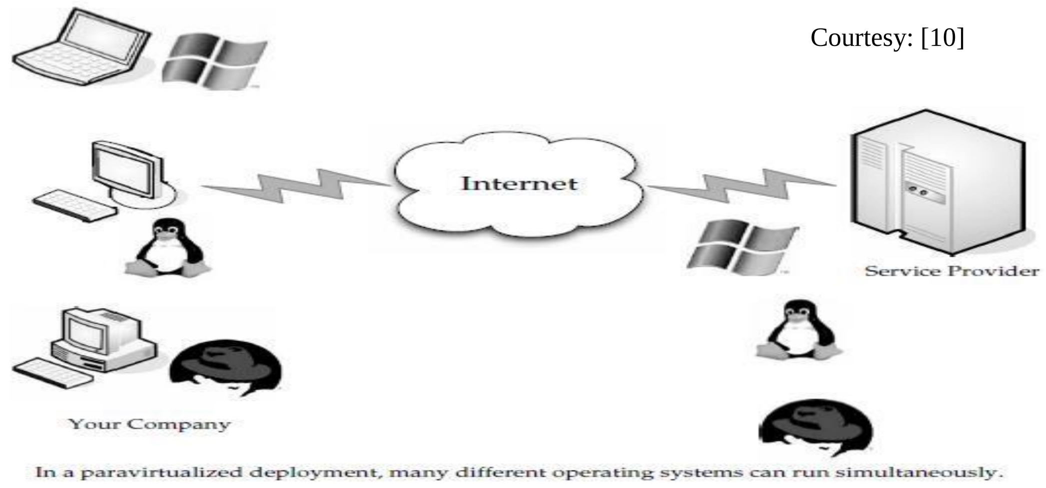

Internet
Service Provider
Your Company
In a paravirtualized deployment, many different operating systems can run simultaneously.

Figure.2.9: Para Virtualization

2.8 ISSUES IN CLOUD COMPUTING

Even though a cloud computing is an emerging technology, it is wide spreading in the industry and gaining lots of attention from business organization, academia and research industry. But the technology is facing many challenges in different areas, which must be resolved to make cloud computing a better technology. Some of the most challenging issues are discussed below:

2.8.1 LOAD BALANCING AND JOB SCHEDULING

In cloud computing researchers look forward to use some better techniques for job scheduling and load balancing other than using the traditional techniques. Resource management in cloud computing is one of the major issues, because in cloud there are generally huge datasets from large number of users belonging to different geographical areas.

2.8.2 LICENSE MANAGEMENT

Cloud service provider rent their resources to the cloud users with proper licensing agreement. Users will choose any cloud service provider in the cloud market, but a provider with better licensing agreement attracts the users more effectively. Also a license agreement may be used in case of any legal disputes.

2.8.3 DATA SECURITY

In cloud, users are storing their important data. This data is completely in the hands of third party so security of the data is important. Generally attackers use public cloud model, because there are some easy chances of cracking the intermediate channel and get the secured data.

2.8.4 SCALABLE STORAGE IMPLEMENTATION

Scalability is one of the important features of cloud, which offers user more convenience. As the demand of the user scales-up or scales-down the cloud service provider must be able to add-on or release the instances of resources created by the user.

2.8.5 DATA LOCK IN AND STANDARD PROTOCOL AND API DESIGN

The API's visible to different users in cloud must be compatible to their own platforms, as some users use windows, some use Linux or Mac operating systems. In the heterogeneous environment a standard API and protocol must be considered to have a better communication among different cloud users.

2.8.6 AVAILABILITY OF SERVICE

When the user is provided with some resources or services from the cloud service provider then it must be available whenever a user required these resources. A cloud service provider must make sure that no intermediate entity should interfere between the user and the server. A denial of service attack is the most common problem in the availability of resources.

2.8.7 CLOUD TO CLOUD AND FEDERATION CONCERNS

In cloud computing there are number of servers geographically distributed over the globe. So when it comes to sharing of information from one server to another or sometimes when the task is to be migrated from one server to the other in the cloud, a cloud service provider should make it confirm that there should not be any problem in sharing or migration of resources over the cloud.

2.8.8 MOBILE CLOUD

Now a day’s mobile devices are considered as a mini computer, almost all the features like accessing the internet, sharing the data, images can be easily done with these devices. A user can browse almost anything over the internet using our mobile. So cloud services and resources can also be accessed through mobile devices. Mobile cloud compatibility with other systems must be addressed carefully.

2.8.9 PERFORMANCE UNPREDICTABILITY DUE TO VIRTUALIZATION

All most all the task execution in cloud computing is done by using virtualization. Virtual machines are deployed over the nodes according to the user requirements and also the configuration of nodes matters in deploying these virtual machines. It is not possible to predict about the virtual machine deployment without knowing about the exact requirements from the user side.

2.8.10 BUG DETECTION

Last but not the least issue in cloud computing is the detection of errors or bugs. Generally, in cloud computing there are larger networks and heterogeneous resources therefore it will be difficult to detect the bug in between the system.

2.9 CLOUD PROJECTS

There are number of researchers from academia and IT industry, who are doing lots of experimentations and analyses which results into some wonderful commercial products and open research projects in the field of cloud computing. These projects are also getting support from industry giants and research organizations to promote cloud at global level. Below mentioned are some of those projects and products.

2.9.1 COMMERCIAL PRODUCTS

Commercial cloud solutions boost dramatically in the last few years and promote organization reallocation from company-owned resources to per-use service-based models. Some of the most popular cloud projects are Amazon Web Service, Eucalyptus, FlexiScale, Joyent, Microsft Azure, Engine Yard, Heroku, Force.com, RightScale, Netsuite, Google Apps, etc.

Amazon EC2 [6] allows users to instantiate a virtual machine, named instance, through an Amazon Machine Image. An instance functions as a virtual private server that contains desired software and hardware. Roughly, instances are classified into 6 categories: standard, micro, high-memory, high-CPU, cluster-GPU and cluster compute, each of which is subdivided by the different memory, number of virtual core, storage, platform, I/O performance and API.

Amazon S3 [7] is meant to provide user with huge, durable data storage and retrieval capabilities over the internet. The computing tasks requiring large storage can be benefitted from this product. This makes the services more scalable in cloud.

Google App Engine [8], released in 2008, owned by Google, is a platform for developing and hosting web applications in multiple servers and data centers. This product mainly provides PaaS, to the users and has some limitation in terms of run-time environment (Python and Java). IaaS, GAP makes it easy to develop scalable applications, but can only run a limited range of applications designed for that infrastructure.

MapReduce [9] is one of the best known programming models introduced by Google. MapReduce supports distributed computing on large clusters. It includes basically two operations: map and reduction. These operations run in parallel. The advantage of MapReduce is that it can efficiently handle large datasets on common servers. Also quick recovery of datasets is possible in case of partial failure.

2.9.2 RESEARCH PROJECTS

Other than the company initiatives there are number of academic projects have been developed. These projects aim at addressing the challenges like stability of test bed, standardization and other implementation issues. Some of the active projects include XtreemOS, OpenNebula, FutureGrid, elasticLM, gCube, ManuCloud, RESERVOIR, SLA@SOI, Contrail, ECEE, NEON, VMware, Tycoon, DIET, BEinGRID, etc.

XtreemOS [10] was started in 2006 by INRIA. It is an open source distributed operation system for grids. XtreemOS is an uniform computing platform integrating various heterogeneous infrastructures, from mobile device to clusters. It provides three services including application execution management, data management and virtual organization management. Even though XtreemOS was originally designed for grids, but it is able to support resource sharing and cloud federation in cloud computing environment.

OpenNebula [11] is an open source project established by Complutense University of Madrid in 2005. Its first software release came in 2008. This project aims at managing datacenter’s virtual infrastructure to build IaaS clouds. It has the capabilities for management of user, virtual network, multi-tier services, and physical infrastructures.

2.10 SUMMARY

In this chapter, we have first discussed the concept of cloud computing. There are number of definitions on cloud computing. We discussed few which are more popular. We have tried to define cloud computing in a more refined manner, so as to generalize the definition of cloud computing. Not only the definition we have discussed the cloud evolution, it's brief history, cloud architecture, various cloud service models and cloud deployment models. Cloud computing is still an evolving paradigm. It integrates many existing technologies. A user can purchase software, infrastructure, platform etc, therefore we can say that cloud computing is a service provisioning model. We analyzed various cloud commercial as well research projects. We discussed number of challenges in cloud computing. These challenges motivate our interest in future research. Load balancing is one such critical issue which we studied in depth in our next chapters.

CHAPTER 3

Load Balancing

Cloud Computing is the most recent emerging standard promising to turn the vision of "computing utilities" into reality, it provides a flexible and easy way to store and retrieve huge data without worrying about the hardware needed. In the previous chapter various aspects of cloud computing was discussed in detail. There are number of issues which must be dealt carefully in order to have an efficient system. Load balancing is one such critical issue in cloud computing. This chapter presents an outline of load balancing, which is the main concern of this thesis.

3.1 INTRODUCTION

As the number of users on cloud increases, the existing resources decreases automatically which leads to the problem of delay between the users and the cloud service providers. Thus, the need of load balancing comes into picture. The traffic over the network must be dealt smartly such that the situation in which some nodes are overloaded and some other are under loaded should never arise. To overcome this situation, many load balancing algorithms are proposed by researchers, with their own pros and cons. Load balancing plays important role in improving the performance and efficiency of the cloud computing systems. At the time of developing a cloud, special attention must be given towards the global load balancing policy of the cloud. Cloud is a container of large number of resources. Load balancing is not a new challenge, earlier in distributed computing or grid computing some efficient load balancers are used to improve the performance of the system. In cloud computing, the idea of load balancing is same but special attention is given to the energy consumption and cost estimation. There has been extensive research going on this field.

3.2 LOAD BALANCING DEFINED

A computer network consists of various computing resources such as nodes (or servers), processors, disks, network links, data, computers etc. When the number of users increases among the network, the workload over the resources also increases abruptly. Sometimes it becomes difficult to manage the resources in the network, which results in poor performance of the system. There comes the need of load balancing. A good load balancing mechanism can effectively improve the resource utilization, throughput, response time and overall efficiency of the system. In simpler terms one can define load balancing as a mechanism for equal distribution of the total workload among all the resources present in the network.

It happens many a times in the network that some resources are given excessive amount of the computing tasks while some resources are just idle or have relatively less number of computing tasks. This scenario leads to a situation in which the service provider as well as the end user will face difficulties like more response time, inefficient utilization of resources, low fault tolerance, bad throughput, etc. As an example look upon the social networking sites, which have millions of users over the globe. It will become difficult to manage the requests from millions of users, at a time from different geographical regions. Therefore, the developers' use various load balancing schemes to tackle such situations, one such schemes is geographical replication of data.

3.3 GOALS OF LOAD BALANCING

Various organizations and IT companies with their self-owned networking systems, implement their own load balancing policies in their computing network. According to their requirements and performance related issues, they set some technical and business goals for their load balancing policy. Let us discuss these technical and business goals of load balancing in detail:

3.3.1 TECHNICAL GOALS

Technical goals of load balancing mainly deals with problems related to the computing network i.e. all the technical problems regarding the computing network. Form technical perspective goals of load balancing are listed below:

- Improving the performance of the computing system.

- To be able to deal with the situation in which there may be a chance of the system failure.
- A load balancer must make sure that the system should be stable throughout the computation work in the network.
- To be able to adopt almost all the future modifications in the system, it may be extending the resources in the network or it may be boosting up the capacity of already existing resources.
- A load balancer must be capable of ensuring the availability of the services or resources whenever required by the user.

3.3.2 BUSINESS GOALS

Apart from the technical aspects, organizations have some business related policies. Every organization wants to be beat its competitors in the market, therefore to achieve better results proper resource management is necessary. From business perspective goals of load balancing are listed below:

- Minimizing the cost of the total operations performed via the cloud computing network.
- Compatibility with all rules and regulations of the industry, like managing the service level agreements, licensing agreement etc.
- To take into account various parameters like geographical location, peak hours of the day, while servicing the user requests.

3.4 LOAD BALANCING IN CLOUD COMPUTING ENVIRONMENT

In cloud computing environment, load balancing is done to achieve the better results in the computing network in terms of resource utilization, throughput and response time. Also the important aim is to minimize the computation cost and reduce the energy consumption in the computing network. An effective load balancing policy enhances the resource management mechanisms as well as the task scheduling mechanism of the computing system. There are three main concerned entities in cloud namely a cloud developer, a cloud service provider and a cloud user. A cloud developer is responsible for designing, developing and deploying the cloud. A cloud user is simply the person or a business group who are using the cloud services as per their own requirements, and a cloud service provider is the middle entity working between the cloud developer and the cloud user, responsible for providing the services to its users.

In a cloud computing network there are datacenters and these datacenters comprise of nodes (or servers). Each node again contains number of processing elements. A user tasks are assigned to various virtual machines. These virtual machines are again allocated to different nodes according to some allocation policy. Figure.3.1 represents the hierarchy of entities and the assignment of these entities to one another in cloud computing network. Various cloud simulation tools like CloudSim [13] and CloudAnalyst [14] use this hierarchy to simulate various cloud scenarios and verify the effectiveness of various load balancing and scheduling algorithms in cloud computing environment.

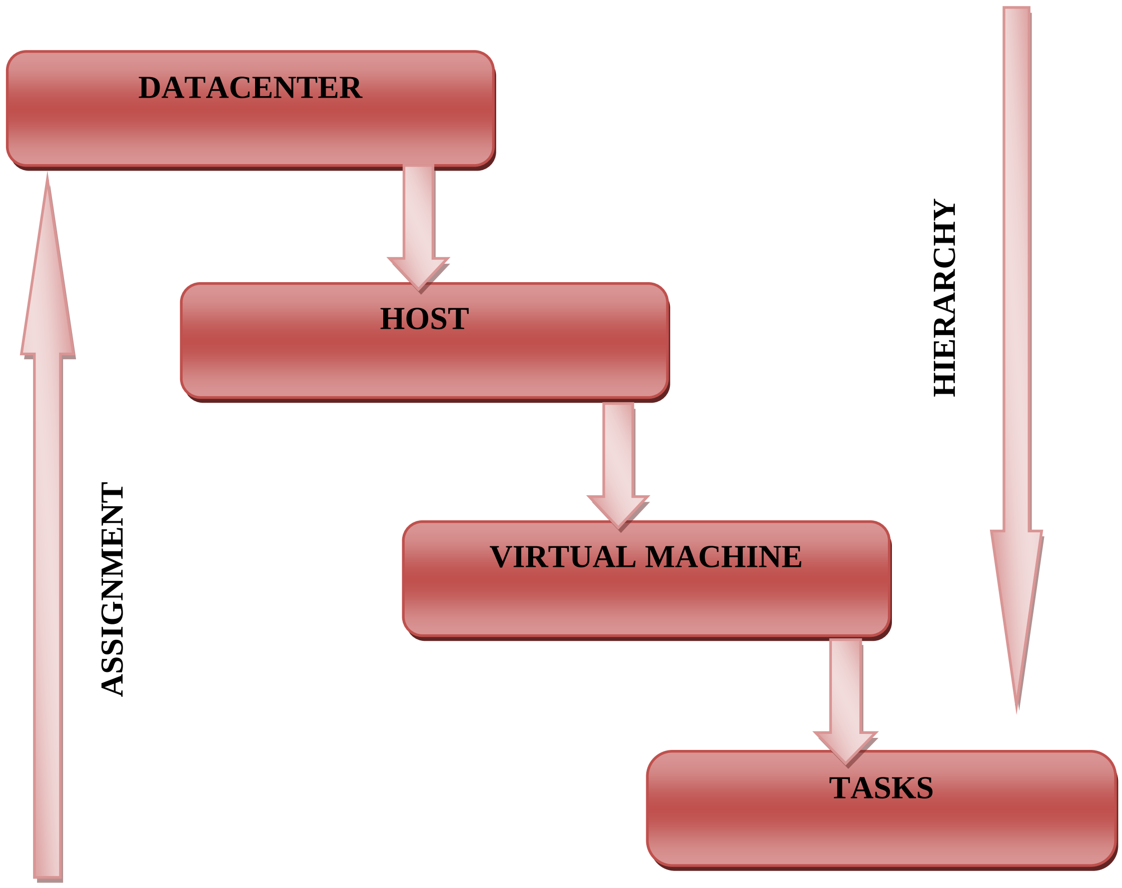


Figure 3.1: Hierarchy and assignment of cloud components

Resource provisioning and task scheduling are two main mechanisms which are together responsible for proper load balancing in cloud computing environment. It is better to work with both the mechanisms simultaneously to achieve proper load balancing within the computing systems. Resource provisioning in cloud is mainly defined as the allocation or mapping of resource to other entity, this entity could be another resource or a user task. Resource provisioning in cloud can be done at two levels: host level or VM (virtual machine). Let us discuss the resource provisioning in detail:

3.4.1 AT HOST LEVEL : ALLOCATING THE HOST TO VM

At this level virtual machines are allocated to the host i.e. physical servers in a datacenter. There is many to one relationship between the virtual machines and the host. More than one virtual machine can be instantiated depending to the user requirements. After instantiation the virtual machines are mapped to the host as per some allocation policy.

3.4.2 AT VM LEVEL : ALLOCATING VM TO USER TASKS

At this level different user tasks are mapped onto virtual machines which are instantiated by the user. This is a many to many relationship i.e. more than one user tasks can be mapped onto the same virtual machine as per the virtual machines and user tasks specifications.

