A
Dissertation
On

Task Scheduling in Cloud Computing

Submitted in Partial Fulfilment of the Requirement

For the Award of the Degree of

Master of Technology

In

Computer Science & Engineering

Submitted By
Veena Kushwaha
2K12/CSE/25

Under the Esteemed Guidance of
Mr. R. K. Yadav
(Assistant Professor)

DEPARTMENT OF COMPUTER ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

JULY, 2014

Department of Computer Engineering
Delhi Technological University
Delhi-110042

CERTIFICATE

This is to certify that the dissertation titled "**Task Scheduling in Cloud Computing**" is a bona fide record of work done at **Delhi Technological University** by **Veena Kushwaha, Roll No. 2K12/CSE/25** in partial fulfilment of the requirements for the degree of Master of Technology in Computer Science & Engineering. This work was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

(Mr. R. K. Yadav)
Assistant Professor & Project Guide
Department of Computer Engineering
Delhi Technological University

Date: _____

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all the people who have supported and encouraged me during the course of this project without which, this work could not have been accomplished.

First of all, I am very grateful to my project supervisor **Mr. R.K. Yadav** for providing the opportunity of carrying out this project under his guidance. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success. I am also thankful to all my friends specially **Komal, Kanishka, Kamal and Varun** for being there for me at all times. Above all, I would like to express my gratitude to my parents for their encouragement and support during the completion of this work. Last but not the least; I am grateful to Delhi Technological University for providing the right resources and environment for this work to be carried out.

Veena Kushwaha
University Roll no: 2K12/CSE/25
M.Tech (Computer Science & Engineering)
Department of Computer Engineering
Delhi Technological University

ABSTRACT

Recently Cloud computing has become successful and developed which, resulted many datacenter around the world. The maintenance of datacenter is necessary to the point of energy consumption and reliable operation. The datacenter maintenance can be associated with cloudlet scheduling approach by using virtualization technology.

In Cloud, virtualization technology is employed to integrate physical machines into a virtual resource pool, to control resources in centralized manner. Recent work considers various strategies with only taking into account, one specific problem of task scheduling without considering the other related problems. Provisioning customer applications in the Cloud while maintaining the application's required quality of service and achieving resource efficiency with power consumption issues are still open research challenges in Cloud computing. Hence, by considering other related problems of the task scheduling while schedule tasks can improve the resource utilization, fault tolerance, energy saving and throughput of the Cloud system.

For this reason, in this thesis, we proposed a new integrated task scheduling approach that takes into account the other related issues such as VM management and datacenter management. At first level the tasks are allocated to the optimal Virtual Machine by taking into account the earliest finish time and at the second level the Virtual Machines are allocated to Server as such manner so that some machines can be switched off in order to save energy. This strategy is also helpful in improving fault tolerance capability of datacenter by switching off server when they go to high thermal state.

This algorithm is implemented on CloudSim platform and the obtained experimental results show that the proposed algorithm runs efficiently, reducing the average execution time, average waiting time of tasks and improving the throughput of the Cloud system. The proposed algorithm can be easily integrated with Virtual Machine management strategies; and is fault tolerant and can efficiently deal with energy consumption problem.

Table of Contents

	Page No
Certificate	**i**
Acknowledgement	**ii**
Abstract	**iii**
Table of Contents	**iv**
List of Figures	**vii**
List of Tables	**viii**
Chapter 1	
Introduction and Problem Statement	
1.1 Introduction	1
1.2 Motivation	2
1.3 Statement of the Problem	3
1.4 Scope of Thesis	4
1.5 Thesis Organization	4
Chapter 2	
Background and Literature Review	
2.1 Virtualization Technology	6
2.1.1 Advantages of Virtualization	6
2.1.2 Virtual Machines	8
2.1.3 Aim of VM Scheduling	8
2.1.4 VM Scheduling Challenges	8
2.1.5 VM migration	9
2.1.6 Comparison the operation mode between the traditional datacenter and virtualized data center	9
2.2 Cloud Computing	
2.2.1 Introduction	11
2.2.2 Service Models	12
2.2.3 Cloud Computing Architecture	13

2.2.4 Types of Cloud 15
2.2.5 Characteristics 15
2.2.6 Cloud Computing Problems, Challenges & Issues 17
2.3 Service-Level Agreement 18
2.4 Energy consumption by data centers: Problems and challenges 18
2.5 Task Scheduling
2.5.1 Introduction 21
2.5.2 Types of Scheduler 21
2.5.3 Scheduling in Cloud Computing Environment 22
2.6 Cloudlet 23
2.7 Literature Review 24
2.8 Research Gaps 28

Chapter 3

Simulator: CloudSim Tool Kit

3.1 Introduction 29
3.2 Cloud Management Platforms 30
3.3 UEC
3.3.1 Introduction 31
3.3.2 Components of a Eucalyptus (UEC) based Cloud 31
3.4 CloudSim
3.4.1 Introduction 35
3.4.2 Features 35
3.4.3 CloudSim Architecture 35
3.4.4 CloudSim work style 37
3.4.5 Advantages 38
3.4.6 Problems 38
3.4.7 Components of CloudSim 39
3.5 Comparison of CloudSim Simulator with Cloud Management Platforms UEC 42

Chapter 4

Proposed Approach

4.1 Cloud Architecture 44

4.2 Integrated Scheduler 45

4.2.1 First Level: Modified Breadth First Search Algorithm 45

4.2.2 Second Level: Energy saving and Fault tolerance 47

4.2.2.1 Energy Saving 48

4.2.2.2 Fault tolerance 48

4.2.3 Flow Chart 49

Chapter 5

Simulation Work and Result Analysis

5.1 Implementation of the Proposed Work 51

5.1.1 Experimental Testbed: CloudSim Simulator 51

5.1.2 Assumptions and Performance Metrics 51

5.1.3 Experimental setup and algorithm parameter 52

5.2 Result and Discussion 53

Chapter 6

Conclusion and Future Work

6.1 Conclusion 58

6.2 Future Work 59

Chapter 7

Publication from Thesis

7.1 Accepted Paper 60

Appendix A

Abbreviations 61

References 62

List of Figures

	Page No
Figure 2.1: Virtualization Technology	7
Figure 2.2: Traditional Datacenter	9
Figure 2.3: Virtualized Datacenter	10
Figure 2.4: Various services on a Cloud	13
Figure 2.5: Cloud computing architecture	14
Figure 2.6: The worldwide datacenter energy consumption 2000-2010	19
Figure 3.1: Eucalyptus Cloud	32
Figure 3.2: CloudSim Architecture	36
Figure 3.3: CloudSim Work style	37
Figure 3.4: CloudSim class design Diagram	39
Figure 3.5: Simulation data flow among CloudSim components	43
Figure 3.6: Communication among components of UEC	43
Figure 4.1: partial diagram of Cloud System	44
Figure 4.2: Flowchart of Proposed Algorithm	50
Figure 5.1: Average execution time with 5 virtual machines and different number of cloudlets	56
Figure 5.2: Average waiting time with 5 virtual machines and different number of cloudlets	56
Figure 5.3: Average execution time with 10 virtual machines and different number of cloudlets	57
Figure 5.4: Average waiting time with 10 virtual machines and different number of cloudlets	57

List of Table

Page No

Table 3.1: Corresponding components of CloudSim and UEC 42

Table 5.1: VM specification with 5 virtual Machines 52

Table 5.2: VM specification with 10 virtual Machines 53

Table 5.3: Execution time (5 virtual machines) 54

Table 5.4: waiting time (5 virtual machines) 54

Table 5.5: Execution time (10 virtual machines) 55

Table 5.6: waiting time (10 virtual machines) 55

CHAPTER 1

Introduction and Problem Statement

1.1 Introduction

Cloud Computing is a new vision of computing as utility in which resources are stored on servers and customers can use them as a service on provisional basis by using laptops, computers or mobile devices over Internet. Cloud computing has emerged as a business revolution of the Information and Communication Technology industry by offering many advantages such as Cost Efficiency, Almost Unlimited Storage, Backup and Recovery to organization etc. But, Cloud computing technology comes with new problems like task scheduling, resource management, security, VM provisioning, load balancing etc. which require better methods than their previous solutions had, to allow it to function properly.

This thesis provides policy for energy efficient and fault tolerance task scheduling in Cloud. In Cloud platforms, the computing power is leased in the form of virtual machine (VM) to users with the help of virtualization technology. Thus, the scheduling algorithm aims to schedule applications on VM s based on the agreed Service Level Agreement (SLA) terms and deploy VM s on physical resources based on resource availability and power consumption.

The development and success of Cloud computing industry, the urgent needs of high performance computing and improvement in parallel and distributed computing have enabled the creation of different Clouds which resulted large-scale data centers establishment around the world containing thousands of computing server. However, Cloud data centers consume huge amounts of electrical energy resulting in high operating costs and carbon dioxide (CO_2) emissions to the environment that significantly contributes to the greenhouse effect. Various surveys have been showed that energy consumption in data centers will continue to grow

rapidly. Thus, advanced energy efficient task scheduling and resource management solutions should be developed and applied.

Moreover, a substantial part of electrical energy consumed by computing hardware is transformed into heat. High temperature led to a number of problems, such as reduced the life time of resources as well as reduced system availability and reliability. Cooling system is used to keep the system hardware within their safe operating temperature and prevent crashes and failures. Although, the cooling system cost is higher than the installation cost, hence there is a need of methods that improve datacenter resources life in cost efficient manner.

Virtualization offers big opportunities and benefits to address above mention issues. VM migration is used for balancing hosts' load and for managing power wastage due to underutilized host so that green house effect can be reduced. VM migration means; to transfer information related to the current VM and create and deploy same replica there. VM migration can also be used for improving fault tolerance capability of datacenter. Thus, VM management is closely related with resource management and task scheduling.

The objective of this thesis is to provide a task scheduling policy for the virtualized Cloud environment with the aim to minimizes the waiting time and makespan of given tasks set with energy saving and fault tolerance capability.

1.2 Motivation

Today the demand of Cloud computing is increased rapidly as it offer dynamic flexible scalable resource allocation. Cloud computing provide computing resources as a reliable services such as IaaS, PaaS and SaaS to users as pay as you go manner. The Cloud provider can gain profit only if it provides services under the terms and condition stipulated between provider and customer mentioned in SLA. Also by efficient management of datacenter; Cloud providers can reduce the cost of maintaining the servers and provide their services at lower cost and make more revenue. The main expenditure has been noticed due to the power consumption by datacenter. It has also been noticed, the life of hardware is reduce if they

work in high temperature continuously. The datacenter emits huge amount of harmful CO_2 gas and tremendous heat. So for the servers to be work reliable require their energy efficient and eco friendly maintenance. Many methods have been proposed to solve the power consumption problem and task scheduling problem in Cloud. These proposed method solved these issues separately however, they are inter-related. There is a need of integrated approach that takes into account these related problems together.

In Cloud computing, each application runs on a virtual machine, where the resources be distributed virtually. The virtualization layer acts as an execution, hosting and management environment for application services. Therefore, the task scheduling problem in Cloud is two step problems. Thus, there is a requirement, when proposing an approach of solving one issue at one level; must also consider the other related issues either same level or another level i.e. some kind of integrated task scheduling approach need to be proposed.

1.3 Problem Statement

The objective of the dissertation is to propose a two level task scheduling problem in Cloud Computing environment that reduces the execution time and waiting time of tasks and can be integrate with VM management methods to solve energy consumption problem and improve reliability and fault tolerance. In particular, the following research problems are investigated:

- The task scheduling problem in Cloud system is a two level problem. At first level: the task is assigned to appropriate VM and at second level: the VM should be allocated to the appropriate Host.
- At first level task is schedule to VM with consideration of user requirement defined in the SLA.
- At the second level VM is deployed to host with taking into account the resource availability, resource utilization, load balancing and recently energy consumption.
- Improving fault tolerance is one of the issue need to be taken into account to maintain datacenter so that it can work reliably.

- Based on the above issue, we design a task scheduling algorithm; find out how much waiting time and execution time get reduced by using proposed algorithm in comparison to existing one.
- We proposed approach to improve fault tolerance and reduce power consumption that can be efficiently integrated with the proposed algorithm.

1.4 Scope of Thesis

This thesis investigates energy-efficient and fault tolerant task scheduling algorithm based on two levels working of Cloud system that applied in virtualized datacenters containing heterogeneous physical resources. The rationale behind the algorithm is earliest finish time, lower average completion time and compatibility with other issues. By doing research and analysis of this problem, the aim of this thesis is to propose the approach that is compatible with other inter-related issues. The energy consumption and CO_2 emission and datacenter maintenance are hot issues so we will check the adaptability of proposed algorithm with these issues. The proposed approach can effectively help in reducing energy consumption and faults in datacenter. As energy consumption is the major component of operating costs so this approach will help Cloud provider to reduce their maintenance cost and improving revenue. Moreover, energy consumption causes CO_2 emissions to the environment, thus reducing energy consumption consequently reduces CO_2 emissions and we can say that the proposed approach is environment friendly. We will also discuss an approach to reduce the fault in datacenter resources by keeping monitor the thermal state of datacenter and alternatively switch on and switch off them.

1.5 Thesis Organization

This thesis report comprises of seven chapters including this chapter that introduces the topic and states the problem. The rest of the dissertation is organized as follows.

Chapter 2: Background and Literature Review

This chapter gives the background of Cloud computing, virtualization technology, VM migration, VM scheduling, Service level agreement, energy consumption problem, task scheduling , scheduling in Cloud environment and a brief literature review of related work including research gaps.

Chapter 3: CloudSim Simulator

In this chapter we will gives the comparative analysis of CloudSim Simulator with real Cloud management platform UEC (Ubuntu Enterprise Cloud) and the details, why we used simulation for evaluating the performance of proposed algorithm.

Chapter 4: Proposed Approach

In this chapter we will propose novel algorithm of task scheduling in Cloud computing environment and the solution of the hot issues like energy saving and fault tolerance then we will show that proposed algorithm can be associated with these solutions.

Chapter 5: Simulation Work and Result Analysis

This chapter gives the implementation details of the proposed algorithm and the details of experiments performed with result discussion.

Chapter 6: Conclusion and Future work

This chapter concludes the dissertation work with a summary of the main findings, discussion of future research directions, and final remarks.

Chapter 7: Publication from the Thesis

This chapter contains the details of publication based this thesis and the conference detail.

Appendix A: Abbreviations.

References: this section gives the list of citations used in the thesis.

CHAPTER 2

Background and Literature Review

2.1 Virtualization Technology [1]

Today's computer hardware was designed to run a single operating system and a single application, leaves the resource utilization of most machines vastly underutilized. To increase resource utilization virtualization technology is used. Virtualization enables you to run multiple virtual machines on a single physical machine, with each virtual machine sharing the resources of that one physical computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer. VM ware and Xen are the hypervisor used in virtualization.

Virtualization technology has become market maker in Cloud after recent advanced in hardware and software technologies. The two main concepts concern with virtualization technology are:

- Virtual Machine (VM)
- Virtual Machine Monitor (VMM)

2.1.1 Advantages of Virtualization

Isolation - Isolation ensures that applications and services that run within a VM cannot interfere with the host OS or other VMs.

Compatibility - Virtual machines are compatible with all standard computers.

Consolidation - virtualization gives the illusion of consolidated resources of multiple distributed hosts.

Migration - Virtual environment can be easily backed up and migrated with no interruption in service.

