JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

Enhanced Throttled Algorithm in Cloud Computing

BETELHEM AKMEL SULEYEMAN

This Research Study Submitted To Department Of Computing, Institute Of Technology, Jimma University, In Meeting Partial Fulfillment For The Award Degree Of Master Science In Computer Networking.

December, 2018

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

BETELHEM AKMEL SULEYEMAN

I, hereby, declare that the materials contained in this thesis have not been previously submitted for a degree in this or any other university. I further declare that this thesis is solely based on my own research work. I also declare that all information contained in this document has been obtained and presented in accordance with academic rules and ethical conduct. I understand that my thesis may be made electronically available to the general public for reference purpose.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: Fisseha Bayu(PhD candidate)	_____	_____
Co-Advisor: Ruth Gashaw (MSc)	_____	_____

Dedication

To almighty

Acknowledgment

First and foremost, I would like to thank God and Virgin Mary for their valuable support to accomplish this thesis and being with me in all directions of my life.

I would like to express my deep, sincere and grateful thanks to my advisor, Fisseha bayu and for my co-advisor, Ruth gashawu for their constant encouragement, honest support, and dedicating their time from starting time till finalizing the work.

A special thanks to my family. Words can't express how grateful I am to my mother and my father for all of the sacrifices that you have made on my behalf. Your prayer for me was what sustained me thus far. I would also like to thank my fiancé and all of my friends. They were always supporting me and encouraging me with their best wishes towards my goal, without whose love, encouragement, and prayer I would not have finished this thesis.

Abstract

Recently, cloud computing become a new global trend of computing. It is a modern style of using the power of Internet and wide area network (WAN) to offer resources remotely. It's a new solution and strategy to achieve high availability, flexibility, cost reduction and on demand Scalability. However cloud computing has many challenges such as poor resource utilization which has deep impact in the performance of cloud computing. These problems arisen due to the huge amounts of information. So the need for efficient and powerful cloud computing load balancing algorithms is one of the most important issues in this area to improve the performance of cloud computing. Many researchers proposed various load balancing and job scheduling algorithms in cloud computing but there is still some inefficiency in the system performance and load still imbalance. Therefore, in this research an enhanced throttled load balancing algorithm is proposed to improve the performance and efficiency of throttled algorithm by distributing the incoming requests to the index table which contains only available (idle) virtual machines. This eliminates the need to queue those requests, which may in turn have the effect on the response time .the proposed algorithm allocates the incoming request in a very efficient and fair way. The performance of proposed algorithm is analyzed using Cloud Analyst simulator and compared with the original Throttled algorithm. The simulation result demonstrate that the proposed algorithm improve the response time of the request and data center processing time and also incoming request is distribute to all available virtual machine with-out over or under utilization

Keywords: Load Balancing, Resource Utilization, Cloud Computing, Virtual Machine, Cloud Analyst.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AMLB	Active Monitoring Load Balancing
DC	Data Center
DCC	Data Center Controller
IAAS	Infrastructure as a Service
NIST	National Institute of Standards and Technology
PAAS	Platform as a Service
RR	Round Robin
SAAS	Software as a Service
SLA	Service-Level Agreement
TLB	Throttled Load Balancing
UB	User Base
VM	Virtual Machine
VMM	Virtual Machine Monitor

Chapter 1: Introduction

1.1Overview of Cloud Computing

Computing is being altered into a model where users access services based on their requirements regardless of where they are hosted. Several computing models have promised to deliver the above mentioned services and cloud computing is one among them. Cloud Computing refers to distribution of pool of resources among users through internet according to use. This technology works on pay per use basis where users have to pay for the services they have used which reduces the cost [1]. The basic services of cloud computing are, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Cloud computing is also having different deployment models. Which are classified into four category. Private Cloud, Public Cloud, Community Cloud and Hybrid Cloud [2, 4, 7]. One of the major challenges of cloud computing is load balancing. Handling and servicing millions of requests from the users arriving at the data center controller and distributing the load fairly across all the available machines is called load balancing. The main aim of load balancing is to achieve high resource utilization, to increase the availability of services, to increase the user satisfaction, to maximize resource utilization, to reduce the execution time and waiting time of task coming from different location, to improve the performance, maintain system stability, build fault tolerance system and accommodate future modification [3, 9]. There are two types of load balancing algorithms: static load balancing algorithm and dynamic load balancing algorithm. In static algorithms, load is assigned to different machines by considering their processing capabilities. Static algorithms are mostly suitable for homogeneous and stable environments and can produce very good results in these environments. However, they are usually not flexible and cannot match the dynamic changes to the attributes during the execution time [1, 6]. In dynamic algorithms, load is assigned to different machines by considering changes in every machine run time conditions and provide better results in heterogeneous and dynamic environments.

1.2 Statement of the Problem

The aim of cloud computing is to reduce capital and operational cost, better performance in terms of response time ,data processing time, and better utilization of its resource. One of the ways to improve this is by providing efficient and effective resource utilization through VM load balancing policies. Load balancing is distributing the load fairly across all the available machines and the main aim of load balancing is to help in achieving high resource utilization and user satisfaction. But, there are various VM load balancing policies which are famous on current cloud environment. The most popular algorithms are Round-robin, Active Monitoring load balancing and Throttled load balancing algorithm.

Round-robin uses the round robin scheme for allocating job. It selects the first node randomly and then, allocates jobs to all other nodes in a round robin fashion so it uses the given resource efficiently but this algorithm doesn't check the current status of VM. That means whether the VM is busy/available or check the current load of VM. It simply put the request to each VM in round way. As a result, most of the requests have to wait more time to execute in addition to execution time. So the overall execution time of the request will be the sum of waiting time and execution time so that this algorithm has maximum response time and data center processing time.[12].

Active Monitoring Load Balancer maintains information about each VMs and the number of requests currently allocated to each VM. When a request to allocate a new VM arrives, it identifies the least loaded VM but each and every time from the first index of table and assigns the task to that VM. This may cause that one VM to be allocated in continuous manner if it is least loaded [5].

Throttled load balancing algorithm is an algorithm when load balancer maintains an index table of virtual machines as well as their states (available or busy). The client/server first makes a request to data center to find a suitable virtual machine (VM) to perform the recommended job. The data center queries the load balancer for allocation of the VM. The load balancer scans the index table from the top until the first available VM is found or the index table is scanned. So there is a huge gap between efficiency of load balancing

algorithm and better resource utilization for better load balancing in the cloud [14].

Most of the existing algorithms may work efficiently but they use resources inefficiently. Other algorithms use the resource efficiently and distribute the incoming request fairly to each VM but they are not fast enough to give response. Thus, there is a need to develop an algorithm due to the aforementioned reasons. The proposed algorithm solves the problem by distributing the incoming requests to the container which contains only available (idle) virtual machines without having to search them. This eliminates the need to queue those requests which may in turn have the effect on the response time .the proposed algorithm allocates the incoming request in a very efficient and fair way. It assign the request efficiently and effective resource utilization. In addition to these the resource can be used effectively so that each VMs have equal load during execution time.

1.3 Objectives

1.3.1 General objectives

The general objective of this research is to propose enhanced throttled load balancing algorithm for cloud computing.

1.3.2 Specific objectives

In order to achieve the general objectives, a set of specific objectives are set. These specific objectives include:

• review literature in the area of cloud computing and more specifically load balancing for cloud computing.

• understand the research gap in the area of load balancing for cloud computing.

• propose load balancing algorithm for cloud computing.

• plan and implement the proposed algorithm

• test and evaluate the proposed algorithm using simulator

• draw conclusion and pass on recommendations for additional research area.

1.4 Research Questions

This study aims to answer the following research question

1. What are the parameters used to evaluate performance of load balancing algorithm in cloud computing?

1.5 Scope and Limitations

The scope of this research is to investigate well known existing load balancing algorithm and discover their shortcomings to modify the algorithm which can take care of the problem of efficiency in load balancing. For benchmarking and evaluation purpose Throttled Algorithm is chosen. The performance of the proposed algorithm will be measured using simulator in homogeneous environment which means each virtual machine had the same bandwidth, processor speed, the same number of CPU . Above and beyond, this research work will not address other issues of cloud computing other than load balancing and also the load balancing algorithm will be developed under data center level that means the request type, size and routing between client node and data centers will not be considered.

1.6 Significance of the study

This research work focused on improving some of the existing load balancing algorithm in cloud computing. The existing load balancing algorithms have their problem. For example: Round Robins load-balancing algorithm does not check the current status of each VMs whether it is busy or available. This leads to load imbalance [7]. So this study is significant for solving such kind of problems. From this point of view the new algorithm is significant to solve such a problem in the existing load balancing policy. So the study provides a good benefit for any cloud service provider as well as cloud service user and also the study will become a motivation for conducting other researches on this area.

1.7 Methods/Approaches

1. Problem identification
2. Review the literature: this includes concepts and theories and also previous research findings
3. Design the proposed solution
4. Select simulation tool and performance evaluation metrics
5. Describe simulation methodology: this includes simulation configuration performance evaluation metrics.
6. Implement the designed approach using simulation tool
7. Analyze the result
8. Evaluate the proposed work based on the selected performance evaluation metrics.

1.8. Organization of the Thesis

This research is organized as follows:

Chapter one is introduction about the study and statement of the problem together with its objective and the methodology used in this study. The second chapter is literatures review around cloud computing in detail and discuss about related works on load balancing for cloud computing. The third chapter discuss about the proposed algorithm including its flow chart and the steps to be followed. The fourth chapter is all about experimentation and results. Therefore in this chapter the study tried to show the analysis of the experimentation in line with the discussion of the results obtained from the CloudAnalyst simulation tool. The fifth chapter is about conclusion and future works or recommendation.

Chapter 2: Literature Review

This Chapter deals with review of important concepts concerning the research work. It provides detailed definition of cloud computing, its main characteristics, cloud computing service model and cloud computing deployment model. In addition to this it discusses cloud computing issues more specifically about load balancing issues in cloud computing and existing load balancing algorithms for cloud computing, as well as it also describes some of the existing cloud simulators.

2.1 Overview of Cloud Computing

Cloud computing is an emerging computing paradigm which promised to provide opportunities for delivering a variety of computer applications in a way that has not been experienced before. It is a relatively recent term even though it was built upon some existing concepts. There are a number of definitions of cloud computing. According to the National Institute of Standard and Technology (NIST) Cloud computing is defined as a computing model used everywhere and provides convenient, on-demand access to a shared pool of computing resources such as networks, servers, storage, applications, etc. These resources can be dynamically assigned and released with minimal management effort or service provider interaction [15, 16]. The difference that cloud computing brings compared to traditional concepts of “grid computing”, “distributed computing”, “utility computing”, or “autonomic computing” is to broaden horizons across organizational boundaries. Cloud computing is so named because the information being accessed is found in the "clouds", and does not require a user to be in a specific place to gain access to it. It is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment [17].

Figure2. 1. Cloud Computing Vision [16]

2.1.1 How does Cloud Computing works?

Cloud computing contains many different components as the service is large and huge but to understand how does cloud computing work; imagine that the cloud consists of layers, the **back-end** layers and the **front-end** layers. The front-end layers appears in the user side and used to see and interact with cloud computing environment. For instance when the client try to access his Facebook account, the client uses software running on the front-end to communicate with cloud environment and this software may be web browser or other[10]. The back-end of cloud consists of the hardware and the software architecture that manages all incoming request and provide cloud service to fulfill the context cloud computing.

Clouds use a network layer to connect users' end point devices, like computers or smart phones, to resources that are centralized in a data center. Users can access the data center via a company network or the Internet or both. Clouds can also be accessed from any location, allowing mobile workers to access their business systems on demand

Applications running on the cloud take advantage of the flexibility of the computing power available. The computers are set up to work together so that it appears as if the applications were running on one particular machine. This flexibility is a major advantage of cloud computing, allowing the user to use as much or as little of the cloud resources as they want at short notice, without assigning any specific hardware for the job in advance [10].