Figure 3.2 presents a framework under which various load balancing algorithms works in cloud computing environment. A framework presents a flow under which the whole process from sending the request to receiving it and then executing it. At first the user cloudlets or tasks are send to the datacenter controller entity. This entity is the first interface between the user and the cloud environment. Afterwards, the datacenter controller entity sends the requests to the load balancer. A load balancer is responsible for all sorts of resource provisioning and the task scheduling activities.

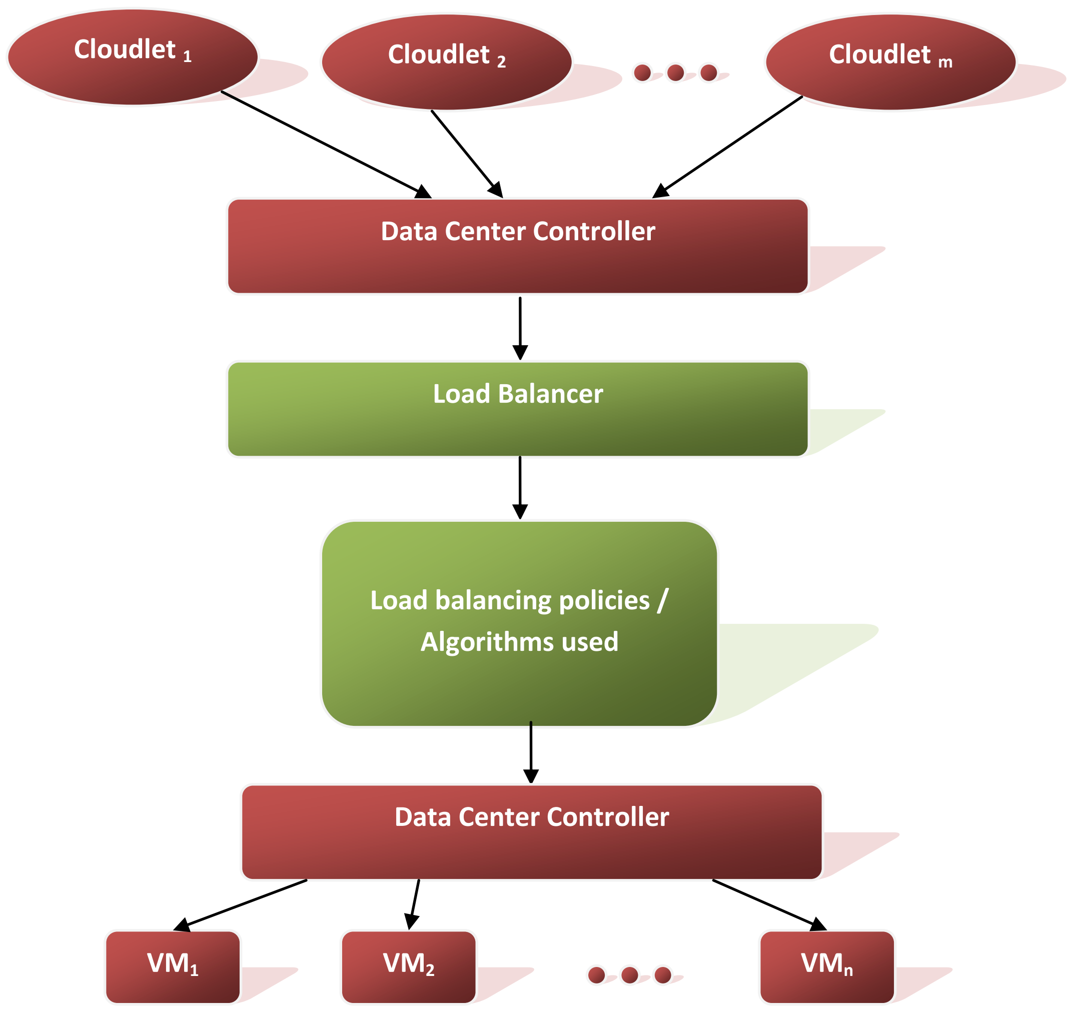


Figure 3.2: Execution of load balancing algorithm

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

There are number of load balancing techniques which are in use since the era of grid computing and distributed computing. In cloud computing too one can use these load balancing techniques to improve the performance of the system. Based on different factors like system topology, system configuration various load balancing techniques are proposed. Basically these load balancing schemes are categorized as shown in Figure 3.3.

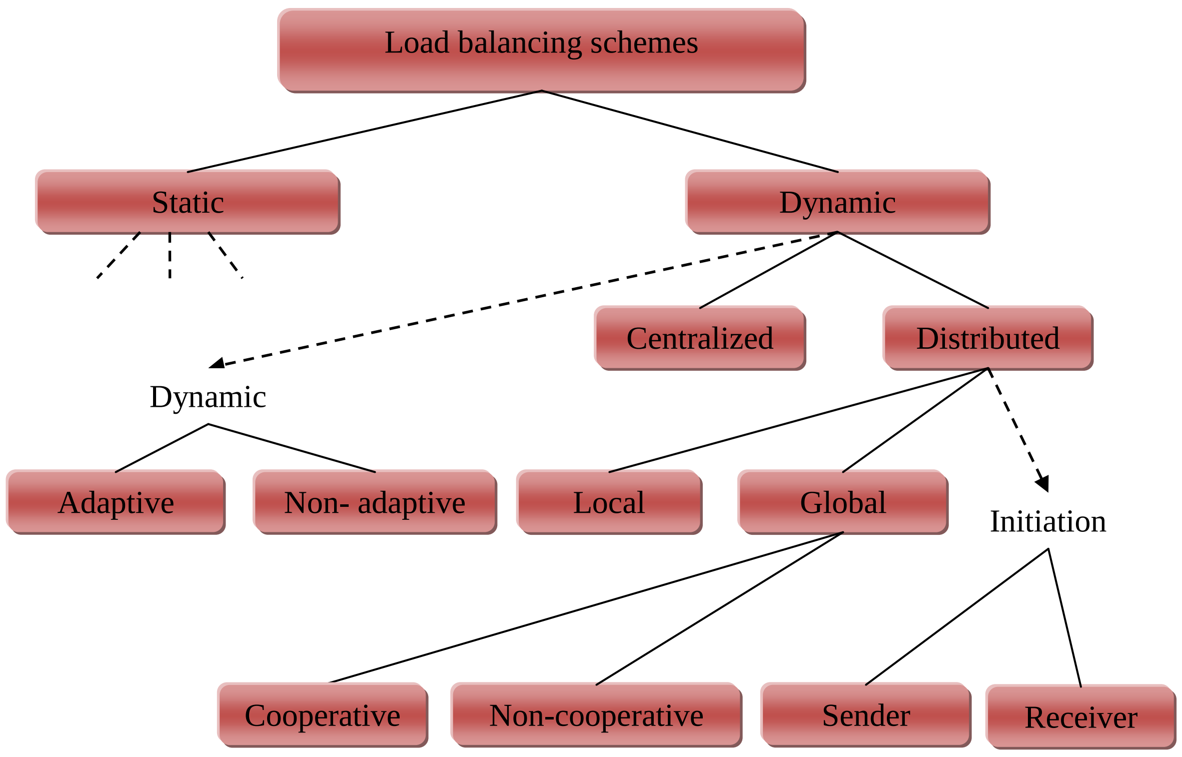


Figure 3.3: Load balancing schemes

Broadly, on the basis of system configuration, there are two categories of load balancing techniques namely static and dynamic. In cloud generally the dynamic techniques are used as in cloud the processing of the requests happens dynamically. Let us discuss in detail these load balancing techniques:

3.5.1 STATIC LOAD BALANCING TECHNIQUES

These techniques as the name defines require prior knowledge of the system. Static load balancing techniques are also known as deterministic techniques as the node upon which the next request is assigned, is already decided. These techniques can’t handle the cases in which some sudden change occurs i.e. these techniques are not suitable for the dynamic environment. Figure 3.4 shows the general outline of the static load balancers.

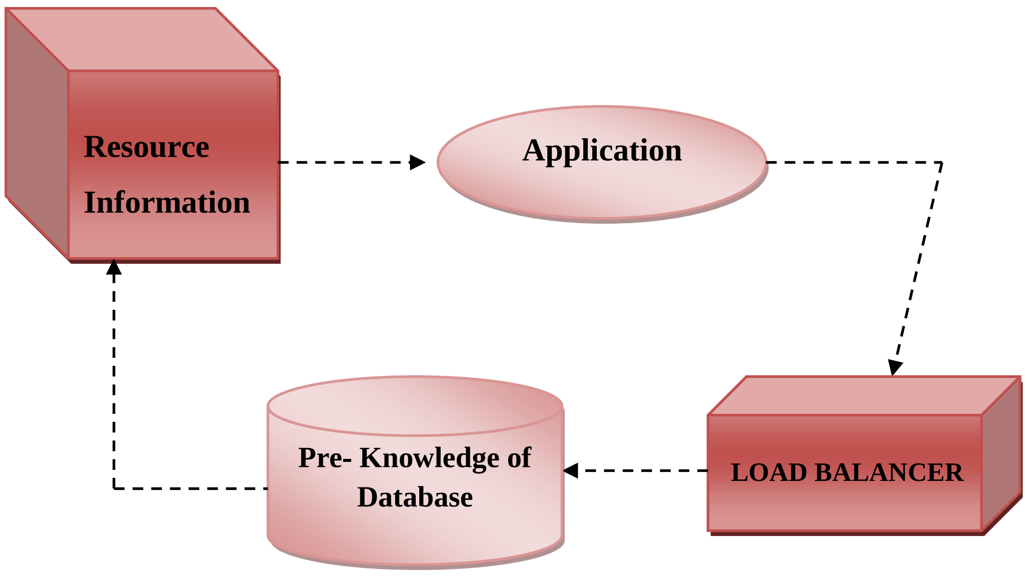


Figure.3.4: Static Load Balancer

As an example let us discuss few static load balancing techniques in cloud computing environment:

- **Round Robin (RR) technique:** As the name suggests Round Robin technique, handles the requests of the clients in a circular manner. It servers every user request on a first come first serve basis. The complexity of this algorithm is less as compared to the other load balancing algorithms.
- **CLBDM (Central Load Balancing Decision Model)**: In [15] author has proposed an improved algorithm over round robin called CLBDM (Central Load Balancing Decision Model). This model uses the basis of round robin. But additionally the algorithm finds out the time duration for establishing connection between the user and the node. The time duration is estimated by calculating overall execution time of task on given cloud resource.

3.5.2 DYNAMIC LOAD BALANCING TECHNIQUES

These are techniques in which the system parameters are not pre-defined. These techniques are widely used in the present condition as in cloud the information processing is generally done in a dynamic environment. Any change in the system is adaptable at run-time. These techniques are preferred over the static one. There are various other categories of load balancing techniques defined within the dynamic techniques. Let us discuss the various categories under the dynamic configuration. Figure 3.5 shows the general outline of the dynamic load balancers.

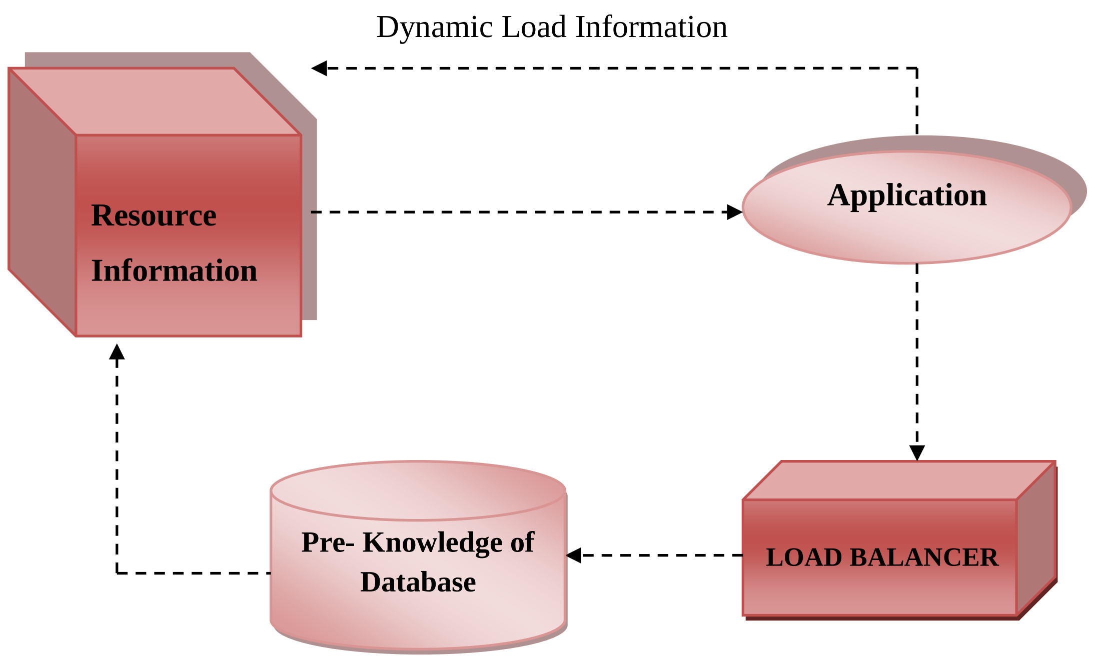


Figure.3.5. Dynamic Load Balancer

Depending upon the system topology there are two types of load balancing techniques: centralized and distributed. In centralized load balancing techniques the decision regarding resource provisioning or scheduling is made at one master or central node, whereas in distributed techniques the decision is made by more than one node distributed in certain topology. As an example let us discuss few dynamic load balancing techniques in cloud computing environment:

- **Equal Spread Current Execution (ESCE) Policy:** This policy equally spreads the current execution load among all the available virtual machines equally. The Load Balancer maintains a record of all virtual machines and the number of allocations assigned to each virtual machine. When a new request comes, Load Balancer searches the least loaded VM.

- **Throttled Policy:** Throttled policy, assigns each VM only one job at one time and other jobs are assigned only when the current job has completed successfully. The job manager maintains a list of all virtual machines. By using this indexed list allocation of virtual machine to the appropriate job is done. If the job is well suited for a particular machine than that job is, assign to the appropriate machine, else job manager waits for the client request and takes the job in queue for fast processing.

3.6 ISSUES IN LOAD BALANCING IN CLOUD COMPUTING SYSTEMS

There are various issues while dealing with load balancing in cloud computing environment. Each load balancing algorithm must be such as to achieve the desired goal. Some algorithms aims at achieving higher throughput, some aims at achieving minimum response time, some other aims to achieve maximum resource utilization while some aims at achieving a trade-off between all these metrics.

Some major issues which must be considered while designing any load balancing algorithm are as follows:

3.6.1 GEOGRAPHICAL DISTRIBUTION OF NODES

The geographical distribution of nodes matters a lot in the overall performance of any real time cloud computing systems, especially in case of large scaled applications like Facebook, Twitter, etc. A well distributed system of nodes in cloud environment is helpful in handling fault tolerance and maintaining efficiency of the system.

3.6.2 ALGORITHM COMPLEXITY

The complexity of any load balancing algorithm affects the overall performance of the system. Sometimes the algorithm is complex, but is better in terms of throughput and resource utilization. On the other hand, the algorithms which are simpler in terms of complexity may give poor performance in terms of fault tolerance, migration time and response time. Therefore, based on the system requirements, care should be taken to decide a better or suitable load balancing algorithm. A trade-off between all the parameters must be set wisely.

3.6.3 DYNAMIC VS STATIC BEHAVIOR OF LOAD BALANCING ALGORITHM

Any algorithm on load balancing is designed, based on the state or behavior of the system, which may be static or dynamic approach. Static algorithms do not depend upon the current state of the system and have prior knowledge regarding system resources, details of tasks in an application. These kinds of algorithms face a major drawback in case of sudden failure of system resource and tasks. On the other hand dynamic algorithms take decision about load balancing based upon the current state of the system and don't need any prior knowledge about the system. This approach is an improvement over the static approach. The algorithms in this category are considered to be complex but have good fault tolerance and overall performance.

3.6.4 TRAFFIC ANALYSES OVER DIFFERENT GEOGRAPHICAL LOCATION

For any load balancing algorithm, it is very important to analyze the traffic flow in real time scenarios over different geographic regions, and then balance the overall workload accordingly. All regions over the globe have different time zones and have certain peak hours during which the network load is supposed to be at its peak. Therefore load balancer must be capable of handling the traffic in peak hours in every location so as to achieve maximum resource utilization and throughput.

3.6.5 STORAGE AND REPLICATION OF DATA IN CLOUD

The task of storage of data is also critical. The data is stored and replicated at some other node. If some failure occurs at one node then the lost data can be retrieved from the node having replicated copy of the lost data. Also sometimes it helps in fast processing of the user requests. Due to its importance many commercial firms have initiated the concept of geographical replication, in which the data is stored at various places over the globe depending upon various factors like number of users, time zones, usability, etc.

3.7 PERFORMANCE METRICS USED TO EVALUATE THE LOAD BALANCING ALGORITHMS

The metrics on which the existing load balancing techniques have been measured are discussed below:

3.7.1 THROUGHPUT

This metric is used to estimate the total number of tasks, whose execution has been completed successfully. High throughput is necessary for overall system performance.

3.7.2 OVERHEAD

Overhead associated with any load balancing algorithm indicates extra cost involved in implementing the algorithm. It may include various factors like the migration time, maintenance cost of the nodes, etc. It should be as low as possible.