Abstraction - The salient feature of virtualization is the ability to hide the technical complexity from users, so it can improve independence of Cloud services.

Fault Tolerance - physical resource can be efficiently configured and utilized; provide quick recovery and fault tolerance.

Encapsulation - Virtual machines encapsulate a complete computing environment.

Hardware independence - Virtual machines run independently of underlying hardware

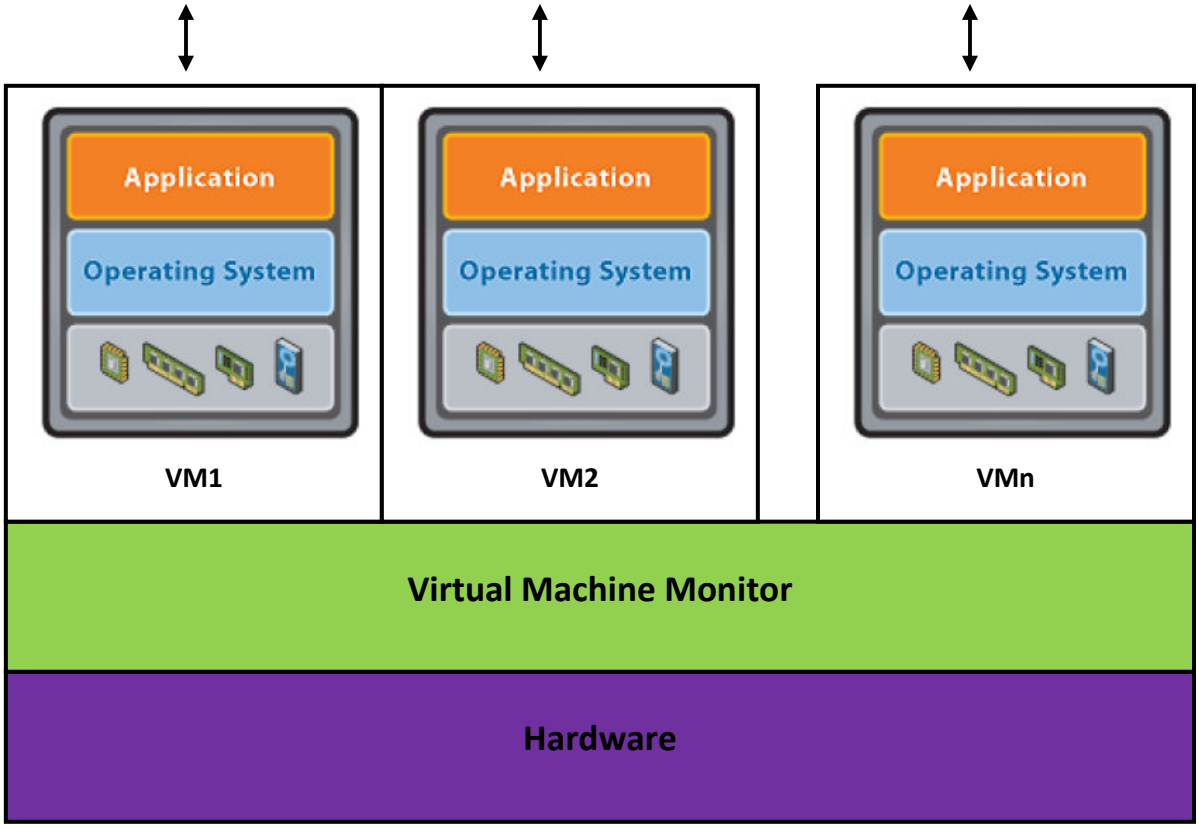

Figure 2.1: Virtualization Technology

2.1.2 Virtual Machines [1]

A virtual machine can be defined as software that can run its own operating systems and applications as like an operating system in physical computer. A virtual machine work almost in similar way as a physical computer and contains its own virtual (i.e., software-based) CPU, RAM hard disk and network interface card.

VMs are created within a virtualization layer, such as a hypervisor or a virtualization platform that runs on top of a client or server operating system. This operating system is known as the host OS. The virtualization layer can be used to create many individual, isolated VM environments. An operating system cannot identify the difference between a virtual machine and a physical machine, nor can applications and about the other computers on a network. Even the virtual machine thinks it is a "real" computer. A virtual machine is composed entirely of software and contains no hardware components. We see, virtual machines provide a number of distinct advantages over physical hardware.

2.1.3 Aim of VM Scheduling

Like any other processing unit, VMs need to be scheduled on the Cloud in order to:

- Maximize utilization
- Do the job faster
- Consume less energy

2.1.4 VM Scheduling Challenges

VM scheduling in Cloud is not free of challenges; the following challenges have to be faced:

- Two level scheduling
- High level of abstraction
- Unpredictable behavior

2.1.5 VM migration

Moving processes from one server to another server known as migration. Likewise VM migration allows moving a virtual machine (and its entire environment) from a host to another. These migrations are not free in terms of energy because every movement requires time and it is also important to take into account the total cost of such action. This technique has been used to balance the load of host or free and turn them off.

2.1.6 Comparison the operation mode between the traditional datacenter and virtualized data center [12]

Figure 2.2 shows, in tradition data center, hosts of the cluster as a single node to provide service to tasks. So it is difficult to satisfy the variegated demands of tasks when the resources in single nodes are fixed. In addition, it is worse to consider the delay caused by the resources competition between loads and hosts, so in tradition data center, the resources utilization is very low.

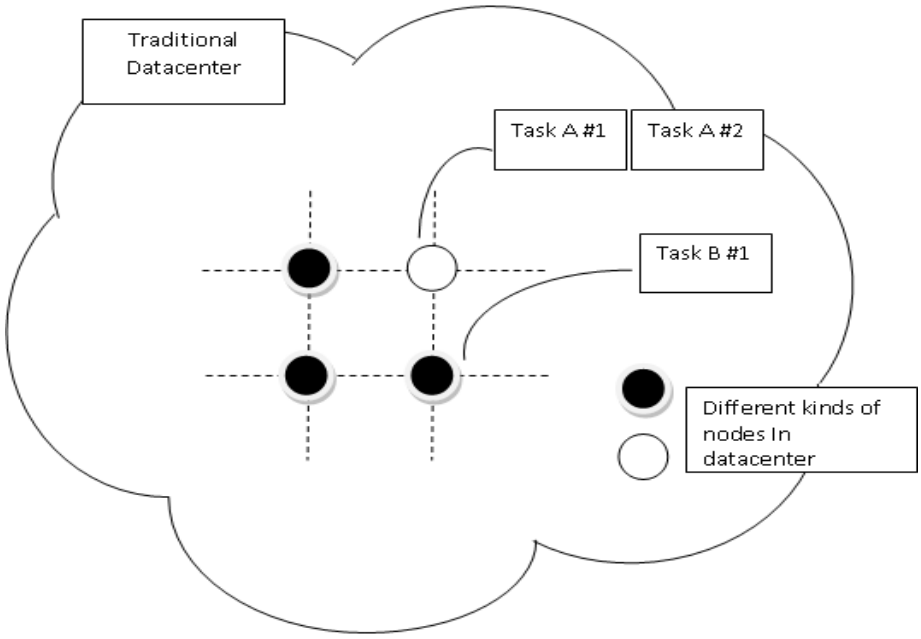

Figure 2.2: Traditional Datacenter

For example, we suppose that there are two kinds of hosts which can respectively execute two kinds of task, such as task A and task B in Figure 2.3. So one host node can only provide services for only one kind of task and competition exists between the same kinds of tasks in host, which lead low resource utilization of the tradition data center.

Figure 2.4 shows virtualization data center creates virtual machines according to the task needs, and all the virtual machines are managed by special unit named virtualization resource pool. In this way, many kinds of virtual machines can run in one host, which can improve the resource utilization of hosts in data center and because the virtual machine is no related to others in one host, it is benefit to improve performance of data center.

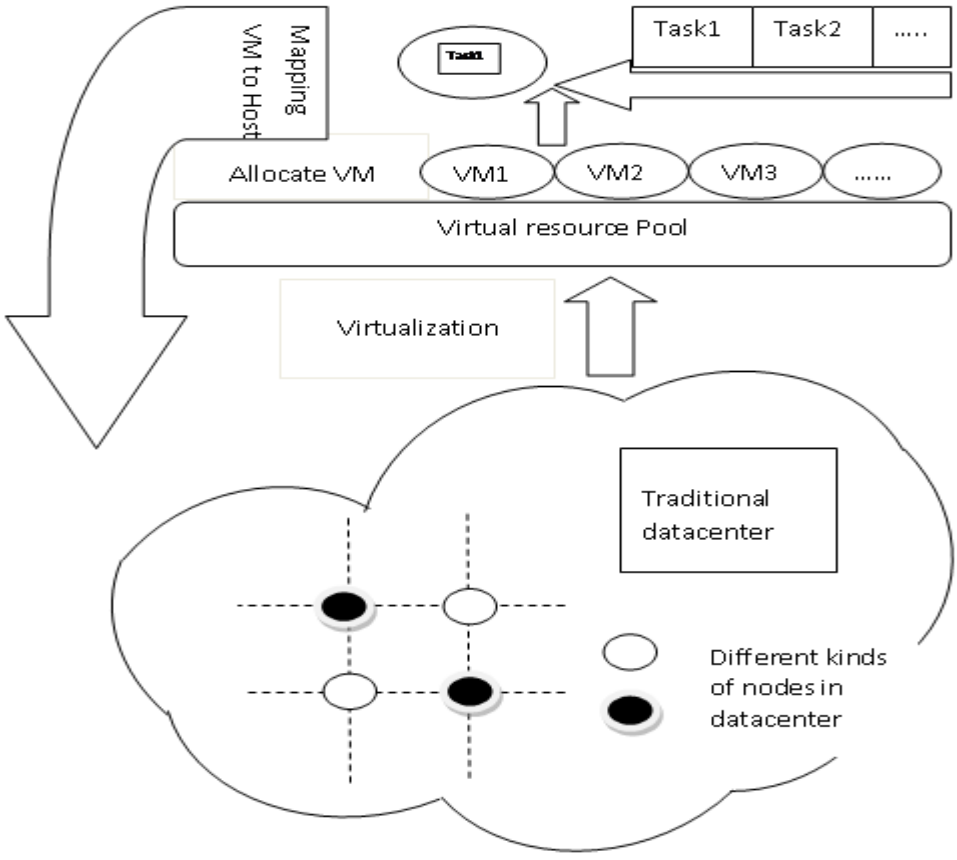

Figure 2.3: Virtualized Datacenter

2.2 Cloud Computing

2.2.1 Introduction

There are different interpretation and views on Cloud Computing. Because, different definition of Cloud computing are given by different author by concerning different issues of Cloud computing. There are various definitions that are centered on virtualization, commercialization, elasticity, scalability, pay-per-use utility etc. Ping Qi and Long-shu Li [3] define, Cloud computing is a user- oriented design which provides varied services to meet the needs of different users. They give insight that the Cloud is a technique emerged to satisfy different needs of different type of users.

The Cloud is the development of distributed computing, parallel computing and grid computing, or defined as the commercial implementation of these computer science concepts [4]. This definition puts Cloud computing into a market oriented perspective and stresses the economic nature of this phenomenon.

Nawfal et al. [5] observe that "Cloud computing, a new concept refer to a hosted computational environment that can provide elastic computation and storage services for user per demand. This definition focuses on the elasticity characteristic of the Cloud.

According to Buaya [6], "A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers. This definition presents both technical perspective (virtualization and dynamism) of Cloud and the market oriented perspective as well.

NIST (National Institute of Standard and Technology) define , the Cloud computing as a model or enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., network, servers, storage, applications, and services) that can be rapidly provisioned and release with minimal management effort or service provider

interaction[7]. This gives light that the Cloud works in on demand, transparent and abstract manner.

One of useful thing comes in a picture by above definitions. Cloud computing has emerged from parallel computing, grid computing, heterogeneous distributed computing, utility computing, and autonomic computing. Cloud provides the ability to deliver software, platform and infrastructure as services. This system aims to offer virtualized, distributed, on demand, salable and elastic resources as utilities to end users such as electricity. The Cloud resources is sharable, different users can use the same resources concurrently with the help of the virtualization technology. Cloud computing refer to the commercialization of computing platform software, where the users pay only for the computing power as much as they used and are usually charged on an hourly basis. The quality of services delivered over Cloud is measured by comparing its service quality with terms and conditions defined in Service Level Agreement (SLA). SLA is a negotiation between customer and Cloud Service Provider (CSP), in which customers and Service Provider settle deal about budget, deadline, reliability, security, penalty in case of failure etc.

2.2.2 Service Models

The Cloud providers offer different types of services to users which include programs, application-development platforms, and storage over the Internet, hardware resources for deploying user friendly platform etc. The Figure 2.4 shows number of services on a Cloud such as services to compute, storage services, OS as a service, Datacenter Applications and many more [8].

The Cloud services can broadly be classified into three categories:

1) IaaS (Infrastructure as a Service): In IaaS, the service providers render physical infrastructure such as CPU, Storage for provisioning computational platform and they pay on a per-use basis. Amazon's Elastic Compute Cloud (EC2) [9], GoGrid [10] is a popular examples of IaaS.

2) PaaS (Platform as a Service): In PaaS, customers allows to rent virtualized servers for using existing applications or developing and testing new ones. Google App Engine [37] is a perfect example of Web platform as a service, where Platform as a service (PaaS) allows Python and Java based Web applications being deployed dynamically and scaled. Amazon Web Services [11], Microsoft Azure [12], Manjrasoft Aneka [13] are some other examples of PaaS.

3) SaaS (Software as a Service): In SaaS, applications are hosted by a service provider delivered as a service to consumer but without controlling the host environment. Google Apps [14], SalesForce [15] are an examples of SaaS.

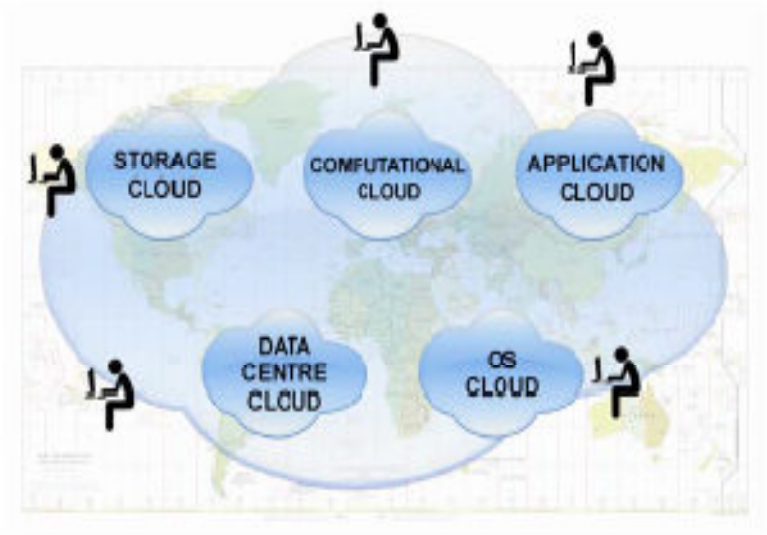

Figure 2.4: Various services on a Cloud [8]

2.2.3 Cloud Computing Architecture

Cloud architecture refers to the components and subcomponents required for constitute Cloud computing. These components distinctively consist of a front end platform called Cloud clients or clients, back end platforms a Cloud based delivery. The front end comprises fat or thick clients, thin clients, zero clients, tablets and mobile devices. The back

end platforms comprise various servers, storage that creates the "Cloud" of computing services. They connect to each other through a network, usually the Internet, Intranet or InterCloud. These client platforms interact with the Cloud data storage via an application or middleware, via a web browser, or through a virtual session. Combined, these components make up Cloud computing architecture. Figure 2.5 shows the layered design of Cloud computing architecture.