2.1.2 Cloud computing architecture

Generally speaking, the architecture of a cloud computing environment can be divided into 4 layers: the hardware/ datacenter layer, the infrastructure layer, the platform layer and the application layer, as shown in Fig.2.2. [40].

a. **The hardware layer**: This layer is responsible for managing the physical resources of the cloud, including physical servers, routers, switches, power and cooling systems. In practice, the hardware layer is typically implemented in data centers. A data center usually contains thousands of servers that are organized in racks and interconnected through switches, routers or other fabrics. Typical issues at hardware layer include hardware configuration, fault tolerance, traffic management, power and cooling resource management [21, 40].
b. **The infrastructure layer:** Also known as the virtualization layer, the infrastructure layer creates a pool of storage and computing resources by partitioning the physical resources using virtualization technologies such as Xen, KVM and VMware. The infrastructure layer is an essential component of cloud computing, since many key features, such as dynamic resource assignment, are only made available through virtualization technologies [21, 40].
c. **The platform layer:** Built on top of the infrastructure layer, the platform layer consists of operating systems and application frameworks. The purpose of the platform layer is to minimize the burden of deploying applications directly into VM containers. For

example, Google App Engine operates at the platform layer to provide API support for implementing storage, database and business logic of typical web applications [21, 40].

d. The application layer: At the highest level of the hierarchy, the application layer consists of the actual cloud applications. Different from traditional applications, cloud applications can leverage the automatic-scaling feature to achieve better performance, availability and lower operating cost. Compared to traditional service hosting environments such as dedicated server farms, the architecture of cloud computing is more modular. Each layer is loosely coupled with the layers above and below, allowing each layer to evolve separately [21, 40].

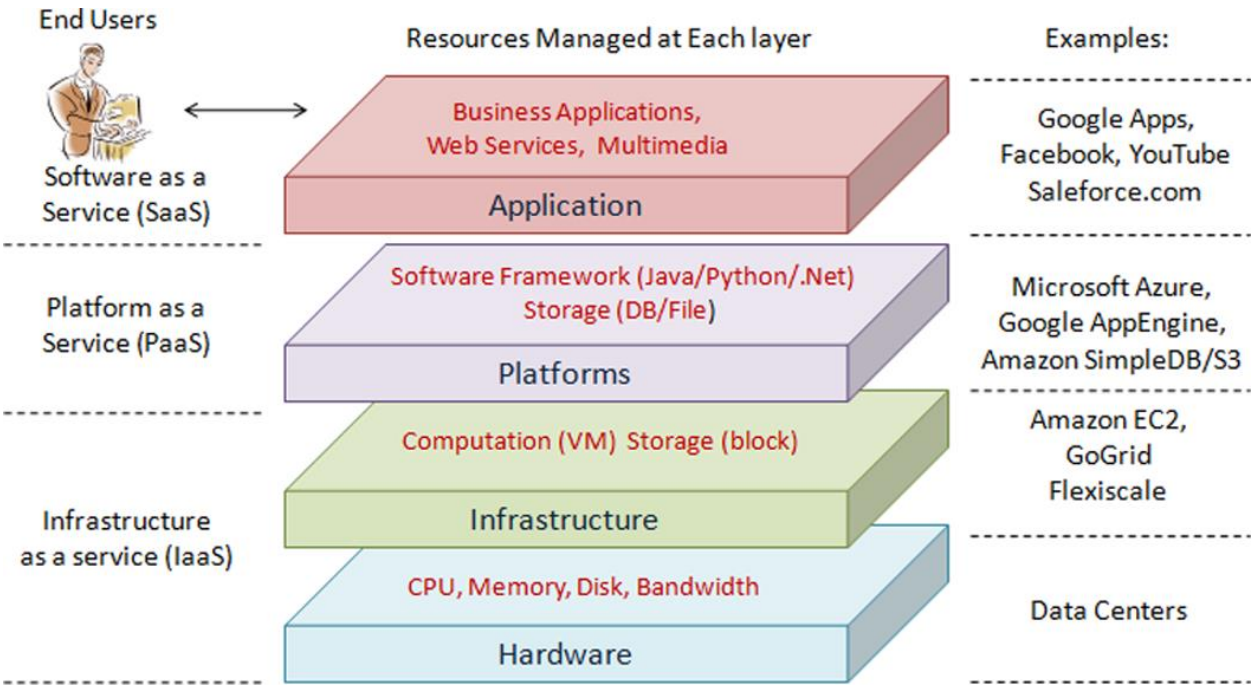

Figure2. 2 cloud computing architecture[21]

2.1.3 Characteristics of Cloud Computing

Advance of Cloud Computing is huge as for individual uses and business employments. Clients of cloud computing can use or keep up the online resources. Among several advantages or benefits few of them are discussed below [17, 22].

a. **Rapid elasticity**: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

b. **Resource pooling**: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

c On-demand self-service: Users can automatically provision their own computing resources as needed and without requiring human intervention, typically through an interactive portal that enables them to configure and manage these services themselves.

d. Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

e. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

2.1.4 Cloud Service Model

Cloud gives services in three diverse ways.

Software as a service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) which are recorded as follows [18, 24].

a. **Software as a service (SaaS):** It is the ability to offer services over the internet to the end user. These services can be used from different devices which can run upon different platforms. The user does not require to manage cloud infrastructure, including storage, servers, network operating systems Google, Sales force, Microsoft, Zoho are some of the company that provides SaaS..

b. **Platform as a service (PaaS):** It is the ability to offer services over the internet to the end user i.e. the intermediary stage is to install on the cloud Infrastructure, the user develop or access applications developed using programming languages and tools supported by the service provider. The user does not require to manage the underlying cloud infrastructure, including storage, servers, network operating systems Google's App Engine, Force.com, etc. are some of the popular PaaS examples.

c. **Infrastructure as a servie (IaaS):** It is the ability to provisioning and re-provisioning the resources such as storage, networks, operating system to the end user as per the requirement. Where the user is capable to install and execute different software, which includes operating systems and applications. Some common examples are Amazon, GoGrid, 3Tera.

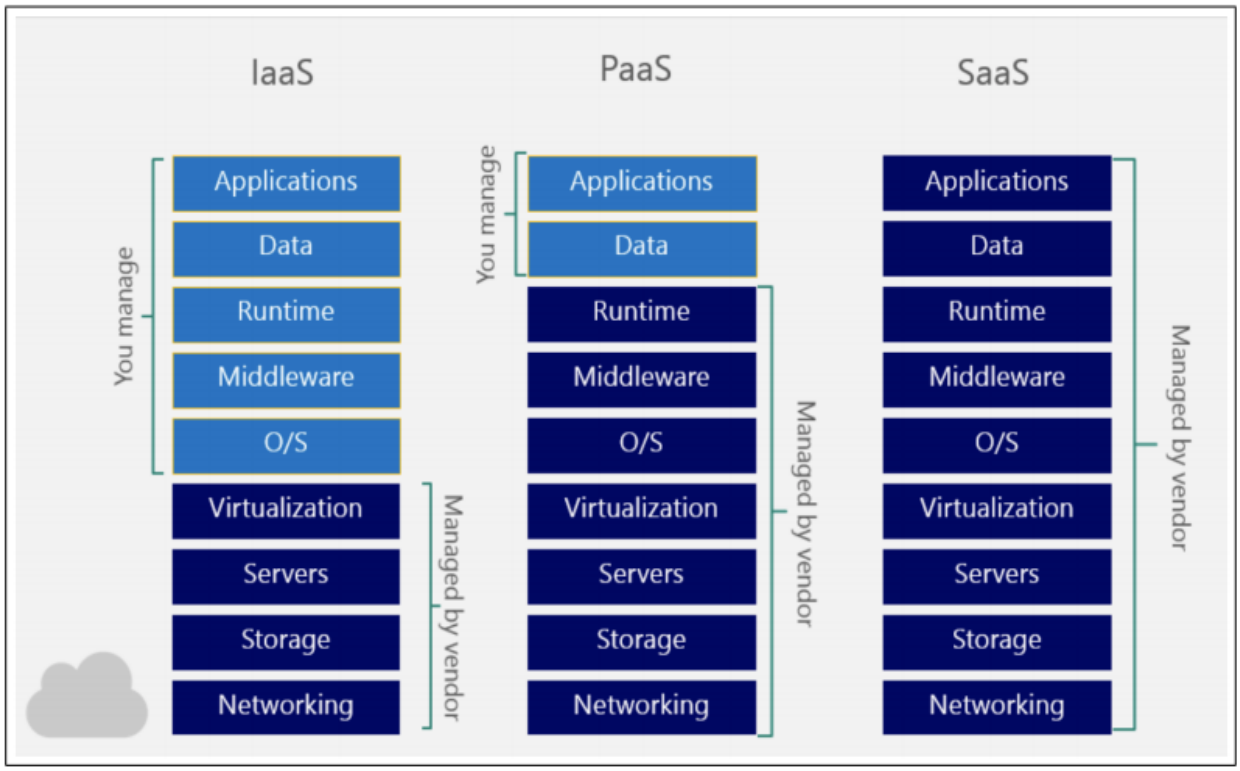

Figure2. 3 cloud computing services[24]

2.1.5 Cloud Deployment Model

Cloud supplier can offer either open or private or hybrid cloud and community cloud. They are In charge of working of the cloud and every one of them are discussed beneath [16, 19, 25].

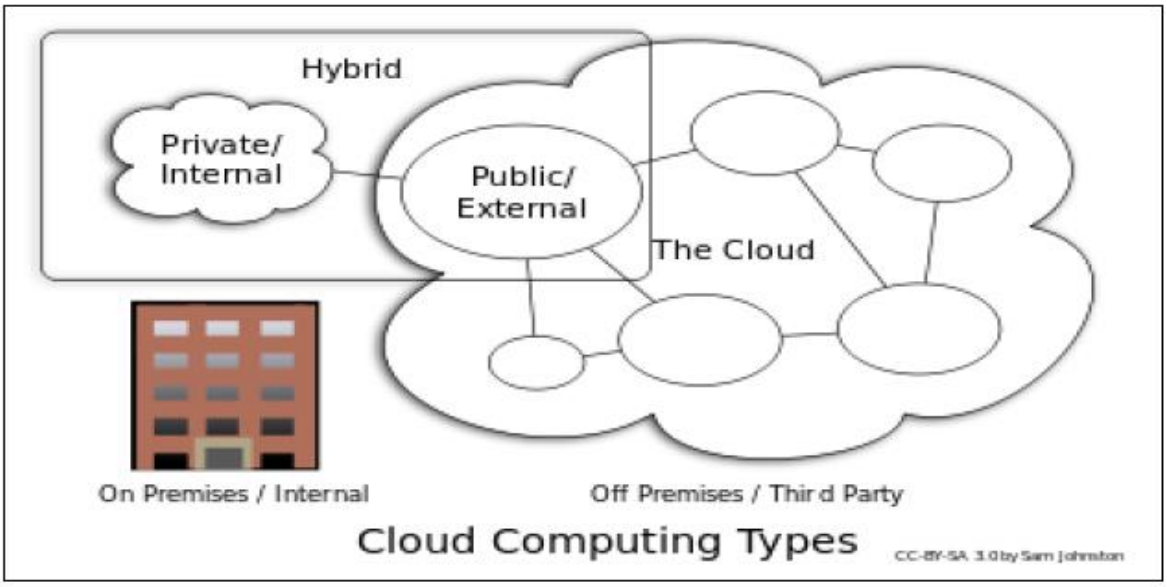

Figure2. 4 Cloud Computing Types[19]

a. **Private cloud:** Also known as internal clouds, private clouds are designed for exclusive use by a single organization. A private cloud may be built and managed by the organization or by external providers. A private cloud offers the highest degree of control over performance, reliability and security. However, they are often criticized for being similar to traditional proprietary server farms and do not provide benefits such as no up-front capital costs.

b. **Public cloud**: cloud in which service providers offer their resources as services to the general public. Public clouds offer several key benefits to service providers, including no initial capital investment on infrastructure and shifting of risks to infrastructure providers. However, public clouds lack fine-grained control over data, network and security settings, which hampers their effectiveness in many business scenarios.

c. **Community cloud:** The cloud infrastructure is shared by several organizations and supports a specific community that has common objective. It may be handled by the organizations or other party to smoothly access the services of cloud.

d. Hybrid cloud: The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability. In this model client typically outsource non business critical information & processing to public cloud, while having control on critical information and data.