3.7.3 FAULT TOLERANCE

It measures the capability of an algorithm to perform uniform load balancing in case of any failure. A point of failure can make your system inefficient. It generally occurs when the system goes through heavy traffic. A good load balancing algorithm must be highly fault tolerable.

3.7.4 MIGRATION TIME

It is defined as, the total time required in migrating the jobs or resources from one node to another. It should be minimized.

3.7.5 RESPONSE TIME

It can be measured as, the time interval between sending a request and receiving a request. It should be minimized, to boost the overall performance.

3.7.6 RESOURCE UTILIZATION

It is used to ensure the proper utilization of all those resources, which comprised the whole system. This factor must be optimized to have an efficient load balancing algorithm.

3.7.7 SCALABILITY

It is the ability of an algorithm to perform uniform load balancing in a system with the increase in the number of nodes, according to the requirements. Algorithm with higher scalability is preferred.

3.7.8 PERFORMANCE

It is used to check, how efficient the system is. This has to be improved at a reasonable cost, e.g., reducing the response time though keeping the acceptable delays.

3.8 SUMMARY

Summarizing this chapter, it was found that we have studied various aspects of load balancing in general as well as from the perspective of cloud computing. We studied how load balancing can be done in cloud computing, it's technical as well as business goals in cloud computing. Also we discussed the different categories of load balancing techniques which helps in analyzing the various pros and cons related to each of the techniques and also help in deciding the suitable load balancing techniques for your own organization's cloud computing network. We also studied different challenging issues in the load balancing. Lastly we discussed different critical parameters in judging the performance of the load balancing techniques.

CHAPTER 4

Related Work

Cloud Computing is the recent emergence of advanced technology in the IT industry leading towards opening of several research avenues in the domain. In this chapter, a discussion on some significant contributions on load balancing in cloud computing was done.

A.Khiyaita ., et al., [16] in their paper gave an overview of load balancing in cloud computing. They have classified load balancing algorithms based on system load and system topology. Different examples in the paper explained the implementation of load balancing algorithms in classical distributed systems. They specified some key technologies in cloud computing system. The authors have stated various research challenges in load balancing. Nidhi Jain Kansal ., et al ., [17] discuss about the existing techniques aimed at reducing associated overhead, service response time and improving performance of the technique. The paper provides details about various parameters used to compare the existing techniques. Each parameter plays an important role in governing the overall performance of the system.

To perform load balancing in cloud computing system, Shu-Ching Wang., et al ., [18] proposed a scheduling algorithm, combining the capabilities of both OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms which is an improvement over Min-Min scheduling algorithm. OLB intends to keep each node busy regardless of the current workload of each node. OLB assigns tasks to available nodes in random order and Minimum Completion Time algorithm (MCT) assigns a task to the node that has the expected minimum completion time over other nodes. A more efficient load balancing algorithm LB3M [19], was proposed by Che-Lun Hung, Hsiao-hsi Wang and Yu-Chen Hu. In [20], idea is to find the best cloud resource by using Co-operative Power aware Scheduled Load Balancing. It has proved to be a good solution to the load balancing challenge in terms of energy efficiency. The algorithm proposed in this paper, utilizes the benefits of both distributed and centralized approach of computing. Inherent efficiency of centralized approach as well as the energy efficiency and fault-tolerant nature of distributed approach is well utilized in this algorithm.

In PALB [21] approach, utilization percentages of each compute node is estimated. This utilization percent helps in deciding the number of compute nodes which must keep operating while other node completely shuts down. The algorithm has three sections: Balancing section determines on the basis of utilization percentages where virtual machines will be instantiated. Upscale section power-on’s the additional compute nodes. Downscale section shut-downs the idle compute nodes. This algorithm is guaranteed to decrease the overall power consumption while maintaining the availability of resources. Raul´ Alonso-Calvo, et al., [22] have discussed about managing large image collections in real world applications. In this paper a cloud computing service and its application for storage and analysis of very large images have been created. Further, data operations are adapted for working in a distributed mode by using different sub-images. These sub-images can be stored and processed separately by different agents in the system. This method facilitates the processing of very-large images in a parallel manner. Availability of resources in cloud environment, as well as other factors like scaling of resources and power consumption, in load balancing in cloud computing environment is one of the important concerns that needs a great attention. Load balancing techniques should be such that to obtain measurable improvements in resource utilization and availability of cloud computing environment [23].

Alexandru Iosup ., et al .,[24] analyzed the performance of cloud computing services for scientific computing workloads. They performed experiment on real scientific computing workloads of Many-Task Computing (MTC) users. MTC users employ loosely coupled applications comprising many tasks to achieve their scientific goals. They have done an empirical evaluation of the performance of four commercial cloud computing services. Srinivas Sethi ., et al ., [25] proposed a load balancing algorithm using fuzzy logic in cloud computing, this algorithm uses two parameters processor speed and assigned load of Virtual Machine to balance the overall load in cloud computing environment, through fuzzy logic though in [26], the authors have introduced a new fuzzy logic based dynamic load balancing algorithm with additional parameters- memory usage, bandwidth usage, disk space usage and virtual machine status and named it as Fuzzy Active Monitoring Load Balancer (FAMLB).

Milan E. Sokile [27], have discussed about different load balancing techniques in distributed environment namely diffusive load, static, round robin and shortest queue in different client environments. Experimental analysis have been done showing diffusive load balancing is more efficient than static and round robin load balancing in dynamic environment. Ankush P.Deshmukh and Prof. Kumarswamy Pamu [28], discussed about different load balancing strategies, algorithms and methods. The research also shows that the dynamic load balancing is more efficient than other static load balancing techniques. Efficient load balancing obviously provides major performance benefits.

A Network Processor consists of a number of on-chip processors to carry out packet level parallel processing operations [29]. This parallel processing ensures fine load balancing among the processors by increasing throughput. However, such type of multiprocessing also gives rise to increased out-of-order departure of processed packets. In this paper, Ordered Round Robin (ORR) scheme is used to schedule packets in a heterogeneous network processor, assuming that the workload is perfectly divisible. The processed loads from the processors are ordered perfectly. This paper analyzes the throughput and derives expressions for the batch size, scheduling time and maximum number of schedulable processors. Jaspreet Kaur [30] has discussed active vm(virtual machine) load balancer algorithm to find suitable vm in less time period. She has done simulation showing comparative analysis of round robin and equal spread current execution policies of load balancing with varying service broker policies for data centers in cloud computing environment, for time and cost.

Zhang Bo ., et al., [31], proposed an algorithm which adds capacity to the dynamic balance mechanism for the cloud. The experiments demonstrate that the algorithm has obtained better load balancing degree and used less time in loading all tasks. Soumya Ray and Ajanta De Sarkar [32] have discussed various algorithms of load balancing like Round robin algorithm, Central queuing algorithm and Randomized algorithm, their analysis is carried out on MIPS vs. VM and MIPS vs. HOST basis. Their results demonstrate that these algorithms can possibly improve the response time in order of magnitude, with respect to number of VMs in Datacenter. Execution analysis of the simulation shows that change of MIPS will affect the response time. Increase in MIPS vs. VM decreases the response time. 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, 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. This algorithm is compared with other existing algorithms namely Greedy, Round Robin and Power-save strategies used in Eucalyptus.

CHAPTER 5

Proposed Methodology

5.1 Introduction

As discussed in previous chapters, choosing a right load balancing technique for a system is one of the critical tasks for developers. There are number of load balancing techniques suggested by various researchers, each having its own pros and cons. There are computing networks in which several applications required the computation work to be done in parallel and with large quantity while some applications only include short and quick tasks. For example scientific activities require huge computation work while a simple user login task requires quick and easy processing over the network. Some load balancing algorithms emphasize on resource utilization , some on having the minimum execution time and some emphasizing on setting a trade-off between the both the parameters. Also there are situations during which the load becomes too high and sometime the load is below average. So considering all the parameters and the need of the organization a load balancing policy is decided. With the proposed methodology main aim is to balance the overall workload among different nodes (server or host) in the network. In chapter 3 various metrics are discussed which are used to judge the performance of any load balancing technique. In this work, it has been tried to maintain a trade-off between different performance parameter so as to achieve the better results. Before moving on to the idea and the concept of the proposed load balancing algorithm let us just have a view on the basic steps of load balancing algorithm in cloud computing environment:

- At a very first step, a load balancer accepts the user requests and verifies if the load is high or low.
- Next the load balancer exchanges the information regarding load between different nodes in the datacenter.
- After that the load balancer exchanges the information regarding load between different virtual machines deployed over the nodes in the datacenter.
- If the demand rises, then a load balancer must look upon the tasks of virtual machine migration or task migration as per the load balancing technique.

An efficient load balancing can definitely boost the performance of the cloud computing network. It not only improves the utilization of resources available in the network but also increases the throughput of the system. Let us discuss the proposed load balancing algorithm in the upcoming sections.

5.2 Description

Let us first discuss about the basic terminologies used in the algorithm.

- **Datacenter:** The datacenter entity is the main entity in the cloud. This entity comprises of many hosts (or physical servers /nodes). There could be more than one datacenter in the cloud depending upon the requirement and the budget of the organization establishing the cloud.
- **Host:** A host in the datacenter is a physical server comprising many processing cores. These processing cores are then further allocated to the virtual machines. Depending upon the requirement a number of hosts can be deployed within the datacenter.
- **Virtual machine (VM):** A virtual machine is instantiated by the user as per his budget and usage. Virtual machines are allocated to different user tasks based on the configuration of the user tasks and the virtual machine.
- **Datacenter controller:** A datacenter controller entity is an intermediate entity which communicates between the cloud users and the load balancer. This entity is responsible for assigning the load to the load balancer and accepts the requests from user.

The proposed load balancing algorithm aims at distributing the total workload from the different cloud users among various nodes in the datacenter. For this a resource provisioning as discussed in chapter 3 is done at host level i.e. the virtual machines which are instantiated by the user are mapped onto the nodes or physical servers in the datacenter. A node based on its configuration allows a virtual machine to be allocated on it. A certain virtual machine allocation policy is required to assign the nodes to different virtual machines. The load balancing algorithm is illustrated step-wise below:

Algorithm: Proposed Load Balancer

Input: 1) Virtual machines < VM_1, VM_2, VM_3, …….. VM_n>

2) Hosts < H_1, H_2, H_3, …….. H_m >

where 'n' and 'm' are the number of virtual machines and hosts respectively.

Output: All virtual machines < VM_1, VM_2, VM_3, …….. VM_n > are allocated to some host < H_1, H_2, H_3, …….. H_m >.

Step.1: Proposed Load Balancer maintains:

- an index table of virtual machines,
- the status of each host (BUSY/AVAILABLE) and
- a "FLAG" containing the highest number of available processing element among all the hosts < H_1, H_2, H_3, …….. H_m >.

Initially all the hosts are available, therefore the status of each host is set to zero and the value of FLAG is set to a smallest number.

Step.2: Data Center Controller receives new request.

Step.3: Data Center Controller queries the Proposed Load Balancer for the assignment of current virtual machine to appropriate host.

Step.4: Proposed Load Balancer starts looking for the availability of host, by checking the status of each host, initially staring from H_1.

4.1: if host is available

4.1.a) Proposed Load Balancer checks for the number of processing elements in that host, saves the host id and updates the value of FLAG accordingly.

4.1.b) Continue from Step.4 for the next host, until all the hosts are traversed one by one.

4.2: else

4.2.a) Continue from Step.4 for the next host, until all the hosts are traversed one by one.

Step.5: After step.4 there are three possible cases:

Case 1: if a host is available with maximum value of flag

a. The Proposed Load Balancer returns the host id to the Data Center Controller.

b. Data Center Controller allocates the virtual machine to the host identified by that id.

c. Data Center controller notifies the Proposed Load Balancer of the new allocation.

d. Proposed Load Balancer updates the allocation table accordingly and the status of the corresponding host is set to one i.e. busy. Also the value of flag is updated.

Case 2: if all the hosts are busy

a. The Proposed Load Balancer will look for the host having maximum number of available processing elements and returns that host id to the Data Center Controller.

b. Data Center Controller allocates the virtual machine to the host identified by that id.

c. Data Center controller notifies the Proposed Load Balancer about the new allocation.

d. Proposed Load Balancer updates the allocation table accordingly.

Case 3: if all the hosts are busy and no one have the available processing elements

a. The Proposed Load Balancer returns null.

b. Data Center Controller sends virtual machine request to the waiting queue for fast processing.

Step.6: When the processing of virtual machines got finished they are destroyed and the status of the corresponding host is changed to zero again.

Step.7: If there are more virtual machine requests, Data Center Controller repeats step 4 until all the hosts get filled.

Step.8: Continue from step 2.

The proposed load balancing algorithm efficiently distributes the virtual machines onto different nodes available in the datacenter. Since the work in this thesis was done at the host level, therefore as an input to the algorithm, all the details of the hosts and the virtual machines was provided. A proposed load balancer initially maintains lists of all virtual machines and the hosts present in the datacenter. A status denotes if the host is busy or available. The status of each host is set to "zero", as it is assumed that initially all the hosts are free. Also a variable named flag is maintained which denotes the highest number of processors among the available nodes, initially it is set to a lowest value.

Now, after the initialization work, move onto the next step. In this step the user requests are submitted to the datacenter controller. And the datacenter controller queries the load balancer for the provisioning of resources and the scheduling process of the user tasks. At the host level one should only concerned about the resource provisioning of the host i.e. allocating the host to some virtual machines. Then according to the proposed load balancing algorithm at first the load balancer have to check the status of each host in the datacenter. When the status of host is available i.e. "zero" then updates the value of flag by checking the available processing element in each node. This process continues till the last node in the datacenter is traversed by the load balancer. This process is illustrated in step.4 of the algorithm.

After having the status and the flag value of each host, one could easily take decision about the provisioning of virtual machines to the host. There are actually three possible cases then.

In case-1, if the host is found available with maximum value of flag i.e. having maximum number of processing cores, then that host is chosen and allocated to the current virtual machine. The status of the host is now changed to "one", and the flag is updated according to the requirement of the processing cores of the current virtual machine.

In case-2, if all hosts are found busy then, simply look for the host with highest number of available processors in the host by examining the value of the flag for each host. In this situation choose the host with highest number of processors available. Now the status of this host is already “one” so need to update the status, so just simply update the value of flag.

In the last case-3, there is a situation in which all the hosts are found busy and there is no available processing cores in each host. In such situation the load balancer notifies the data center controller about this situation by returning null. Then data center controller put that request in the weighting queue.

After all three cases are tested for the current virtual machine, if the processing of user requests by the virtual machine got finished then the data center controller destroys the virtual machine and updates the status and the flag value of the host acquired by that virtual machine. Afterwards the data center controller looks upon the next virtual machine request in the queue and repeats the same process for all the virtual machines.

In figure 5.1 the flow of execution of the algorithm is shown clearly. First the request from the user comes to the datacenter controller. Datacenter controller entity passes on the requests to the load balancer entity. Then load balancer entity performs the load balancing at host level as well as at the VM level. In this case the load balancing algorithm works at the host level in which a suitable host is allocated for each virtual machine. In the figure a variable “curr Flag” indicates the number of available processing cores in the host. The process is shown for one virtual machine requests for the host. Also when there is no available host only then the busy hosts are considered for allocation. Whenever a new request for host allocation comes from the virtual machine the same process goes on repetition.

