

SCHEDULING IN CLOUD COMPUTING

Thesis Submitted in Partial Fulfillment of the Requirement for the Award of the degree of

**MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEM**

**SUBMITTED BY
DEEPTI SARASWAT
(2K12/ISY/10)**

**UNDER THE GUIDANCE OF
DR. N. S. RAGHAVA
ASSOCIATE PROFESSOR**



**DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
BAWANA ROAD, DELHI-110042
(2012-2014)**

CERTIFICATE

This is to certify that the thesis entitled “**Scheduling in Cloud computing**” submitted by **Deepti Saraswat (2K12/ISY/10)** to Delhi Technological University, Delhi for the award of the degree of **Master of Technology** is a bonafide record of research work carried out by her under my supervision.

The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. N.S. Raghava
Project Guide
Associate Professor
Department of Information Technology
Delhi Technological University, Delhi

Date:

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and **Delhi Technological University**. I would like to extend my sincere thanks to all of them.

I am highly indebted to Dr. N. S. Raghava, *Project guide* for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

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

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

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

Deepti Saraswat

Roll No.: 2K12/ISY/10

Dept. of Information Technology

Delhi Technological University

ABSTRACT

Cloud computing is a way through which a user can access all types of resources from the location that is different from the location where his system presents. Basically cloud computing is a combination of grid computing, utility computing (Complete package of computer resources that can be used as a metered service) and autonomic computing (ability to do self management). Job scheduling is one of the most necessary tasks in a cloud computing environment because cost is associated with every resource based upon the time that is used by the user.

Generally such type of job scheduling algorithm that gives optimized solution in cloud computing environment. Also, the utilization of resources must be efficient, for that purpose, we have to build such type of job scheduling algorithm that does better utilization of resources. There have been various varieties of scheduling algorithm applied in a distributed computing system. By making certain changes, we can apply them in a cloud computing environment. The main goal of the job scheduling algorithm is to gain high computation and the best system throughput. In our proposed scheduling algorithm, Jobs are submitted to the virtual machine by checking the best allocation condition through which maximum no of jobs are executed and also best utilized the resources.

In our algorithm, a predefined number of resource instances have been taken, but the number of jobs for execution is given by the user as an input, while considering that arrival time of all jobs are same. This algorithm is implemented in cloudsim platform with cloudletspaceshared environment in which migration of jobs between the virtual machine is not possible. In this algorithm a class is created that takes a number of jobs and their requirement of resources as input and allocate them to the virtual machine so that maximum number of jobs will be execute is our output. While at the level of host and datacenter we use timeshared environment for all virtual machines. Spaceshared environment is used according to our requirement. If there is no virtual machine is available for remaining number of jobs then appropriate message is displayed to the user and those jobs are discarded for execution.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION.....	1
1.1 Evolution Of Cloud Computing.....	1
1.2 Cloud.....	2
1.2.1 Private Cloud.....	4
1.2.2 Public Cloud.....	4
1.2.3 Hybrid Cloud.....	6
1.2.4 Community Cloud.....	6
1.3 Virtualization.....	7
1.4 Cloud Computing.....	8
1.4.1 Grid Computing.....	9
1.4.2 Utilitycomputing.....	9
1.4.3 Autonomic Computing.....	10
1.5 Service Models.....	11
1.5.1 Platform As A Service.....	11
1.5.2 Software As A Service.....	11
1.5.3 Infrastructure As A Service.....	12
1.5.4 Database As A Service.....	12
1.5.5 Network As A Service.....	12
1.5.6 Communication As A Service.....	13
1.6 Architecture.....	14
1.7 Components of Cloud Computing.....	15
1.8 Different Roles In Cloud Computing.....	18

LIST OF FIGURES

Figure 1.1: Evolution of cloud computing.....	2
Figure 1.2: Conceptual view of cloud computing.....	3
Figure 1.3: Private cloud vs. public cloud.....	5
Figure 1.4: Types of cloud.....	6
Figure 1.5: Virtualization.....	7
Figure 1.6: Cloud computing features.....	9
Figure 1.7: Overview of Cloud computing.....	10
Figure 1.8: Services of cloud.....	13
Figure 1.9: Cloud computing sample architecture.....	14
Figure 1.10: Components of Cloud Computing.....	16
Figure 1.11: Layered architecture	17
Figure 1.12: Benefits of cloud computing.....	19
Figure 1.13: Cloudsim architecture.....	21
Figure 1.14: Detailed architecture of cloudsim.....	22
Figure 1.15: Class Diagram of cloudsim.....	22
Figure 4.1: Flowchart of Proposed Algorithm.....	38

LIST OF TABLES

Table 4.1: Number of cpu available in virtual machine.....	30
Table 4.2: Number of cpu required by jobs.....	30
Table 4.3: Updated table after execution of job1.....	31
Table 4.4: Updated table after execution of job2.....	32
Table 4.5: Updated table after execution of job3.....	32
Table 4.6: Updated table after execution of job4.....	33
Table 4.7: Updated table after execution of job5.....	33
Table 4.8: Updated table after execution of job6.....	33
Table 5.1: Number of cpu available in virtual machine.....	35
Table 5.2: Number of cpu required by jobs in case 1.....	35
Table 5.3: Number of cpu required by jobs in case 2.....	38
Table 5.4: Number of cpu required by jobs in case 3.....	40
Table 5.5: Number of cpu required by jobs in case 4.....	43
Table 5.6: Number of cpu required by jobs in case 5.....	45

CHAPTER 1

INTRODUCTION

1.1 EVOLUTION OF CLOUD COMPUTING

Cloud computing has several benefits over previous technologies such as grid computing, parallel computing, autonomic computing etc. During comparison between grid computing with cloud computing we find several changes while delivering the services. Cloud computing is a form of distributed computing. cloud computing become famous day by day due to its several feature for example it provides high scalability while delivering so many services. Many cloud provider companies such as Amazon, Microsoft etc is developing new things day by day in cloud computing. Cloud computing definition and benefits vary from people to people. For example, for IT managers, it means to focus on reducing the capital expenditure by outsourcing maximum number of the hardware and software resources. For tie-in multiple universities into a massive computing grid long distance optical network are used in Europe so that resources could be shared and scaled for large scientific calculations.