2.2 Virtualization

Virtualization separates resources and services from the underlying physical delivery environment[13]. Virtualization is considered as a core of cloud computing technologies and one of the most important technologies that enabled this paradigm [25] [29]. Virtualization hides a computing platform's physical characteristics from users [25] [29]. It allows abstraction and isolation of lower level functionalities and underlying hardware. This enables portability of higher level functions and sharing and/or aggregation of the physical resources [20].Virtualization means "something which isn't real", but gives all the facilities of a real[23]. It is the software implementation of a computer which will execute different programs like a real machine[21].

Virtualization has three characteristics that make it very related with cloud computing which are[13]:

1- Partitioning:

By partitioning the available resources, many applications and operating systems can run in a single physical system.

2- Isolation:

By isolation, each virtual machine can run in its host with others virtual machine without effect on others. So, if one virtual instance failed, it doesn't affect the other virtual machines.

3- Encapsulation:

A virtual machine encapsulated and stored as a single file, so a virtual machine can be presented to an application as a complete entity without interfere with another application.

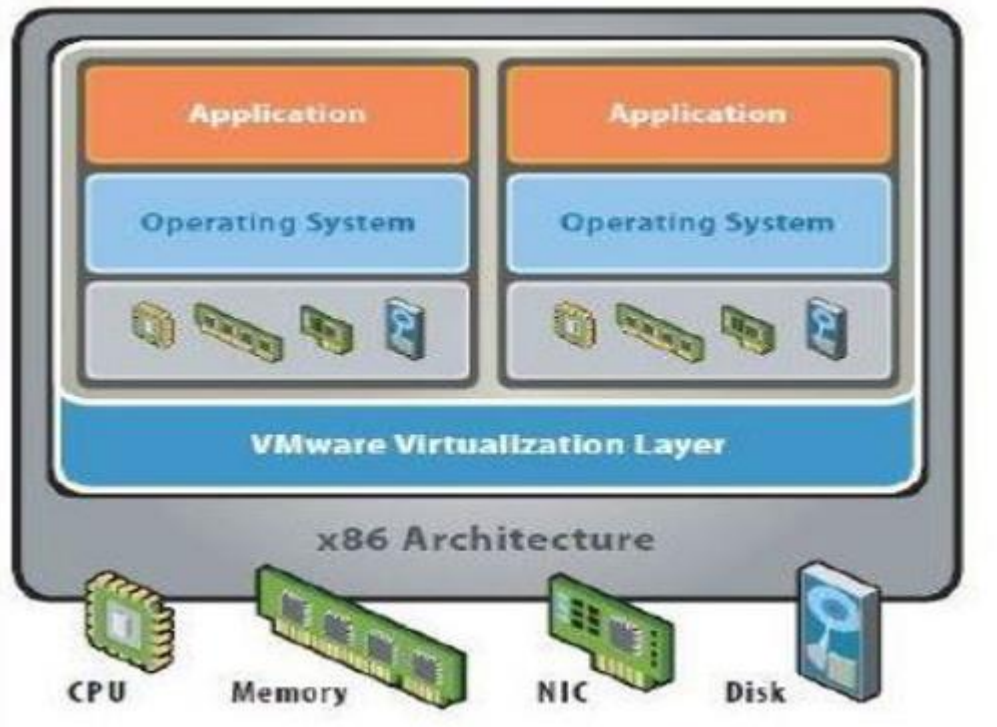

Figure2. 5. Virtualization Architecture[13]

There are two sorts of virtualization on account of cloud computing:

a. **Full virtualization:** In Full virtualization a total establishment of one machine is finished on another machine. It will bring about a virtual machine which will have all the software i.e. introduce in the actual server. It serves to sharing a PC system among various clients and it additionally gives copying equipment on another machine [19]. Full virtualization has been successful for several purposes as pointed out in [19].
    - Sharing a computer system among multiple users
    - Isolating users from each other and from the control program
    - Emulating hardware on another machine

b. Para virtualization: In Para virtualization the hardware permits multiple operating systems to keep running on single machine. It likewise permits effective utilization of system resources, for example, memory and processor. Again in Para virtualization every one of the services are not completely accessible rather the administrations are given partially it has the following advantages as given in [19]:

- Disaster recovery: In the event of a system failure, guest instances are moved to hardware until the machine is repaired or replaced.

- Migration: As the hardware can be replaced easily, hence migrating or moving the different parts of a new machine is faster and easier.

- Capacity management: In a virtualized environment, it is easier and faster to add more hard drive capacity and processing power. As the system parts or hardware can be moved or replaced or repaired easily, capacity management is simple and easier.

2.3 Cloud computing issues and challenges

There are issues and difficulties that rose up out of cloud computing that should be tended to. The major one is security which assume the most essential part in Cloud computing. Security issues such as information misfortune, phishing, protection and different dangers, whether at the enterprise level or person level that utilization the pooled processing resources in cloud computing, has presented new security challenges. Along these lines, there is a need for novel systems to decrease the effect of the unlimited risks in the cloud computing condition and the second one is performance, in cloud computing poor performance can be caused by absence of resources, for example, disk space, constrained bandwidth, lower CPU speed, memory, network connections and so on. The information escalated applications are more testing to give legitimate resources and poor performance can bring about end of service delivery, loss of clients and reduces incomes. Performance can be founded on various strategies, tools and simulations for cloud environments and tools like Cloud Analyst. There is a progression of factor that influence the execution, for example, Security, Recovery and Fault resilience, Service level agreements, Bandwidth,

Storage capacity[26,27]. The other real issue of cloud computing is load balancing which is the focal point of this investigation.

2.4 Load balancing in Cloud Computing

Load balancing is one of the major issues in cloud computing environment. Load balancing is a mechanism that distributes the workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. The increase in web traffic and different application in the web world is increasing day by day where millions of data are created every second and load balancing is needed on CPU load, memory capacity and network. If there is a failure of any node or host system in the network, it will lead to isolation of web resource in the web world. Load balancing in such situation should be able to provide availability and scalability [22].

Balancing work load on cloud helps to achieve a high user satisfaction and resource utilization ratio, Hence improving the overall performance and resource utility of the system. Load balancing is the process of assigning the total loads to the individual nodes of the collective system to make the best response time and also good utilization of the resources [23]. Some measurement parameters that can be used to evaluate the load balancing techniques, which allow us to check whether the given technique or algorithm of load balancing is good enough to balance the load or not are [27];

a. **Scalability:** is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. [28].

b. **Resource Utilization:** is used to check the utilization of re-sources. It should be optimized for an efficient load balancing [28].

c. **Performance** is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays [28].

d. **Response Time:** is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. And for efficient load balancing, this parameter

should be minimized. For minimum response time, the resource utilization should be improved [28].

e. **Overhead Associated:** determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and intercrosses communication. This should be minimized so that a load balancing technique can work efficiently [28].

f. **Throughput:** It is the number of task executed in the fixed interval of time. To improve the performance of the system, throughput should be high [28].

g. **Point of Failure**: the system should be designed in such a way that the single point failure does not affect the provisioning of services. Like in centralized system, if one central node is fail, then the whole system would fail, so load balancing system must be designed in order to overcome this problem [28].

h. **Data Center Processing Time**: is the amount of time taken by the data center to execution a single task. It should be minimum with effective resource utilization [28].

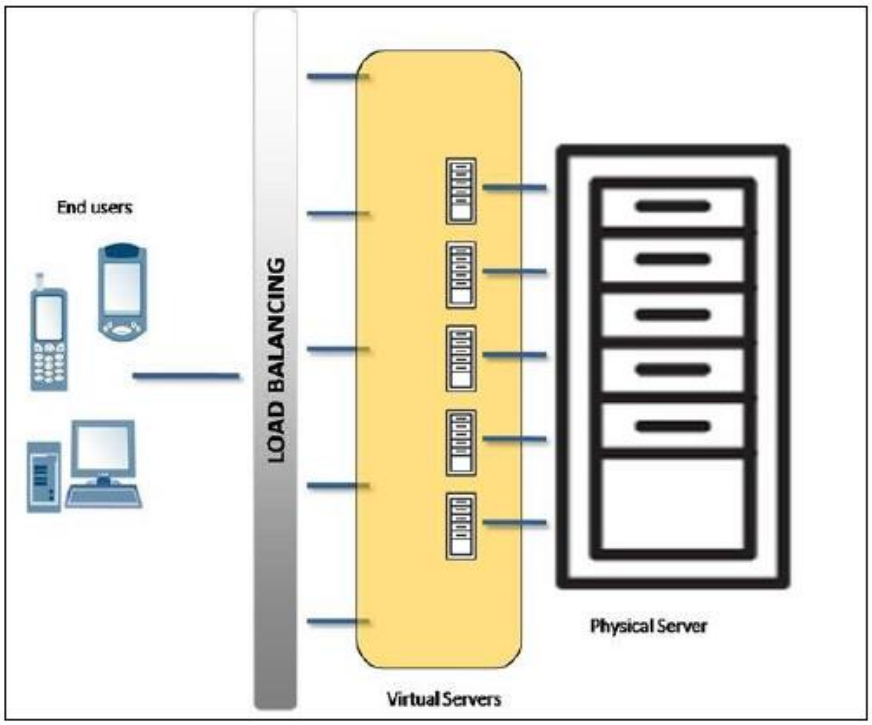

Figure2. 6. Load Balancing In Cloud Computing[22]

There are two main categories of load balancing algorithm.

a. **Static:** In this approach load is assigned to different machines by considering their processing capabilities. Static load balancing algorithm divides the traffic equivalently between all users. It use only information about the average behavior of the system, ignoring the current state of system. Static load balancing algorithms are simpler because there is no need to maintain and process system state information [29].

b. **Dynamic:** Dynamic load balancing algorithm collects information and run time conditions of machines and according to gathered characteristics assign and dynamically reassign the load among machines [29].

2.5 Cloud Simulators

The main aim of simulator is to test the implementation work in the absence of the required environment [35]. The following are some of the simulators used in the cloud environment and this section discuss about them.

a. CloudSim: CloudSim is a framework developed by the GRIDS laboratory of University of Melbourne which enables seamless modelling, simulation and experimenting on designing Cloud computing infrastructures. CloudSim is a self-contained platform which can be used to model data centers, service brokers, scheduling and allocation policies of a large scaled Cloud platform. It provides a virtualization engine with extensive features for modelling the creation and life cycle management of virtual engines in a data center. CloudSim framework is built on top of GridSim framework also developed by the GRIDS laboratory [16].

b. GridSim: GridSim toolkit was developed by Buyya et al to address the problem of near impossibility of performance evaluation of real large scaled distributed environments (typically Grid systems but also P2P networks) in a repeatable and controlled manner. The GridSim toolkit is a Java based simulation toolkit that supports modelling and simulation of heterogeneous Grid resources and users spread across multiple organizations with their own policies. It supports multiple application models and provides primitives for creation of application tasks, mapping of tasks to resources and managing such tasks and resources [26, 37].

c. CloudAnalyst: CloudAnalyst seems to be easy as it has a graphical user interface with which it makes easy to use tool and also have a level of visualization capability which is even better than just a tool-kit. CloudAnalyst separates simulation set up environment exercise and supports the modeler to focus on the parameters used for simulation purposes rather than the programming technicalities only. It also supports a modeler to perform simulations continually by modifying the parameters easily, quickly and in less time [30]. Cloud Analyst Simulator has the following Features

i. Simple to Use: CloudAnalyst is easy in setting up as it comes with the java package. [36, 37, 38].

ii. GUI based Output: GUI based output which comprises of tables (rows and columns), graphs and charts are highly desired to review a number of results which are obtained at the time of Cloud Analyst simulation. Such GUI based presentation helps in understanding and identifying the important outlines of the parameters and also helps in their comparison [36, 37, 38].