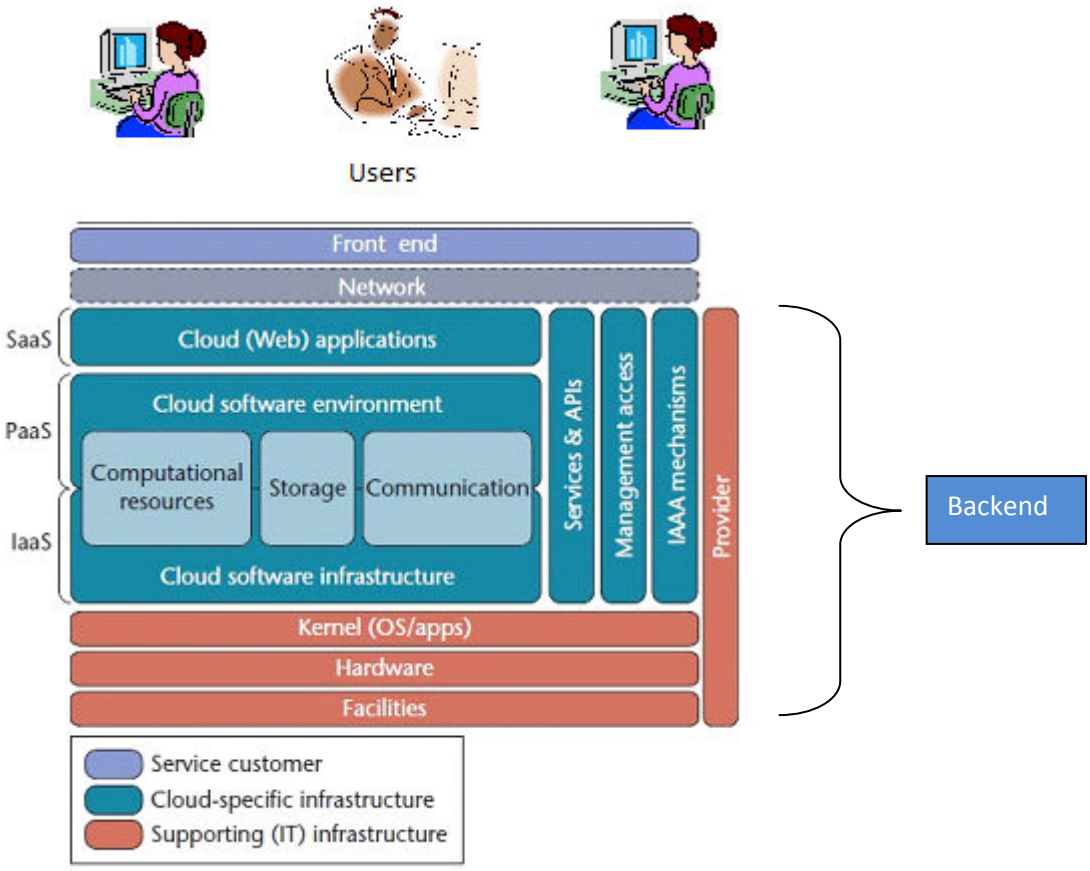

Figure 2.5: Cloud computing architecture

2.2.4 Types of Cloud

The Cloud can be differentiating on the type of ownership and access rights that they support; these are as follows:

- **Private Cloud:** Access to private Cloud resources is restricted to the users belonging to the organization that owns the Cloud.
- **Public Cloud:** In public Cloud; resources are available on the Internet to any interested user under pay-as-you-go model.
- **Hybrid Cloud:** A hybrid Cloud model is a combination of private Clouds with public Clouds.

2.2.5 Characteristics [7]

Cloud computing exhibits the following key characteristics

Empowerment – In Cloud computing end-users of computing resources has its own control on available resource, in traditional computing system control is in hand of a centralized IT service.

Application programming interface (API) - Cloud computing provide accessibility to software that enables machines to interact with Cloud software in the same way the user interface facilitates interaction between humans and computers.

Cost - Cloud computing use pay as you go model, so user does not need to pay whole money to buy the costly resource, maintain datacenters. In old computing scenario user has to maintain data centre with large number of powerful and is claimed to be reduced and in a public Cloud delivery model capital expenditure is converted to operational expenditure.

Device and location independence - Cloud computing enable users to access systems using a web browser regardless of their location or what device they are using like PC and mobile

phone. As infrastructure is typically provided by a third-party and accessed via the Internet, users can access resource from anywhere at any time.

Virtualization- This technology allows servers and storage devices to be shared and utilization be increased. Applications can be easily migrated from one physical server to another in order to manage the load balancing among the datacenter.

Reliability - It is improved if multiple redundant sites are used, which makes well-designed Cloud computing suitable for business continuity and disaster recovery.

Performance - In Cloud performance is monitored, and consistent and loosely coupled architectures are constructed using web services as the system interface.

Security - Cloud security is one the challenging issue, could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford.

Maintenance - Cloud computing applications is easier to manage, because they do not need to be installed on each user's computer and can be accessed from different places.

On-demand self-service - A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access - In Cloud computing Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms like, mobile phones, tablets, laptops, and workstations.

Resource pooling - The provider's computing resources are continuously pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. Cloud computing

provide location transparency in that the customer (user) generally has no control over the exact location of the provided resources however user may be able to specify location at a higher level of abstraction like, country, state, or datacenter. Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity - Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

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 services like, storage, processing, bandwidth, and active user accounts. Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

2.2.6 Cloud Computing Problems, Challenges & Issues

As Cloud Computing is enhancement of distributed computing, parallel computing and Grid Computing. The previously available methods cannot be used in Cloud because Cloud exhibits different new characteristics. Cloud computing technology comes with new problems like job scheduling, VM provisioning, load balancing, resource management, virtualization, security & Privacy etc. which require better methods than their previous solutions had, to allow it to function properly.

While solving the above mentioned problems, one have to deal with many issues like Quality of Service (QoS), Service Level Agreements(SLA), Resource Metering, Pricing, Billing, Scalability, Reliability, Energy Efficiency, Utility & Risk Management, Provisioning on Demand, Legal & Regulatory, Trust, Software Engineering Complexity, Programming Environment & Application Development.

2.3 Service-Level Agreement (SLA)

Service provisioning in Clouds is based on Service Level Agreements. Service Level Agreements (SLAs) representing a contract set up between customers and Cloud service providers. SLAs act as a warranty for users, so that they can more comfortably move their business to the Cloud. An SLA stating the details of the service to be provided in terms of metrics agreed upon by all parties.

Some terms of the agreement including in SLA are:

- Non-functional requirements of the service specified as Quality of Service (QoS)
- Obligations
- Penalties in case of agreement violations

SLA violation should be prevented to avoid costly penalties, loss of goodwill and growth of business.

2.4 Energy consumption by data centers: Problems and challenges

The population of Cloud computing and advancement in technology have resulted in the large-scale data centers establishment all over the world containing thousands of computing servers. However, Cloud data centers disburse huge amounts of electrical energy resulting pollution by emissions of carbon dioxide (CO_2) to environment and high operating costs. As shown in Figure 2.6, energy consumption by data centers worldwide has risen by 56% from 2005 to 2010, and in 2010 is accounted to be between 1.1% and 1.5% of the total electricity use [16]. As the use of Cloud is growing rapidly, energy consumption in data centers will continue to grow rapidly. A significant part of energy waste because of the inefficient usage of computing resources and by the cooling system to maintain the temperature of server. Thus, advanced energy efficient task scheduling and resource management solutions need to be developed and applied immediately.

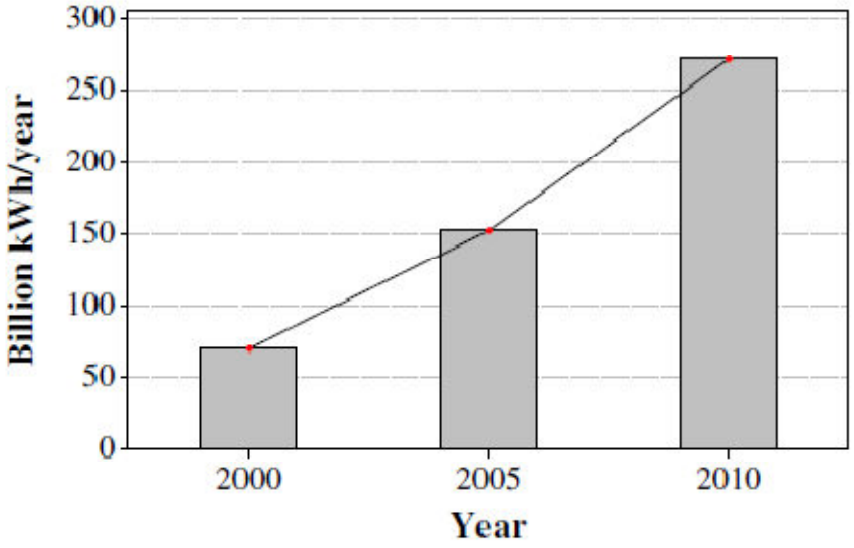

Figure 2.6: The worldwide Datacenter energy consumption 2000-2010 [16]

To address the problem of high energy use, it is necessary to eliminate inefficiencies and waste in the way electricity is delivered to computing resources, and in the way these resources are utilized to serve application workloads. This can be done by improving both the physical infrastructure of data centers, and the resource allocation and management algorithms. Recent advancement in the data center design has resulted in a significant increase of the infrastructure efficiency. As reported by the Open Compute project, Facebook's Oregon data center achieved a Power Usage Effectiveness (PUE) of 1.08 [17], which mean that approximately 91% of the data center's energy consumption is consumed by the computing resources.

A source of energy waste lies in the inefficient usage of computing resources. Data accumulated from more than 5000 servers over a 6 month period have shown that although servers are usually not idle even then utilization rate rarely approaches 100%. Mostly servers operate only at 10-50% of their full capacity which leading extra expenses on over-provisioning, and thus extra Total Cost of Acquisition (TCA). [18]

Moreover, managing and maintaining over provisioned resources results in the increased Total Cost of Ownership (TCO). In addition, the problem of low server utilization is exacerbated by narrow dynamic power ranges of servers: even completely idle servers still consume up to 70% of their peak power [19]. Therefore, keeping servers underutilized is highly inefficient from the energy consumption perspective. This report focuses on the solution of energy-efficient resource management in Cloud data centers, i.e., ensuring that computing resources are efficiently utilized to serve application workloads to minimize energy consumption, while maintaining the required Quality of Service.

A substantial part of electrical energy consumed by computing hardware is transformed into heat. High temperature led to a number of problems, such as reduced the life time of resources as well as reduced system availability and reliability. In order to keep the system components within their safe operating temperature and prevent failures and crashes, the emitted heat must be dissipated. Cooling system is used to keep the system hardware within their safe operating temperature and prevent crashes and failures. The cooling problem becomes extremely important for modern blade and 1U rack servers that pack computing resources with high density and complicate heat dissipation. Although, the cooling system cost is higher than the installation cost for example, in a 30,000 ft2 data center with 1,000 standard computing racks, each consuming 10 kW, the initial cost of purchasing and installing the infrastructure is $2-$5 million; with an average cost of $10/MWh, the annual costs of cooling alone are $4-$8 million [20].

Therefore, apart from the hardware improvements, it is essential to address the problem of optimizing the cooling system operation from the resource management side. One of the ways to minimize the cooling operating costs is to continuously monitor thermal state of physical nodes and reallocate VM s from a node when it becomes overheated. In this case, the cooling system of the offloaded node can be slowed down allowing natural heat dissipation. Therefore, it is necessary to investigate how and when to reallocate VM s to minimize the power drawn by the cooling system, while preserving safe temperature of the resources and minimizing the migration overhead and performance degradation. However,

VM migrations are not free in terms of energy because every movement requires time and it is also important to take into account the total cost of such action [21]. VM migration is also not free in terms of energy, so some kind of remedy need to be investigated.

2.5 Task Scheduling

2.5.1 Introduction

Task scheduling, one of the most famous combinatorial optimization problems, plays a key role to improve flexible and reliable systems. The main purpose is to schedule tasks to the adaptable resources in accordance with adaptable time, which involves finding out a proper sequence in which tasks can be executed under transaction logic constraints. [22]

Scheduling aims to allocate the tasks to the best optimal resources with taking into account of some restrictions, dynamic parameters and demands, such as resource processing capability, network restriction and as well as waiting time. A task scheduler is activated only at the scheduling points to make decision to which the task to be run next and on which resource. Thus, a task scheduler does not need to be run continuously, instead it is discrete.

"Task Scheduling is mapping of tasks to resources according to a certain principle of optimization."

2.5.2 Types of Scheduler

There are the following types of scheduler based on the scheduling environment:

Priority Scheduler - This scheduler assigns tasks to resources according to priorities. Every task in priority scheduling is given a priority based on some policy.

Dynamic Scheduler - A scheduler is called dynamic if it makes scheduling decisions at run time i.e. scheduling plan is depend on not only the bygone tasks and environment but also the current system state and current set of ready tasks.

Static Scheduler - Static Scheduler uses so called pre-scheduling technology to schedule known tasks in foregone environment.

Preemptive scheduler - Preemptive scheduler interrupt the running process for some time and preempt the resources from currently running process. The currently running process will resume later.

Non-preemptive scheduler - Non-preemptive scheduler does not allow a task to be preempted. The newly arrived tasks will queued in waiting queue.

Real time Scheduler - In real time scenario, task scheduling algorithms play an important role where the aim is to schedule the tasks effectively so as to reduce the turnaround time, waiting time and improve resource utilization while meet the deadline constraint. So, Real time task scheduling must determines the order in which the various task are taken for execution, so that the timing constraints of more tasks can be satisfied. Hence, Priority is applied to the scheduling of these periodic tasks with deadlines. A real-time scheduler must ensure that processes must meet deadlines, regardless of system load or makespan.

2.5.3 Scheduling in Cloud Computing Environment

The Cloud computing uses virtualization technology to map host resources to the virtual machine layer. Cloud computing makes required resources of task manifest in the form of a virtual machine. Therefore, the job scheduling in Cloud computing is implemented in application layer and virtual machine layer [3]. The aim of scheduling at application layer is to schedule tasks to optimal VM. At virtual machine layer, task scheduling involves the VM management and resource allocation and management.

In the Cloud environments on the one hand, applications require guaranteeing numerous SLA objectives to achieve their QoS goals and on the other hand, resource utilization is preponderant importance to the Cloud provider. These objectives contradict each other and make scheduling problem more challenging in Cloud. Dynamism is the main constraints need to be consider because, provisioning resources in a scalable on-demand manner and

uncertainties require dynamic scheduling in Cloud computing. Resources' capability, current load, interests and tasks' requests, which can affect the scheduling a lot, are dynamic too. The availability of resources and user behavior are also uncertain. There are mainly two factors of uncertainties. First one the users can request more resources later and the second one is resources are also uncertain. These issues make task scheduling a complex and challenging problem in Cloud. Recently energy crisis, CO_2 emission and reliability of datacenter have been added new challenges to researcher. These are the some hot issues attracting more attention.