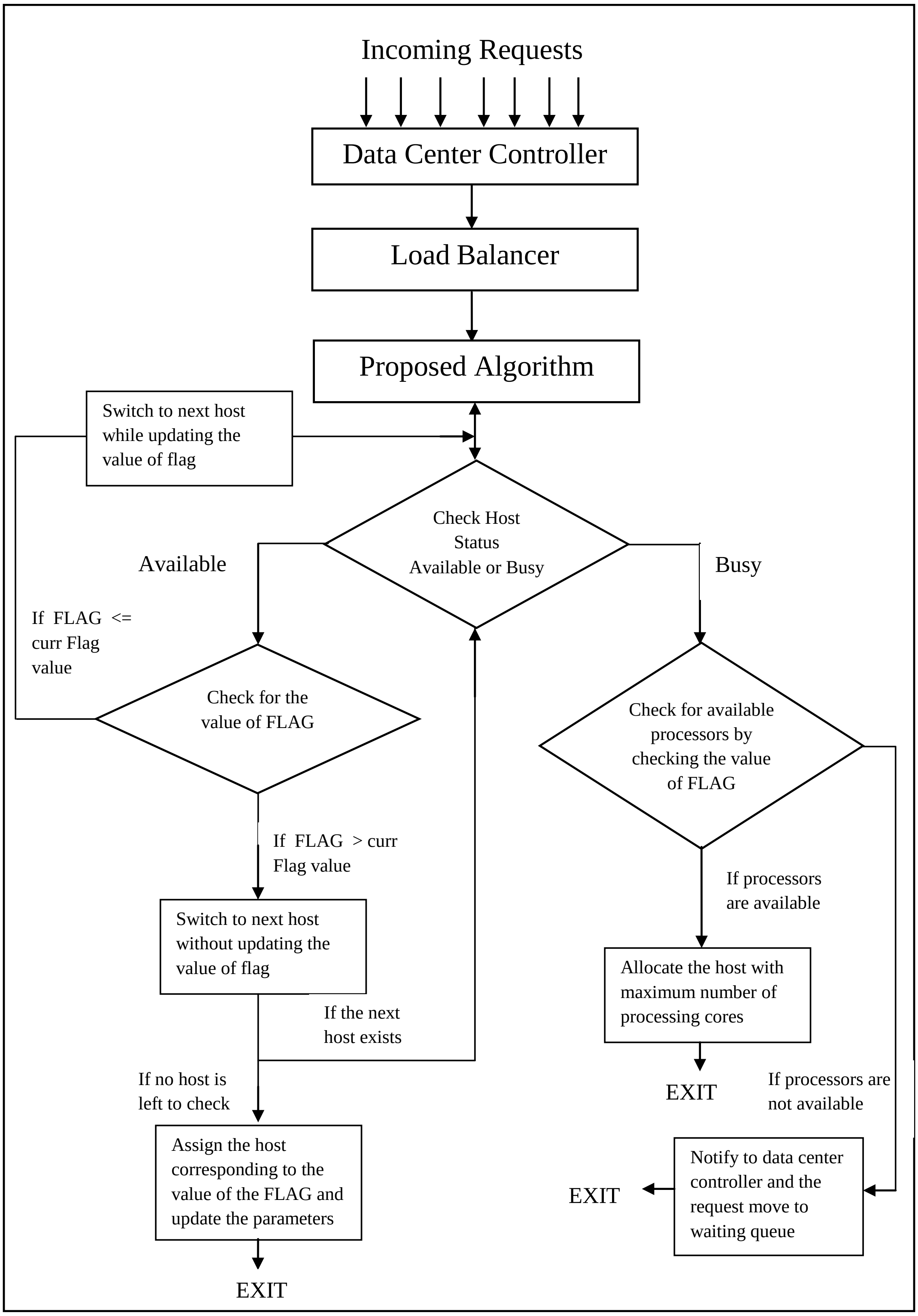

Incoming Requests
Data Center Controller
Load Balancer
Proposed Algorithm
Switch to next host while updating the value of flag
Check Host Status Available or Busy
Available
Busy
If FLAG <= curr Flag value
Check for the value of FLAG
Check for available processors by checking the value of FLAG
If FLAG > curr Flag value
If processors are available
Switch to next host without updating the value of flag
Allocate the host with maximum number of processing cores
If the next host exists
If no host is left to check
EXIT
If processors are not available
Assign the host corresponding to the value of the FLAG and update the parameters
EXIT
Notify to data center controller and the request move to waiting queue
EXIT

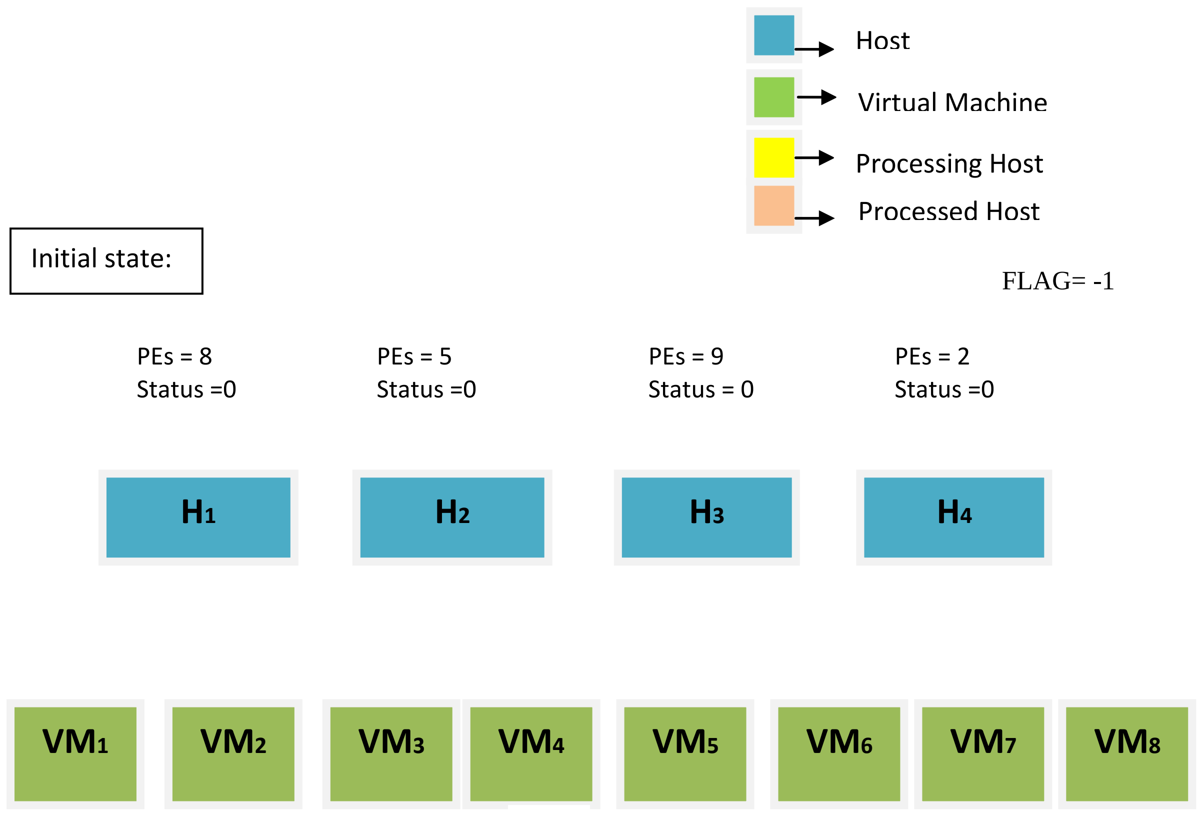
Figure 5.1: Execution flow of the proposed algorithm

5.3 Process of Operation

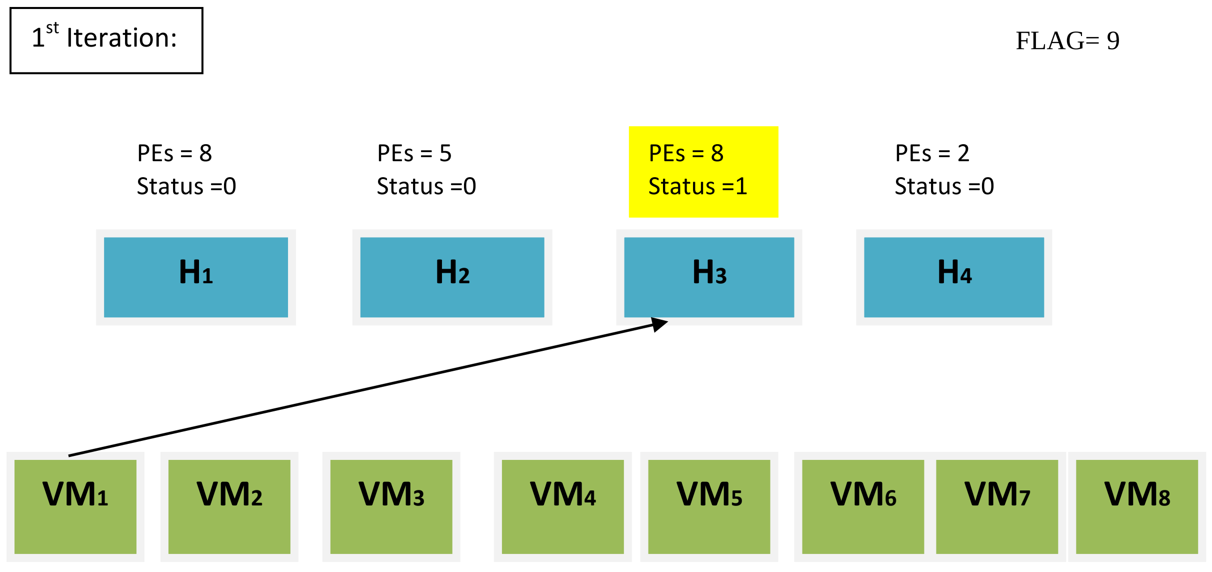
The proposed algorithm can be better understood by the following example. In the given example, consider one datacenter, four hosts and eight virtual machines for clear understanding of the proposed algorithm. The initial configuration of the system is shown in figure 5.1 a). Initially each host is available for all virtual machines therefore the status of each host is set to “0”. The number of processing elements in each host is chosen randomly, as it will provide more realistic framework for simulation. In figure 5.1 b), first iteration is shown. According to algorithm, first virtual machine finds that the third host is available and has maximum numbers of processing elements, therefore first virtual machine is allocated to third host. Number of processing element on host H_3 is deducted as per the requirement of virtual machine VM_1 and the status of host is changed to busy, denoted by “1”.

In second iteration, next virtual machine, VM_2 will search for appropriate host. Now again following the procedure of algorithm, the load balancer finds that the first host is busy so moving on to next host it finds that it is available, so now check the value of flag. From previous iteration the value of flag is 8, therefore searching for the host is such that the host is available and the flag value of that host should be maximum. In this iteration it is found that H_1 is the available host with maximum number of available processing cores. Though H_3 is also having the 8 processing cores but it is already busy. Therefore, VM_2 is assigned to H_1. After the assignment the value of status for H_1 is changed to busy and the flag value also got updated. This process is shown in Figure 5.2 c).

In the third iteration, VM_3 is searching for the appropriate host machine. Now it will look upon the available hosts and found that H_2 is available with maximum number of processing cores i.e.5. Therefore VM_3 is assigned to host number 2. The exact process for this iteration can be seen in Figure 5.2 d). The value of status is updated again.

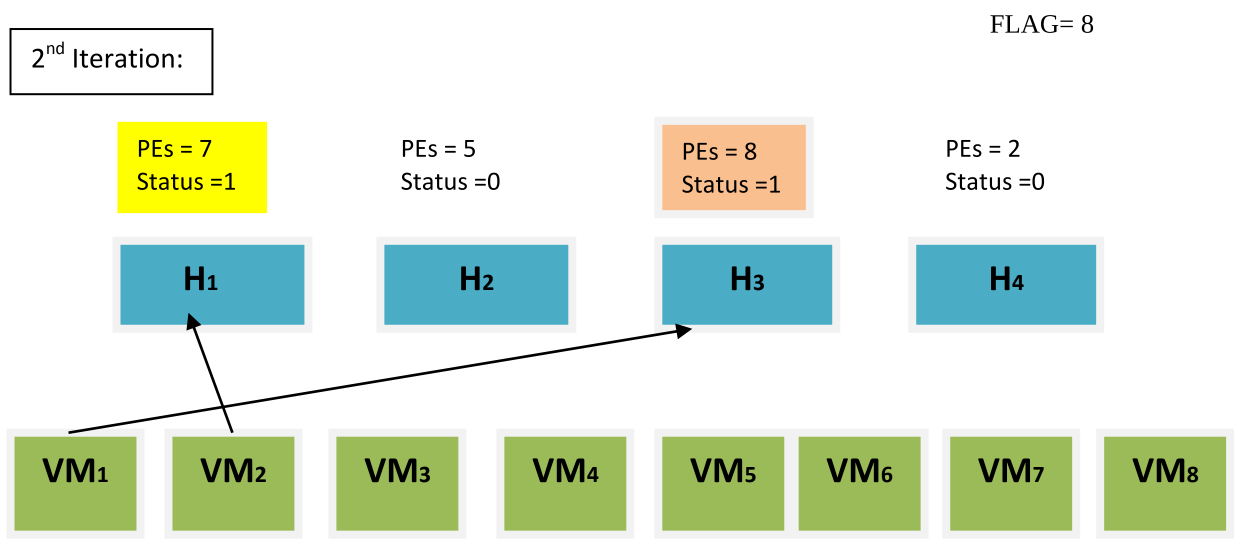


a)

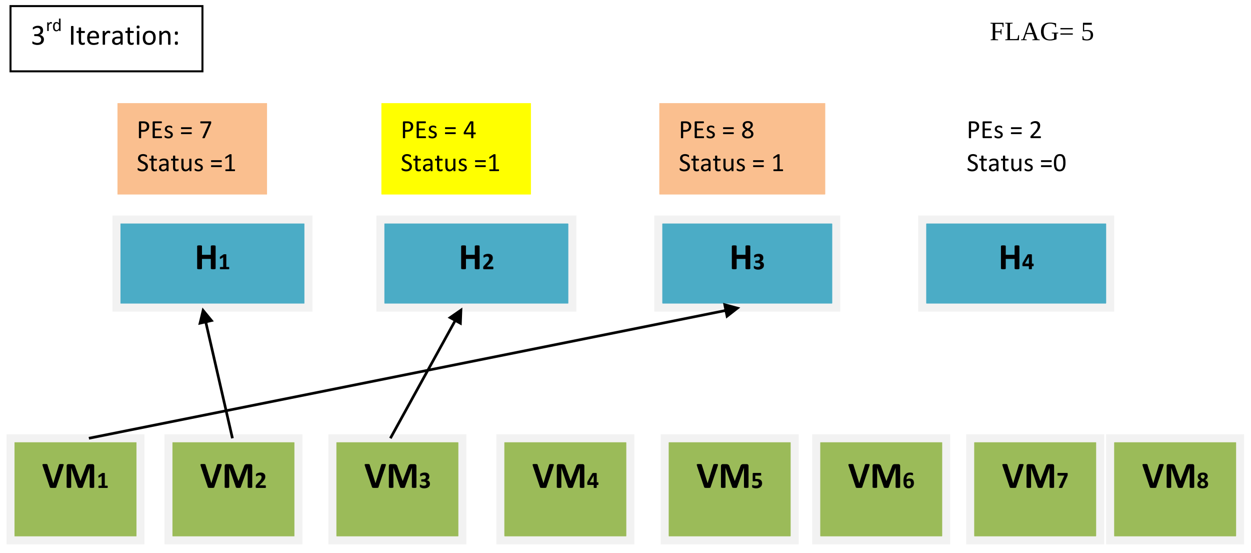


b)

Figure 5.2 Operation of algorithm: a) Initially, all Hosts are available for Virtual Machines. b) In first iteration, third host having maximum number of free processors and status "available" is allocated to first virtual machine. Now the status of host is changed to "busy" and number of processors are updated accordingly.



c)



d)

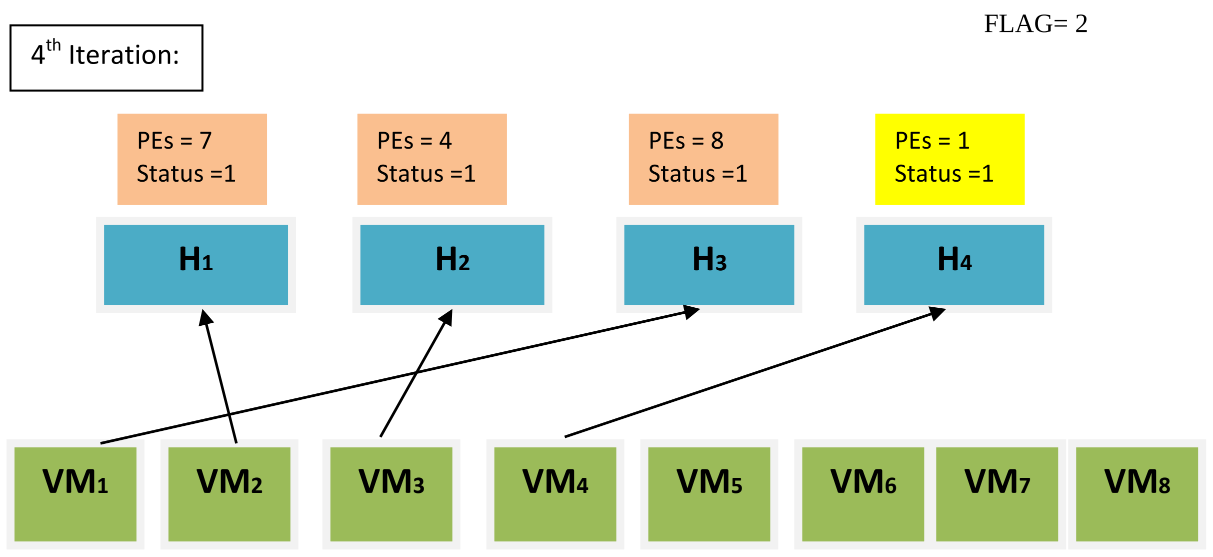
Figure 5.2, continued c) In second iteration, first host having maximum number of free processors and status "available" is allocated to second virtual machine. Now the status of host is changed to "busy" and number of processors are updated accordingly. d) In a similar fashion second host is allocated to third virtual machine.

In the next iteration i.e. fourth one, fourth virtual machine, VM_4 is looking for suitable host. Following the procedure of algorithm it is found that only left available host is H_4. Therefore the host 4 is allocated to the virtual machine VM_4. The status of all the host get updated to “one” i.e. now all the hosts are allocated some virtual machines. The process can be understood from Figure 5.2 e).