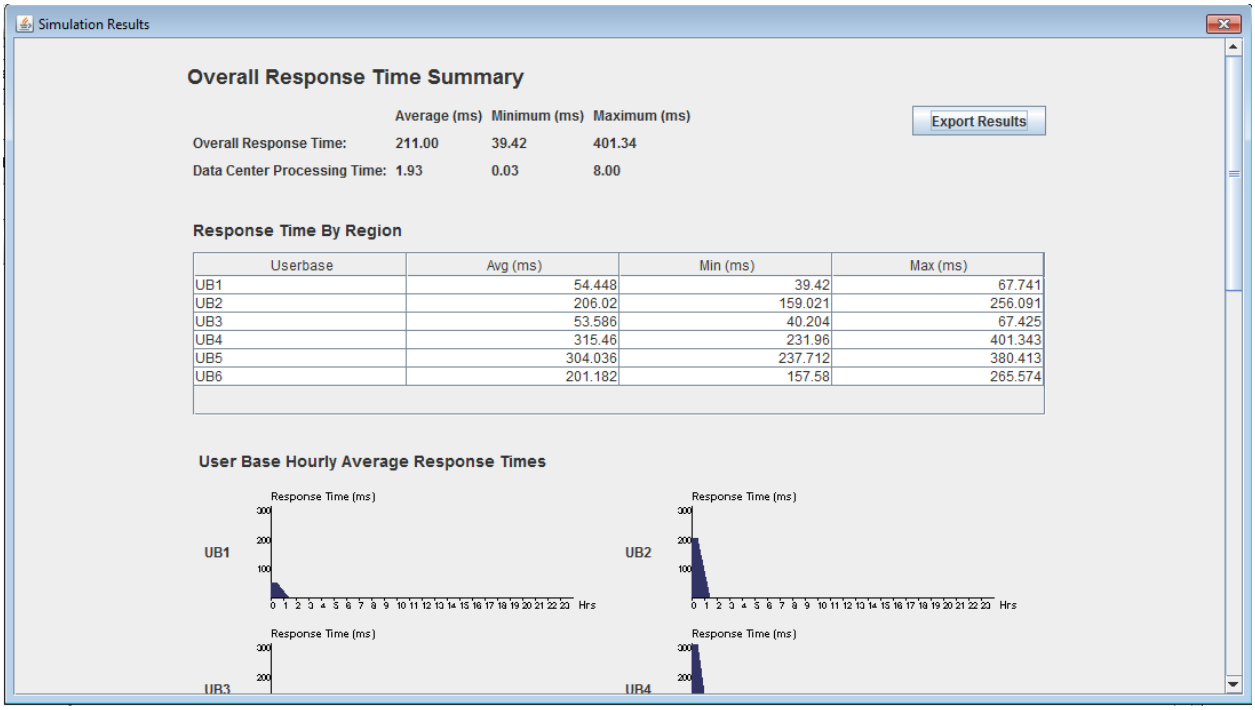

Figure2. 7 Sample Cloud Analyst simulation output window[37]

iii. Ability to Repeat: CloudAnalyst has the ability to repeat the experiments. This is a very significant requirement of any simulator. With cloudAnalyst simulator, if one experiment having some parameters, on simulation, produces some results then these results will be same each time the same simulation is executed with same parameters in the same experiment. Without this, the simulation just seems to be a random sequence of events and not a controlled experiment [36].

iv. Ability to Save the Results: CloudAnalyst also has the option to save the results. This is helpful as it can save the experiment (along with the set of all input parameters and values taken during simulation) as a file. This file can then be saved on the system (PC) or the same can be taken into flash drives or pen drives to some other computers at different locations [36].

Main Components of CloudAnalyst Simulator are listed as follow:

i. **Regions:** CloudAnalyst simulator splits the whole world into six regions. These regions coincide with six main continents in the World. Data centers and user bases, which are also other main entities, related to some of these areas. [37,38].

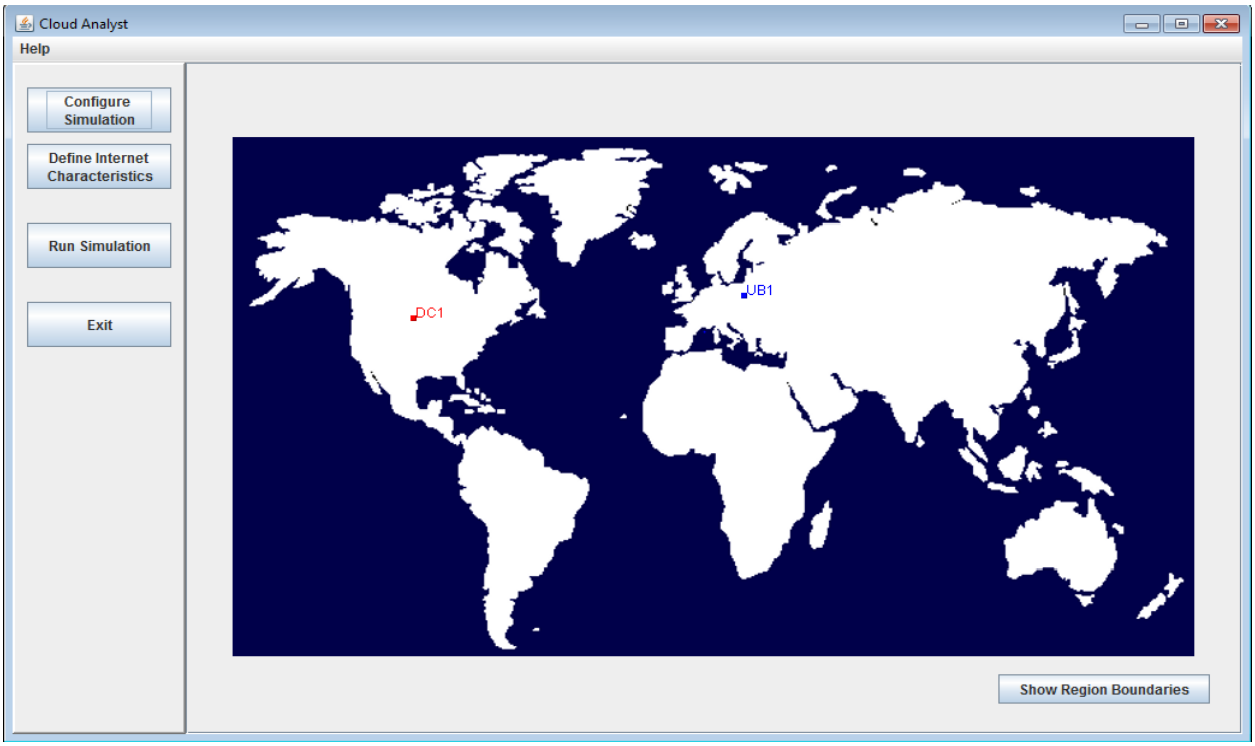

Figure2. 8. Main Screen with Simulation Panel[38]

ii. Internet Settings: Internet in the cloudAnalyst simulator is the idea for the actual real world Internet. This implements all those features which are important only to that particular simulation. CloudAnalyst also manages to create the replicas of the Internet traffic, which seems to be routing around all over the sphere through hosting appropriate amount of data

transmission delays and the transfer latency. The user can also configure the transmission latency and all the existing bandwidth among six areas (the whole world is divided into regions) [37].

iii. Service Broker: Service Broker manages all traffic that is moving between the data center's and the user bases. The decision about servicing of the data center from each user base is done by the Service Broker. CloudAnalyst equipment presently provides three types of service brokers and all these three types of service brokers implement different routing policies. The three service brokers are: closest data center, optimize response time and dynamically reconfigured [37].

iv. User Bases: The collection of users, considered to be as one individual unit in the cloudAnalyst, which are involved in the simulation, are said to be the User Bases. The main responsibility of user bases is generating traffic for cloudAnalyst simulation, with which the simulation considers being in the real time. Thousands of users can be represented as a single User Base but all these users are configured as a single unit i.e. a single user base. Traffic generated in these user bases will be the simultaneous bursts depending on the capacity of the respective user base. User Base can be chosen by the modelers in order to signify each user to increase the efficiency of the simulation, generally, the user bases are used to be considered as they represent a large number of users and user base configuration is shown in figure 2.5 [37].

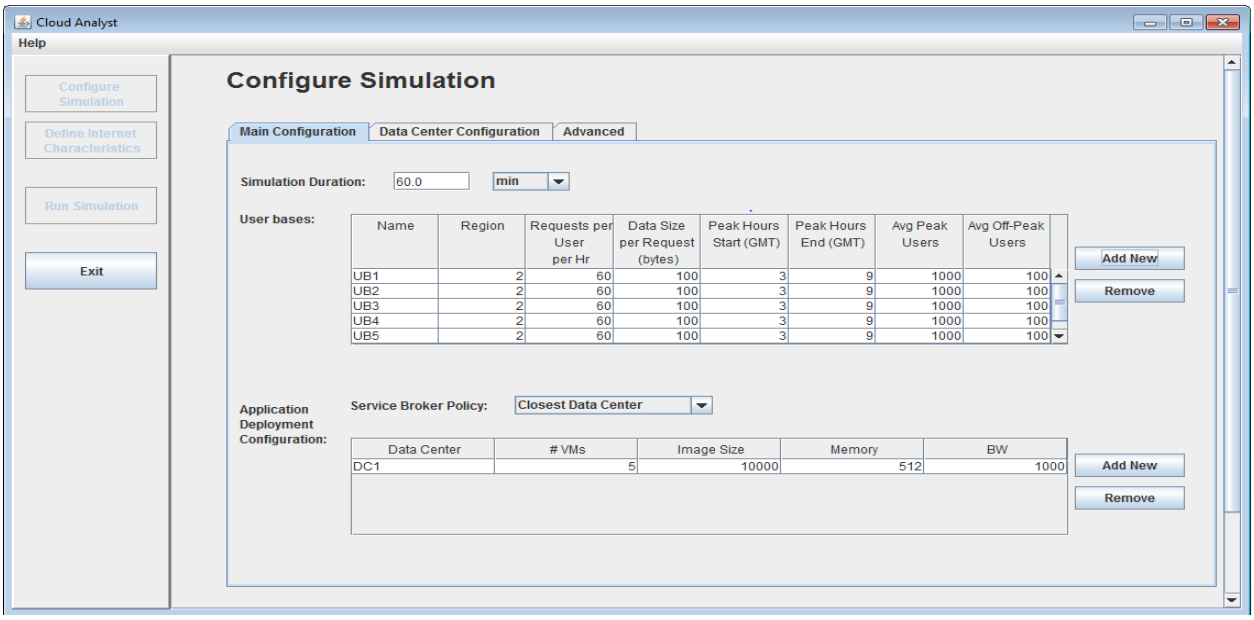

Figure2. 9 User base configuration and Application Deployment Configuration[37]

v. Data Center Controller: This Controller proves to be very indispensable part in cloudAnalyst. One particular cloudSim is mapped to each data center controller. Data center controller manages the activities performed by the data center. These activities are like the creation of VM's etc. Data center controller also routes user requests that are received from respective user bases with the help of the Internet to the virtual machines. This can also be observed as the facade which is used by the cloudAnalyst for accessing and functioning the main part of cloudSim toolkit and the data center configuration is shown in figure 2.10 with its physical hardware details [37].