Application schedulers have different policies to efficiently schedule the tasks and data of applications onto the Cloud computing environments that vary according to the objective function. The objective function can be based on one of the following criteria or combination of two or more criteria:

- Minimize total execution time.
- Power consumption & energy efficiency
- Minimize total cost to execute.
- Balance the load on resources used while meeting the deadline constraints of the application, and so forth

2.6 Cloudlet [23]

In CloudSim, tasks are packaged as Cloudlets, which contain jobs requirements (MI), size of job input and output data (in bytes) and other various parameters related with execution when tasks are deployed to corresponding resources by broker. These Cloudlets simulate Cloud-based application services, such as content transfer, social networking, and so on. Each application complexity is described by computational requirements. Therefore, each application has a pre defined processing requirements component which is inherited from Cloudlets and amount of data transfer which is related with input file size and output file size.

2.7 Literature Review

We have studied several papers of Cloud computing[3]-[7] , these gives us the idea about fundamentals of Cloud computing like, architecture of Cloud, deployment models, service models and essential characteristics of Cloud, job scheduling, resource allocation, security, virtualization.

The most popular scheduling approaches in Cloud are based on multiple resources pool algorithm, heuristic algorithm, load balance algorithm, random integer programming algorithm and the resources scheduling algorithm. Since, task scheduling in Cloud is an NP-complete optimization problem and heuristic algorithms are useful for the solution of combinatorial optimization problem. So, many meta-heuristic algorithms have been proposed to solve this problem. Some heuristic algorithms are:

- GA(Genetic Algorithm)
- Ant colony optimization (ACO) problem
- Particle Swarm Optimization (PSO)

Genetic algorithm has been widely and successfully applied in scheduling problems in [22] and [24]. Genetic algorithm is one of the popular interdisciplinary technologies of the artificial intelligence. Genetic algorithm is based on inheritance theory and natural selection and the idea behind it is to extract optimization strategies and transform them in mathematical optimization theory to find the global optimal solutions in a defined phase space for application. In [22], authors take into account both time utilization and resource utilization. The proposed algorithm scheduled divisible and independent tasks and able to adjust with different computation and memory requirement.

In [24], Author propose two genetic algorithms, Energy consumption time unify genetic algorithm (ETU-GA) and Energy consumption time double fitness genetic algorithm (ETDF-GA). These two algorithm use the method of unify and double fitness to define the fitness function and select individuals. The algorithm schedule tasks to physical resources when

virtual resources meet QoS requirement of tasks. Author assumes the independent task with same QoS requirement. However several tasks have different QoS requirement.

In [4] and [25], several different scheduling algorithms using the concept of ACO (Ant Colony Optimization) Algorithm were proposed. In [4], author used Ant colony optimization (ACO) algorithm with load balancing issue. The rationale behind the algorithm is to minimize the makespan and balancing the load of the system. The LBACO is an improvement over the ACO algorithm [25].

Particle Swarm Optimization (PSO) is an optimization technique based on self-adaptive global search introduced by Kennedy and Eberhart. The PSO algorithm is similar to other algorithms that are based on population like Genetic algorithms but, PSO has fewer primitive mathematical operators and faster convergence rate than GA [26]. In [26], author used PSO based heuristic to schedule application to Cloud resources in such manner that the distribution of tasks to all available resources in proportion to their usage costs to ensure load balancing. This algorithm focused to minimize the total cost of execution of application workflows by considering both computation cost as well as data transmission cost.

Cloud computing is a commercialization of distributed computing and parallel computing. There are many Cloud service providers in market; they compete. Hence the pricing of Cloud service also have paramount importance. The tasks have needed various resources to be complete and the price of all resources is not same. In [27] and [28] tasks are scheduled based on calculation of cost incurred by task. The work in [27] has been extended in [28] algorithm. In improved ABC the communication cost is also taken in consideration and based on the communication overhead the task are grouped so that the cost can be minimized.

The business perspective of Cloud gives a new vision to the task scheduling problem of Cloud that the services are provided based on negotiation between customer and Cloud provider. The success of deal depends on the accomplishment of the terms and condition discussed between the both parties, which are stated in SLA. Thus the key issues of Cloud computing is to ensure the QoS while schedule application.

Many author have been taken SLA terms as objective in their work. Most of the work [26, 27, 18, 29] are usually fitted toward one single SLA objective such as execution time and cost of execution etc. In [30, 31], authors considered multiple SLA parameters such as CPU, network bandwidth, storage for deploying application in Cloud. The algorithm included load balancing mechanism and automatically starts a new VM when no existing VM is appropriate for application deployment [30]. In [31], author used preemption to respond the fluctuating work load. If preemption is not possible because of tie in priority, then new VM is created from globally available resources.

Because of emergency of energy crisis, more and more attention has been paid for energy consumption aware and power control for Cloud datacenter. [24, 2, 21] focused on the energy awareness task scheduling. In [24], author took task scheduling as a bi-objective optimization problem with makespan and energy consumption as the scheduling criteria and proposed the way to find compromises between these two conflicting objectives. To reduce energy consumption author used dynamic voltage supply scheduling (DVS) that enables processors to dynamically adjust voltage supply level. The decrease in the supply voltage and frequency reduces the energy consumption consumed by resources. Author consider 3 power supply strategies (voltage relative frequency pairs), and 16 DVS levels.

Jingling Yuan, Xing Jiang and Luo Zhong_Hui Yu [2], focused onto reduce the energy consumption of the datacenter by idle resources and improve the datacenter resource utilization. The scheduling strategy is based on reinforcement learning and greedy thought, which uses N:1 mapping virtualization technology and Q learning to explore the global energy saving mapping. In [21], author used autonomic approach to provide energy aware scheduling and migrating task. Autonomic computing paradigm provides systems with self-managing capabilities helping to react to unstable situation. To do energy-aware scheduling, author combines two tools, a generic automatic computing framework Frameself and simulator CloudSim. The Frameself provide autonomic middle-ware which detects specific events and CloudSim is used to take information about energy consumption that helps to take a scheduling decision. This algorithm uses DVFS (Dynamic Voltage and Frequency Scaling)

policy that allows to dynamically changing the voltage and frequency of a host in relation to their CPU load.

In [3], the proposed algorithm optimizes the selection process of resources based on fuzzy quotient space theory to satisfy different user needs. Different task require different combination of abilities due to the different characteristics of user application, for example, some tasks with more network service require the virtual machine with higher bandwidth. At the same time, some tasks with more data processing require the virtual machine with high performance. Fuzzy equivalence partition and distance function are Combined with the theory of fuzzy quotient space for matching of tasks with resources in Cloud environment. The first step is to coarsen granulation spaces, to produce a candidate collection of VM s which meet the task execution conditions. . The second step is to refine granulation spaces partly and to select a virtual machine from candidate collection of VM s according to the general expectation preference. The last step is to choose the appropriate virtual machine to satisfy task expectations based on the distance between the vector of general expectation resources and actual allocation resources d(x,y). Lower value of d(x,y) indicate "better" candidates according to their QoS.

In CNC system (computer numeric control system) [32] task scheduling based on Cloud computing has been studied. This article provides a scheduling-algorithm based on the analysis of the particularities (real time requirement, different task have different resource requirement and complicate calculation) of the CNC system with two objectives: First the feasibility of the distributed system and the second one is reducing the average response time. For this distributed CNC system has been established based on the architecture of the Cloud computing. The algorithm is mainly based on dynamic feedback and adjustable weight value, and the node weight value is described by a Vector to express the different request of the different resource. Quantization analysis of the resource request is used to select the best node to accomplish the task. This article shows that the scheduling algorithm can perfectly realize the basic functional of CNC system. Likewise other industrial field computing as distributed system can associate with Cloud Computing and can utilize resources efficiently.

In [5], the proposed scheduler is location-aware, sharing-aware and QoS-aware. The execution time of tasks has been reduced by reducing the data transfer time. They used data sharing property of jobs to reduce data transfer time by schedule jobs with same data requirement to same VM. This leads to reduce the total execution time, execution cost, makespan and jobs failure. Author also considers transfer time of output file. CSA algorithm uses the benefit of both the algorithms MCT and MET. MCT is used to achieve good load balance.

2.8 Research Gaps

As we studied, very few papers proposed: integrated solution of the task scheduling and resource allocation problem of Cloud computing. [33], [34] and [35] presented integrated solution of task scheduling problem. In [34], the first scheduler is responsible to give specification of the required VM, according to resource demand of tasks. And the second scheduler find appropriate host for VM. In addition to handle dynamic demand of task, VM migration strategy has been used. This paper [35] has proposed a two-level scheduler; a meta scheduler based on QoS and a VM scheduler based on backfill strategy. In [33], scheduling heuristic proposed an integrated resource provisioning strategy in which, application are scheduled based on the agreed SLA terms and VM s are deployed based on resource availability. They considered VM deployment as the second level problem. However, deploying VM on an appropriate host has various issues. Moreover, in recent energy crisis, CO_2 emission and fault tolerance capability are of paramount importance. So they should be tackled properly to make datacenter work efficiently. Thus, VM management comprises of creation of VM, deployment of VM, VM migration, VM destruction and VM scaling. So, these related issues should be taken into consideration while scheduling tasks.

Thus we are proposing an algorithm Modified Breadth First Search that will be easily and efficiently integrated with VM management strategy to solve problem related to energy consumption, improving fault tolerance and reliability. The proposed algorithm will also focus on improving the throughput and reducing the execution time and waiting time.

CHAPTER 3

Simulator: CloudSim Tool Kit

3.1 Introduction

Performance Evaluation of Proposed Algorithm and Protocols can be done in one of the following manner:

- Using real infrastructures, such as Microsoft Azure or Amazon EC2
- By creating and managing Cloud by freely available Cloud Management Platforms.
- By using simulator such as CloudSim, TeachCloud etc.

Quantifying the performance of newly proposed approaches and provisioning policies such as scheduling and allocation in private Cloud or in real Cloud computing environment (Amazon EC2 [9], Microsoft Azure [36], Google App Engine [37]) for different application under various conditions is challenging because:

(i) Clouds present varying resources, demands, supply and patterns.
(ii) Users have dynamic, heterogeneous and contending QoS requirements.
(iii) Applications also have fluctuating workload, performance and dynamic application scaling requirements.

It is difficult and even not possible to perform standard experiments in repeatable, scalable and dependable environments by using real-world Cloud environments.

The use of real infrastructures for benchmarking the application performance (throughput, cost benefits) under variable conditions (availability, workload patterns) is often constrained by the rigidity of the infrastructure. So, this makes the reproduction of results that can be relied upon, an extremely difficult undertaking. Further, it is tedious and time consuming to re-configure benchmarking parameters across a massive-scale Cloud computing

infrastructure over multiple test runs. Such limitations are caused by the conditions prevailing in the Cloud-based environments that are not in the control of developers of application services. Thus, it is not possible to perform benchmarking experiments in repeatable, dependable, and scalable environments using real-world Cloud environments. [23]

A more feasible alternative is use of simulation tools. These tools put on the possibility of evaluating benchmarking study of the application in a controlled environment in which one can easily reproduce simulation results. Simulation-based approaches provide substantial benefits to researcher by allowing them to: [38]

(i) Test their approaches in repeatable, scalable and controllable environment.
(ii) Tune the system bottlenecks before deploying on real Clouds.
(iii) Experiment with different workload mix and resource performance scenarios on simulated infrastructures for developing and testing adaptive application provisioning techniques.

This chapter is organized as follows. Section 5.1 give the details of the Components of one of the Freely available Cloud Management Platforms Ubuntu Enterprise Cloud (UEC) Section 5.2 introduce the CloudSim, section 5.3 presents the comparison between them CloudSim and UEC

And then finally conclude why we chose CloudSim Simulator to evaluate our Algorithm.

3.2 Cloud Management Platforms (CMPs)

Cloud computing is a computing model, where resources such as computing power, storage, network and software are abstracted and provided as services on the internet in a remotely accessible fashion. Billing models for these services are generally similar to the ones adopted for public utilities. On-demand availability, ease of provisioning, dynamic and virtually infinite scalability are some of the key attributes of Cloud computing. [39]

Setup of an infrastructure using the Cloud computing model is generally denoted as the “Cloud". There are various Open Source freely available Cloud Management Platforms (CMPs) that, addressing different Cloud niches. Any CMP can be used to build either private or public Clouds, all of CMPs implement Cloud APIs. Since the entire CMPs enable infrastructure Cloud computing, there is always some overlapping in the features that they provide. Some of the open-source CMPs are:

- Opennebula
- Cloudstack
- Eucalyptus
- Openstack
- UEC

3.3 UEC (Ubuntu Enterprise Cloud)

3.3.1 Introduction

Ubuntu Enterprise Cloud, UEC for short, is a stack of applications from Canonical included with Ubuntu Server Edition. UEC includes Eucalyptus along with a number of other open source software. UEC makes it very easy to install and configure the Cloud. Eucalyptus is a software available under GPL that helps in creating and managing a private or even a publicly accessible Cloud. It provides an EC2 compatible Cloud computing platform and S3-compatible Cloud storage platform. Elastic Compute Cloud (EC2) and Simple Storage Service (S3) are popular Cloud services provided by Amazon Web Services (AWS). These services are available through web services interfaces. The client tools can use EC2 and S3 APIs to communicate with these services. [39]

3.3.2 Components of a Eucalyptus (UEC) based Cloud

The architecture of Eucalyptus, which is the main component of Ubuntu Enterprise Cloud, has been designed as a set of five simple elements that is shown in figure 3.1.