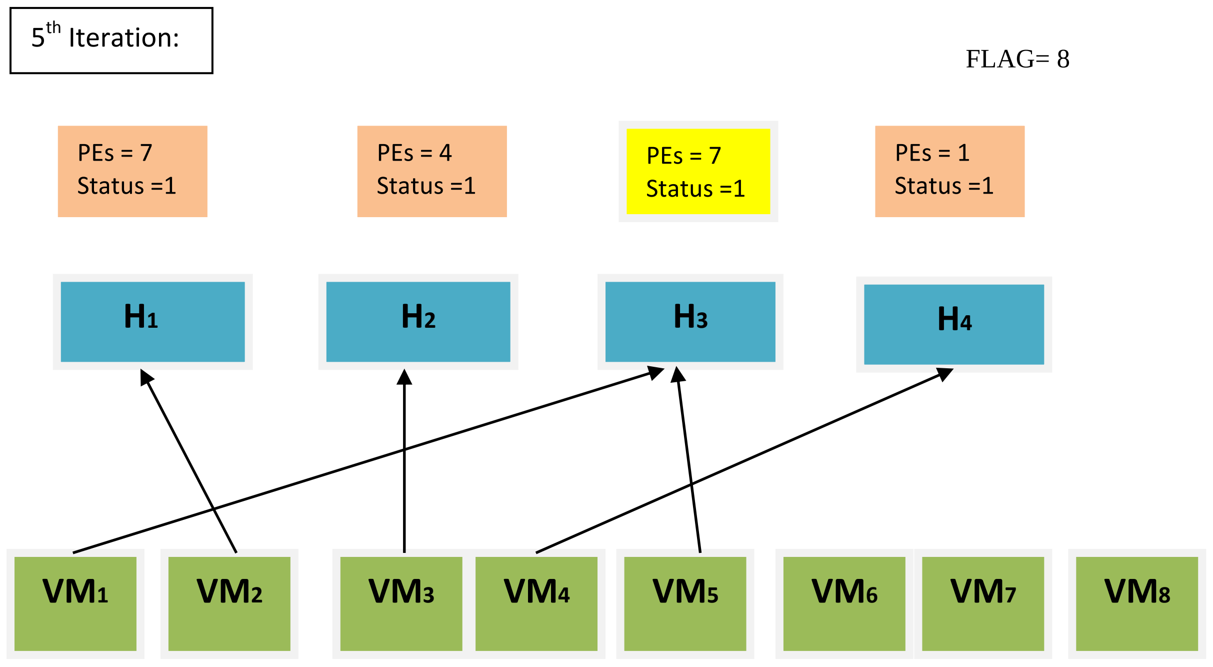
In the next iteration i.e. fifth VM_5 is looking for some suitable host. Now see that all the hosts are busy, so applying the second case of the algorithm just look for the host with maximum number of processing cores without worrying about the status of the host. So after applying this procedure it was found that third host is having the maximum number of available processors i.e. eight by verifying the value of flag. After the VM_5 is assigned to H_3 the value of flag is updated for next iteration. The process of execution for this iteration can be seen in Figure 5.2 f).

Now only three more virtual machines are left. Let us see how these are getting placed onto suitable hosts. In the sixth iteration Figure 5.2 g), VM_6 is looking for the suitable host. At this instant it is found that the status of all the hosts is “one”, i.e. all are busy with some virtual machines deployed on them. In this situation just apply case-2 of the algorithm and look for the host with maximum number of available processing cores. Therefore, here host one is found having maximum available processors seven. H_1 is allocated to VM_6.

In a similar fashion VM_7 is allocated to H_3 and VM_8 is allocated to H_1. The process is shown in Figure 5.2 h) and Figure 5.2 i) respectively. And finally all the virtual machines are assigned to some processors. In Figure 5.2 j), see the complete output of the processing which is applied initially on the taken example. It is observed that all the hosts are busy with some virtual machines. Processing their requests and doing the computation work of the user tasks assigned to the virtual machines.

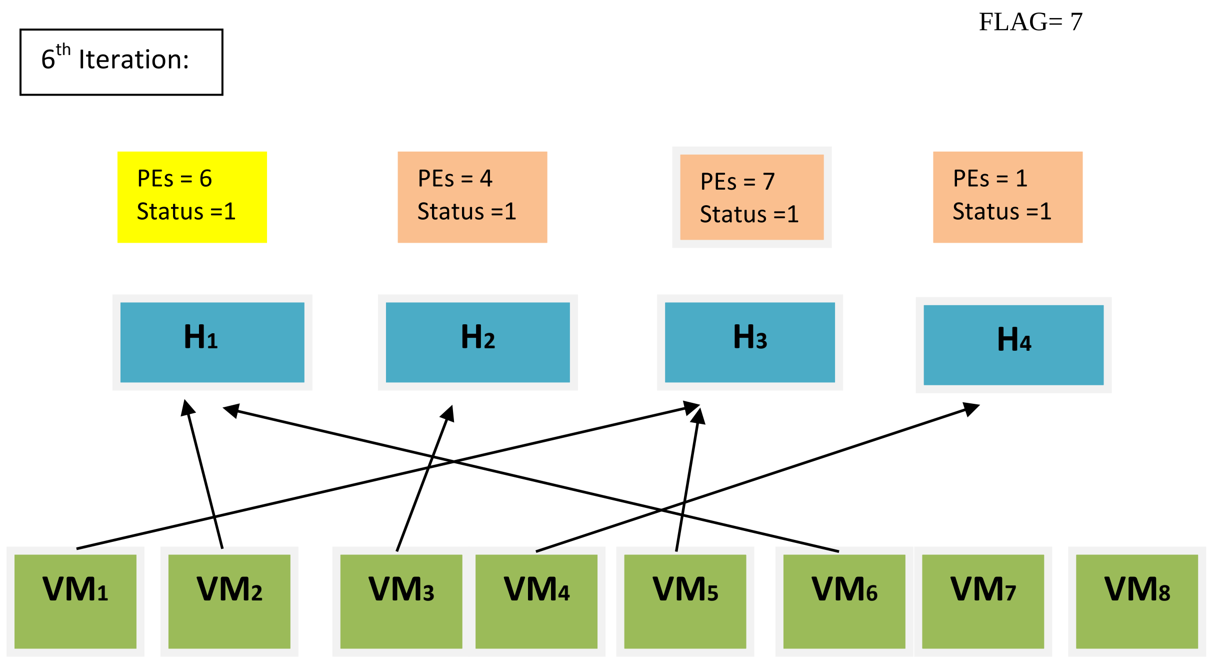


e)

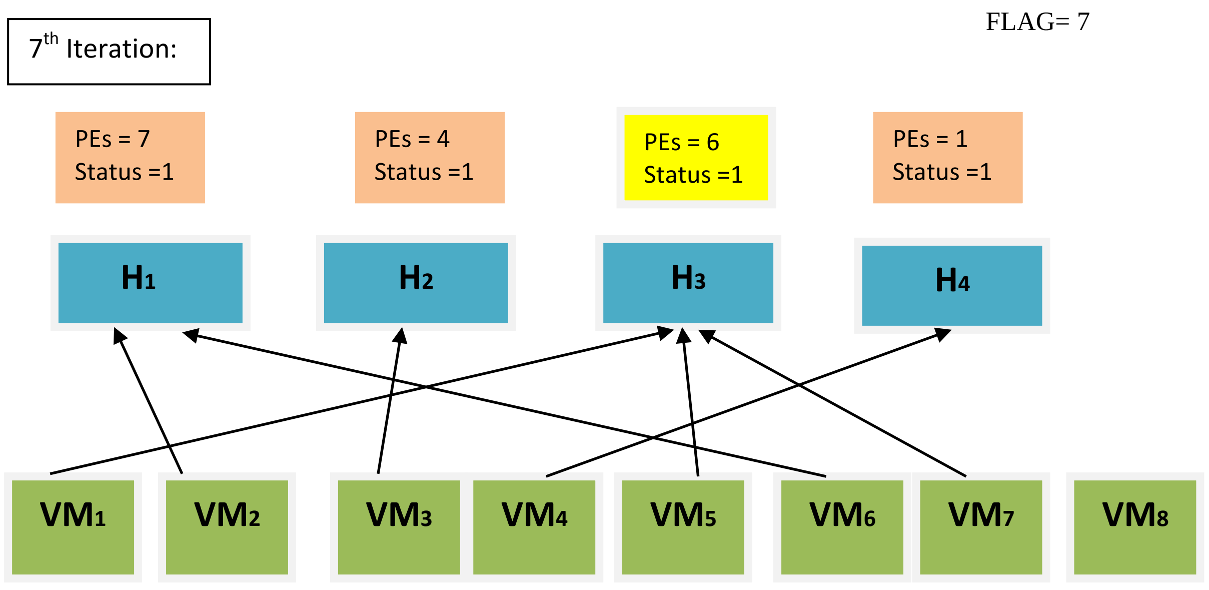


f)

Figure 5.2, continued e) In fourth iteration, only fourth host is having status "available", therefore fourth virtual machine is instantiated on H4. Status of the host is changed to "busy" and the number of processors are updated accordingly. Since the status of all the hosts has changed to busy, choose the host with maximum number of processors. f) H3 is allocated to fifth virtual machine.

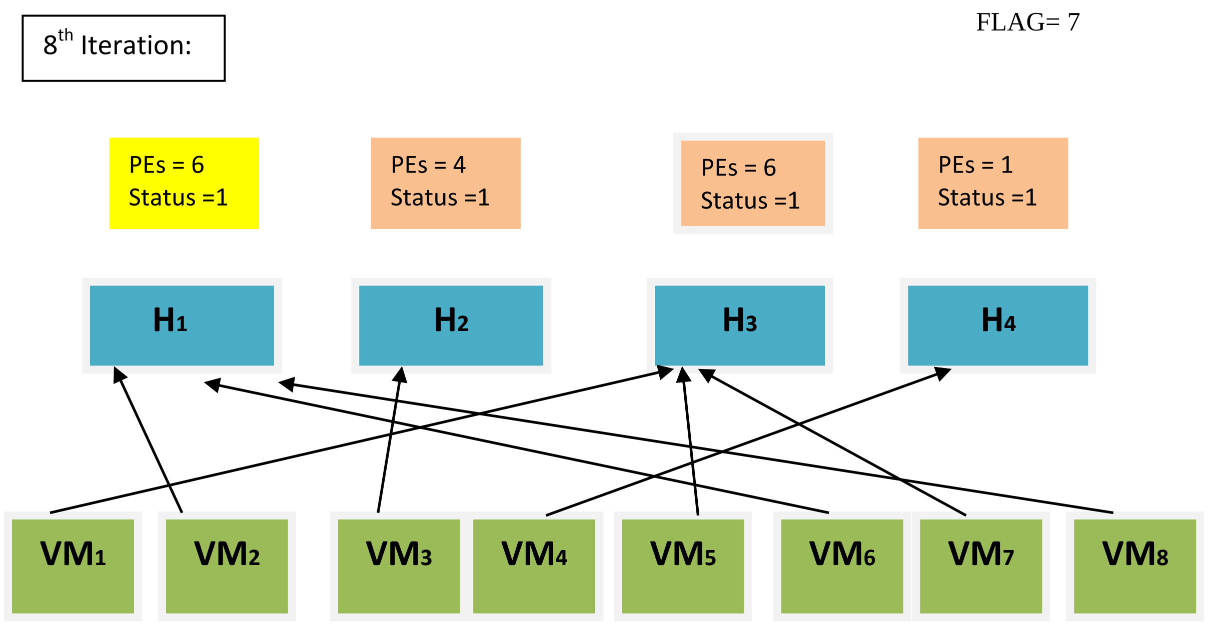


g)

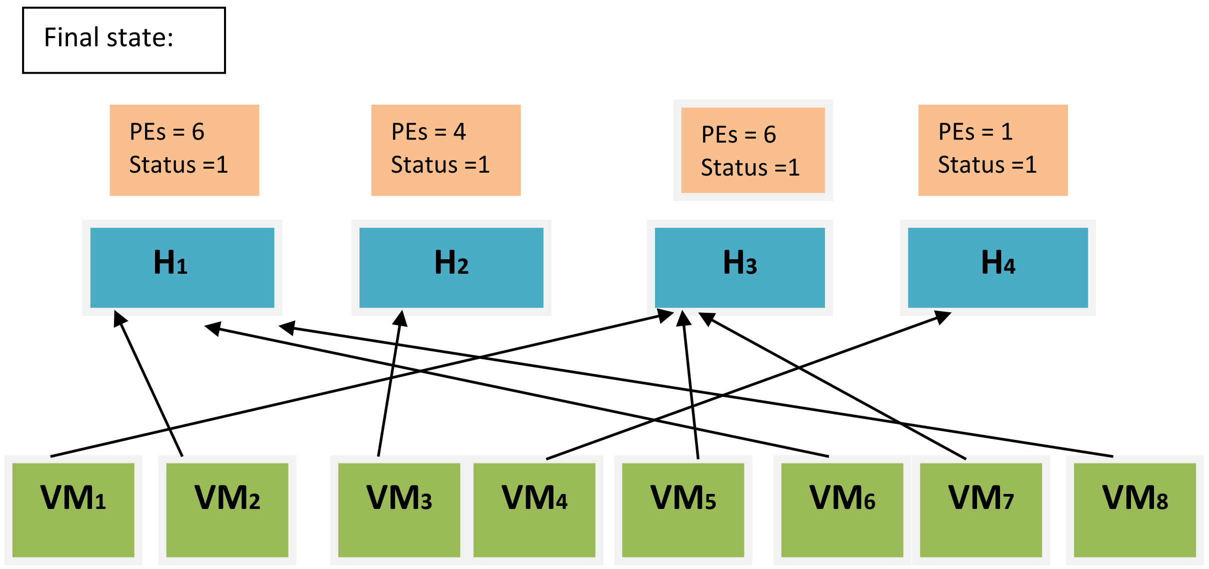


h)

Figure 5.2, continued g) In sixth iteration, since all hosts are busy, again choose the host with maximum free processors. H1 is allocated to sixth virtual machine. Update the number of processors. h) In similar fashion, seventh virtual machine is instantiated on H3.



i)



j)

Figure 5.2, continued i) Since all the hosts are busy, choose the host with maximum free processors. VM8 is assigned to first host having maximum processors available. Update the number of processors. j) Finally all the virtual machines are assigned to specific hosts.

After the computation work by each virtual machine gets done then the virtual machine is destroyed by the data center controller. And the new virtual machines which are in waiting queue were provided the freed hosts which are just released by the previous virtual machines.

From the above process it is now very much clear that the proposed algorithm efficiently distribute the virtual machines requests. Virtual machines are loaded with the user tasks by some other resource provisioning policy.

CHAPTER 6

Experimental Setup and Results

As discussed in previous chapters, choosing a right load balancing technique for a system is one of the critical tasks for developers. There are number of load balancing techniques suggested by various researchers, each having its own pros and cons. With the proposed methodology the main aim is to balance the overall workload among different nodes (server or host) in the network. In chapter 3 discussions about various metrics which are used to judge the performance of any load balancing technique have been done. Also with the proposed work it was tried to maintain an appropriate trade-off between different performance parameter so as to achieve the better results.

6.1 USED TECHNOLOGIES

This section presents a brief overview on the technologies used for the proposed work. The main technologies which have played an important role in implementing the proposed methodology are discussed below:

6.1.1 JAVA

Java has many attractive features which made it so popular among its users. 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 bytecode 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. The core idea was implemented in java only.

6.1.2 CLOUDSIM

A CloudSim [13] is just like other simulators which simulate certain scenarios as made by the users to test any policy or working of any other entity in a well-configured environment. For simulating the cloud computing scenarios we use CloudSim. CloudSim provides a vast number of choices for its user to test and model cloud computing system with different combination of provisioning policies and hardware resources. With the CloudSim one can model the cloud components with respect to system as well as behavior. 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. Big organizations like HP Labs in U.S.A., are using CloudSim for their research and development work. The proposed load balancing algorithm is also tested using CloudSim toolkit. Figure 6.1 shows the layered architecture of CloudSim.

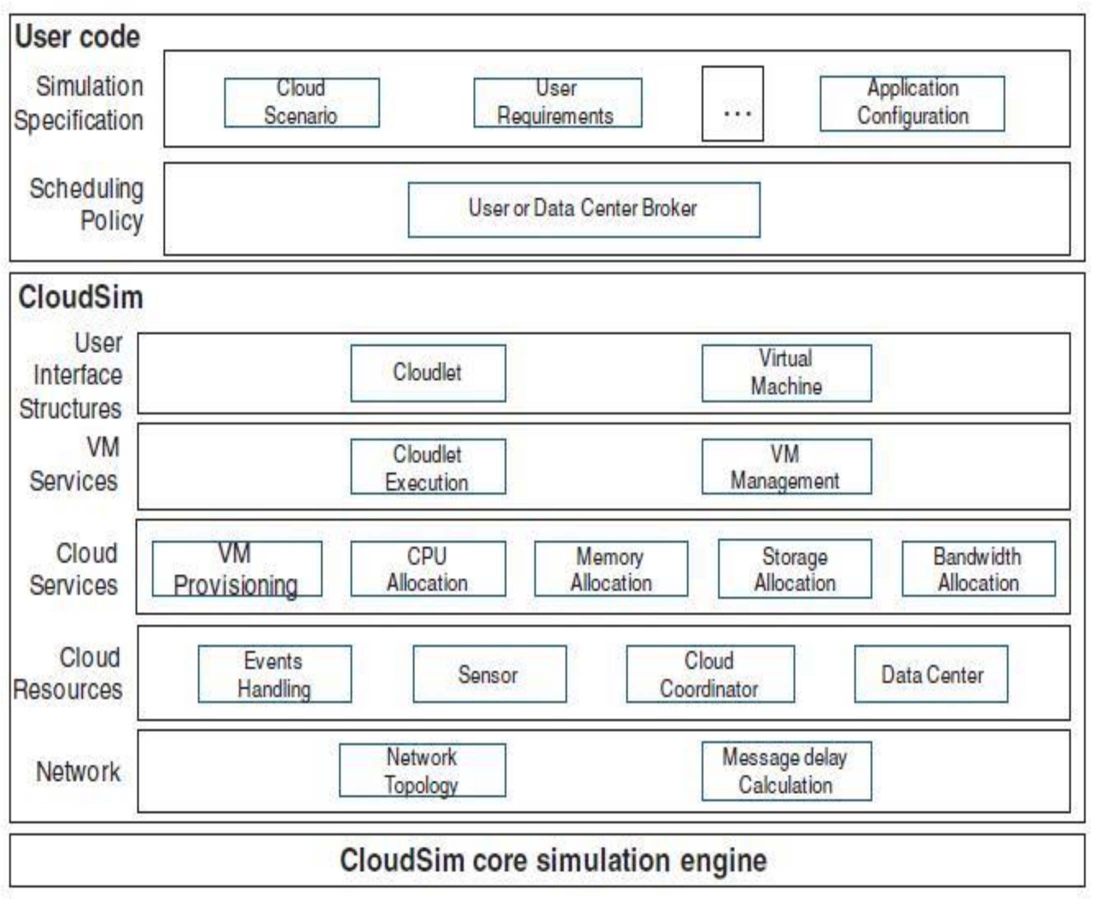


Figure 6.1: Layered architecture of CloudSim

6.1.3 CLOUDANALYST

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. A CloudAnalyst provides more user-friendly environment for the researchers. Also a new concept of dividing the whole area over the globe into six major continents and then also considering the peak-hours of each continent provides more realistic framework for doing the experiments. In CloudAnalyst a detailed report is generated for each piece of experiment. This report shows the graphical analyses of each of the scenarios. A response time and the total cost of the datacenter is also gets as an output. 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.