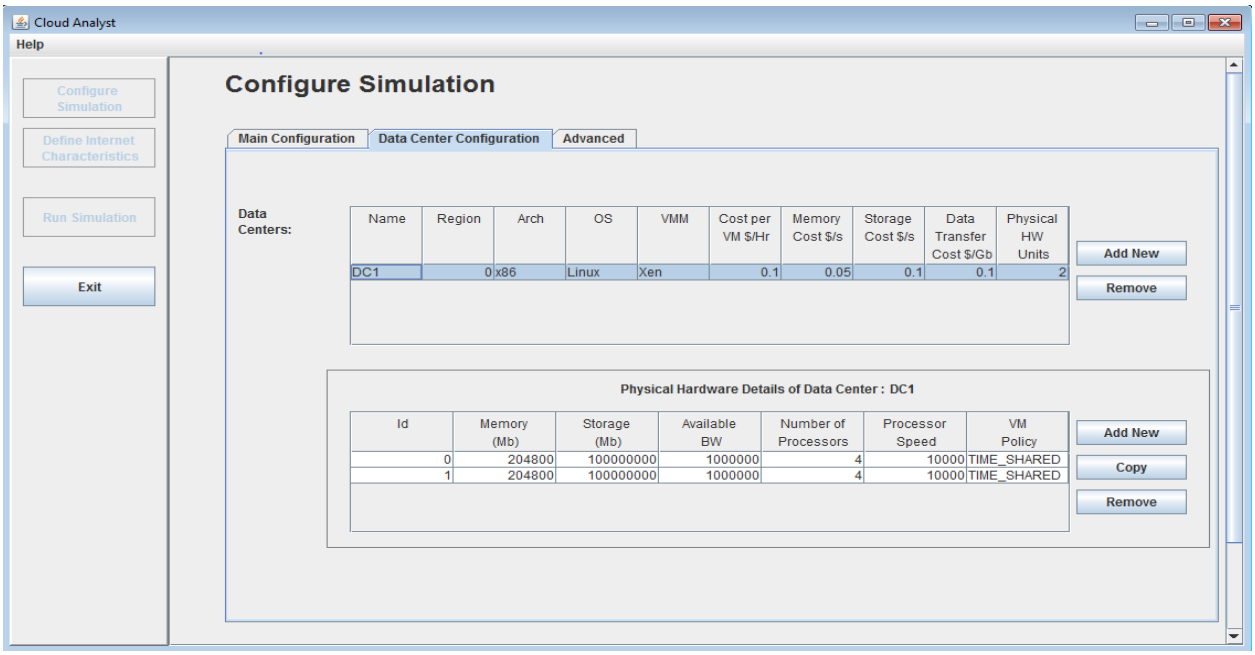

Figure2. 10 Data center Configuration in Cloud Analyst[37]

vi. VM Load Balancer: The data center utilizes the virtual machine load balancer. The data center controller makes use of virtual machine load balancer to govern the particular virtual machine allocated for processing of the upcoming cloudlet. Presently cloud Analyst has three VmLoadBalancers which implements three different load balancing procedures and their policies can be further selected by the modeler according to the requirement. The three different VM load balancers are active monitoring, round-robin and throttled load balancer [37].

vii. Application Deployment Configuration: This is a table which lists the number of VMs that are allocated for the application in each data center from the Data Centers tab, along with the details of a VM. The fields included in this table are Data Center which is a drop down listing the names of data centers created in the data center tab, number of VMs to be allocated to the application from the selected data center, image size a single VM image size in bytes, amount of memory available to a single VM, and amount of bandwidth available to a single VM [38].

Viii. Advanced tap: in advanced tap the simulator has user grouping factor in User bases which enable us to set the number of simultaneous users from a single user base and the

parameter must be one in the ideal scenario but it increase the response time unrealistically. The parameter must be one in the ideal scenario but it increase the response time unrealistically.

Request grouping factor in Data Centers: which is used to set the number of simultaneous requests a single application server instance can support. In the ideal scenario this should be equal to 1. Executable instruction length per request which is used to set the length of instruction Load balancing policy: in this option the available load balancing policy will be listed and it enable us to select the load balancing policy for the required activity.

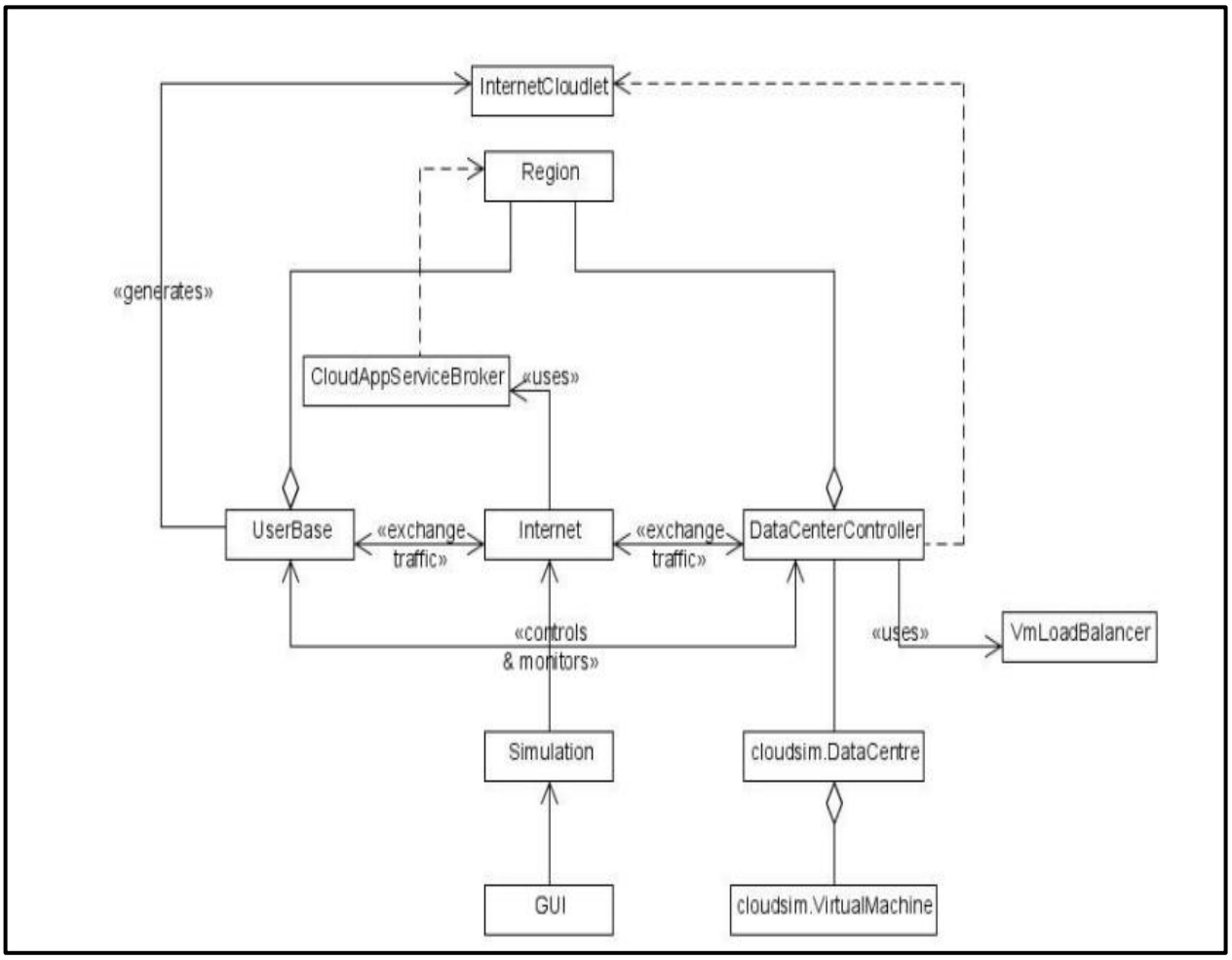

Figure2.11. Summary Of The Main Components And Domain Entities Of The Cloud Analyst[36]

Table2. 1. Summary of cloud simulators[37]

Cloud Simulators	**Description**
GridSim [26,27]	Supports modelling and simulation of heterogeneous Grid resources or it mainly focus on Grid computing systems
CloudSim[16]	Built on the top of GridSim and it covers most of the activities taking place within a Data Center. CloudSim is a self-contained platform which can be used to model data centers, service brokers, scheduling and allocation policies of a large scaled Cloud platform.
CloudAnalyst [37]	CloudAnalyst which is built on the top of CloudSim and cloudsim features are also directly used in the CloudAnalyst by adding new extensions and some of them are Application users, Internet, Simulation defined by time period, Service Brokers, GUI based output and Ability to save simulations and results in the form of pdf

2.5 Summary

In this chapter cloud computing is defined, its main characteristics and cloud deployment models are defined. In addition to this, cloud computing issues specifically load balancing techniques and challenges are discussed.

For better performance of cloud computing load balancing plays a very important role as result of this different authors propose various types of load balancing algorithm for efficient resource utilization and user satisfaction by improving some of the measurement parameters.

Chapter 3: Related Work

Different researchers propose various load balancing algorithm to address load balancing issues of cloud computing. In this section the existing well known load balancing algorithm with their advantage and drawbacks is discussed.

The author in [31] propose Round Robin (RR) load balancing algorithm and this algorithm works in a round robin fashion to improve resource utilization. When request arrives to the data center then the first request is assign to the virtual machine randomly but the subsequent requests are assigned in a circular order and once the virtual machine is assigned request it goes to the end of the virtual machine list. The main advantage of this algorithm is work load distributions between processors are equal. And its main drawback is the job processing time for different processes are not the same and again if the VM is not free then incoming job should wait in the queue and this leads to higher response time.

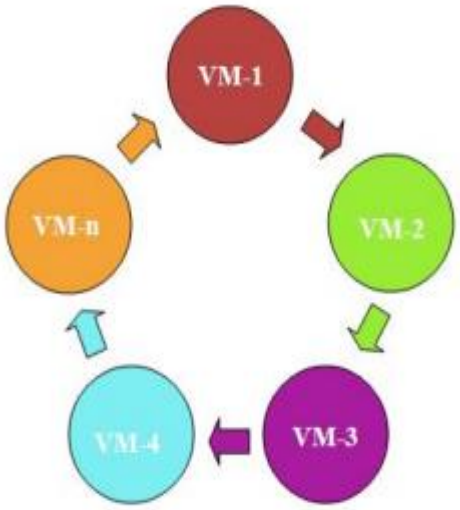

Figure3. 1 Round Robin algorithm(RR)[31]

The work in [33] propose new load balancing algorithm based on round robin algorithm, they made a modification on round robin algorithm by implementing a dynamic time

Quantum based on algorithm execution round. The result shows an improvement in response time as compared to normal round robin algorithm. But they need to compares their results with other algorithms such as active monitoring and Throttled in order to evaluate the results.

Similarly the work in [42] proposed Weighted Round Robin: It is the modified version of Round Robin in which a weight is assigned to each VM so that if one VM is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, the Data Center Controller will assign two requests to the powerful VM for each request assigned to a weaker one. The major issue in this allocation is same as that of Round Robin that is it also does not consider the advanced load balancing requirements such as processing times for each individual requests .