- Cloud Controller (CLC)
- Walrus Storage Controller (WS3)
- Storage Controller
- Cluster Controller (CC)
- Node Controller (NC)

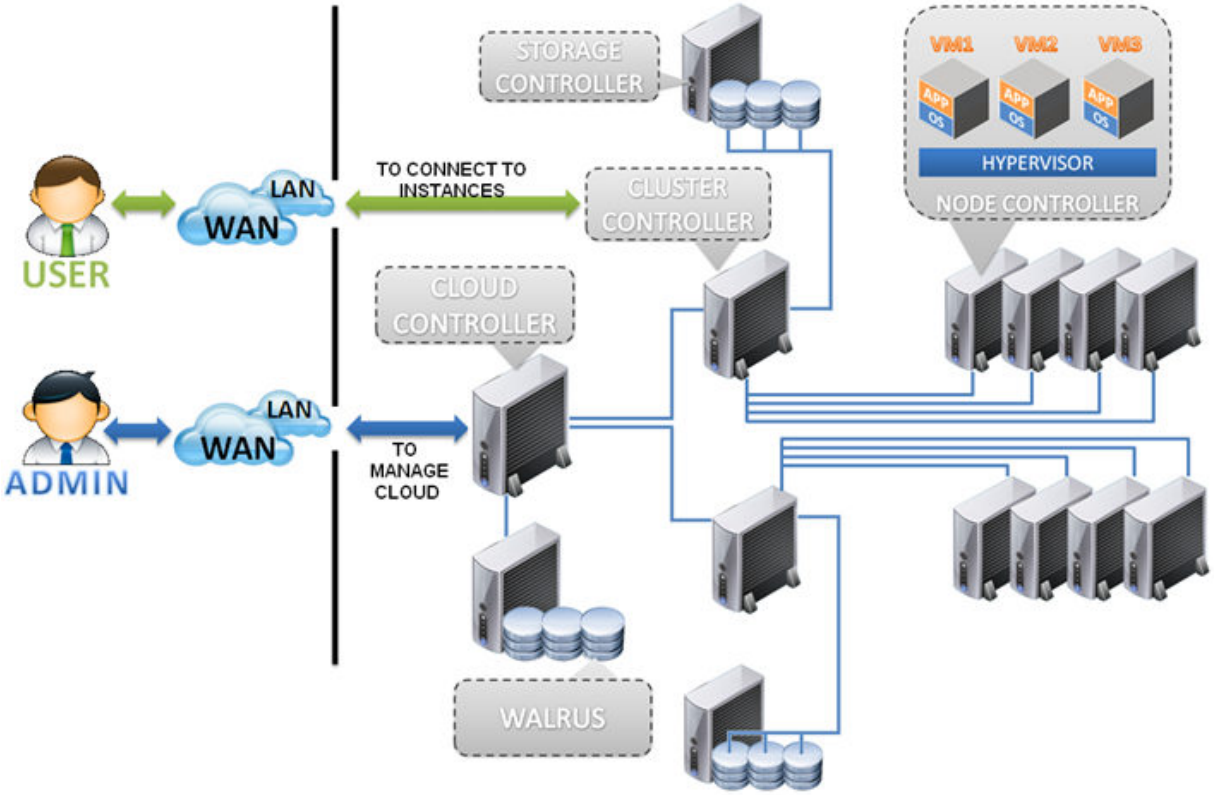

Figure 3.1: Eucalyptus Cloud

Node Controller - A UEC node is a Virtualization enabled server that is capable of running KVM /Xen as the hypervisor. The VMs are called instances, running on the hypervisor and controlled by UEC.

The Node Controllers' software runs on the physical machines where the Machine Image is instantiated. The role of NC software is to interact with the Operating System and hypervisor running on that node, as instructed by the Cluster Controller to discover available resources like type and number of cores, disk space, memory of the node in which it run and to learn about the VM instances' state running on the node and propagates this information up to the Cluster Controller.

Functions [39]:

- Collection of data related to the resource availability and utilization on the node and reporting the data to CC
- Instance life cycle management

Cluster Controller - The Cluster Controller operates between the Node Controller and the Cloud Controller. It receives requests to allocate VM Images from the Cloud Controller and then decides which Node Controller will be used to run VM instance.

Functions [39]:

- Receive requests from CLC to deploy instances
- Decide which NCs to use for deploying the instances on
- Control the virtual network available to the instances
- Collect information about the NCs registered with it and report it to the CLC

Cloud Controller (CLC) - The Cloud Controller is mainly visible element that offers EC2/S3-compatible interfaces to the client tool on the one side and interacts with the rest of the components of the Eucalyptus infrastructure on the other side. The Cloud Controller interacts with the Cluster Controllers and takes the top level decision to allocate new instances.

Functions [39]:

- Monitor the availability of resources on various components of the Cloud infrastructure, including hypervisor nodes that are used to actually provision the instances and the cluster controllers that manage the hypervisor nodes
- Resource arbitration; Deciding which clusters will be used for provisioning the instances
- Monitoring the running instances

Walrus Storage Controller (WS3) - Walrus provide persistent storage for all of the virtual machines that can be deployed on Cloud There are no data type constraints for Walrus, and it can contain images, volume and application data. There can exist only one Walrus per Cloud. Currently, the machine on which the Cloud Controller runs also hosts the Walrus Storage Controller (WS3).

Functions [39]:

- Storing the machine images
- Storing snapshots
- Storing and serving files using S3 API

Storage Controller (SC) - Storage Controller offers persistent block storage to be used by the instances. It communicates with the Node Controller and Cluster Controller and manages Eucalyptus block volumes and the snapshots for instances within its specific cluster. If an instance needs to write persistent data to storage outside of the cluster, it would need to write to Walrus that is available for any instance in any cluster.

Functions [39]:

- Creation of persistent EBS devices
- Providing the block storage over AoE or iSCSI protocol to the instances
- Allowing creation of snapshots of volumes

3.4 CloudSim

3.4.1 Introduction

Basically CloudSim is a simulator. CloudSim is an open Source product coded & designed in JAVA language, which is used in the field of Cloud computing for simulation. CloudSim is a new, generalized, and extensible simulation framework that allows seamless modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services [23]. CloudSim provide event-based simulation, where different system entities communicate via sending events. Compile the CloudSim Example Codes is very simple; we need to go through Command prompt we may also Add CloudSim with Eclipse, Netbeans etc. for making work easy.

3.4.2 Features [23]

- Availability of a virtualization engine.
- Flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services.
- Support for modeling and simulation of large scale Cloud computing environments, including data centers, on a single physical computing node.
- A self-contained platform for modeling Clouds, service brokers, provisioning, and allocation policies.
- Support for simulation of network connections among the simulated system

3.4.3 CloudSim Architecture [23]

CloudSim Architecture consists of two layers User Code and CloudSim layer and CloudSim core simulation engine SimJava and GridSim as shown in figure 3.2. The top-most layer in the CloudSim stack is the User Code that exposes basic entities for hosts (number of machines, their specification, and so on), applications (number of tasks and their

requirements), VMs, number of users and their application types, and broker scheduling policies.

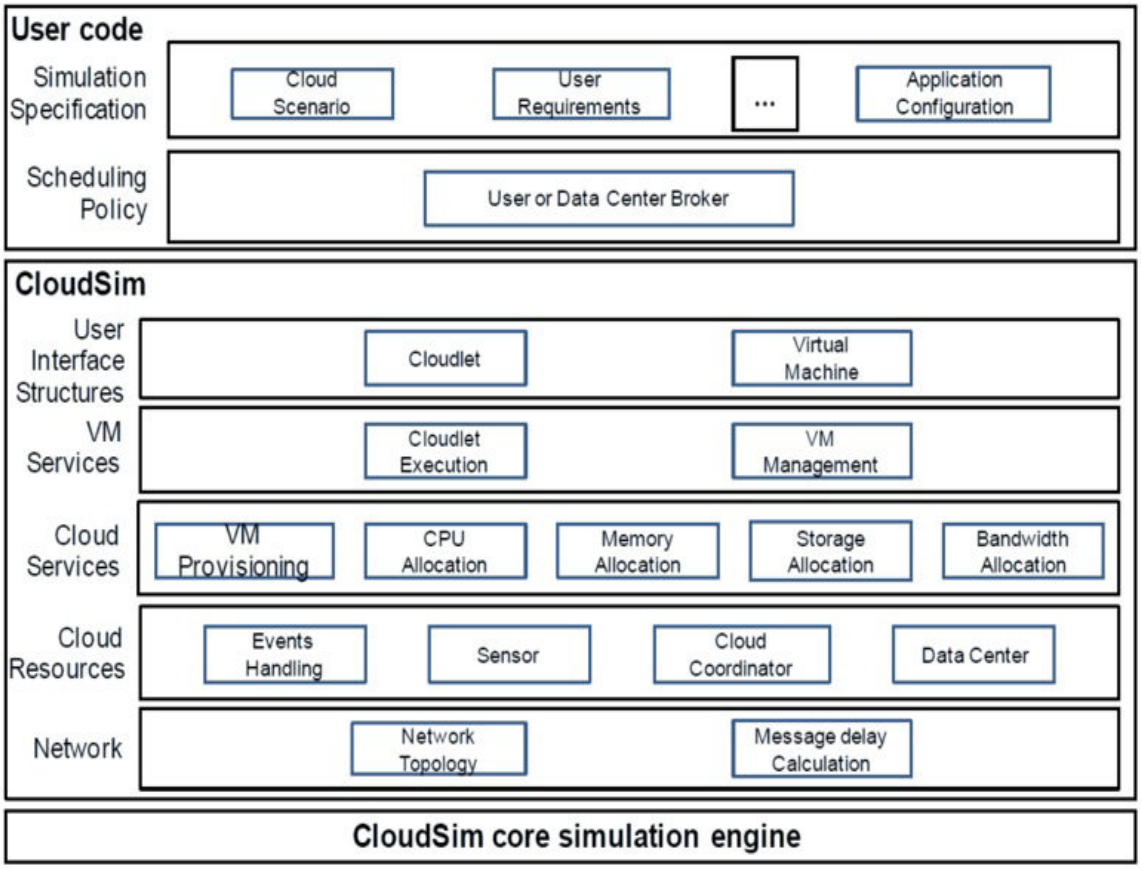

Figure 3.2: CloudSim Architecture [23]

By extending the basic entities given at top layer, a Cloud application developer can perform the following activities:

- Generate a mix of workload request distributions, application configurations.
- Model Cloud availability scenarios and perform robust tests based on the custom configurations.
- Implement custom application provisioning techniques for Clouds and their federation.

The CloudSim simulation layer provides support for modeling and simulation of virtualized Cloud-based data center. The fundamental issues, such as provisioning of hosts to VM s, managing application execution, and monitoring dynamic system state, are handled by this layer. A Cloud provider, who wants to study the efficiency of different policies in allocating its hosts to VM s (VM provisioning), would need to implement his strategies at this layer.

3.4.4 CloudSim work style

In [23], author proposed CloudSim work style that resembles the work style of real Cloud. The distributed resources are unified in the virtualized resource pool; the resources are allocated to application in the form of virtual machine. These VMs can be deployed to any of the host of datacenter; it is transparent to the users. The VM can be created, destroyed, migrated dynamically according to the application requirement and system load or work condition.

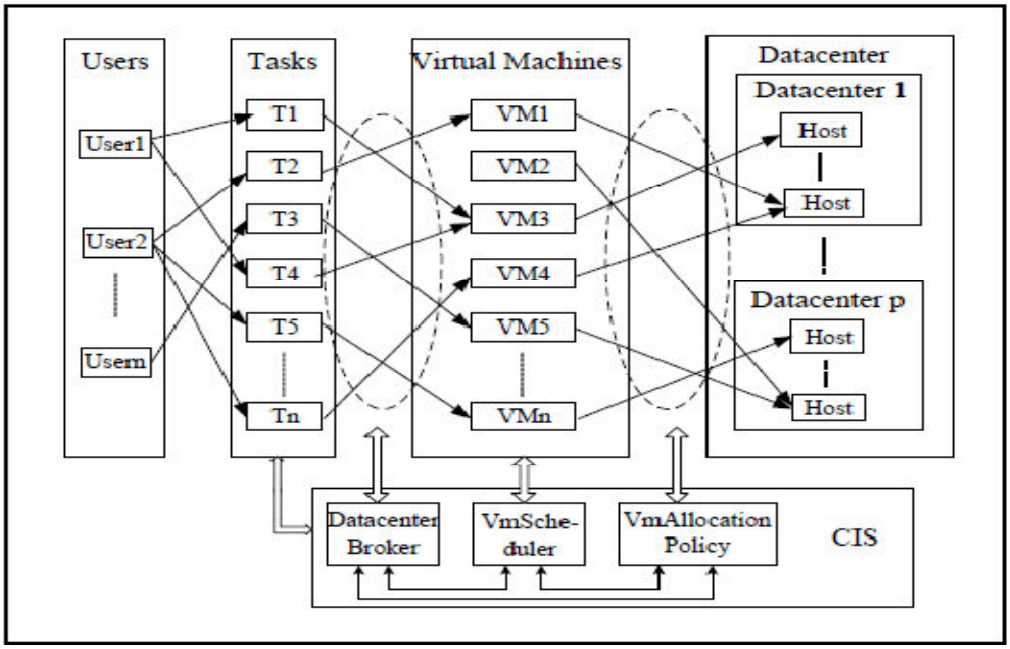

Figure 3.3: CloudSim Workstyle [4]

In the figure 3.3, author presented different component responsible for specific task. It is cleared that these components are inter-related to each other. Hence, while scheduling task to optimal VM, scheduling at these different related components should also be taken into consideration, so that the whole system works efficiently.

3.4.5 Advantages [23]

By using CloudSim, researchers and industry-based developers can test the performance of a newly developed application service in a controlled and easy to set-up environment. Based on the evaluation results reported by CloudSim, they can further fine-tune the service performance. The main advantages of using CloudSim for initial performance testing include:

- Time effectiveness: it requires very less effort and time to implement Cloud-based application provisioning test environment.
- Flexibility and applicability: developers can model and test the performance of their application services in heterogeneous Cloud environments (Amazon EC2, Microsoft Azure) with little programming and deployment effort.
- It is easy to use.
- CloudSim does not enforce any limitation on the service models or provisioning techniques that developers want to implement and perform tests with.

3.4.6 Problems

- Very limited, basic Cloud components modeling
- Hard to use in education and research (No GUI),
- No real workload Modeling
- It does not have SLA components.

3.4.7 Components of CloudSim

Figure 3.4 shows the class design diagram of CloudSim. In the context of CloudSim, a CloudSim component can be an abstract class or complete class or set of classes that represent one CloudSim model such as data center, host and an entity is an instance of any component.

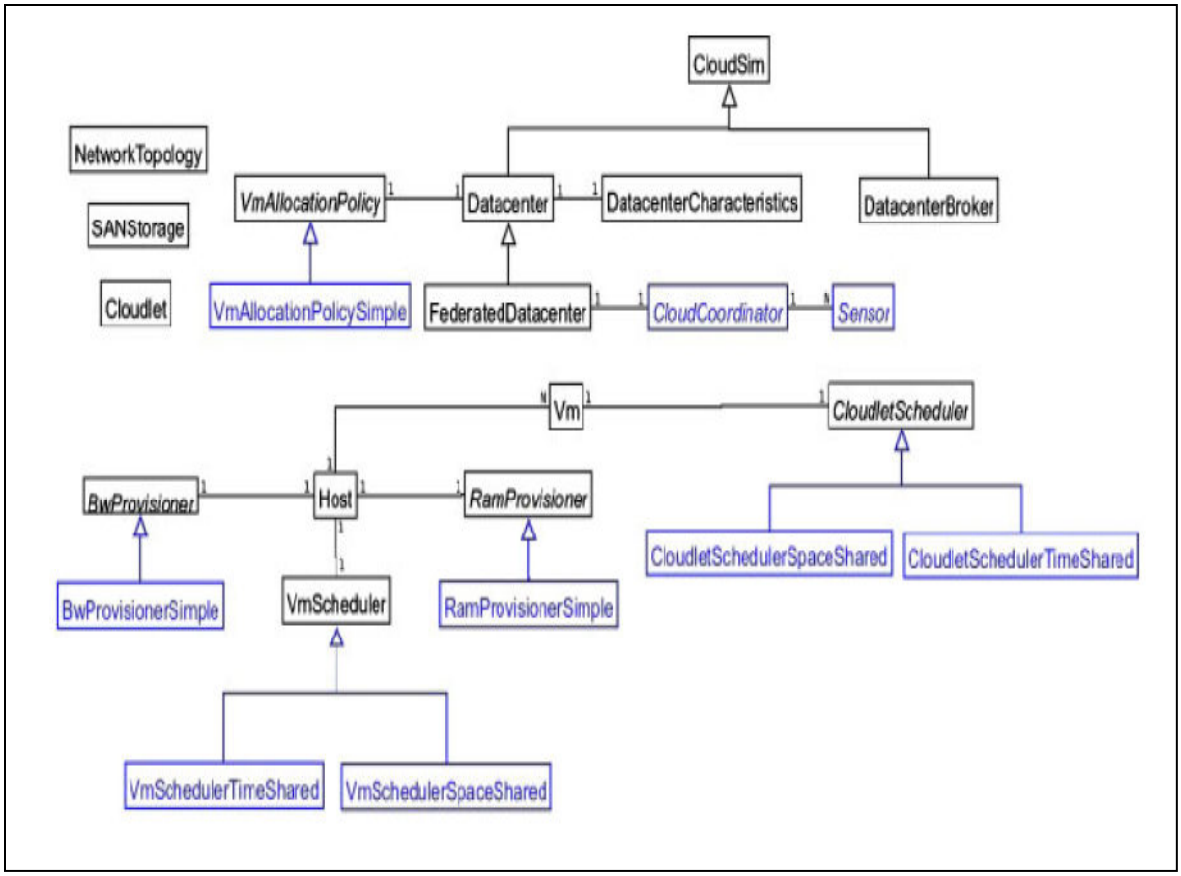

Figure 3.4: CloudSim class design Diagram [23]

CIS (Cloud Information Service) - The CIS (Cloud Information Service) provides database level match-making services; it maps user requests to suitable Cloud providers.