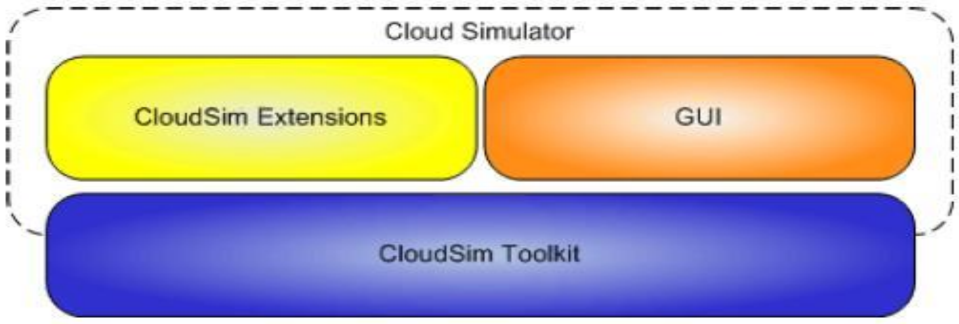


Figure 6.2: CloudAnalyst component

6.2 IMPLEMENTATION DETAILS

Implementation is done using the above technologies. The simulation work has been categorized in two sections. The first one is done using the CloudAnalyst. The specifications for the first simulation are mentioned below:

Facebook user statistics has been used for simulating different load balancing policies in combination with different service broker policies. Registered users on Facebook on 31-03-2012 are listed in Table 6.1. [39]

Table 6.1: Registered users on Facebook as on 31-03-2012

Major Geographic Region	Registered Users	CloudAnalyst Region Id
North America	173,284,940	R0
South America	112,531,100	R1
Europe	232,835,740	R2
Asia	195,034,380	R3
Africa	40,205,580	R4
Oceania/Australia	13,597,380	R5

A similar system has been considered, but at 1/10th of the scale of the Facebook, for the simulation task. Six user bases have been created defining the above mentioned six major geographic regions. Table 6.2 represents these user bases with some important parameters used in simulation.

Table 6.2: User bases with parameters

User base	Region Id	Time Zone	Peak Hours (GMT)	Simultaneous online users during peak hours	Simultaneous online users during off- peak hours
UB1	R2	GMT + 1.00	20:00 – 22:00	400,000	40,000
UB2	R3	GMT + 6.00	01:00 – 03:00	300,000	30,000
UB3	R0	GMT - 6.00	13:00 – 15:00	150,000	15,000
UB4	R1	GMT – 4.00	15:00 – 17:00	100,000	10,000
UB5	R4	GMT + 2.00	21:00 – 23:00	50,000	5,000
UB6	R5	GMT + 10.00	09:00 – 11:00	40,000	4,000

In order to provide the simulation framework more real environment certain assumptions regarding network configurations and pricing of resources have been done similar to one of the most popular cloud service provider Amazon EC2[7]. Assumptions are done according to the user statistics for different number of requests from each user in each hour and also for different data size request per user so as to achieve results closer to real environment. Certain parameters which are fixed during entire simulation are also used. These parameters are listed in Table 6.3.

Table 6.3: Fixed parameters used during simulation

Parameters	**Value Used**
Simulation duration	10 Hours
Data centers	2
User Bases	6
Number of Hosts(Each having 4 processors)	20
Virtual Machine	50 for each Datacenter
User grouping factor in user bases	1000
Request grouping factor in datacenters	100
Executable instruction length per request	250 bytes
Virtual Machine- image size	100000 Mb
Virtual Machine- available bandwidth	10000 Mb
Virtual Machine- memory	1024 Mb
VM policy for datacenter	Time-shared
Processor speed for datacenter	100 MIPS
VMM (Virtual machine monitor)	Xen
Memory Machine	100000 Mb

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.

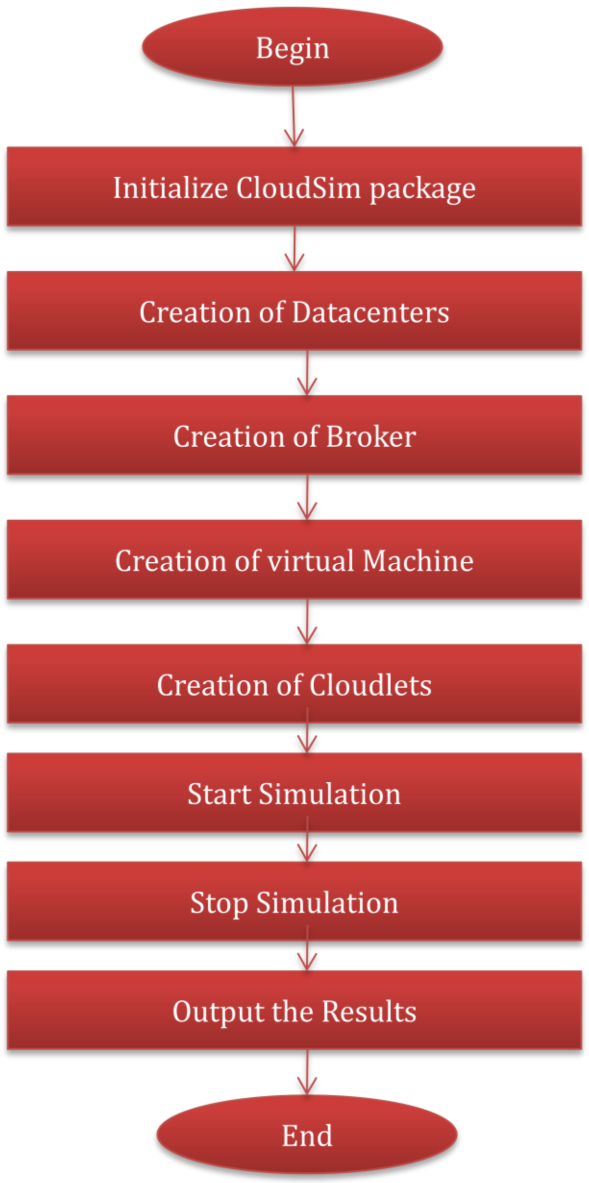


Figure 6.3: Flow of program execution in CloudSim

6.3 EXPERIMENTAL RESULTS

The results are 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. The results are shown as the screen shots of completed simulation process.

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 also done in order to have clear understanding of both the approaches. Let us discuss these results in detail.

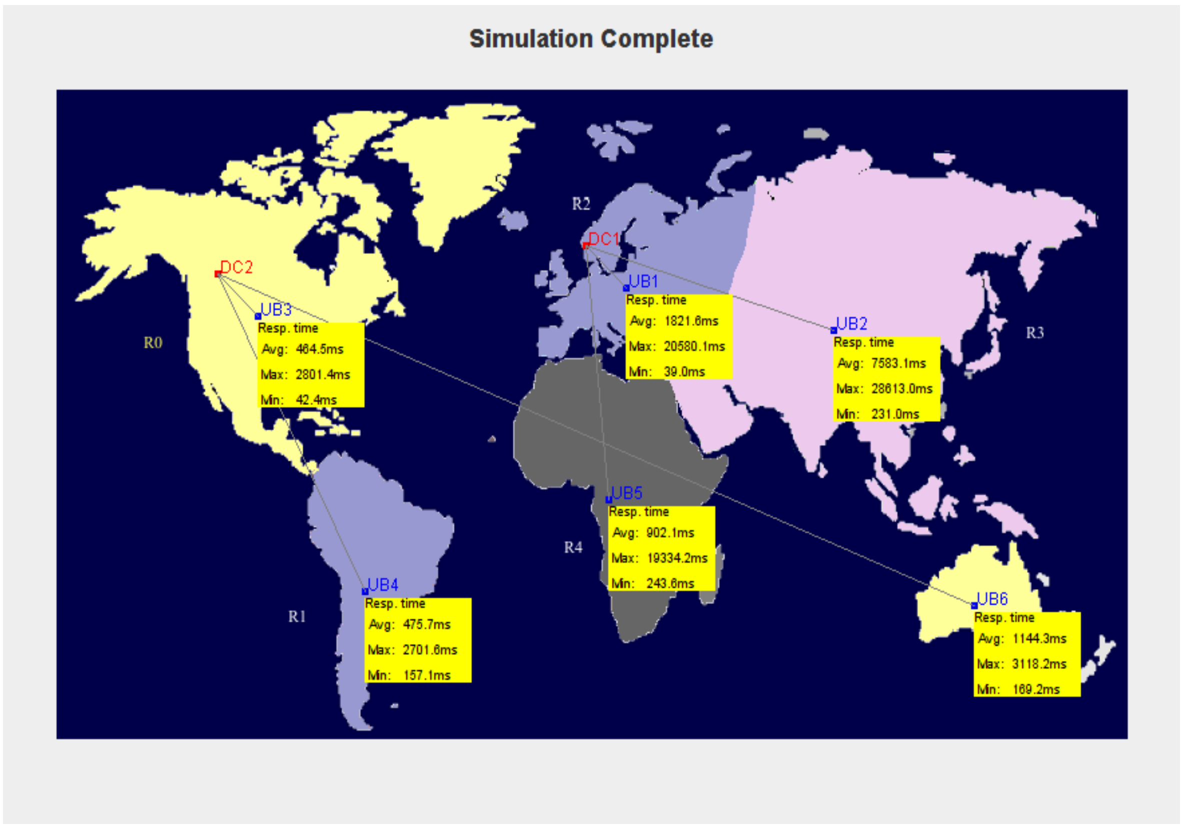
6.3.1 SIMULATION RESULTS OF EXISTING LOAD BALANCING ALGORITHMS

Results are analyzed for six different cases based upon the configuration stated in the previous section. Three load balancing policies namely Round Robin (RR), Equal Spread Current Execution (ESCE) and Throttled combined one by one with two service broker policies namely Closest Datacenter and Optimized Response Time. All of these policies with their mechanisms are discussed in [40]. All the six cases are discussed below:

CASE-1: Round Robin policy with Closest Datacenter policy

In this case load balancing policy used is Round Robin and service broker policy is Closest Datacenter, a simulated output is shown in Screen Shot 6.1. Region-wise response time is generated for each user base. In this configuration total cost for datacenter one

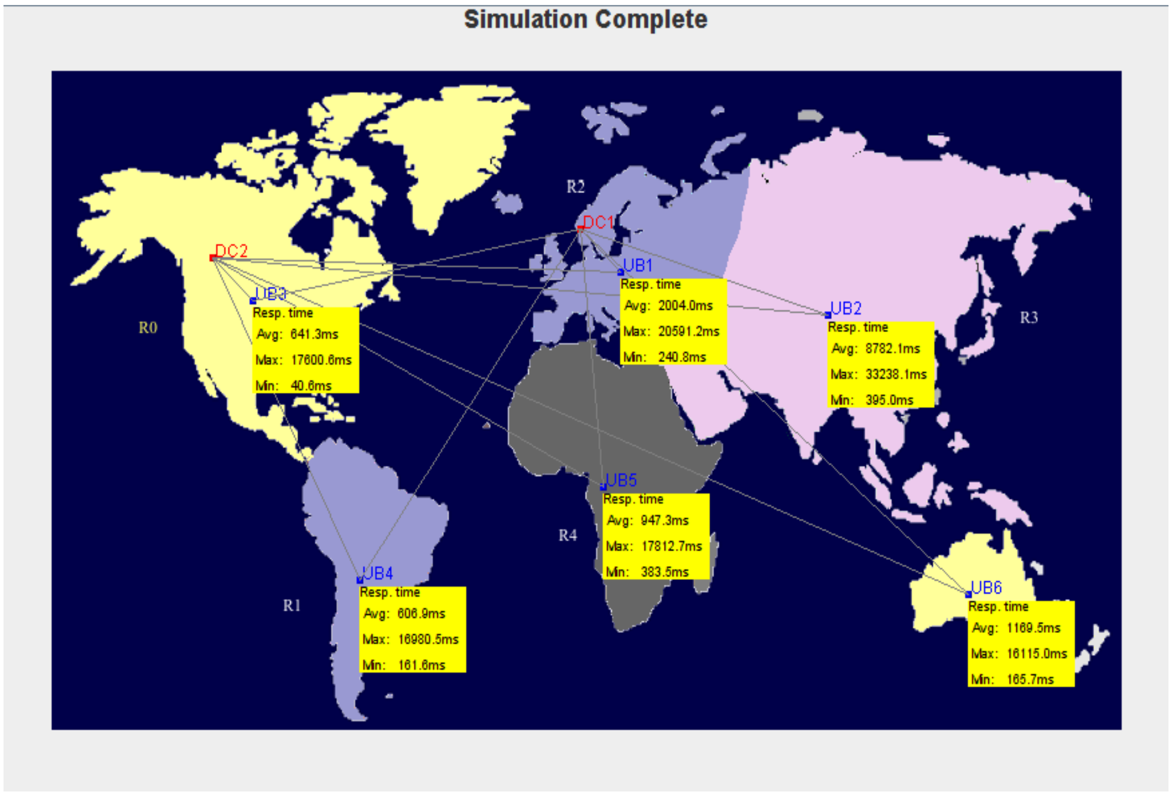
(DC1) comes out to be 1124.30 $ much higher than the cost of datacenter two (DC2), which is only 267.23 $.


Simulation Complete
R0
R1
R2
R3
R4
DC1
DC2
UB1 Resp. time Avg: 1821.6ms Max: 20580.1ms Min: 39.0ms
UB2 Resp. time Avg: 7583.1ms Max: 28613.0ms Min: 231.0ms
UB3 Resp. time Avg: 464.5ms Max: 2801.4ms Min: 42.4ms
UB4 Resp. time Avg: 475.7ms Max: 2701.6ms Min: 157.1ms
UB5 Resp. time Avg: 902.1ms Max: 19334.2ms Min: 243.6ms
UB6 Resp. time Avg: 1144.3ms Max: 3118.2ms Min: 169.2ms

Screen Shot 6.1: Simulation completed for Round Robin policy with Closest Datacenter policy

CASE-2: Round Robin policy with Optimize Response Time policy

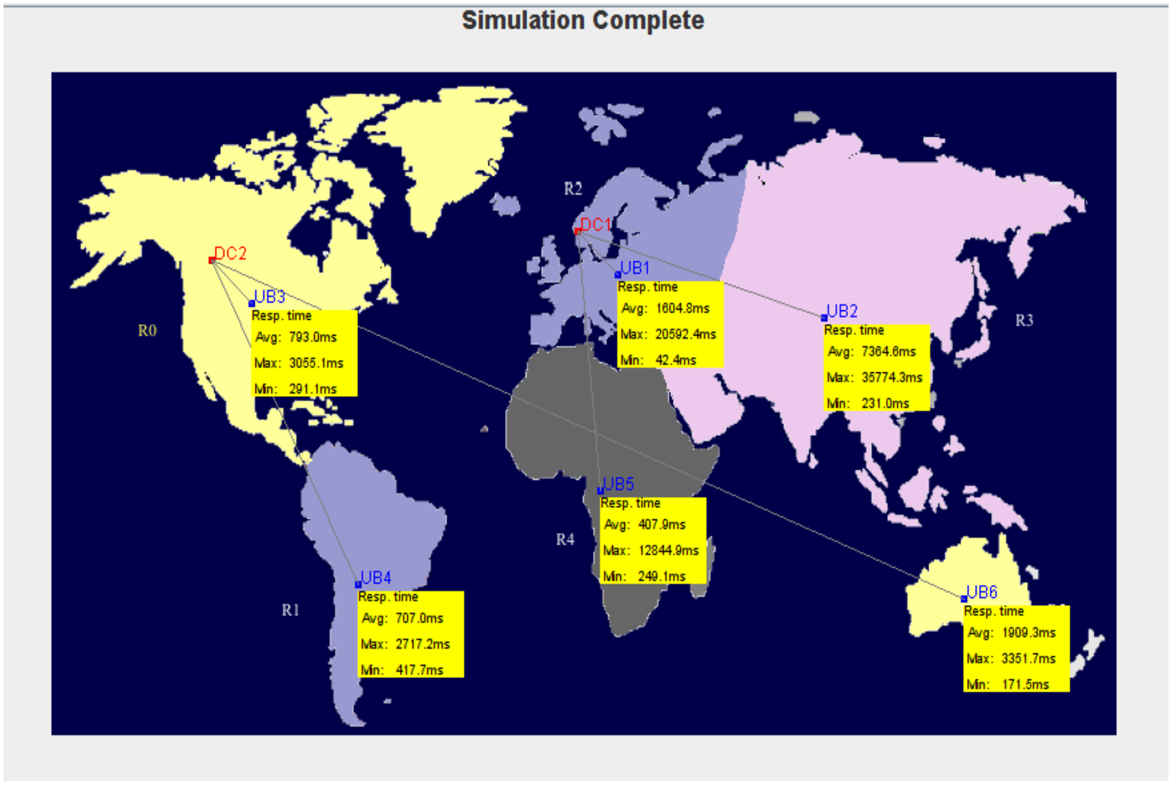
In this case service broker policy is changed to Optimize Response Time and load balancing for the application is done through Round Robin policy. Screen Shot 6.2 shows the simulated output for this configuration. Fine lines in the Screen Shot 6.2 represent several requests from different user bases to the specific datacenter. As compared to previous case overall response time in this situation is higher than the previous case, but the total cost for both the datacenter is balanced.