Grid computing also provides a virtual pool of resources for computation, but it's different than cloud computing. Grid computing particularly refers to running several computers in parallel to solve a particular, individual problem, or to run a specific application. Cloud computing, on the other hand, refers to using multiple resources, including computing resources, to deliver a unified “service” to the end user. In case of grid computing, computers coordinates and communicates by passing messages to each other.

ERP (Enterprise Resource Planning) applications are only afford by big enterprises because they have big IT budgets. But it is very difficult for medium class companies because it is very costly. This ability of SaaS helps small companies because they deliver expensive applications at affordable cost.[7] Now a days Cloud computing becomes a hotspot technology in the world. It provides group of shared pool of hardware and software resources on the basis of demand over Network on pay per use. Figure 1.1 has taken from Google which gives overview of cloud computing evolution.

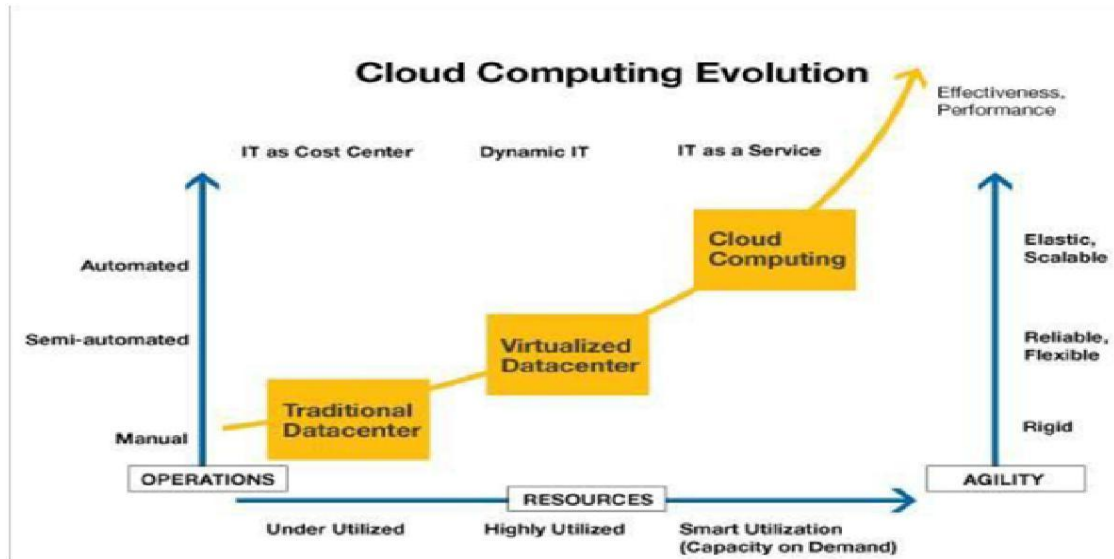


Figure 1.1: Cloud Computing Evolution

1.2 CLOUD

The cloud is just a synonym for the internet. Cloud consists of datacenter, clients and distributed servers. While constructing cloud we have to take care of certain parameters such as fault tolerance, scalability, reliability, sustainability, high availability, flexibility etc. Cloud is like a big pool of resources whose size is dynamically scalable. According to business needs, they grow their storage and computing capacity to meet consumer resources or services on the cloud expansion of demand. [1] Various dynamic services are provided by the cloud computing environment with high scalability and reliability. The definition varies according to the prospective of service provider and the customers Also the scope of cloud computing varies according to customer and service provider. Figure 1.2 has taken from Google which explains conceptual view of cloud computing.