DataCenter - The infrastructure-level service related to the Clouds is simulated by extending the data center entity of CloudSim. A data center manages a number of hosts that in turn manages VM s during their life cycles (VM creation, VM destruction).

Host - It represents a physical computing server in a Cloud. It is assigned a pre-configured memory, processing capability expressed in millions of instructions per second (MIPS), storage, and a provisioning policy to allocate processing cores to VMs.

VM - The fundamental hardware and software configuration parameters related to VMs are defined in the VM class.

Cloudlet - The core Cloudlet object can used to implement the users' application services.

VM scheduler - Each host component also instantiates a VM scheduler component, which can either implement the space-shared or the time-shared policy for allocating cores to VM s. Cloud system/application developers and researchers, can further extend the VM scheduler component for experimenting with custom allocation policies. [23]

VM allocation policy - The VM policy means operations and control policies related to VM life cycle such as: provisioning of a host to a VM, VM creation, VM migration and VM destruction. This component brings into the open a number of custom methods for developers and researchers who can aid in the implementation of new policies.

Cloud Coordinator (Modeling federation of Clouds) - This entity is used to model and manage the federation of Cloud-based data centers.

Sensor - The Cloud Coordinator triggers the inter-Cloud load adjustment process based on the state of the data center. The specific sets of events that affect the adjustment are implemented via a specific sensor entity. Each sensor entity implements a particular parameter (such as under provisioning, over provisioning, and SLA violation) related to the data center. For enabling online monitoring of a data center host, a sensor that keeps track of the host status (utilization, heating) is attached with the Cloud Coordinator. At every monitoring step, the Cloud Coordinator queries the sensor. [23]

Utilization Model - CloudSim uses Utilization Model entity that brings to light variables and methods for defining the resource and VM -level requirements of a SaaS application at the instance of deployment. CloudSim users are required to override the method, getUtilization(), whose input type is discrete time parameter and return type is percentage of computational resource required by the Cloudlet. [23]

Power Model (Modeling the DataCenter power consumption) - This abstract class should be extended for simulating custom power consumption model of a PE. CloudSim users need to override the method getPower() of this class, whose input parameter is the current utilization metric for Cloud host and return parameter is the current power consumption value. This capability enables the creation of energy-conscious provisioning policies that require real-time knowledge of power consumption by Cloud system components. Furthermore, it enables the accounting of the total energy consumed by the system during the simulation period. [23]

Application provisioning - one or more application services can be provisioned within a single VM instance, referred to as application provisioning in the context of Cloud computing. [23]

Modeling Cloud Market - Market is an essential component of the Cloud computing system; costs and economic policies modeling are important aspects so that researchers can study and correctly assess the cost-to-benefit ratio of Cloud computing platforms.

In CloudSim, the Cloud market model is based on a multi-layered (two layers) design. The first layer contains the economics of features related to the IaaS model such as cost per unit of memory, cost per unit of storage, and cost per unit of used bandwidth. The second layer models the cost metrics related to SaaS model. [23]

Modeling Network Behavior - Inter-networking of Cloud entities (data centers, hosts, SaaS providers, and end-users) in CloudSim is based on a conceptual networking abstraction. Network latency that a message can experience on its path from one CloudSim entity to

another is simulated based on the information stored in the latency matrix. The topology description is stored in BRITE format. [23]

The CloudSim's event management engine employs the inter-entity network latency information for inducing delays in transmitting message to entities. This delay is expressed in simulation time units such as milliseconds. [23]

Modeling Dynamic Entities - Developer can create dynamically user, broker and data center entities. This functionality is useful for simulating dynamic behavior of entities where they can join, fail and leave the system randomly. After creation of new entities, they automatically register themselves in the Cloud Information Service (CIS) to enable dynamic resource discovery.

3.5 Comparison of CloudSim Simulator with Cloud Management Platforms UEC

Table 3.1 shows the corresponding components of CloudSim and UEC superficially; the functionality of UEC component is accomplished by different functions of different classes of CloudSim API. Figure 3.5 and figure 3.6 illustrates communication flow among components of CloudSim and UEC respectively. By seeing these figure we can conclude that the CloudSim works as real Cloud so that it can be preferred for evaluating new approaches.

S No.	UEC	CloudSim
1	Cloud Controller	CIS
2	Cluster Controller	Data Center
3	Node Controller	Host
4	Virtual Machines	VM
5	User	DataCenter Broker

Table 3.1: Corresponding components of CloudSim and UEC

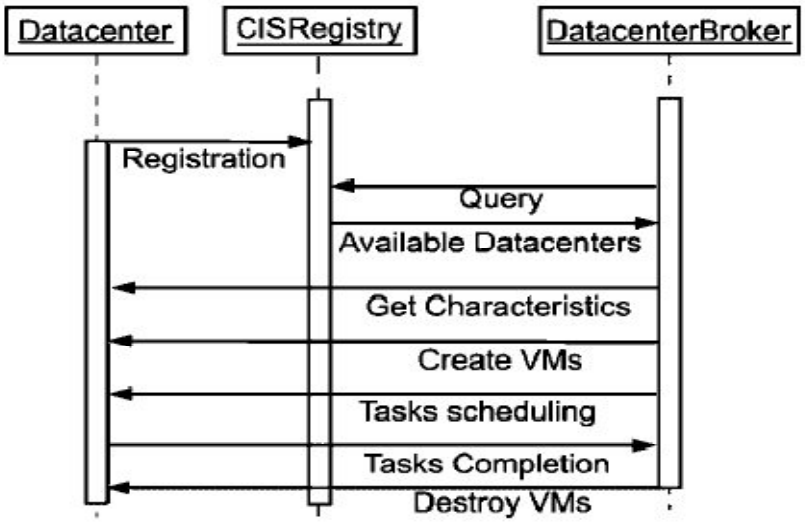

Figure 3.5: Simulation data flow among CloudSim components

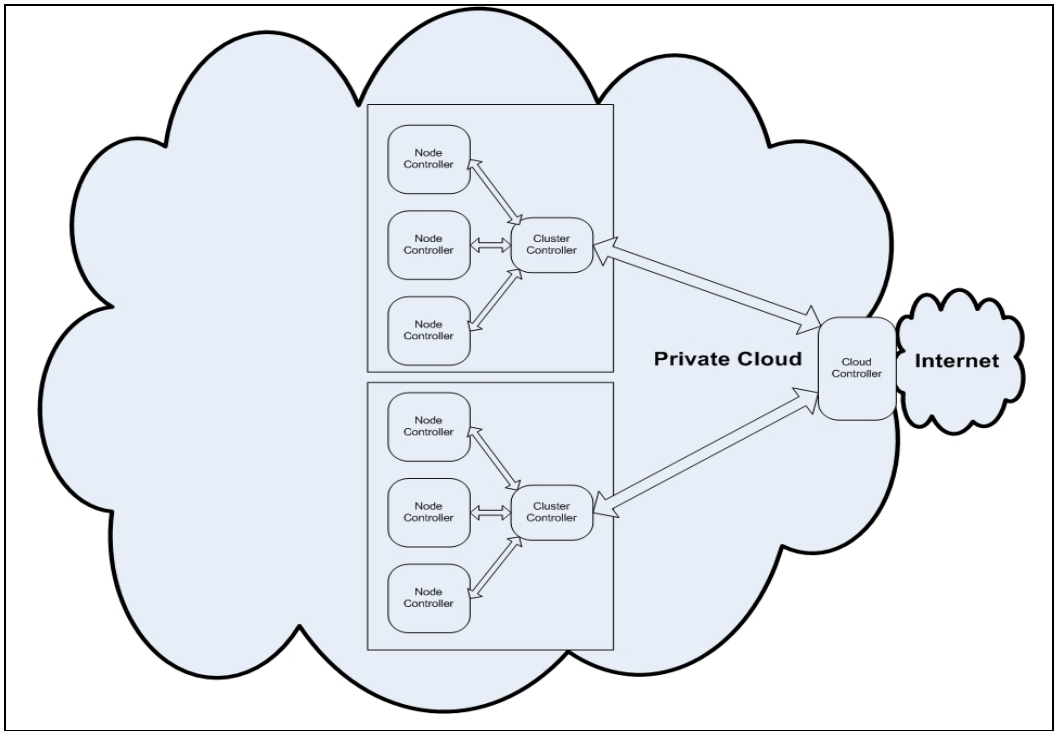

Figure 3.6: Communication among components of UEC

CHAPTER 4

Proposed Approach

4.1 Cloud Architecture

Figure 4.1 shows the generalized partial diagram of Cloud system. Cloud architecture consists of several layers as shown in fig, application layer, virtualization layer and infrastructure layer. Cloud offers the benefits of virtualization layer. Hence task is not scheduled directly on actual resources in Cloud. Instead, Task scheduling problem in Cloud is a two level problem. At first level: the task is assigned to appropriate VM and at second level: the VM should be allocated to the appropriate Host.

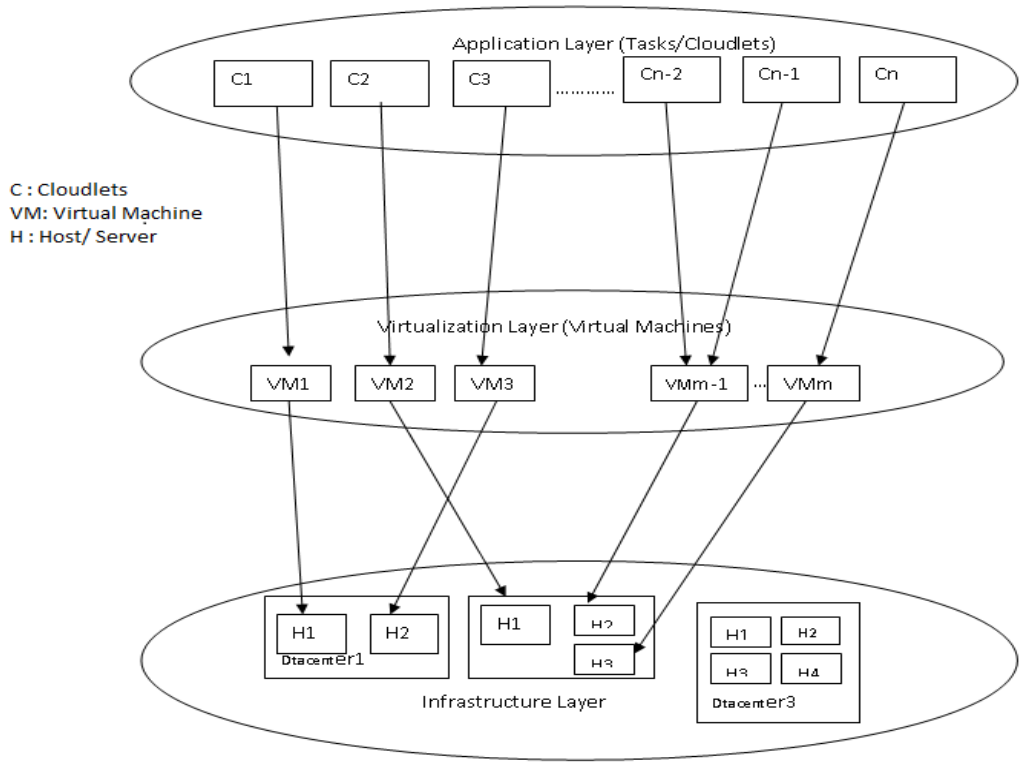

Figure 4.1: Partial diagram of Cloud System

In addition VM migration, scaling and destruction are also handled at virtualization layer. A typical application specific metrics include response time, waiting time, processing time, length of a task and number of tasks in request or waiting queue. A typical infrastructure specific metrics include system load, resource utilization. But, when energy consumption, tolerance capability, reliability, temperature maintenance and CO_2 emission are taken into account, the objective of task scheduling must be adjusted accordingly.

These different management strategies of VMs and task scheduling are inter-related to each others. So, when task is scheduled to VM; these relevant issues should be taken into account and this is one of the rationale factors behind our proposal.

4.2 Integrated Scheduler

This section gives the detail of proposed work that is based on the layers work style of cloud system. The scheduler works in two level and the details are given in the following sections.

4.2.1 First Level: Modified Breadth First Search Algorithm

Proposed scheduling approach is based on earliest finish time, high processing capabilities and compatibility with other inter-related issues. We used the binary tree based data structure called virtual machine tree (VMT) for efficient execution of task as used by Modified Depth First Search (MDFS) [40]. We prioritized task according to their MIPS. Then we used Modified BFS to identify the suitable Virtual Machine, for which the submitted task will be executed. In CloudSim task is known as cloudlet so, we are using the term cloudlet in algorithms. In algorithm 1 the tasks are prioritized based on their size.

Algorithm 2 create a binary tree of VMs based on prioritized order of Virtual Machines from left to right such that the MIPS of VM at level L is greater than or equal to node value at level L+1 where $L \geq 0$ and their right sibling.

Algorithm 1: Prioritize tasks

Input: cloudletList

Output: cloudletList is arranged in prioritized order

Step 1: Cloudlets are received by the scheduler

Step 2: Sort the cloudlets based on their size in decreasing order

Step 3: Store the cloudlets in a list, cloudletList

Algorithm 2: VM Tree Construction

Input: the list of VMs

Outputs: the root of VM Tree

Step 1: Obtain the all already created virtual machines

Step 2: Sort the virtual machines based on their MIPS

Step 3: Construct binary tree based on prioritized order of Virtual Machines from left to right

Step 4: Return the root of VM Tree

Algorithm 3 MBFS selects the appropriate Virtual Machine, at which the submitted task will be executed. In MBFS, the tasks are crowded towards more powerful VMs. Since the capacity of parent node either equal or high than child node, so in step 4 of algorithm 3, the task are submitted to the parent node in case of tie; hence it ensures that the finishing time will be fast of the next tasks to be submitted to VM Tree. So the waiting time and the execution time are reduced. In our approach the VM having more power, is utilized more and the other VM can be freed, this will improve datacenter management efficiency.

Algorithm 3: Modified BFS

Input: VM Tree and cloudletList

Output: mapping among tasks and virtual machiness

Step 1: insert root in queue

Step 2: for all the cloudlets available in cloudletList

Step 3: remove a node from queue and store it in current node

Step 4: check the estimated finishing time of cloudlet on current node as well as on parent of the current node

Step 5: submit the cloudlet to the node whose finishing time is less and in case of a tie submit task to parent node. And start process with next cloudlet.