Screen Shot 6.2: Simulation completed for Round Robin policy with Optimized Response time policy

CASE-3: Equal Spread Current Execution policy with Closest Datacenter policy

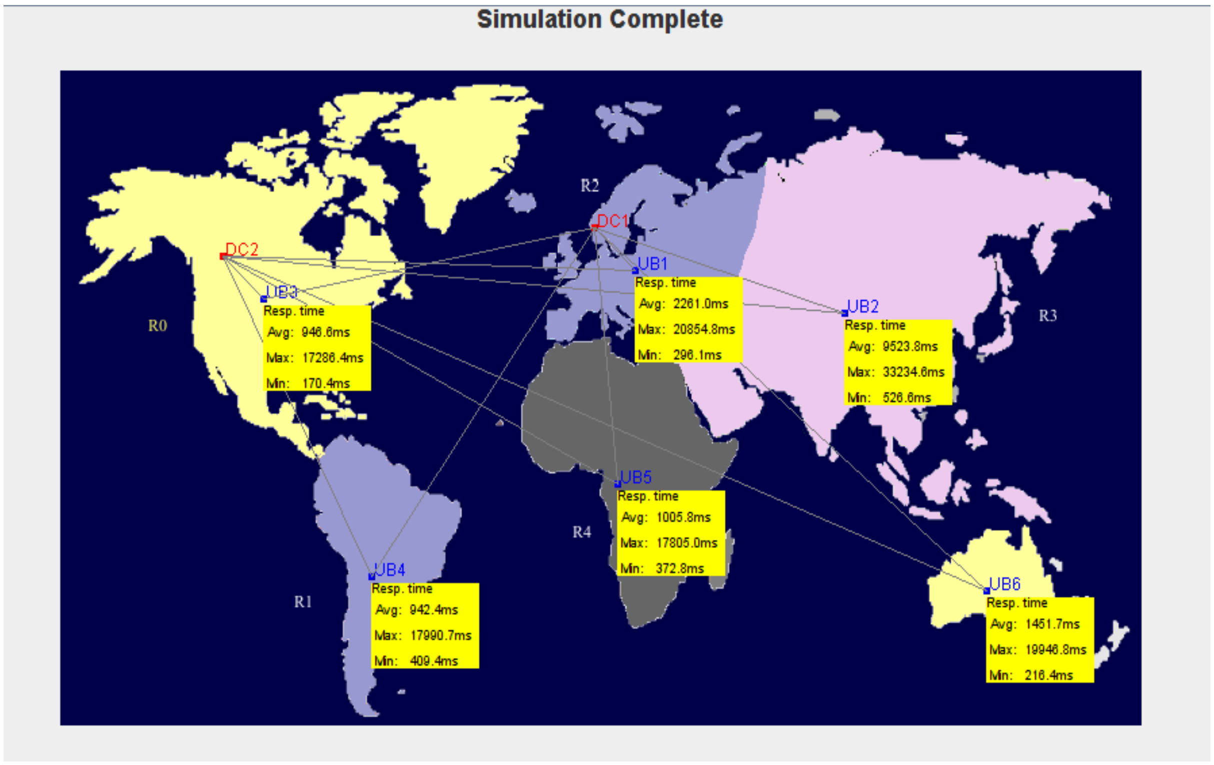
Here the load balancing policy was changed to ESCE and service broker policy was chosen as Closest Datacenter policy. The methodologies of all these policies have been discussed earlier. Overall response time and data processing time in this case is better than the previous cases and the total cost is similar to CASE-1. In Screen Shot 6.3 the whole situation for this configuration can be analyzed clearly.



Screen Shot 6.3: Simulation completed for Equal Spread Current Execution policy with

CASE-4: Equal Spread Current Execution policy with Optimize Response Time policy

In this case, load balancing policy is same as in CASE-3, but service broker policy is changed to Optimize Response Time. Overall response time and data processing is much higher than all the cases. Also the cost for virtual machine and data transfer is high. Therefore, it can be concluded that this configuration will not provide any benefit to end users and Cloud service providers. Screen Shot 6.4 shows the simulated output for this case.



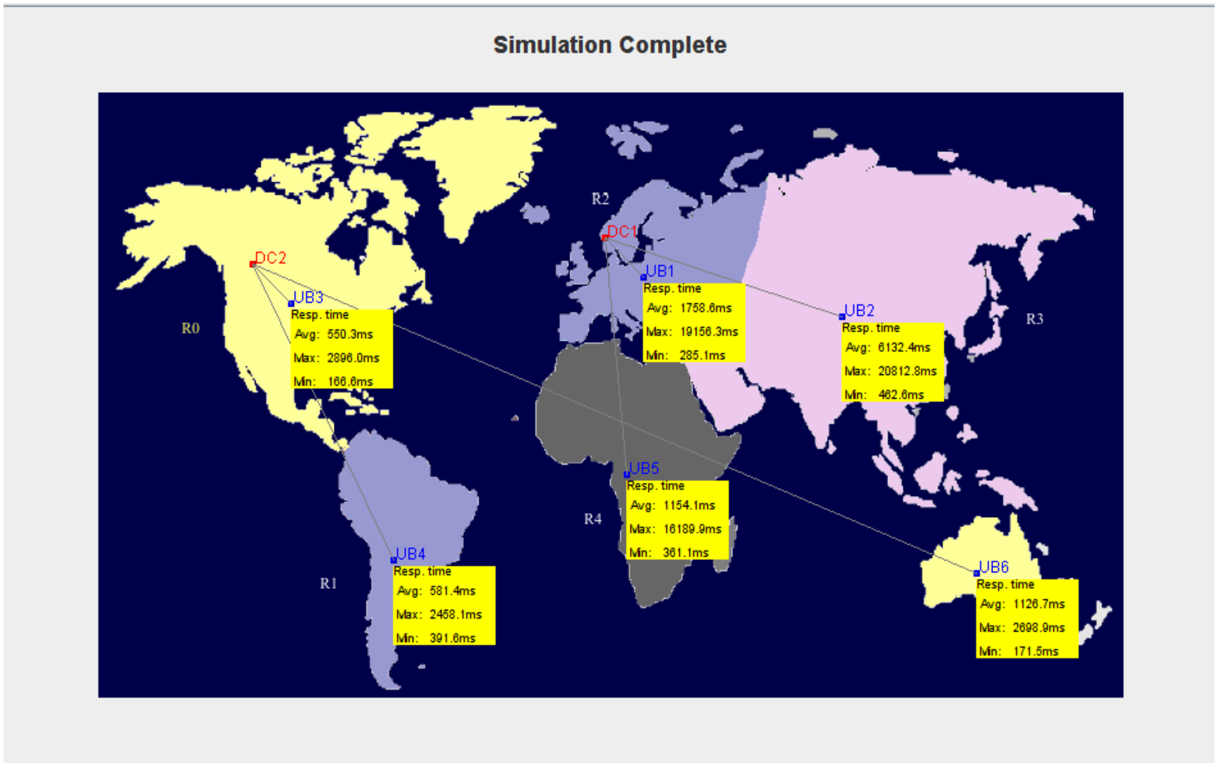
Simulation Complete

```
UB1
Resp. time
 Avg:  2261.0ms
 Max:  20854.8ms
 Min:  296.1ms

UB2
Resp. time
 Avg:  9523.8ms
 Max:  33234.6ms
 Min:  526.6ms

UB3
Resp. time
 Avg:  946.6ms
 Max:  17286.4ms
 Min:  170.4ms

UB4
Resp. time
 Avg:  942.4ms
 Max:  17990.7ms
 Min:  409.4ms

UB5
Resp. time
 Avg:  1005.8ms
 Max:  17805.0ms
 Min:  372.8ms

UB6
Resp. time
 Avg:  1451.7ms
 Max:  19946.8ms
 Min:  216.4ms
```

Screen Shot 6.4: Simulation completed for Equal Spread Current Execution policy with Optimize Response Time policy

CASE-5: Throttled policy with Closest Datacenter Policy

Here the load balancing is done through Throttled policy and service broker policy is Closest Datacenter. This configuration gives the much better result than the previous cases. Screen Shot 6.5 shows the response time for different user bases. There is major downfall in response time as well as in data processing time at each datacenter. Total cost is same for all the cases where service broker policy used is Closest Datacenter.



UB1 Resp. time Avg: 1758.6ms Max: 19156.3ms Min: 285.1ms

UB2 Resp. time Avg: 6132.4ms Max: 20812.8ms Min: 462.6ms

UB3 Resp. time Avg: 550.3ms Max: 2896.0ms Min: 166.6ms

UB4 Resp. time Avg: 581.4ms Max: 2458.1ms Min: 391.6ms

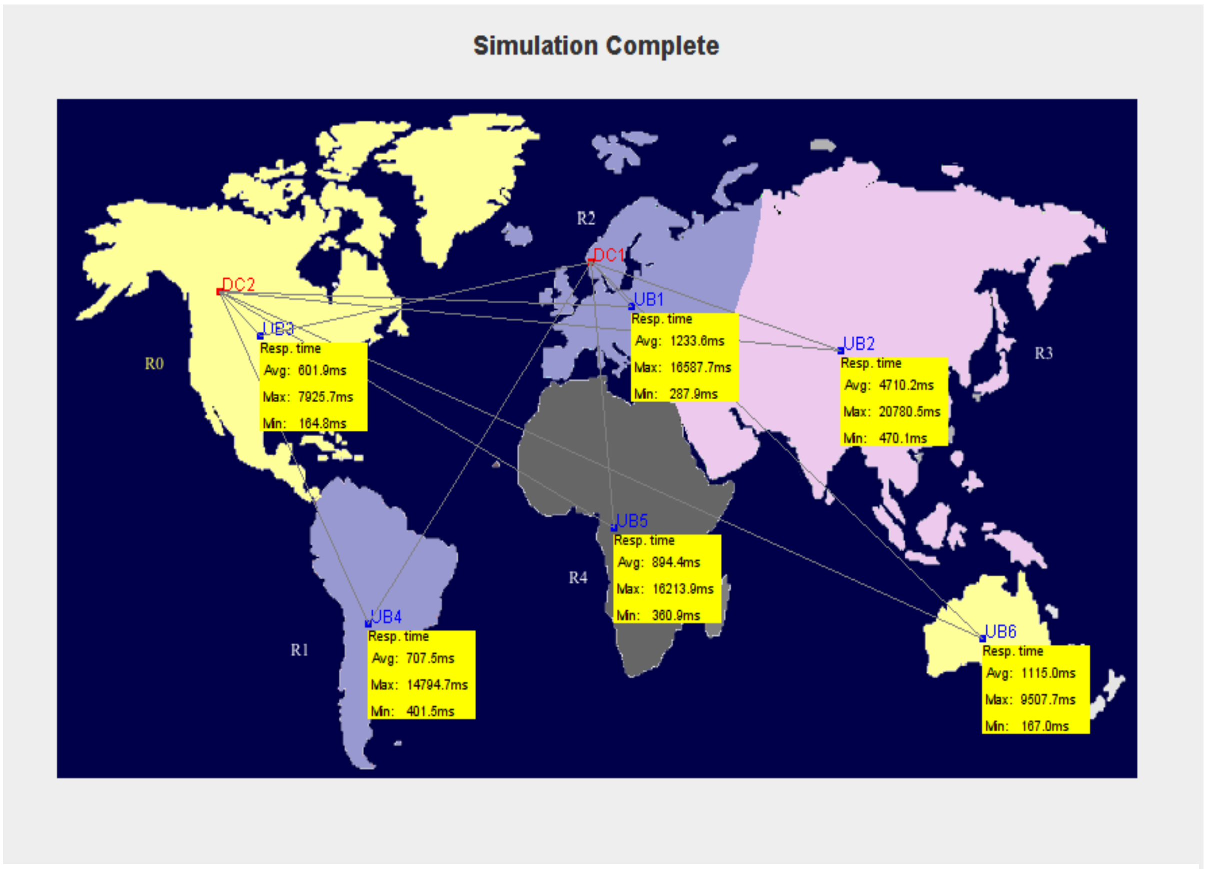
UB5 Resp. time Avg: 1154.1ms Max: 16189.9ms Min: 361.1ms

UB6 Resp. time Avg: 1126.7ms Max: 2698.9ms Min: 171.5ms

Screen Shot 6.5: Simulation completed for Throttled policy with Closest Datacenter Policy

CASE-6: Throttled policy with Optimize Response Time policy

In this case, load balancing policy is Throttled, but service broker policy has changed to Optimize Response Time. Overall response time and the data processing time are 3034.47 ms and 2760.83 ms respectively, which is lowest as compared to all the previous cases. And the total cost for the virtual machine and the data transfer is also the minimum. Therefore, it can be stated that this configuration is best for our application. It will benefit end-users with minimum response time and also profit cloud service providers with minimum cost for operation. Screen Shot 6.6, shows the simulated result for this case.



Screen Shot 6.6 Simulation completed for Throttled policy with Optimize Response Time policy

The overall response time in each case can be analyzed graphically through Figure 6.4. The minimum, 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 results 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.

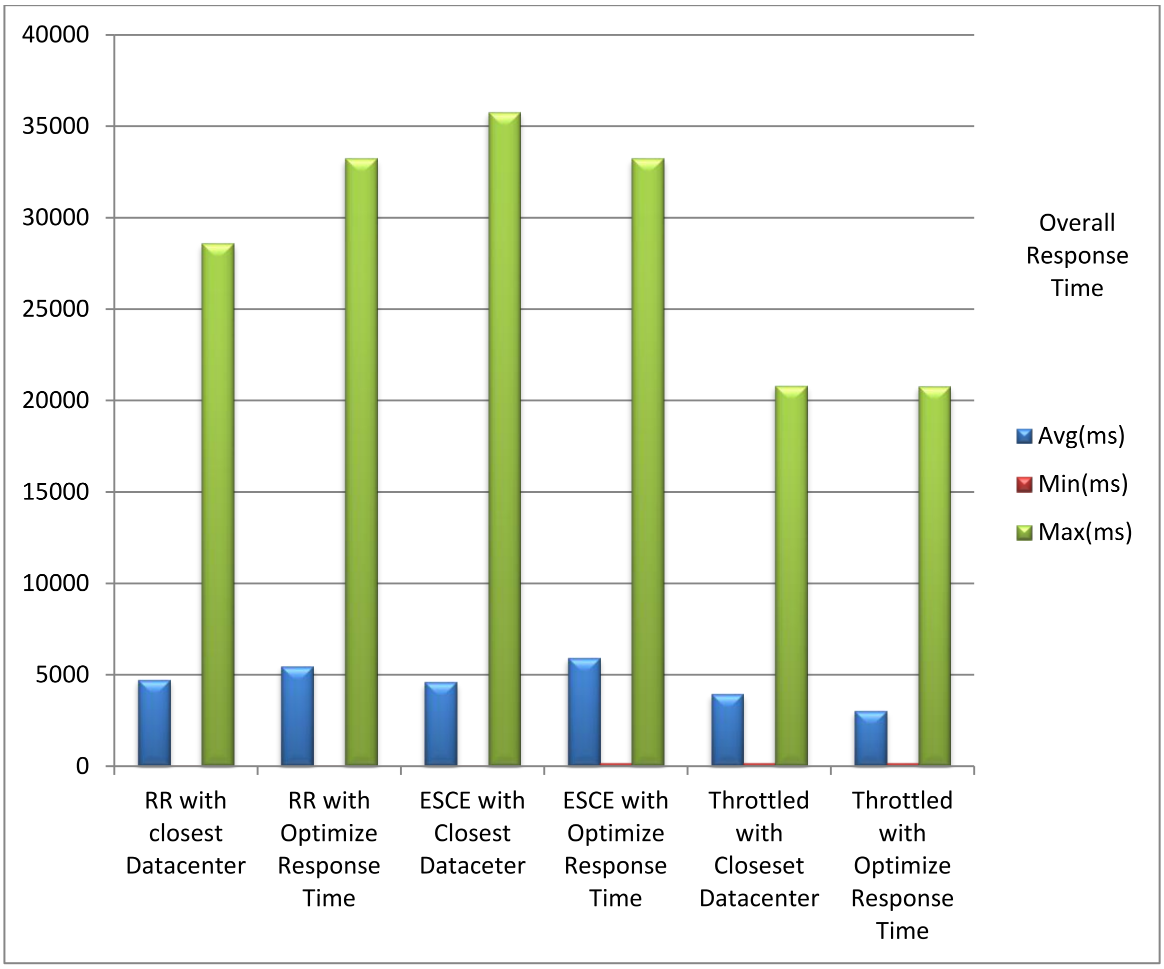


Figure 6.4: Overall Response Time for all the cases

6.3.2 SIMULATION RESULTS OF PROPOSED LOAD BALANCING ALGORITHM

In this section discussion about the simulation results of the proposed load balancing algorithm have been done. Certain assumptions have been made regarding the experiment setup required for the verification and validation of the proposed load balancing algorithm. A set of twenty virtual machines and five hosts all with different configurations have been modeled and configured for the experiment purpose in CloudSim.