Fuzzy logic with Round-robin algorithm was proposed in [41]. The algorithm maintains the information about each VM with number of tasks currently allocated to each VM and the processor speed about each VM. When a new request arrives, the proposed algorithm search for the least loaded VM and assign the incoming task to that VM, but if there are more than one list loaded VM, then the proposed algorithm compare with their processor speed and select the best VM which have best processor and assignment is performed using fuzzy logic. The proposed algorithm is evaluated using CloudSim simulator. This algorithm improved the data center processing time and response time. In addition, the results referred that its performance is better than RR algorithm. The drawback of this paper is the researchers evaluate the performance of proposed algorithm with only Round-robin algorithm because Round-robin algorithm is improved and enhanced by many researchers

The author in [32] propose Random load balancing algorithm. The idea of this algorithm is to randomly assign the selected jobs to the available Virtual Machines (VM). when request arrives to the data center, the data center request the load balancer for appropriate virtual machine then load balancer peak the virtual machine randomly and send id of that virtual machine to the data center, the data center allocate the request to the virtual machine identified by that id. The advantage of this algorithm is that its complexity is quite low since it does not need any overhead or pre-processing in order to assign requests to the virtual

machine. The main problem of this algorithm is it does not take into considerations the status of the VM which will either be under heavy or low load and this may result in the selection of a VM under heavy load and in this algorithm resource utilization become bad and also the job requires a long waiting time before service is obtained.

The author in [36] proposes an algorithm which is called Throttled load balancing algorithm to assign requests to the available VM. In this load balancing algorithm, an index table having the id of VM along with its state either as available or busy is maintained. When request arrives to data center then the data center queries the load balancer for allocation of the task. Then the load balancer scan index table from top until the first available VM is found or it will scan the index table fully. After that it will return id of that available VM to data center controller and assign the request to the VM identified by that id. The data center controller acknowledge the load balancer for the allocation and the load balancer update the status of that VM to busy and when processing the task finished then the data center controller again inform the load balancer about deallocation then the load balancer update status of the virtual machine to available. While processing the request, if no available VM is found then the load balancer return -1 to the data center controller and the task has to wait in queue. Advantage of throttled algorithm is it checks status of the virtual machine before assigning request and its main drawback is always scan the index table to find the available virtual machine from its first index each and every time and this maximize the response time.

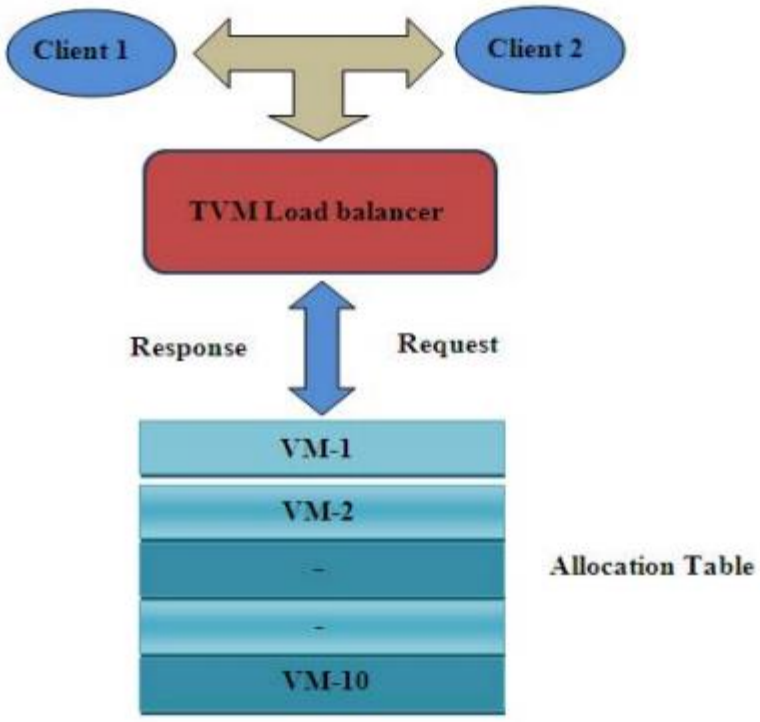

Figure3. 2 Throttled Algorithm(TLB)[33]

The work in [39] proposed modified throttled algorithm. It also maintains an index table containing a list of virtual machines and their states. The first VM is selected in the same way as in Throttled. When the next request arrives, the VM at index next to already assigned VM is chosen depending on the state of the VM and the usual steps are followed, unlike that of the Throttled algorithm, where the index table is parsed from the first index every time the Data Center Queries Load Balancer for allocation of VM. It gives better response time compare to the previous one. But in index table the state of some VM may change during the allocation of next request due to deallocation of some tasks. So it is not always beneficial to start searching from the next to already assigned VM.

The work of [23] propose an algorithm that combine the advantages of three algorithms and overcomes their disadvantages. These three algorithms were: greedy, round robin, and

power saver algorithm. The algorithm focused on best utilization of resources and minimizing the power consumption. It scheduled the VMs to the nodes depending on their priority value, which varies dynamically based on their load factor. When a request is received, the node with the maximum available resource is determined and then it is checked whether the node had a load factor less than 80%. If the highest priority node had a load factor less than 80%, then the VM is scheduled to that node, otherwise it checks the next maximum resource. The idle nodes (which, no VM is allocated to them) are turned off to save power. The main drawback of this algorithm is in some case such as the high workload, the power saver algorithm will be inactive because all the VM would be busy in most of the processing time and this would affect the performance.

In [37] the authors propose an algorithm that depend on ant colony technic. Ants depend on the strength of the ant's pheromone to select the optimal path that leads to their destination. In the same way each node in the network has a pheromone. Each row in the pheromone table represents the routing preference for each destination, and each column represents the probability of choosing a neighbor as the next hop. If an ant is at a choice point when there is no pheromone, it makes a random decision. If the pheromone exists, the node with high probability is selected and then the pheromone table is updated by increasing the probability of this node and decreasing other nodes probabilities. The main drawback of this algorithm is that it does not consider the current workload information for each node. So in some case some nodes may be heavily loaded whereas others remain idle.

According to the work in [34] the author propose load balancing algorithm called Active Monitoring Load Balancing (AMLB) Algorithm. This algorithm maintains information about each VM and the current number of load each virtual machine holds. When a request to allocate a new VM arrives, data center controller queries the load balancer for allocation of the task and the load balancer scan index table from top to find least loaded VM among the list of virtual machine. Then it will return id of that least loaded VM to data center controller and the data center controller assign request to the virtual machine identified by that id after that data center controller acknowledge the load balancer about allocation then the load balancer increment current load of that virtual machine by one. When processing

the request is finished then data center controller again acknowledge about the deallocation to the load balancer and load balancer decrement current load of that virtual machine by one.

Advantage of AMLB is it maintains information about current load of each virtual machine. But its main drawback is when request arrives to data center it will scan the index table again and again from its first index to end of the table to find least loaded VM as result of this associated overhead is high which in turn increase response time and in addition to this, AMLB it does not check whether the VM is recently used or not as result of this one VM is assigned in a continuous manner if it is least loaded and this will result in unfair distribution of load among the virtual machine.

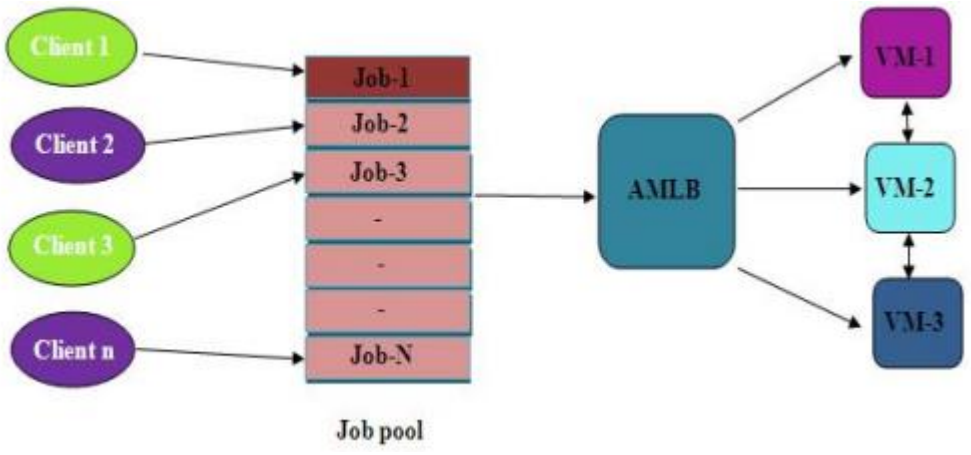

Figure3. 3. Active monitoring VM load balancing algorithm (AMLB) [34]

Similarly, the work in [35] improves Active monitoring VM load balancing algorithm, Weighted Active monitoring load balancing algorithm . This algorithm uses the concept of weights in active monitoring. The VM are assigned varying (different) amount of the available processing power of server/ physical host to the individual application services. To these VMs of different processing powers; the tasks/requests (application services) are assigned or allocated to the most powerful VM and then to the lowest and so on according to its weight and its availability. The drawback of this algorithm is it starts scanning from the first index of the table again and again to find the Powerful VM as result of this the overhead computation becomes high and this in turn increases response time.

Summary

In this chapter some of the existing load balancing algorithm with their advantages and drawbacks is discussed. It is shown that existing load balancing algorithm has some deficiency and this affect the performance of cloud computing. So in order to overcome this problem there is a need to develop efficient load balancing algorithm and in this research efficient enhanced throttled load balancing algorithm is proposed to overcome response time problem.

Chapter 4: Proposed Algorithm

4.1 Overview

This chapter discusses about the proposed Enhanced Throttled load balancing algorithm for cloud computing.

.In Throttled load balancing algorithm, requests are assigned to the available VM. In this load balancing algorithm, an index table having the id of VM along with its state of either available or busy is maintained. When request arrives to the datacenter, then the datacenter queries the load balancer for allocation of the task. Then the load balancer scans the index table from top until the first available VM is found or it will scan the index table fully.

4.2Proposed Algorithm (Enhanced Throttled Load Balancing Algorithm)

The Enhanced Throttled Load Balancer carry out load balancing by taking the set of available virtual machines in an available group. When a new request comes just check for best suited virtual machine. Once the request is bound with the virtual machine, then remove this VM index from the group of available virtual machines so it will not be considered for any future request until it finishes its assigned workload and becomes available again by setting its status to be free.when the table is empty (all VMs unavailable) Enhanced Throttled Load Balancer will return a value of -1 to the Data Center Controller to wait until Available VM is found in the table.