Step 6: delete the front node of queue and insert its children nodes.

4.2.2 Second Level: Energy saving and Fault tolerance

The urgent needs of high performance computing and development and success of cloud computing has resulted large-scale datacenters establishment around the world. To compete in market, uncertainty in user demands and provide required QoS parameter, the performance and energy consumption of datacenter have paramount importance.

Reliable data shows, in 2005 the total energy consumption and cooling system of the data centers was projected at 1.2% the total U.S. energy consumption and doubling every five years [41]. In 2006, it was 1.5% of all U.S. energy consumption, projected at that point to double by 2011 [41]. Thus, energy consumption and efficient maintenance of datacenter needs to be paid attention and immediately solution.

4.2.2.1 Energy Saving

In 2010, data centers has consumed 0.5% of the world's total electricity usage and if the demand of energy continues, is projected to quadruple by 2020 [41]. And also, Earlier studies have showed that, the average resources utilization of data center is usually less than 20%. Even at a very low load, such as 10% CPU utilization, the power consumtion is over 50% of the peak power, because a large amount of energy is wasted by the idle resources [2]. To address the high energy use proble, elimination of waste and inefficiencies in the way electricity is used by computing resources.

This problem can be solved by allocating host resources at max then go to next host for allocation of resources. In MBFS approach, the tasks are crowded towards more powerful VM. So these crowded VM can be deployed to same host while, others at other hosts. So, later on the other VMs eventually become free and the host can be switched off to save energy.

4.2.2.2 Fault tolerance

A substantial part of electrical energy consumed by computing hardware is transformed into heat. High temperature led to a number of problems, such as reduced the life time of resources as well as reduced system availability and reliability. Cooling system is used to keep the system hardware within their safe operating temperature and prevent crashes and failures. Although, the cooling system cost is higher than the installation cost for example, in a 30,000 ft2 data center with 1,000 standard computing racks, each consuming 10 kW, the initial cost of purchasing and installing the infrastructure is $2-$5 million; with an average cost of $10/MWh, the annual costs of cooling alone are $4-$8 million [20].

One of the methods to reduce the cooling operating costs is to continuously monitor temperature of servers and reallocate VMs to other server when it becomes overheated. In this case, the offloaded server's cooling system can be slowed down and allowing natural heat dissipation. However, VM migrations are not free in terms of energy because every

movement requires time and it is also important to take into account the total cost of such action [21].

The proposed method is suitable with this kind of remedy. In our approach we are using VM Tree to identify the suitable mapping among tasks and VMs. The VM Tree is constructed by the all VMs of datacenter. The hosts can be alternatively switched on and off at second level without affecting the task and VM mapping at first level. We have to Create and deploy new VMs at the host that was switched off when a currently running host goes at high thermal state. When next time tasks need to be schedule construct VM Tree with these newly created VMs and discards the old VMs. And subsequently the host can be switched off.

4.2.3 Flow Chart

Figure 4.2 shows the flow chart of the proposed algorithm MBFS.

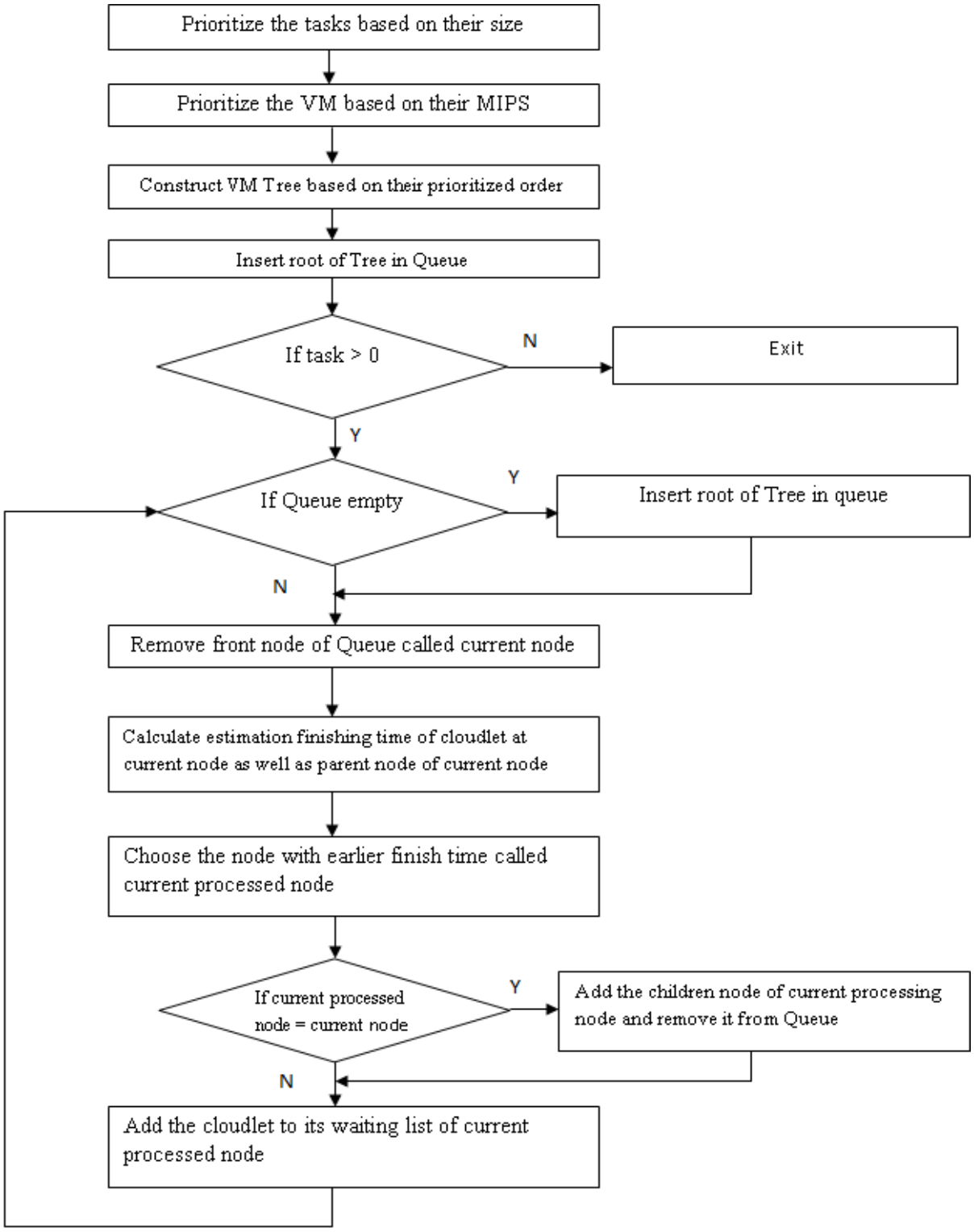

Figure 4.2: Flowchart of Proposed Algorithm

CHAPTER 5

Simulation Work and Result Analysis

5.1 Implementation of the Proposed Work

In this section, the CloudSim tool kit discussed in chapter 3 is applied to evaluate First Come Fist Serve (FCFS) Algorithm, Modified Depth First Search (MDFS) [40] Algorithm and Modified Breadth First Search (MBFS) Algorithm discussed in chapter 4.

5.1.1 Experimental Testbed: CloudSim Simulator

To evaluate and compare the proposed algorithms, it is extremely difficult to conduct repeatable large-scale experiments on a real infrastructure. Therefore, for ensuring the repeatability of experimentation, simulations have been chosen as a way to evaluate and compare the performance of the proposed algorithm with FCFS and MDFS.

CloudSim is an extensible modeling tool kit that provides a series of core classes and functions to establishment and simulation of cloud. The cloudSim toolkit support of modeling the cloud components such as data centers, host, virtual machine, cloudlets, scheduling and resource provisioning policies. Several researchers from organizations, such as HP Labs in U.S.A., are using CloudSim in their investigation on Cloud resource provisioning and energy efficient management of data center resources [23].

5.1.2 Assumptions and Performance Metrics

We are assuming the following assumption:

- Tasks are non preemptive i.e. no new tasks preempt the current running task
- Tasks are mutually independent, i.e., there is no precedence relation between tasks
- Tasks are computationally intensive.

In order to evaluate and compare the performance of the algorithms, several performance metrics were used. One of the metrics was the waiting time, which is the time the tasks spend in waiting queue of VM or the time tasks waited to actually acquire the VM. Another metric was the execution time, which is the time task employs the VM. In addition, the energy consumption and fault tolerance issues have been taken as the metrics that was discussed in the second level of the proposed algorithm. The scheduling problem aims to minimize the average waiting time and average execution time of tasks as well as second level VM management to improve fault tolerance and reducing power consumption.

5.1.3 Experimental setup and algorithm parameter

The number of experiments is conducted by varying number of tasks and virtual machines. First set of experiments is conducted with 5 Virtual Machines with MIPS shown in table 5.1 and the size of all Virtual Machine has been set as 512 MB. Another set of experiments is conducted with 10 Virtual Machines with MIPS shown in table 5.2 and RAM size of all Virtual Machine as 512 MB.

To ensure tasks are similar to real tasks in practice, Gaussian distribution function was used to create simulated tasks. Gaussian functions are used in statistics where they depict the normal distributions. The tasks processing requirements (MI) follows Normal Distribution and ensuring their processing requirements larger than or equal to 1000. The number of tasks (cloudlets) varies 5 to 50 with 5 step size.

S. No	VM ID	MIPS
1	VM1	250
2	VM2	500
3	VM3	1000
4	VM4	250
5	VM5	250

Table 5.1: VM specification with 5 virtual Machines

S. No	VM ID	MIPS
1	VM1	2000
2	VM2	5000
3	VM3	2000
4	VM4	700
5	VM5	250
6	VM6	500
7	VM7	1000
8	VM8	1024
9	VM9	250
10	VM10	250

Table 5.2: VM specification with 10 virtual Machines

5.2 Result and Discussion

In this section, we will show comparative simulation results of proposed algorithm. In this experiment, we considered execution time and waiting time as the main metrics. We have tabulated the results obtained by FCFS, MDFS and MBFS algorithms in table 5.3 and table 5.4 with 5 virtual machines for execution time and waiting time respectively. Table 5.5 and table 5.6 are showing the execution time and waiting time with 10 virtual machines respectively.

The results of simulated experiments are graphically depicted in the following Figure. Figure 5.1 and figure 5.2 compare the result of MDFS, MBFS and FCFS for execution time and waiting time of tasks based on the values in table 5.3 and table 5.4 respectively. And the Figure 5.3 and 5.4 compare the result of algorithms for execution time and waiting time of tasks based on the values in table 5.5 and table 5.6 respectively.

From Figureures, it can be seen that the execution time and waiting time taken to complete the tasks is less in MBFS as compared to MDFS and FCFS. The results are getting improve as the number of cloudlets are increased.

No of Cloudlets	FCFS	MDFS	MBFS
5	100.25	66.5	34.25
10	251.25	200.3333	110
15	139.5	162.4167	104.6667
20	177.5	296.75	125.3333
25	398.25	500.1667	172.5833
30	727.5	663.5833	341.75
35	1454	1327.25	786.3333
40	1637.25	1443.167	1323.583
45	2374.5	2625	1663.083
50	1595	2177.167	1066.083

Table 5.3: Execution time (5 virtual machines)

No of cloudlets	FCFS	MDFS	MBFS
5	0	41.5	0.166667
10	75.5	107.9167	58.75
15	86.25	102.0833	69.16667
20	73.5	210.6667	71.75
25	235.25	404.9167	109.25
30	530.25	534.0833	261.5
35	1082.25	1081.833	597.8333
40	1289	1196.5	1152.5
45	1851.75	2236.667	1323.5
50	1302	1879.083	873.1667

Table 5.4: waiting time (5 virtual machines)

No of cloudlets	FCFS	MDFS	MBFS
5	37.69405	15.1	5.166733
10	75.85051	29.55	17.29749
15	158.4871	74.29282	57.0166
20	226.4624	113.6028	53.38779
25	283.8303	187.1903	100.3295
30	240.0265	185.2306	135.0629
35	190.7926	240.2824	107.1661
40	445.6102	464.2436	265.4187
45	408.1355	325.6553	161.6276
50	833.1107	554.8806	322.5546

Table 5.5: Execution time (10 virtual machines)

No of cloudlets	FCFS	MDFS	MBFS
5	0	11.71667	1.750033
10	**0**	21.80833	0.214714
15	35.21905	42.41365	12.7419
20	95.91881	49.289	21.5962
25	121.0724	86.39983	48.0466
30	125.9981	114.5237	82.63767
35	101.3374	169.2535	53.21937
40	264.0761	269.5274	156.6352
45	247.9316	222.5916	118.7939
50	551.8069	388.4452	264.4992

Table 5.6 waiting time (10 virtual machines)

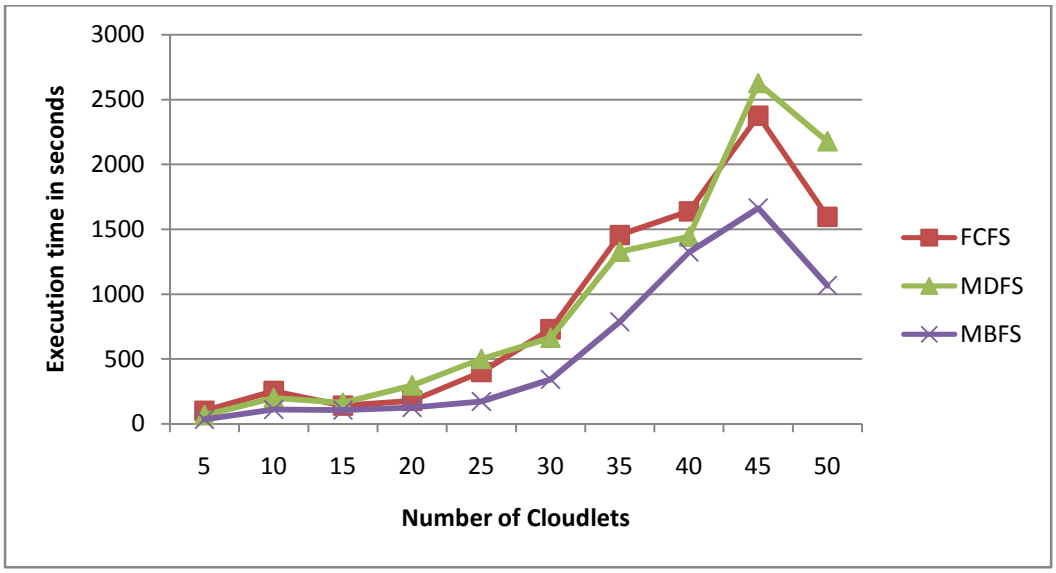

Figure 5.1: Average execution time with 5 virtual machines and different number of cloudlets