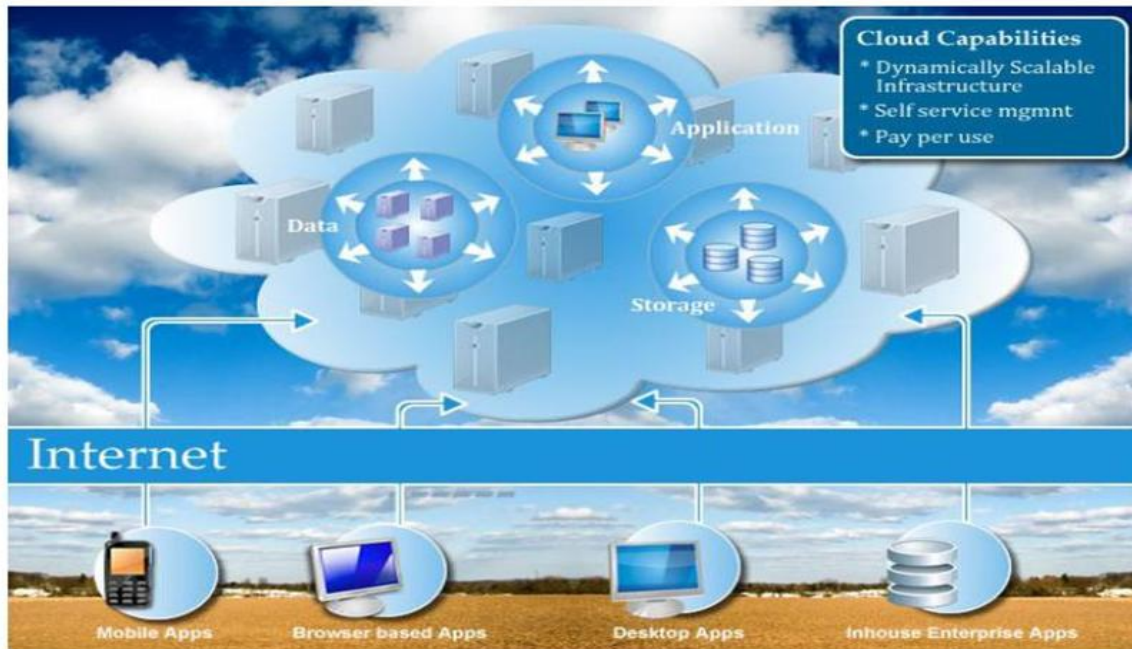


Figure 1.2: Conceptual View of Cloud Computing

CHARACTERISTICS OF CLOUD:-

Infinite scalability:

From the customer point of view, cloud is like a black box in which things are hidden which are not visible to the user. The user only sees the specific services that are provided by the cloud, but not aware about its internal working. It's easy to expand the cloud when it is required.

Low-cost expansion:

Cloud architecture must be able to use its ability to integrate various types of software and hardware resources. It also includes the integration of infrastructure. However, one thing is to note that: low-cost expansion and low-cost structure is different, low-cost expansion is mainly focused on the characteristics of the cloud structure, cloud components that not all low-cost equipment, expansion and the cost is inversely proportional to the unit cost as the cloud continues to expand gradually reduced, which is to develop the concept of cloud and

the presence of the cornerstone.

1.2.1 PRIVATE CLOUD

Private cloud is used only within the organization i.e. access is restricted only for some people. A private cloud is mainly used by an organization to main its privacy and security,for that purpose cloud is only accessible by its authorized employees. We can also develop our own private cloud by using certain platform that is available online.

Private cloud can be classified into two categories:-

1) On-premise:

These are also remembered by internal clouds. Here the term on-premise means that the private cloud is hosted within its own datacenter. Those applications which requires total control, configurability of the infrastructure and security of an organization is used this type of cloud.

2) Externally hosted:

Here externally hosted term means that these are not require to be hosted within its own datacenter. In this type of cloud the provider provides a separate cloud environment by which we can achieve high level of privacy.

1.2.2 PUBLIC CLOUD

In public cloud all the applications are shared to the public. User can easily use services from the internet. Also low cost is given to the service provider for accessing application from the cloud because physical and software resources are distributed among various clients.

Following are the characteristics of public cloud:

High scalability: - Hardware and software resources are easily available to the customer from the pool of resources according to their requirement i.e. dynamically.

Cost effective: - public clouds combine together greater levels of resource that provides benefits for the organization at large scale.

Various examples of public clouds are:

1. Amazon company provides EC2 public cloud for users.
2. Google provides Google App Engine
3. Microsoft Azure platform is also used widely.

Figure 1.3 has taken from Google that represents difference between private cloud and public cloud.

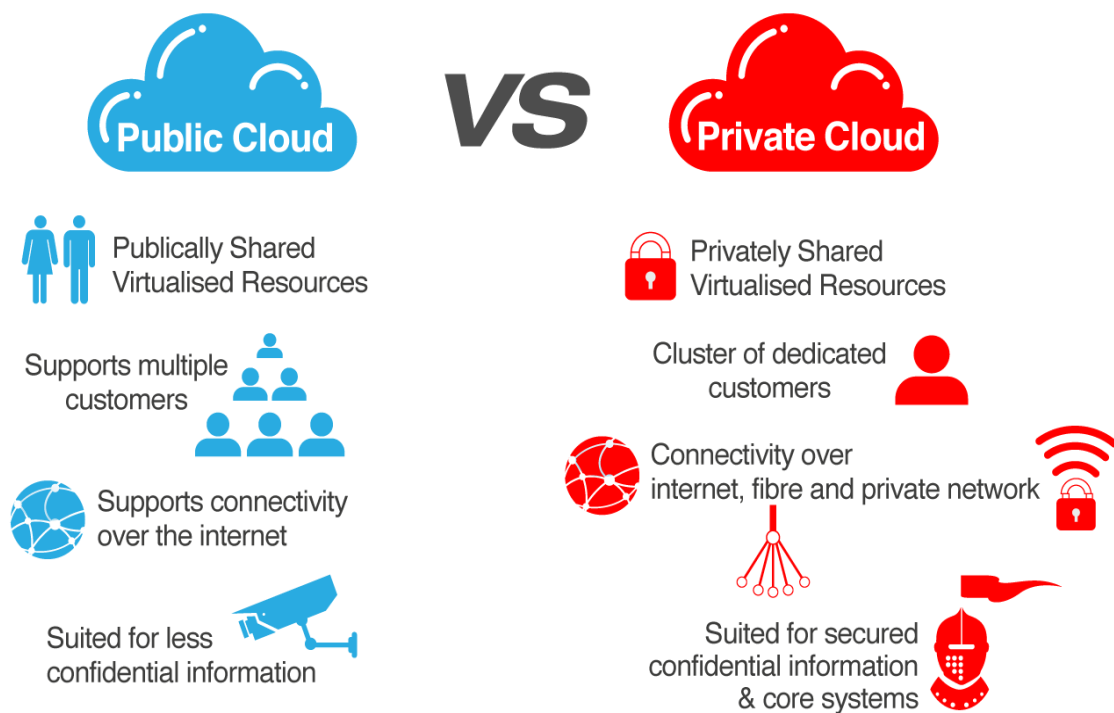


Figure 1.3: Private Cloud vs. Public Cloud

1.2.3 HYBRID CLOUD

It is a combination of private and public models. It utilizes 3rd party Cloud Providers in a full or partial manner which increases the flexibility of computing.