So, the proposed algorithm avoid searching from the first index until it gets the first available virtual machine unlike the throttled algorithm which starts scanning the index table from the first index every time request comes to the data center .The process is shown in Figure 4.1

```
Input: available VM List(VM_id),maintain an index of
available VMs ,
     assigned VM List(VM_id),maintain the index of busy Vms
     TempVMid is  the id selected for assignment
Output: VMid is the VM id that is selected to assign the
load.
1. Initialize, available VM List (0..n-1) ← All VM's are not
allocated yet ,VM_id ←-1, TempVMid ← -1;
2.    If available VMid exist in available VM List(VM_id )
then
     VMid ←TempVMid
     TempVMid ← remove(available VMList())
```

Figure4. 1Pseudocode for proposed algorithm

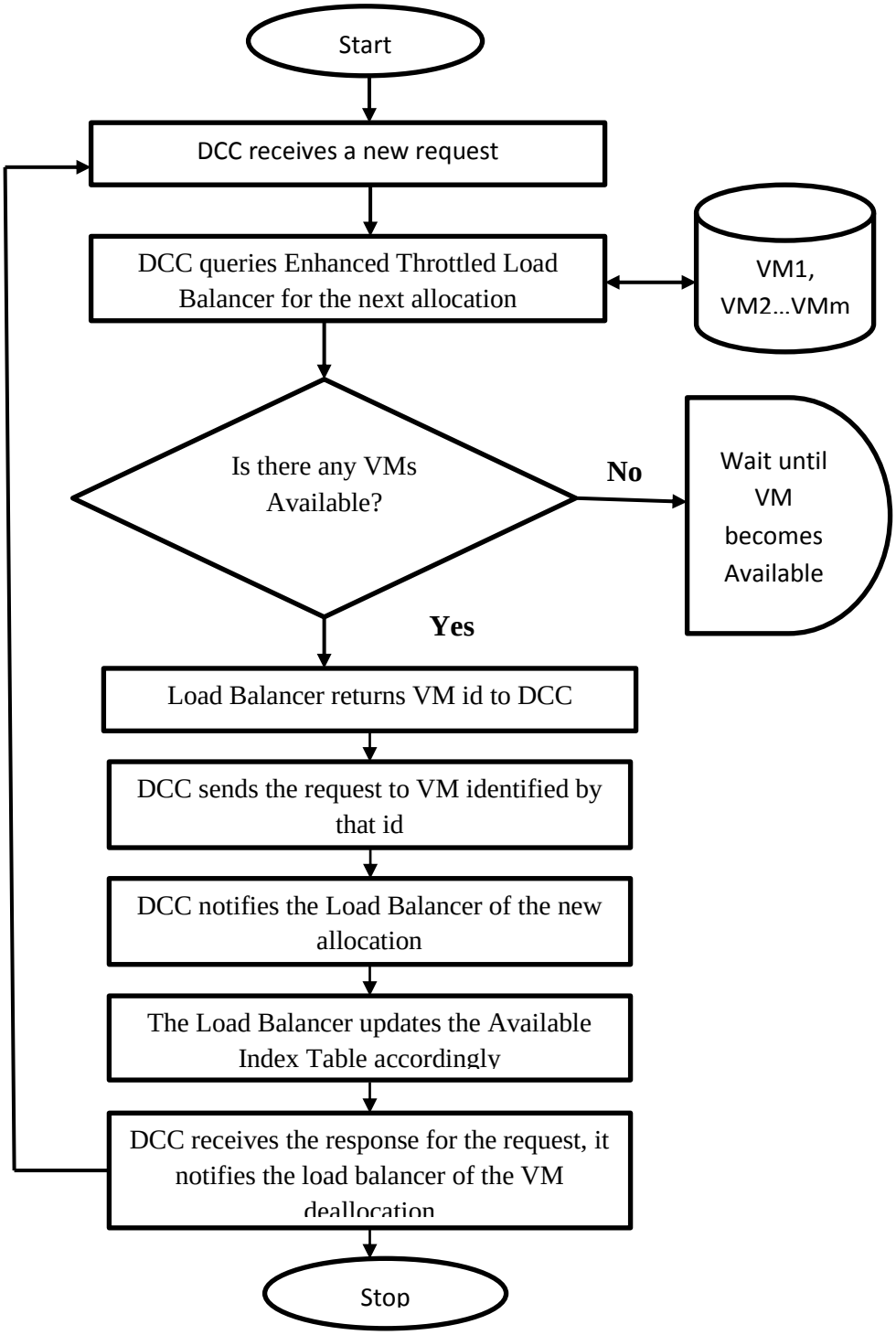

Figure4. 2. Flow chart of the proposed algorithm

4.3 Implementation

The proposed enhanced throttled load balancing algorithm has been implemented in CloudAnalyst simulator by programmatically extending the core framework provided in the CloudAnalyst. After implementing the proposed load balancing algorithm, configure the CloudAnalyst simulator for testing and evaluation. And in order to do this first we configure user base with their geographic distribution, and other properties like the number of users during peak and off-peak hour, peak hour time, number of request per hour and request size in byte in Configuration window on Main Tab then define the data centers that is used in the simulation with all the hardware and accounting aspects of the data centers in data Centers tab of the configuration screen. Once the data centers have been created, VMs are allocated in them for the simulated application using the Main tab of the Configurations screen. A data center defined above does not get included in the simulation unless it is allocated in this step after reviewing and adjusting the advanced parameters in the advanced tab of the configuration screen and run the simulation after filling the required data and select the VM load balancer. The algorithm is tested in homogeneous environment and each virtual machine had the same bandwidth, processor speed, the same number of CPU and tested the algorithm by considering the effect of large number of user request from each user base. Finally the efficiency of the algorithm is tested in terms of response time and processing time. original throttled algorithm were implemented as well.

4.4 Evaluation

Different parameters are available to evaluate the performance of load balancing algorithms and in our research, two measurement parameters were used.

1. Response time: it is the time taken by the load balancing technique to respond to a certain request from the client.

2. Data processing time: which is the time taken by the load balancing technique to process a certain request.

Chapter 5: Experiments, Results and Discussion

5.1 Introduction

In this chapter the proposed experiment results which are conducted in order to test efficiency of Enhanced throttled load balancing algorithm in terms response time is dicussed. For analysis the data from Facebook users is used because, the present generation has embraced the use of social networks in many roles both personal and commercial. In order to manage such a high traffic, cloud technology has been adapted.

A typical large scale application on the Internet that can benefit from Cloud technology is social networking applications. This application may benefit from Clouds because they typically present non-uniform usage patterns. Access to such services varies along the time of the day, and geographic location from sources of service requests also varies. Furthermore, a new functionality in the service may cause a sudden increase interested by the service, leading to an increase in number of requests arriving to servers that may be only temporary. Cloud allows infrastructures to dynamically react to increase in requests, by dynamically increasing application resources, and reducing available resources when the number of requests reduces. So, SLAs between Cloud providers and consumers are met with a minimal cost for consumers.

Hypothetical applications like Facebook users, Twitter users, Internet users are considered for experimentation. Six different geographical locations (six different continents of the world) are considered. One well-known social networking site is Facebook, which has over 2.19 billion registered users worldwide. On December 31 2017,the approximate distribution of the Facebook user base across the globe is shown in the table.

Table5. 1 Approximate distribution of fb user

Regions Divided in GUI Screen	**Region Id In Cloud Analyst**	**Number Of Users (Million)**
North America	0(R_0)	379,080,100
South America	1(R_1)	266,583,100
Europe	2(R_2)	340,891,620
Africa	3(R_3)	935,421,000
Asia	4(R_4)	177,005,700
Oceania	5(R_5)	194,630,548

A single time zone is considered for all user locations. For simplicity one hundredth of the total users from each continent is considered and it is assumed that only 1% of total users are on line during peak hours and and 10% of this number are on the network during off peak hours, as shown in the table 5.2.

Table5. 2 Define 6 user bases representing the 6 region

User Base	**Region**	**Simultaneous Online Users During Peak Hours**	**Simultaneous Online Users During Off- Peak Hours**
North America	0	37908	3791
South America	1	26658	2666
Europe	2	34089	3409
Asia	3	93542	9354
Africa	4	17701	1770
Oceania	5	19463	1946

5.2 Experimentation

This experiment investigates the performance of enhanced throttled algorithm with respect to different number of request from each user.

Table5. 3 Sample User base Configuration

Name	Region	Requests per User per Hr	Data Size per Request(bytes)	Peak Hours start(GMT)	Peak Hours End(GMT)	Average peak User	Average Off-Peak Users
UB1	0	100	1000	3	9	37908	3791
UB2	1	100	1000	3	9	26658	2666
UB3	2	100	1000	3	9	34089	3409
UB4	3	100	1000	3	9	93542	9354
UB5	4	100	1000	3	9	17701	1770
UB6	5	100	1000	3	9	1946	195

Table5. 4 sample Application Deployment Configuration

Data Center	**#VMs**	**Image Size**	**Memory**	**BW**
DC1	20	10000	1024	1000

Table5. 5 sample Data center Configuration

Physical hardware unit	**Memory(Mb)**	**Storage(Mb)**	**Available BW**	**Number of Processors**	**Processor speed**	**VM Policy**
1	204800	100000000	1000000	4	10000	TIME_SHARED

Table5. 6.Result of average response time

Algorithms	**Number of cloudlets**					
	100	**200**	**300**	**400**	**500**	**600**
Throttled	426.51	433.26	437.20	443.42	452.77	463.22
Enhanced Throttled	425.83	431.50	433.90	438.62	446.44	450.32

Table5. 7Result of Average Processing time

Algorithms	**Number of cloudlets**					
	100	**200**	**300**	**400**	**500**	**600**
Throttled	106.56	110.02	117.75	126.42	133.77	141.92
Enhanced Throttled	105.84	108.26	114.65	122.62	128.44	135.56