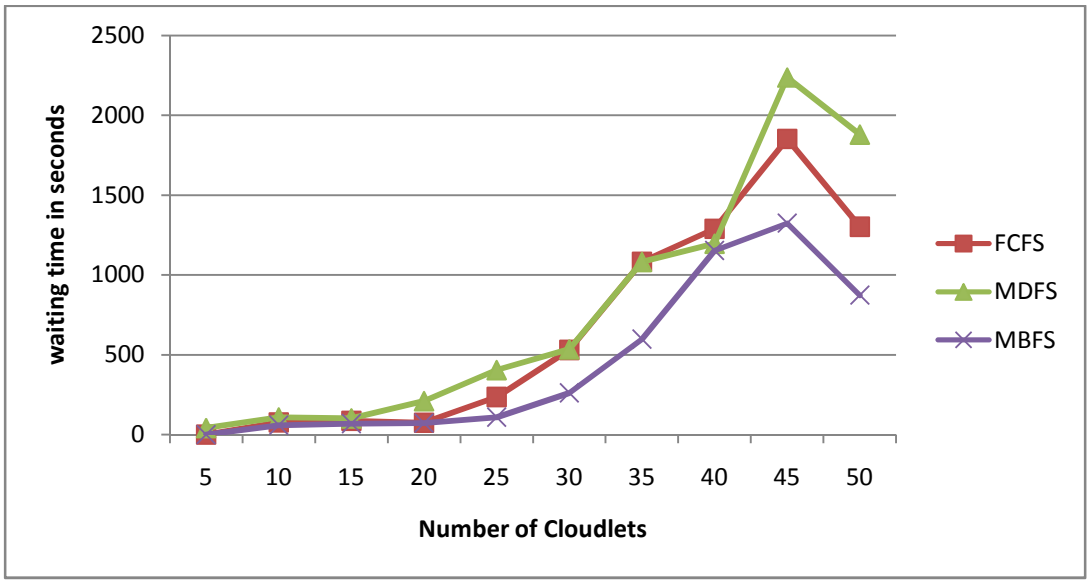

Figure 5.2: Average waiting time with 5 virtual machines and different number of cloudlets

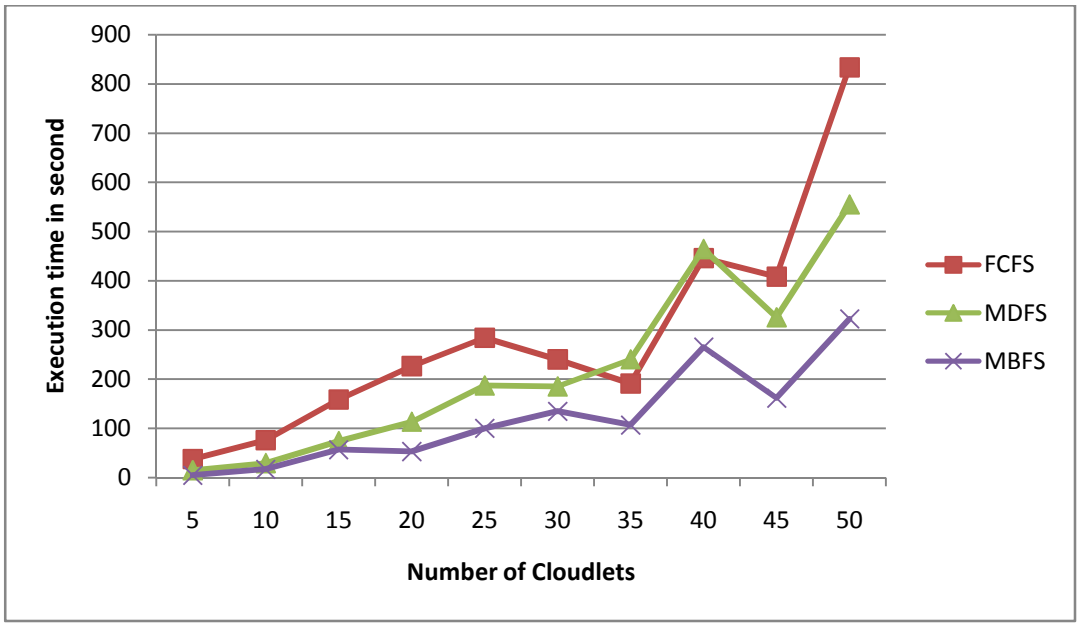

Figure 5.3: Average execution time with 10 virtual machines and different number of cloudlets

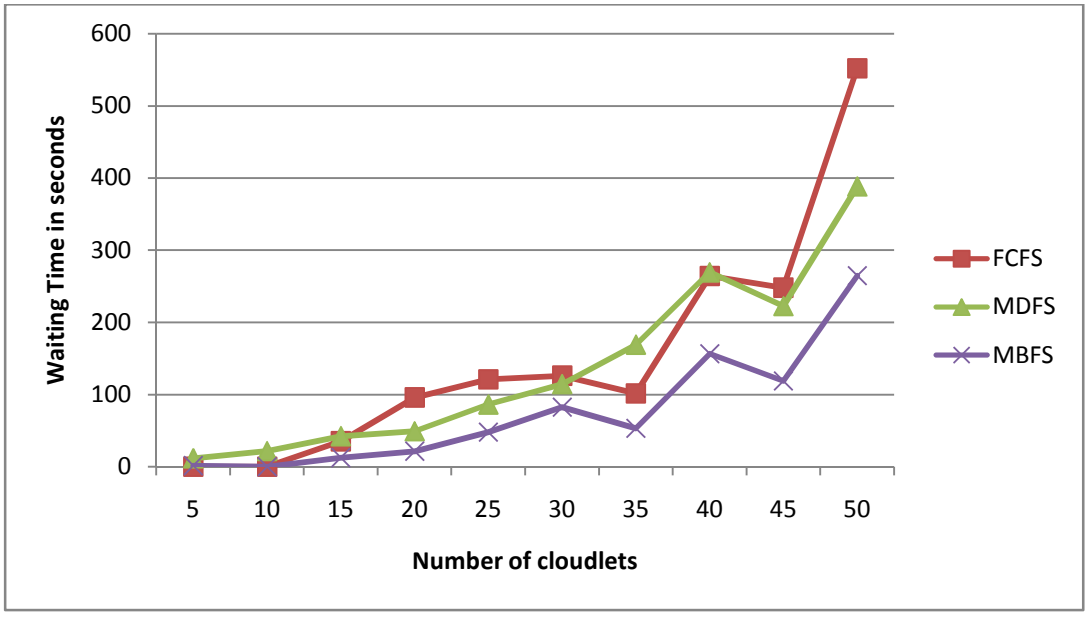

Figure 5.4: Average waiting time with 10 virtual machines and different number of cloudlets

CHAPTER 6

Conclusion and Future work

6.1 Conclusion

In this thesis, we discussed cloudlets scheduling in cloud environment by taking inter-related factors relative to two level of cloud system. First one to allocate appropriate VM to task by taking into account earliest finish time, which leads lower of average execution time , waiting time and improvement of the throughput. Second one to allocate the VM to hosts by taking into account the energy consumption and thermal state of the hosts. The tasks are allocated to the appropriate VM by constructing the VM Tree from the available virtual machines and then Modified Breadth First Search Algorithm is used to find the suitable VM for the task to be allocated. The tasks are assigned the VM that is able to finish them early at each step of algorithm.

The energy consumption, CO_2 emission and datacenter maintenance are hot issues. These factors need to be taken into consideration because avoidance of these issues leads high expense and lower profit as well as intolerable to avoid.

By doing research and analysis of cloudlet scheduling problem in Cloud computing, it is a two level problem that is associated with each other. Hence, the aim of this thesis is to propose the approach that takes into account other associated issues. So, we checked the compatibility of proposed algorithm with these issues. The rationale behind the algorithm is earliest finish time, lower average completion time and compatibility with other issues. CloudSim 2.2 is employed to carry out and simulate the algorithm. The results are then compared with FCFS and MDFS. The conclusion is that the scheduling algorithm MBFS is better than MDFS.

6.2 Future Work

In the performed experiments, the tasks are prioritized based on their size, which is not realistic for cloud systems. As a future work, the availability vector should be extended to incorporate information about task requirements like deadline, budget So that, the priority of task can be decided more accurately. Virtual machines are prioritized based on the Million Instructions per Second (MIPS). In future the other characteristics such as memory and bandwidth can also be taken into consideration to prioritize the VMs.

CHAPTER 7

Publication from Thesis

7.1 Accepted Paper

Conference name: IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014).

Paper Title: "An Energy Preserving and Fault Tolerant Task Scheduler in Cloud Computing".

Authors: Prof R.K. Yadav and Veena Kushwaha.

Status: Paper is accepted at the conference and will be published in IEEE Xplore.

Location: Dr. Virendra Swarup Group of Institutions, Unnao, India.

Conference Date: 1-2 August, 2014.

Publisher/Proceedings: Paper will be published by IEEE Xplore.

APPENDIX A

Abbreviations

ACO - Ant colony optimization

AWS - Amazon Web Services

CC - Cluster Controller

CIS - Cloud Information Service

CLC - Cloud Controller

CMP - Cloud Management Platform

CSP - Cloud Service Provider

EC2 - Elastic Compute Cloud

FCFS - First Come Fist Serve

GA - Genetic Algorithm

IaaS - Infrastructure as a Service

MBFS - Modified Breadth First Search

MDFS - Modified Depth First Search

MIPS - Millions instructions per second

NC - Node Controller

OS - Operating System

PaaS - Platform as a Service

PSO - Particle Swarm Optimization

QoS - Quality of Service

S3 - Simple Storage Service

SaaS - Software as a Service

SC - Storage Controller

SLA - Service Level Agreement

UEC - Ubuntu Enterprise Cloud

VM - Virtual Machine

VMM - Virtual Machine Monitor

WS3 - Walrus Storage Controller

REFERENCES

[1] http://www.vmware.com/virtualization (December 5, 2013)

[2] Jingling Yuan, Xing Jiang and Luo Zhong_Hui Yu, "Energy Aware Resource Scheduling Algorithm for Data Center using Reinforcement Learning", IEEE Fifth International Conference on Intelligent Computation Technology and Automation. , 2012

[3] Ping Qi and Long-shu Li, "Job Scheduling Algorithm Based on Fuzzy Quotient Space Theory in Cloud Environment", IEEE International Conference on Granular Computing, 2012.

[4] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang, "Cloud Task scheduling based on Load Balancing Ant Colony Optimization", IEEE 2011.

[5] Nawfal A. Mehdi, Bryn Holmes, Ali Mamat, and Shamala K. Subramaniam, "Sharing-Aware InterCloud Scheduler for Data Intensive Jobs", IEEE 2012.

[6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.

[7] Peter Mell and Tim Grance, Definition of Cloud Computing. Version 15,. National Institute of Standards and Technology, Information technology vol.24, Special Publication 800-145, 10-7-09.

[8] Ravin Ahuja, Asok De and Goldie Gabrani, "SLA Based Scheduler for Cloud for Storage & Computational Services", IEEE International Conference on Computational Science and Its Applications 2011

[9] Amazon Elastic Compute Cloud (EC2), http://aws.amazon.com/ec2 (January 10, 2014)

[10] http://www.gogrid.com/ (January 10, 2014)

[11] http://www.aws.amazon.com/ (January 10, 2014)

[12] Windows Azure Platform, http://azure.microsoft.com/en-us/ (January 10, 2014)

[13] http://exchange.gogrid.com/partnergsi/manjrasoft-aneka (January 20, 2014)

[14] http://www.google.com/apps/ (January 20, 2014)

[15] http://www.salesforce.com/ (January 20, 2014)

[16] "Growth in data center electricity use 2005 to 2010," Analytics Press, Tech. Rep., 2011

[17] Open Compute Project, "Energy efficiency," (accessed on 21/9/2013). [Online Available]: http://opencompute.org/about/energy-efficiency/

[18] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," Computer, vol. 40, no. 12, pp. 33–37, 2007.

[19] X. Fan,W. D.Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA), 2007, pp. 13–23.

[20] C. D. Patel, C. E. Bash, R. Sharma, M. Beitelmal, and R. Friedrich, "Smart cooling of data centers," in Proceedings of the Pacific RIM/ASME International Electronics Packaging Technical Conference and Exhibition (InterPACK), 2003, pp. 129–137.

[21] Tom Gu´erout and Mahdi Ben Alaya, "Autonomic energy-aware tasks scheduling", Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2013.

[22] Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie and Jicheng Hu, "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing", IEEE 2009

[23] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms,"Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[24] Ying Chang-tian and Yu Jiong, "Energy-aware Genetic Algorithms for Task Scheduling in Cloud Computing", IEEE Seventh ChinaGrid Annual Conference, 2012.

[25] Liu Yong, Wang Xinhua, Xing Changpling and Wang Shuo, " Resources Scheduling Strategy Based on Ant Colony Optimization Algorithms in Cloud Computing", Computer Technology and development, 2011

[26] Suraj Pandey, LinlinWu, Siddeswara Mayura Guru and Rajkumar Buyya, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

[27] QI CAO, ZHI-BO WEI and WEN-MAO GONG, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing", IEEE 2009

[28] Mrs.S.Selvarani and Dr.G.Sudha Sadhasivam, "IMPROVED COST-BASED ALGORITHM FOR TASK SCHEDULING IN CLOUD COMPUTING", IEEE 2010

[29] M. Salehi and R. Buyya. Adapting market-oriented scheduling policies for cloud computing. In Algorithms and Architectures for Parallel Processing, volume 6081 of Lecture Notes in Computer Science, pages 351–362. Springer Berlin / Heidelberg, 2010.

[30] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer and Ivan Breskovic, "SLA-Aware Application Deployment and Resource Allocation in Clouds", 35th IEEE Annual Computer Software and Applications Conference Workshops, 2011.

[31] Chandrashekhar S. Pawar and Rajnikant B. Wagh, “Priority Based Dynamic resource allocation in Cloud Computing”, IEEE International Symposium on Cloud and Services Computing, 2012.

[32] WANG Han, TANG Xiao-qi, SONG Bao and TANG Yu-zhi, “Dynamic Task-Scheduling Algorithm in CNC System Based on Cloud Computing”, IEEE Second International Conference on Instrumentation & Measurement, Computer, Communication and Control, 2012.

[33] Yiqiu Fang, Fei Wang, and Junwei Ge, “A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing”, Springer-Verlag Berlin Heidelberg, 2010.

[34] R. JEYARANI, N. NAGAVENI, and R. VASANTH RAM, ” Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment”, 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010.

[35] Amit Kumar Das, Tamal Adhikary, Md. Abdur Razzaque and Choong Seon Hong, “An Intelligent Approach for Virtual Machine and QoS Provisioning in Cloud Computing”, IEEE 2013.

[36] Chappell D. Introducing the Azure services platform. White Paper, October 2008.

[37] Google App Engine, Available at: http://appengine.google.com (January 10, 2014)

[38] Quiroz A, Kim H, Parashar M, Gnanasambandam N, Sharma N. Towards autonomic workload provisioning for enterprise grids and clouds. Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (Grid 2009), Banf, AB, Canada, 13–15 October 2009. IEEE Computer Society: Silver Spring, MD, 2009; pp. 50–57.

[39] Eucalyptus Beginner's Guide

[40] Raghavendra Achar, P. Santhi Thilagam, Shwetha D, Pooja H, Roshni_ and Andrea, “Optimal Scheduling of Computational Task in Cloud using Virtual Machine Tree”, Third

International Conference on Emerging Applications of Information Technology (EAIT), 2012.

[41] W. Forrest. How to cut data centre carbon emissions [EB/OL]. http://www.computerweekly.com/Articles/2008/12/05/233748/How-to-cut-data-centre-carbon-emissions.htm. August 2011.