1.2.4 COMMUNITY CLOUD

Community cloud is used by organizations who want to share physical and software resources and belongs to same community.

Example:

All government hospitals within Delhi may share some resources on the cloud to handle and maintain data related to patients. Community clouds are the hybrid form of private cloud. This cloud is only used if some groups have a common goal and they want to maximize their profits from the customer point of view. Figure 1.4 has taken from Google that represents types of cloud .

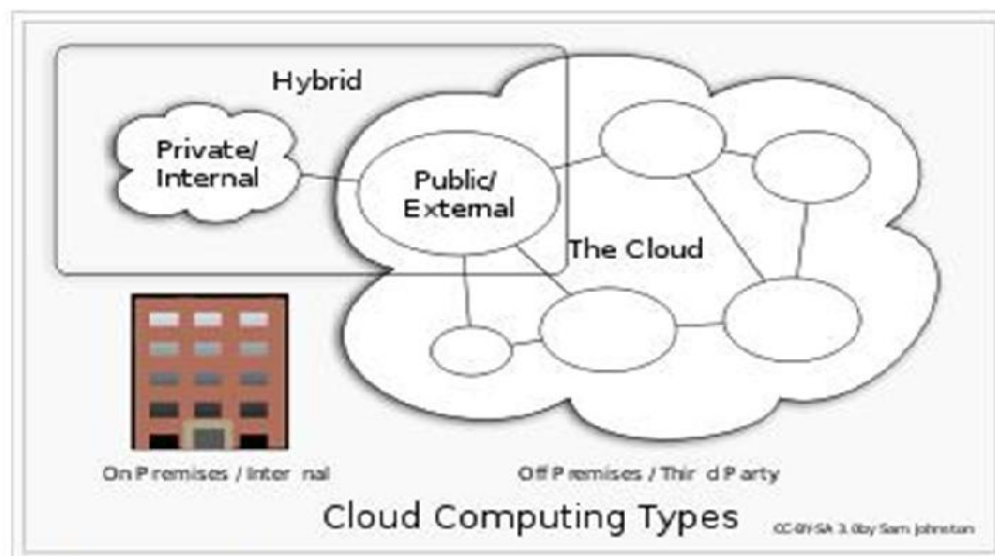


Figure 1.4: Types of Cloud

1.3 VIRTUALIZATION

Virtualization is the key concept of cloud computing. It is the process of creating virtual version of something that may be hardware or software resources. In case of infrastructure as a service we have to do hardware virtualization and in case of software as a service we have to do software virtualization. We can create virtual version of operating system, storage device or computer network resources. With the help of virtualization we can allow to share servers and storage devices through which utilization is automatically increased. Application can be easily migrated from one physical server to another. The technology starts when the mainframe computer comes due to which expensive power of processing reduces. Figure 1.5 has taken from Google that gives overview of virtualization.