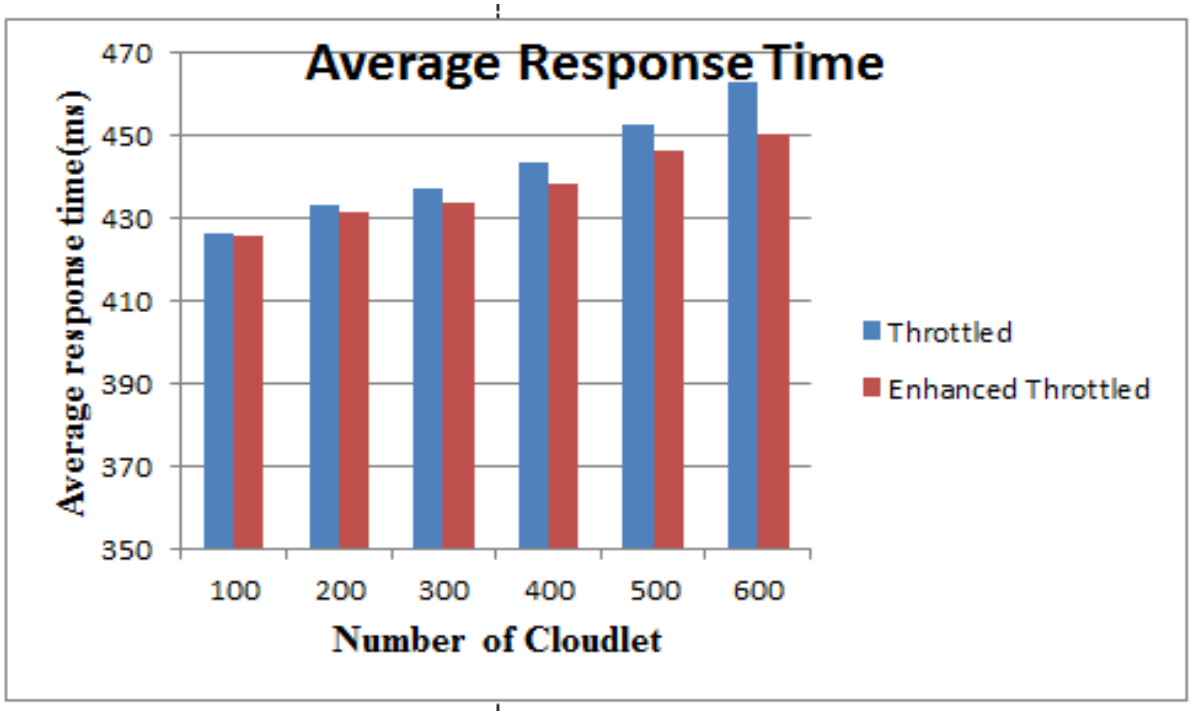

Figure5. 1 .Average Response Time Chart

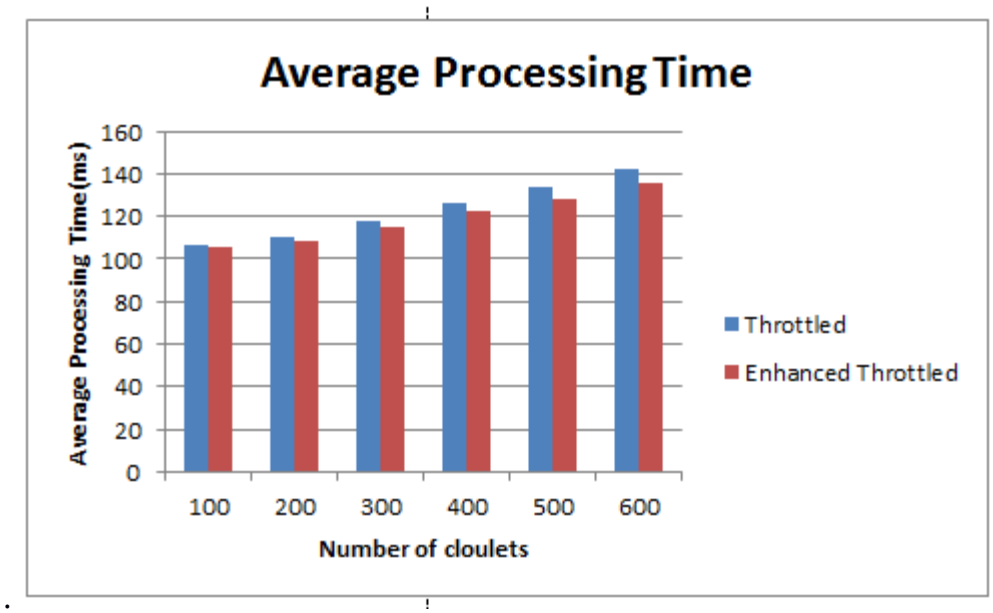

Figure 5. 2.Average Processing Time Chart

Discussion

The results showed that for experiment two as well the enhanced throttled algorithm has better response time and data center processing time than throttled algorithm. This was due to the efficiency of the algorithm. The algorithm became more efficient even when the number of request is large. Data center processing time is improved on the enhanced throttled algorithm as compared to throttled algorithm. This is because it is obvious that whenever the response time is improved it also decrease the processing time and also the enhanced throttled algorithm has no complex computation to takes decision to allocate VM. As for the Throttled algorithm, the scanning of virtual machines in the state index table from the beginning of the table will lead to the status of requests to queue when the system has large number of requests. With the Enhanced throttled algorithm, it distributes requests to the available virtual machines without having to search for them. This eradicates the need to queue up the system, improving the processing time of the data center.

From experimental results conducted, it clearly shows that with the enhanced throttled algorithm, the number of requests that have to queue has decreased, as well as the processing time of the data center and the response time of the system is improved than the throttled algorithm.

Chapter 6: Conclusion and Future Work

6.1. Conclusion

In this thesis, a load balancing algorithm which handles the response time problem for the incoming request has been introduced. The enhanced throttled algorithm improves the performance and efficiency of throttled algorithm by distributing the incoming requests to the index table which contains only available (idle) virtual machines. Thus it avoids scanning the queue again and again.

To test the enhanced throttled algorithm Cloud Analyst simulator had been used. Several experiments had been conducted by considering different issues. For evaluation, two metrics had been chosen: Response time, processing time. The results had been compared with Throttled algorithm.

The experiment in this work is conducted for different number of requests. At first the response time were measure and next the processing time is measure. The results showed that the enhanced throttled algorithm have better response time and processing time as compared to the throttled algorithm. This is because the enhanced throttled algorithm selects the VM from only the available index table which contains only available virtual machine.

6.2. Future Work

Here are suggestions to be done in the future.

- For the enhanced Throttled Algorithm it is good to have a load migration technique in addition.
- It is good to test the proposed algorithm in a heterogeneous environment

Reference

[1] Malhotra, Rahul, and Prince Jain. 2013a. "Study and Comparison of CloudSim Simulators in the Cloud Computing."*The SIJ Transactions on Computer Science Engineering and its Applications (CSEA)* 1(4): 111–15.

[2] Aditya, Abhijit, UddalakChatterjee, and Snehasis Gupta. 2015. "A Comparative Study of Different Static and Dynamic Load Balancing Algorithm in Cloud Computing with Special Emphasis on Time Factor." *International Journal of Current Engineering and Technology* 5(3): 1898–1907.

[3] Cse, M Tech. 2012. "Comparison of Load Balancing Algorithms in a Cloud JaspreetKaur."*International Journal of Engineering Research and Application* 2(3): 1169–73.

[4] Sharma, Tejinder, and Vijay Kumar Banga. 2013. "Efficient and Enhanced Algorithm in Cloud Computing." *International Journal of Soft Computing and Engineering* 3(1): 385–90.

[5] Zhang, Qi, Lu Cheng, and RaoufBoutaba. 2010. "Cloud Computing: State-of-the-Art and Research Challenges." *Journal of Internet Services and Applications* 1(1): 7–18.

[6] Begum.S and Prashanth.C.S.R. 2013. "Review of Load Balancing in Cloud Computing." *IJCSI International Journal of Computer Science Issues* 10(1): 343–52.

[7] Dave, Stuti, and PrashantMaheta. 2014. "Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Environment." *International Journal of Computer Applications* 94(4): 23–29.

[8] DharmeshKashyap, JaydeepViradiya, ― A Survey Of Various Load Balancing Algorithms In Cloud Computing,‖ International Journal of Scientific & Technology Research, volume 3, issue 11, november 2014. Structure of cloud computing environment

[9] Rakesh Kumar Mishra ,SreenuNaikBhukya," Service Broker Algorithm for Cloud-Analyst," Rakesh Kumar Mishra et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014

[10] Hr, Ragini. 2015. "An Analytical Model to Efficiently Assess Data Centre Performance and QOS in Cloud." *International Journal of Multidisciplinary Research Development* 2(8): 140–42.

[11] Olivier B., Thomas B., Heinz D. 2013. "SATW White Paper Cloud Computing -." *Swiss Academy of Engineering Science*.

[12] DharmeshKashyap, JaydeepViradiya." A Survey Of Various Load Balancing Algorithms In Cloud Computing," International Journal of Scientific & Technology research,vol. 3, issue 11, november 2014

13. Sareen, P., *Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of*

*IT Governance in Cloud.*International Journal of Advanced Research in Computer Science and

Software Engineering, **3**(3): pp. 533-538,(2013).

[14] ShridharG.Damanal and G. Ram Mahana Reddy," Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines," 2014 IEEE.

15. Sajid, M. and Raza, Z. *Cloud Computing: Issues & Challenges*. in*International Conference on Cloud*.pp. 35-41, (2013).

[16] Sajjan R.S, BiradarRekhaYashwantra," Load Balancing and its Algorithms in Cloud Computing: A Survey ,"International Journal of Computer Sciences and Engineering,2017,Vol.1,pp.2347-2693.

[17] Dothang Truong, "How Cloud Computing Enhances Competitive Advantages: A Research Model for Small Businesses " Fayetteville State University,2010.

[18] Nikhil Rajeshirke, RohanSawant, SumeetSawant, HasibShaikh," Load Balancing In Cloud Computing,"International Journal of Recent Trends in Engineering & Research (IJRTER),Vol.03,Issue 03, *March 2017*

[19] Amandeep, VandanaYadav, Faz Mohammad," Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study," International Journal of Scientific Research Engineering & Technology,2014,Vol.3,Issue 1.

[20]. Vouk, M.A., *Cloud Computing – Issues, Research and Implementations.* Journal of Computing and

Information Technology - CIT: pp. 235-246,(2008).

[21]. Padhy, R.P. and Rao, G.P., *Load Balancing in Cloud Computing Systems*: Orissa, India, (2011).

[22] Qi Zhang, Lu Cheng, RaoufBoutaba," Cloud computing: state-of-the-art and research challenges ," The Brazilian Computer Society ,2010.

[23]. Tiwari, M., Gautam, K., and Katare, K., *Analysis of Public Cloud Load Balancing using Partitioning Method and Game Theory.* International Journal of Advanced Research in Computer Science andSoftware Engineering, **4**(2): pp. 807-812,(2014).

[24] Soumya Ray and Ajanta De Sarkar," execution analysis of load balancing algorithms in cloud computing environment**,**" International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2012,Vol.2, Issue 5

[25]. Marisol, G.-V., Cucinotta, T., and Lu, C., *Challenges in real-time virtualization and predictable cloud computing.* Journal of Systems Architecture (ELSEVIER): pp. 1-15,(2014).

[26] AmandeepKaurSidhu, SupriyaKinger," Analysis of Load Balancing Techniques in Cloud Computing,"
International Journal of Computers & Technology,2013,Vol.4, Issue 2.

[27] Hafiz JabrYounis,"Effcient load balncing algorithm in cloud computing," MSC. Thesis, Islamic University,Gaza,2015.

[28] DharmeshKashyap, JaydeepViradiya," A Survey Of Various Load Balancing Algorithms In Cloud
Computing," INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH,2014,Vol.3,Issue.11.

[29]. Kumar, S. and Aramudhan, M., *Performance Analysis of Cloud under different Virtual Machine*

*Capacity.*International Journal of Computer Applications, **68**(8): pp. 1-4,(2013).

30. Sethi, S., Anupama, S., and Jena, K., S, *Efficient load Balancing in Cloud Computing using Fuzzy*

*Logic.*IOSR Journal of Engineering (IOSRJEN), **2**(7): pp. PP 65-71,(2012).

31. Pathak*Computing Using Extended Honey Bee Algorithm.* IJRREST: International Journal of Research Review in Engineering Science and Technology, **1**(3): pp. 12-19,(2012).

, K.K., Yadav, P.S., Tiwari, R., and Gupta, T., *A Modified Approach for Load Balancing in Cloud*

[32] AthokpamBikramjit Singh, SathyendraBhat J, Ragesh Raju1, Rio DSouza," Survey on Various Load Balancing Techniques in Cloud Computing," Advances in Computing, 2017,Vol.2,pp.28-34.

[33] Harish Chandra, HimanshuBahuguna," A SURVEY OF LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING ," International Journal of Computer Engineering and Applications,2017,Vol.11, Issue 12.

[34] IsamAzawiMohialdeen," COMPARATIVE STUDY OF SCHEDULING AL-GORITHMS IN CLOUD COMPUTING ENVIRONMENT ," Journal of Computer Science,2013.

[35] Subhadra Bose Shaw,A.K. Singh," A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment," 5th International Conference on Computer and Communication Technology,2014.

[36] Manan D. Shah, Amit A. Kariyani, Dipak L. Agrawal," Allocation Of Virtual Machines In

Cloud Computing Using Load Balancing Algorithm," International Journal of Computer Science and Information Technology &Security,2013,Vol.3,No.1.

[37] Sharma, Prof Meenakshi. 2012. “Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm.” *International Journal of Advanced Computer Science and Applications* 3(2): 86–88.

[38] Patel, Durgesh. 2015. “International Journal of Modern Trends in Engineering and Research Efficient Throttled Load Balancing Algorithm in Cloud Environment.” *International Journal of Modern Trends in Engineering and Research* 02(03): 463–81.

[39] Shridhar G. Domanal, G. Ram Mohana Reddy, “Load Balancing in Cloud Computing Using Modified Throttled Algorithm”, IEEE,

International conference.CCEM 2013.

[40] Arif Ahmed, AbadhanSaumyaSabyasachi," Cloud Computing Simulators: A Detailed Survey and Future Direction ," 2014 IEEE International Advance Computing Conference.

[41] Malhotra, Rahul, and Prince Jain. 2013a. “Study and Comparison of CloudSim Simulators in the Cloud Computing.”*The SIJ Transactions on Computer Science Engineering and its Applications (CSEA)* 1(4): 111–15.

[42] Wickremasinghe, Bhathiya, Rodrigo N. Calheiros, and RajkumarBuyya. 2010. “CloudAnalyst: A Cloudsim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications.” *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*: 446–52.

[43] B. Rajkumar. 2009. “CloudAnalyst : A CloudSim-Based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments.” *Distributed Computing Project, Csse Dept.*, *University of Melbourne*: 433–659

[44] “Cloud Computing Simulators: A Detailed Survey and Future Direction” AbadhanSaumyaSabyasachi, Arif Ahmed *2014 IEEE International Advance Computing Conference (IACC)*