For the present experimentation work a most realistic scenario has been considered. 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 twenty virtual machines are created which are to be deployed on any of the five hosts created in the datacenter based upon the load balancing algorithm. Each host is having different number of processing cores. Here, different numbers of processing cores are assumed in order to judge the proposed algorithm more precisely in a heterogeneous environment. Here host H_1 is having eight processing cores, similarly H_2 six, H_3 five, H_4 nine and H_5 four. Also each host is configured differently, each parameter like memory, bandwidth, secondary storage, number of processing cores are assigned randomly to have a more real touch.

Since for the work mentioned in this dissertation load balancing at a host level is performed 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.

Before discussing the output of the proposed algorithm and the traditional existing algorithm let us discuss a brief upon the traditional approach. In traditional approach of load balancing the allocation of host for a given virtual machine is done by the rule which states that “a host with maximum number of available processing element is allocated to a given virtual machine whenever a virtual machine requests a host allocation”. This approach doesn’t checks whether a host with maximum number of available processing element is already running many virtual machines or not.

OUTPUT:

The output for the above stated scenario in the case of traditional approach is shown below:

```
Starting MyTest...
Initialising...
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 #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: 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: Trying to Create VM #9 in Datacenter_0
0.0: Broker: Trying to Create VM #10 in Datacenter_0
0.0: Broker: Trying to Create VM #11 in Datacenter_0
0.0: Broker: Trying to Create VM #12 in Datacenter_0
0.0: Broker: Trying to Create VM #13 in Datacenter_0
0.0: Broker: Trying to Create VM #14 in Datacenter_0
0.0: Broker: Trying to Create VM #15 in Datacenter_0
0.0: Broker: Trying to Create VM #16 in Datacenter_0
0.0: Broker: Trying to Create VM #17 in Datacenter_0
0.0: Broker: Trying to Create VM #18 in Datacenter_0
0.0: Broker: Trying to Create VM #19 in Datacenter_0
0.0: Broker: VM #0 has been created in Datacenter #2, Host #3
0.0: Broker: VM #1 has been created in Datacenter #2, Host #0
0.0: Broker: VM #2 has been created in Datacenter #2, Host #3
0.0: Broker: VM #3 has been created in Datacenter #2, Host #0
0.0: Broker: VM #4 has been created in Datacenter #2, Host #3
0.0: Broker: VM #5 has been created in Datacenter #2, Host #0
0.0: Broker: VM #6 has been created in Datacenter #2, Host #1
0.0: Broker: VM #7 has been created in Datacenter #2, Host #3
0.0: Broker: VM #8 has been created in Datacenter #2, Host #0
0.0: Broker: VM #9 has been created in Datacenter #2, Host #1
0.0: Broker: VM #10 has been created in Datacenter #2, Host #2
0.0: Broker: VM #11 has been created in Datacenter #2, Host #3
0.0: Broker: VM #12 has been created in Datacenter #2, Host #0
0.0: Broker: VM #13 has been created in Datacenter #2, Host #1
0.0: Broker: VM #14 has been created in Datacenter #2, Host #2
0.0: Broker: VM #15 has been created in Datacenter #2, Host #3
0.0: Broker: VM #16 has been created in Datacenter #2, Host #4
0.0: Broker: VM #17 has been created in Datacenter #2, Host #0
0.0: Broker: VM #18 has been created in Datacenter #2, Host #1
0.0: Broker: VM #19 has been created in Datacenter #2, Host #2
0.0: Broker: Sending cloudlet 0 to VM #0
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 #6
0.0: Broker: Sending cloudlet 6 to VM #7
```

```
0.0: Broker: Sending cloudlet 7 to VM #8
0.0: Broker: Sending cloudlet 8 to VM #9
0.0: Broker: Sending cloudlet 9 to VM #10
0.0: Broker: Sending cloudlet 10 to VM #11
0.0: Broker: Sending cloudlet 11 to VM #12
0.0: Broker: Sending cloudlet 12 to VM #13
0.0: Broker: Sending cloudlet 13 to VM #14
0.0: Broker: Sending cloudlet 14 to VM #16
0.0: Broker: Sending cloudlet 14 to VM #16
0.0: Broker: Sending cloudlet 15 to VM #17
0.0: Broker: Sending cloudlet 16 to VM #18
0.0: Broker: Sending cloudlet 17 to VM #19
80.0: Broker: Cloudlet 1 received
80.0: Broker: Cloudlet 3 received
80.0: Broker: Cloudlet 7 received
80.0: Broker: Cloudlet 11 received
80.0: Broker: Cloudlet 15 received
80.0: Broker: Cloudlet 8 received
80.0: Broker: Cloudlet 12 received
80.0: Broker: Cloudlet 16 received
80.0: Broker: Cloudlet 9 received
80.0: Broker: Cloudlet 13 received
80.0: Broker: Cloudlet 17 received
80.0: Broker: Cloudlet 0 received
80.0: Broker: Cloudlet 2 received
80.0: Broker: Cloudlet 4 received
80.0: Broker: Cloudlet 6 received
80.0: Broker: Cloudlet 10 received
160.0: Broker: Cloudlet 5 received
160.0: Broker: Cloudlet 5 received
160.0: Broker: Cloudlet 14 received
160.0: Broker: Cloudlet 14 received
160.0: Broker: All Cloudlets executed. Finishing...
160.0: Broker: Destroying VM #0
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
160.0: Broker: Destroying VM #9
160.0: Broker: Destroying VM #10
160.0: Broker: Destroying VM #11
160.0: Broker: Destroying VM #12
160.0: Broker: Destroying VM #13
160.0: Broker: Destroying VM #14
160.0: Broker: Destroying VM #15
160.0: Broker: Destroying VM #16
160.0: Broker: Destroying VM #17
160.0: Broker: Destroying VM #18
160.0: Broker: Destroying VM #19
```

```
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   Time   Start Time   Finish Time
    1          SUCCESS          2            1       80          0             80
    3          SUCCESS          2            3       80          0             80
    7          SUCCESS          2            8       80          0             80
    11         SUCCESS          2           12       80          0             80
    15         SUCCESS          2           17       80          0             80
    8          SUCCESS          2            9       80          0             80
    12         SUCCESS          2           13       80          0             80
    16         SUCCESS          2           18       80          0             80
    9          SUCCESS          2           10       80          0             80
    13         SUCCESS          2           14       80          0             80
    17         SUCCESS          2           19       80          0             80
    0          SUCCESS          2            0       80          0             80
    2          SUCCESS          2            2       80          0             80
    4          SUCCESS          2            4       80          0             80
    6          SUCCESS          2            7       80          0            80
    10         SUCCESS          2           11       80          0            80
    5          SUCCESS          2            6      160          0           160
    5          SUCCESS          2            6      160          0           160
    14         SUCCESS          2           16      160          0           160
    14         SUCCESS          2           16      160          0           160
*****PowerDatacenter: Datacenter_0*****
User id              Debt
3                    2248
**********************************
MyTest finished!
```

Allocation for the host can be clearly seen in the above output. According to the traditional 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 only one processing element from the host for performing its operation.

The output for the above stated scenario in the case of proposed approach is shown below:

```
Starting MyTest...
Initialising...
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 #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: 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: Trying to Create VM #9 in Datacenter_0
0.0: Broker: Trying to Create VM #10 in Datacenter_0
0.0: Broker: Trying to Create VM #11 in Datacenter_0
0.0: Broker: Trying to Create VM #12 in Datacenter_0
0.0: Broker: Trying to Create VM #13 in Datacenter_0
0.0: Broker: Trying to Create VM #14 in Datacenter_0
0.0: Broker: Trying to Create VM #15 in Datacenter_0
0.0: Broker: Trying to Create VM #16 in Datacenter_0
0.0: Broker: Trying to Create VM #17 in Datacenter_0
0.0: Broker: Trying to Create VM #18 in Datacenter_0
0.0: Broker: Trying to Create VM #19 in Datacenter_0
0.0: Broker: VM #0 has been created in Datacenter #2, Host #3
0.0: Broker: VM #1 has been created in Datacenter #2, Host #0
0.0: Broker: VM #2 has been created in Datacenter #2, Host #1
0.0: Broker: VM #3 has been created in Datacenter #2, Host #2
0.0: Broker: VM #4 has been created in Datacenter #2, Host #4
0.0: Broker: VM #5 has been created in Datacenter #2, Host #3
0.0: Broker: VM #6 has been created in Datacenter #2, Host #0
0.0: Broker: VM #7 has been created in Datacenter #2, Host #3
0.0: Broker: VM #8 has been created in Datacenter #2, Host #0
0.0: Broker: VM #9 has been created in Datacenter #2, Host #3
0.0: Broker: VM #10 has been created in Datacenter #2, Host #0
0.0: Broker: VM #11 has been created in Datacenter #2, Host #1
0.0: Broker: VM #12 has been created in Datacenter #2, Host #3
0.0: Broker: VM #13 has been created in Datacenter #2, Host #0
0.0: Broker: VM #14 has been created in Datacenter #2, Host #1
0.0: Broker: VM #15 has been created in Datacenter #2, Host #2
0.0: Broker: VM #16 has been created in Datacenter #2, Host #3
0.0: Broker: VM #17 has been created in Datacenter #2, Host #0
0.0: Broker: VM #18 has been created in Datacenter #2, Host #1
0.0: Broker: VM #19 has been created in Datacenter #2, Host #2
0.0: Broker: Sending cloudlet 0 to VM #0
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 #6
0.0: Broker: Sending cloudlet 6 to VM #7
```

```
0.0: Broker: Sending cloudlet 7 to VM #8
0.0: Broker: Sending cloudlet 8 to VM #9
0.0: Broker: Sending cloudlet 9 to VM #10
0.0: Broker: Sending cloudlet 10 to VM #11
0.0: Broker: Sending cloudlet 11 to VM #12
0.0: Broker: Sending cloudlet 12 to VM #13
0.0: Broker: Sending cloudlet 13 to VM #14
0.0: Broker: Sending cloudlet 14 to VM #16
0.0: Broker: Sending cloudlet 14 to VM #16
0.0: Broker: Sending cloudlet 15 to VM #17
0.0: Broker: Sending cloudlet 16 to VM #18
0.0: Broker: Sending cloudlet 17 to VM #19
80.0: Broker: Cloudlet 1 received
80.0: Broker: Cloudlet 3 received
80.0: Broker: Cloudlet 7 received
80.0: Broker: Cloudlet 11 received
80.0: Broker: Cloudlet 15 received
80.0: Broker: Cloudlet 8 received
80.0: Broker: Cloudlet 12 received
80.0: Broker: Cloudlet 16 received
80.0: Broker: Cloudlet 9 received
80.0: Broker: Cloudlet 13 received
80.0: Broker: Cloudlet 17 received
80.0: Broker: Cloudlet 0 received
80.0: Broker: Cloudlet 2 received
80.0: Broker: Cloudlet 4 received
80.0: Broker: Cloudlet 6 received
80.0: Broker: Cloudlet 10 received
160.0: Broker: Cloudlet 5 received
160.0: Broker: Cloudlet 5 received
160.0: Broker: Cloudlet 14 received
160.0: Broker: Cloudlet 14 received
160.0: Broker: All Cloudlets executed. Finishing...
160.0: Broker: Destroying VM #0
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
160.0: Broker: Destroying VM #9
160.0: Broker: Destroying VM #10
160.0: Broker: Destroying VM #11
160.0: Broker: Destroying VM #12
160.0: Broker: Destroying VM #13
160.0: Broker: Destroying VM #14
160.0: Broker: Destroying VM #15
160.0: Broker: Destroying VM #16
160.0: Broker: Destroying VM #17
160.0: Broker: Destroying VM #18
160.0: Broker: Destroying VM #19
```

```
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   Time   Start Time   Finish Time
    1         SUCCESS         2            1      80        0            80
    7         SUCCESS         2            3      80        0            80
    9         SUCCESS         2            8      80        0            80
    12        SUCCESS         2           12      80        0            80
    15        SUCCESS         2           17      80        0            80
    2         SUCCESS         2            9      80        0            80
    10        SUCCESS         2           13      80        0            80
    13        SUCCESS         2           18      80        0            80
    16        SUCCESS         2           10      80        0            80
    3         SUCCESS         2           14      80        0            80
    17        SUCCESS         2           19      80        0            80
    0         SUCCESS         2            0      80        0            80
    6         SUCCESS         2            2      80        0            80
    8         SUCCESS         2            4      80        0            80
    11        SUCCESS         2            7      80        0           80
    4         SUCCESS         2           11      80        0           80
    5         SUCCESS         2            6      160       0           160
    5         SUCCESS         2            6      160       0           160
    14        SUCCESS         2           16      160       0           160
    14        SUCCESS         2           16      160       0           160
*****PowerDatacenter: Datacenter_0*****
User id              Debt
3                    2248
**********************************
MyTest finished!
```

In case of the proposed algorithm it is clearly visible from the output that each host is allocated to a given virtual machine as per the proposed methodology, which is comparatively more distributed in nature. The order of host allocation is clearly visible in the above output. Now after running both the load balancing algorithm a difference can clearly be seen in the allocation of host to a given virtual machine. 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 virtual machines 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.

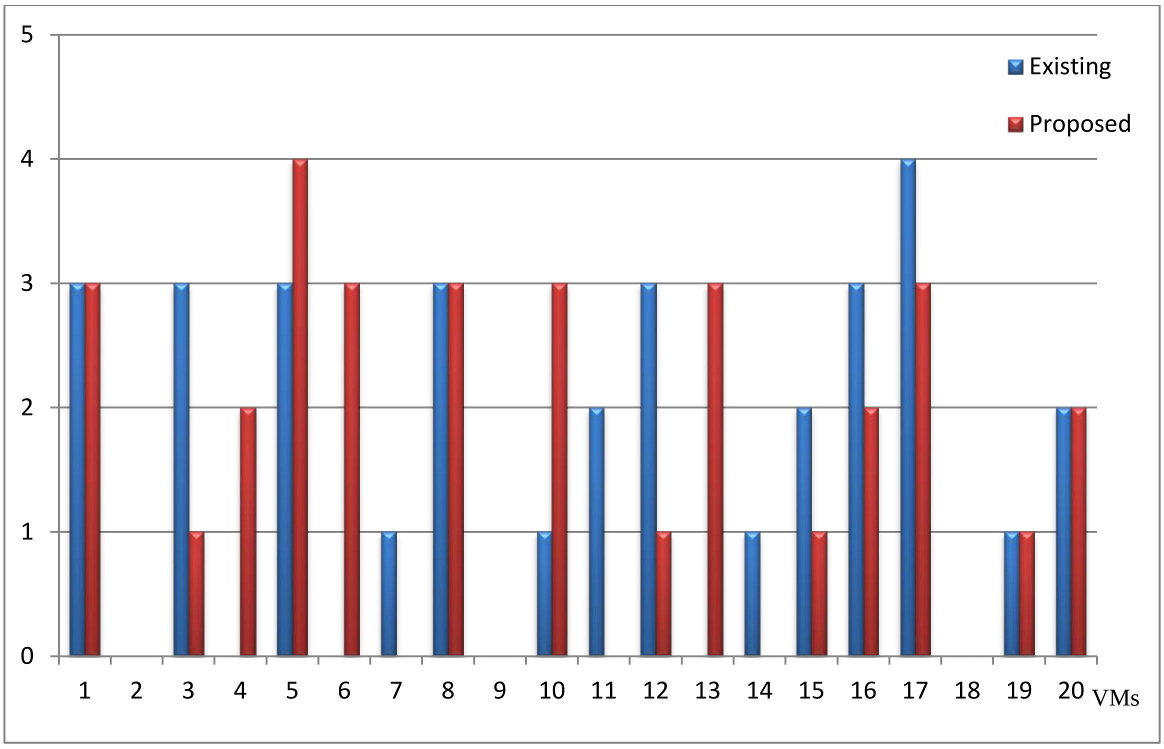


Figure 6.5: Comparision between the traditional approach and the proposed approach

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

Cloud computing is one of the hot topics in the information and communication industry. It has become popular among a large number of users in a very short span of time. 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. 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.

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. A cloud service provider must be able to deliver services to its user in minimum time as well as performing the efficient utilization of all the resources present within the cloud network.

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 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
40. Deepti Singh, Deepti Sarsawat and Dr. N. S. Raghava, “Analyzing Large-scaled Applications in Cloud Computing Environment”, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR), 2014.

LIST OF PUBLICATIONS

1. N. S. Raghava and Deepti Singh, “Comparative Study on Load Balancing Techniques in Cloud Computing”, Open journal of mobile computing and cloud computing, 2014.

2. Deepti Singh, Deepti Sarsawat and Dr. N. S. Raghava, “Analyzing Large-scaled Applications in Cloud Computing Environment”, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR), 2014.

3. Deepti Singh, N. S. Raghava, “Cloud Computing in Large-scaled Applications”, National Conference on Emerging trends in Communication and Biomedical Engineering (NCECB), 2014.