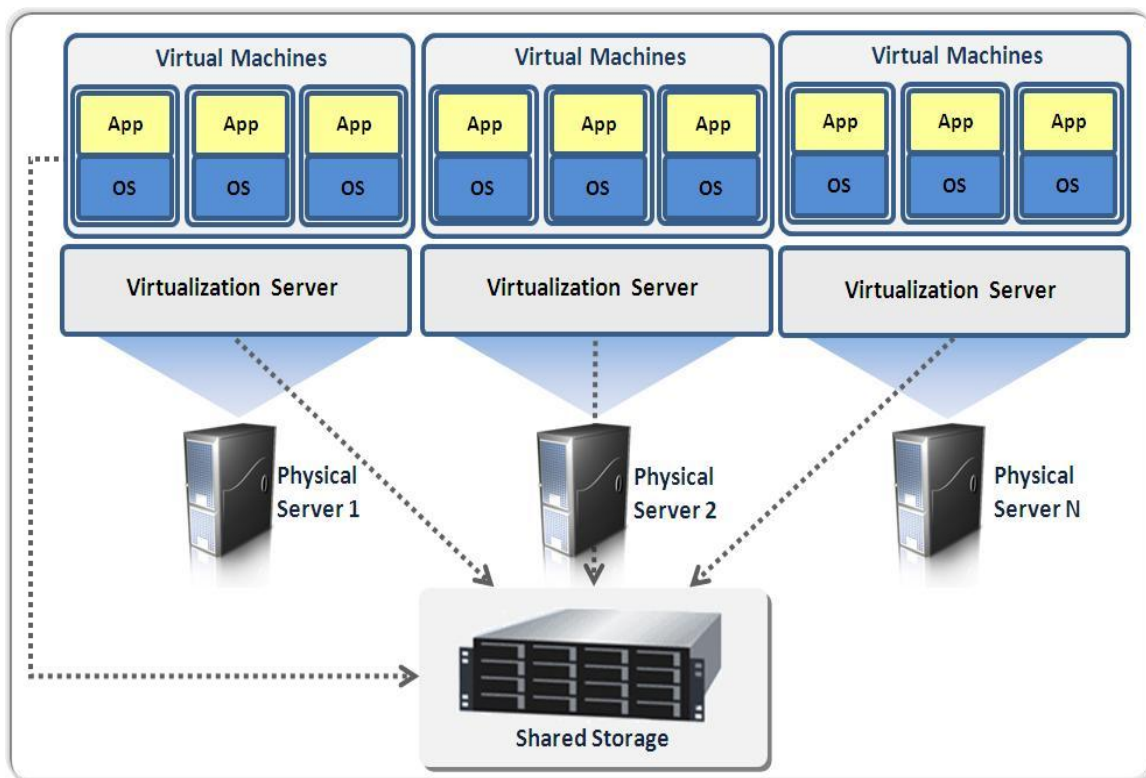


Figure 1.5: Virtualization

1.4 CLOUD COMPUTING

Cloud computing comes into picture when IT needs to increase the capacity and capabilities without spending money in new infrastructure, training new employee or licensing new software. Cloud computing provides everything over the internet as a service to the customer and take specific cost related to services that is used by the customer.

In Cloud computing, computing performs over the entire network, where a program or an application may run on many computers simultaneously that are connected to each other. A server is a collection of computing hardware machine or a single hardware machine. These machines are connected through a network of communication through internet, intranet, local area network, metropolitan area network. Only an authorized person can use the processing power of server for running application, storing data or performing any task. Therefore, it provides benefits to personnel that he/she run their application from any location of the world. [12] For the purpose of scientific researches, a large number of resources and equipments for storage, computation can be done with the help of cloud computing. Job scheduling algorithms are to be applied in submission of jobs so that they will give more efficient and optimized solution. For example [18] Amazon Elastic Compute Cloud (EC2) is an IaaS cloud that is most famous now a days.

Cloud computing is a more young,smart and promising phenomenon than either of its precedent core technologies as its offers best of all of them along with commercial driving force to further enhance it right from its inception.cloud computing has been able to draw attention from the users,industry and developers equally for it has got something to offer everyone.cloud computing is very beneficial option for enterprises as they save cost,have the advantages for users ,more efficient and productive development environment for developers.

Figure 1.6 has taken from Google explains the features



Figure 1.6: Cloud Computing Features

1.4.1 GRID COMPUTING

Through Grid computing a common goal is achieved by the collection of computer resources from multiple locations in the world. [16] Grid computing controlled and managed various problems in dynamic way in multi- institutional virtual organizations. We can compare grid with the distributed system with non-interactive workloads that contains a varieties of files.

1.4.2 UTILITY COMPUTING

Utility computing is mainly serving a package that contains various types of computing resources for computation, storage and services. The main advantage of this model is that it has low or no initial cost for acquiring computer resources. There are certain company that is leaders in the utility computing such as IBM, HP and Microsoft.

1.4.3 AUTONOMIC COMPUTING

Self-managing features of distributed computing resources are come onto the autonomic computing. One of the features is dynamicness that is it hides internal complexity to operators and users while adapting any unpredictable changes. It is started by IBM in 2001. The main motive of this computing is to develop such type of computer system which have capability of self management and removes the hindrance that comes in the growth of computing system self management.

There are certain policies which are of high level through which system takes decision by its own. It will periodically check and optimize its status and automatically adapt itself to changing situations. Figure 1.7 has taken from Google which gives overview of cloud computing.

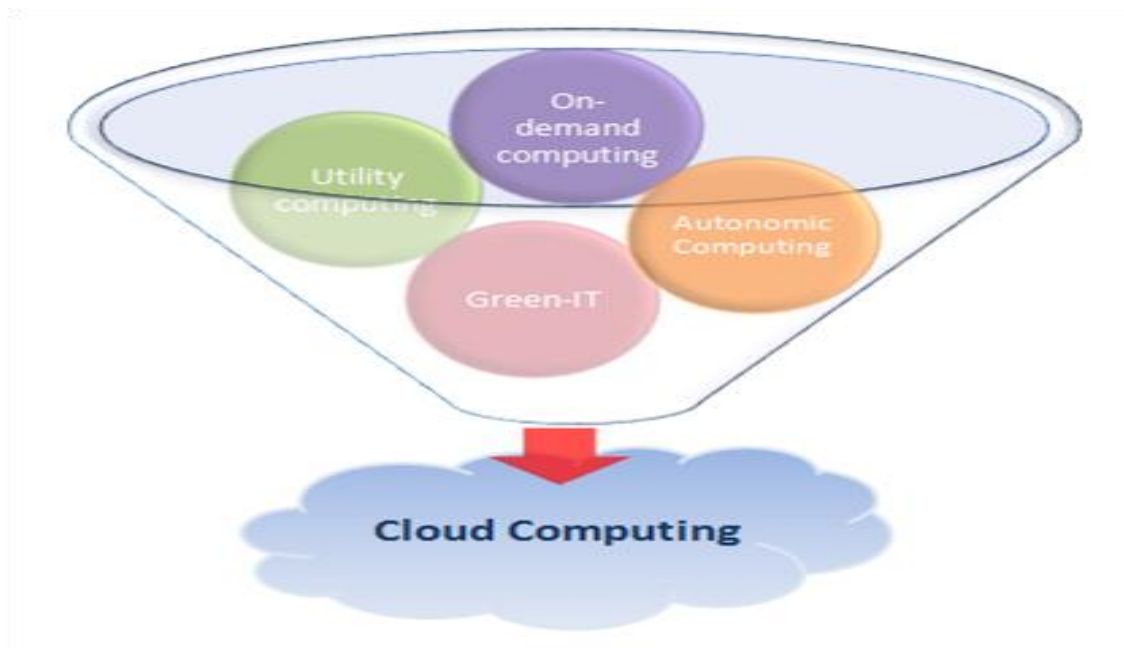


Figure 1.7: Overview of Cloud Computing

1.5 SERVICE MODELS

Service delivery in Cloud Computing comprises three different service models, namely Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS) and Infrastructure-as-a-Service (IaaS).

1.5.1 PLATFORM AS A SERVICE

Platform as a service means that a user can access application on any platform that is available from the cloud by paying some cost. So user doesn't require to purchase its own platform. Two components of Paas:-

PRODUCER:- Someone produces PaaS by integrating an OS, middleware, application software, or can be developed a environment that is given as a service to customer.

CONSUMER:- They use the concept of API. Encapsulated services are available in the form of API to the uses platform as a service.

1.5.2 SOFTWARE AS A SERVICE

In software as a service, a complete application is given as a service on demand. For providing services to multiple users or client organizations, only a single instance is required to run in the cloud. Authorized user can access multiple varieties of application in the cloud.

Characteristics of SaaS

- Commercial software can be accessed thorough web.
- Centralization concept is used for managing the software.
- “one to many” model is used for the delivery of software.

- Software upgrades and patches are not required to be handled by the users.
- With the help of API we can integrate various components of software.

Ex: salesforce.com, Google Apps.

1.5.3 INFRASTRUCTURE AS A SERVICE

Various types of software and hardware resources that are used for computation, storage etc are delivered as a service to users.

Characteristics of IaaS

- Distribution of resources is available in the form of service.
- If number of users increases then it automatically changes its distribution of resources that is it also provides scalability.
- Its cost is varying according to the utilization.
- Common hardware is shared among various users.

1.5.4 DATABASE AS A SERVICE

With a DBaaS, developer of application doesn't require to have the knowledge of database. Also they don't require hiring third party to handle administration of database. Cloud service provider handle all things related to the database.

1.5.5 NETWORK AS A SERVICE

Naas sometimes includes the provision of a virtual network service by the owners of the network infrastructure to a third party. Often this includes network virtualization using a protocol such as OpenFlow.

1.5.6 COMMUNICATION AS A SERVICE

Communication as a Service (CaaS), allows the consumer to utilize Enterprise level VoIP, VPNs, PBX and Unified Communications without doing investment in purchasing, hosting and managing the infrastructure. The other advantage is that the consumer doesn't require their own trained personnel, bringing significant OPEX as well as CAPEX costs for management and running the services.

Other Services in cloud computing are listed as follows:

- Network as a service(NaaS)
- Storage as a service (STaaS)
- Security as a service (SECaaS)
- Data as a service (DaaS)
- Desktop as a service (DeaaS)
- Testing as a service (TaaS)

Figure 1.8 has taken from Google represents various services of cloud.

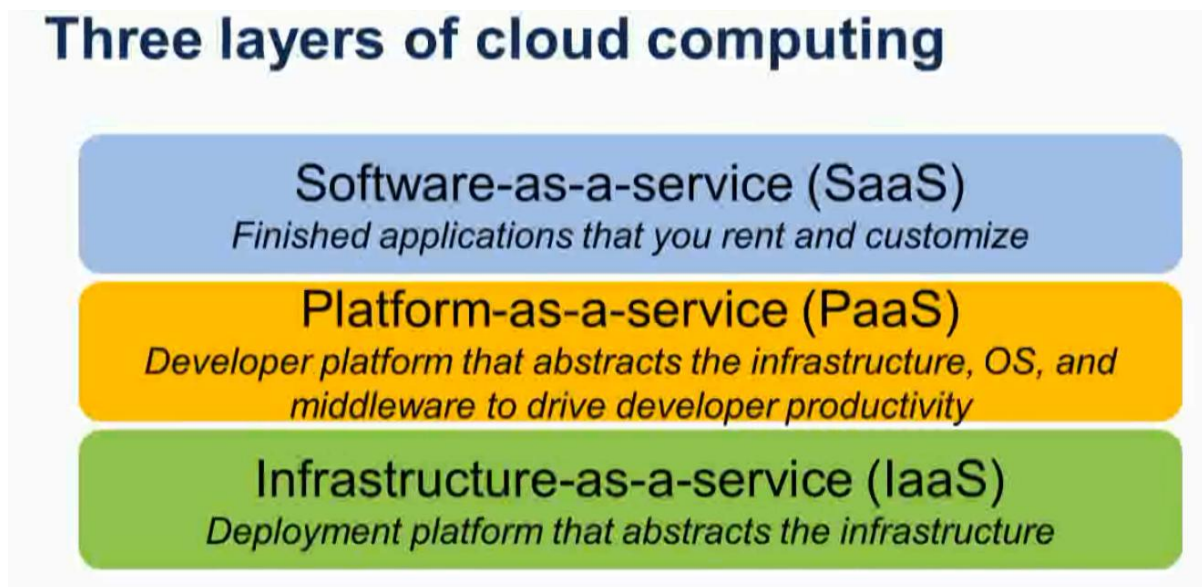


Figure 1.8: Services Of Cloud

1.6 ARCHITECTURE

It's mainly composed of four components cloud service, cloud platform, cloud infrastructure and cloud storage. Clients can access cloud application with the help of web browsers. Architecture of cloud storage is loosely coupled. In this metadata operations are centralized due to which data nodes are scaled into the thousands and each separately delivers data to various users. Figure 1.9 has taken from Google that describes the complete architecture of cloud computing.

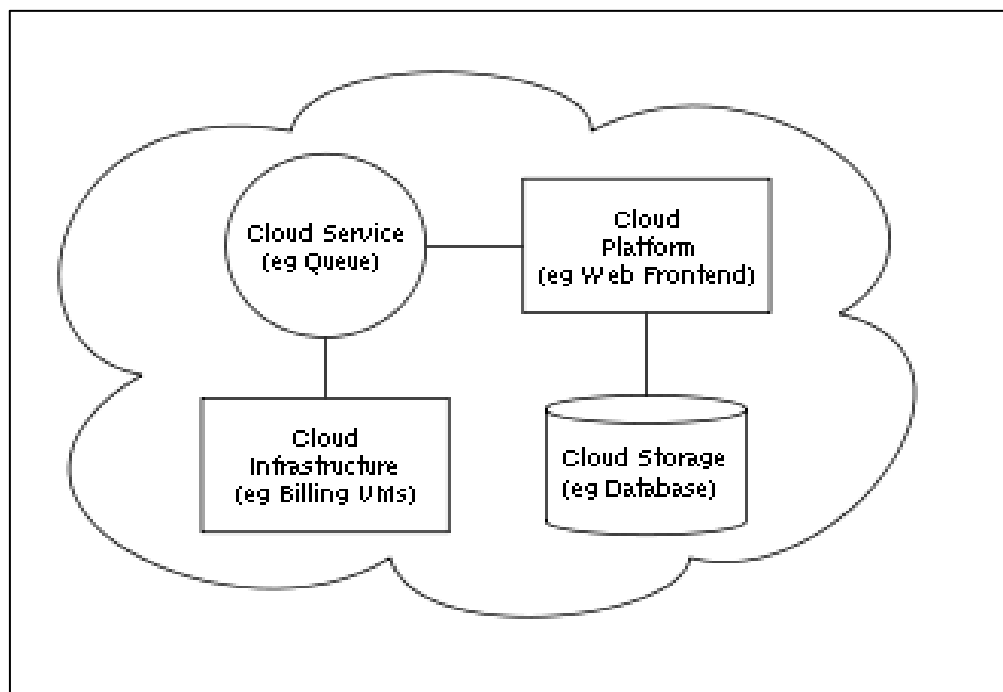


Figure 1.9: Cloud Computing Sample Architecture

1.7 COMPONENTS OF CLOUD COMPUTING

Various components of cloud computing are discussed below:

Client:

- A cloud client may be hardware or software which uses cloud computing for accessing application.

Examples

- Cell phones using various operating systems such as android, Microsoft windows.
- Thin client.
- Different types of web browsers such as googlechrome, Mozilla Firefox, internet explorer etc.

Service:

- A cloud service includes "products, services and solutions that are delivered and consumed in real-time over the Internet".

Some examples of services are:-

- For unique identification of users we can use opened.
- Google maps are used for searching any location in the world.

Application:

A cloud application is accessed by multiple numbers of users simultaneously from any location in the world. Also user doesn't require to download, install application in his system had taken the burden of handling the maintenance. For example:

- Skype software launch by Microsoft is used for videochatting
- Facebook
- Sap and Google Apps

Figure 1.10 has taken from Google represents all component of cloud cloudcomputing.

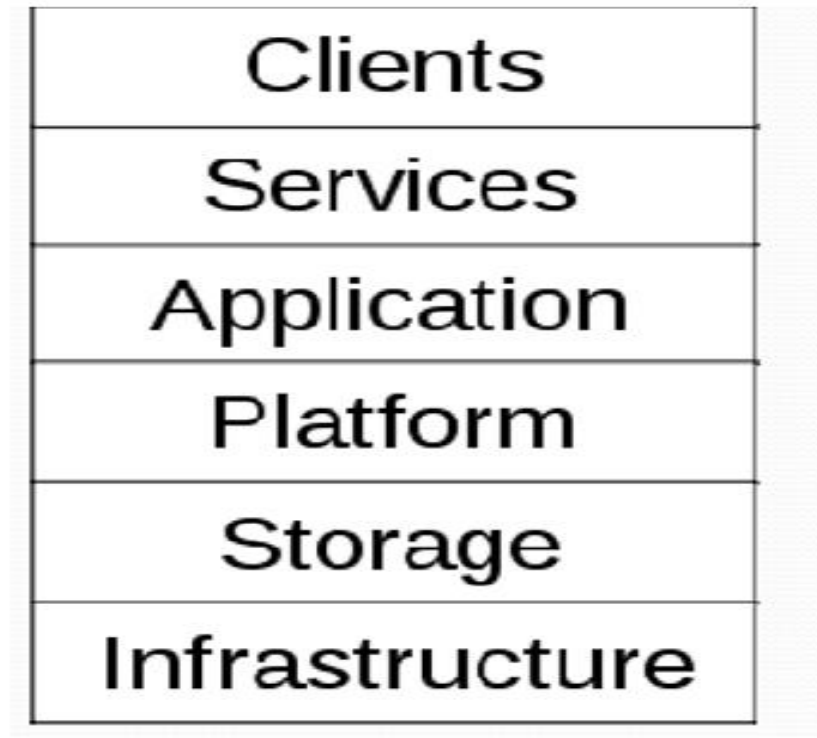


Figure 1.10 Components of Cloud Computing

Platform:

A platform is given as a service in cloud computing environment. We require some specific configuration to run our application so we do not require to purchase it. We just use platform from the cloud as a service and run our application. For using the platform as a service we have to pay some cost according to utility of resources which have been already decided by cloud service provider.

Storage:

Storage for data of user is given as a service by cloud computing. For using this storage user must pay same cost on the basis of utility.

Infrastructure:

Complete infrastructure for running any application is given by the cloud service provider. For that user pay same cost according to amount of infrastructure used and also includes duration of using infrastructure.

Figure 1.11 has taken from Google that represents the layered architecture

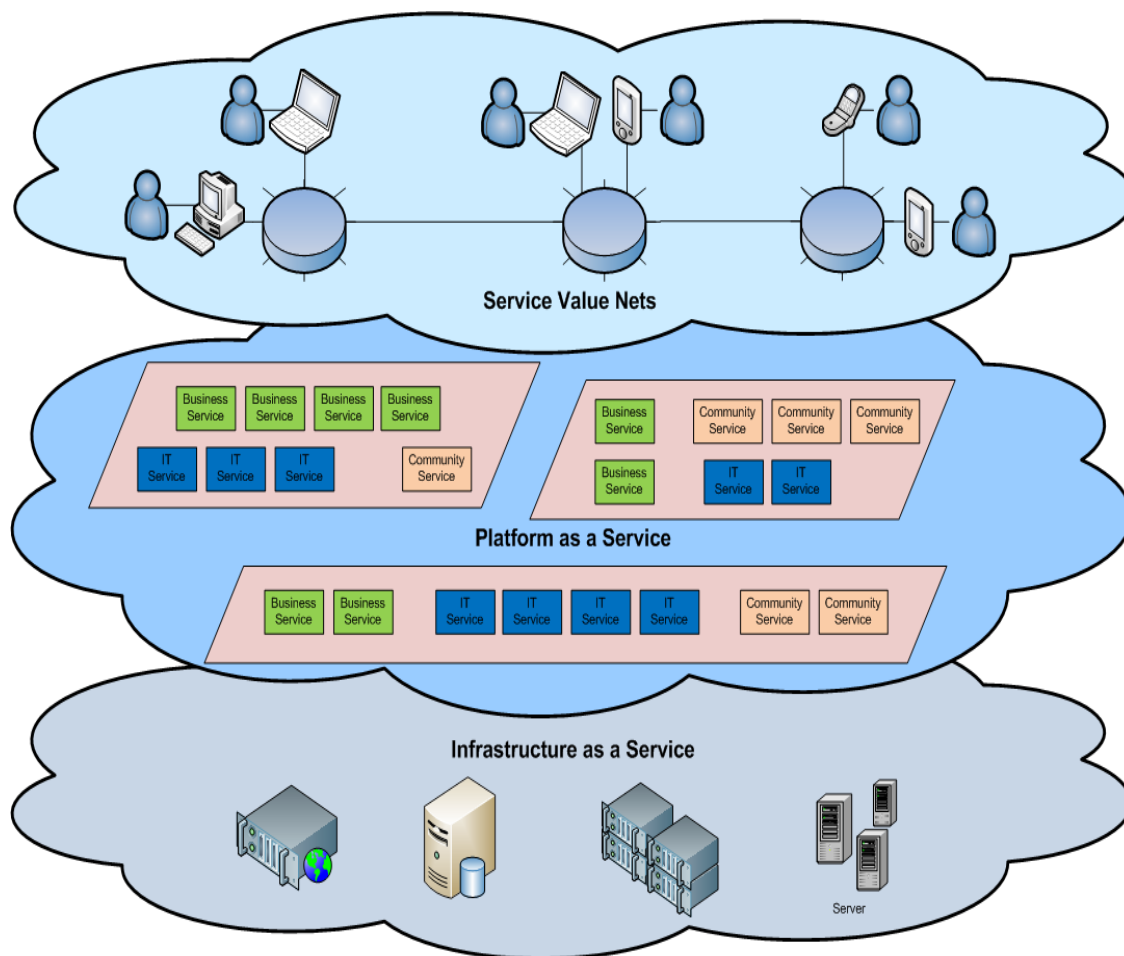


Figure 1.11: Layered Architecture

1.8 DIFFERENT ROLES IN CLOUD COMPUTING

1. PROVIDER:

The main aim of cloud computing service provider is to operate and manage live cloud computing systems for delivering various types of services to third parties.

2. USER:

A user uses various services of cloud computing. The main concern in cloud computing is to provide privacy to the user and also provide security to the data that is stored in the cloud.

3. VENDOR:

A vendor is an intermediate between sells products and services that provides the delivery, adoption and use of cloud computing.

1.9 ISSUES IN CLOUD COMPUTING

- Job scheduling
Job scheduling is very important issue in cloud computing for the proper utilization of resources. By proper scheduling of jobs we can increase the efficiency of cloud computing environment.
- Load balancing.
Load balancing is used for the proper distribution of task on the resources. so none of the resource remains idle or also none of the resource will get overload.
- License management
There is an agreement of license between cloud service provider and the client before delivering their resources as a service to the client
- Scalability

In cloud instance of resources are added or deleted according to the demand of user. if demand of user increases then automatically instance of that particular resources increases and also viceversa happens.

- Availability of services

Cloud provides every type of resources in the form of service. user can use that service whenever he/she requires by paying some amount to the cloud service provider.

- Unpredictable performance due to virtualization

We can't predict about the virtual machine deployment without knowing about the exact requirements from the user side.

- Mobile cloud

Now a day's mobile devices are considered as a mini computer, almost all the features like accessing the internet, sharing the data, images can be easily done with these devices

Figure 1.12 has taken from Google that explains several benefits of cloud computing

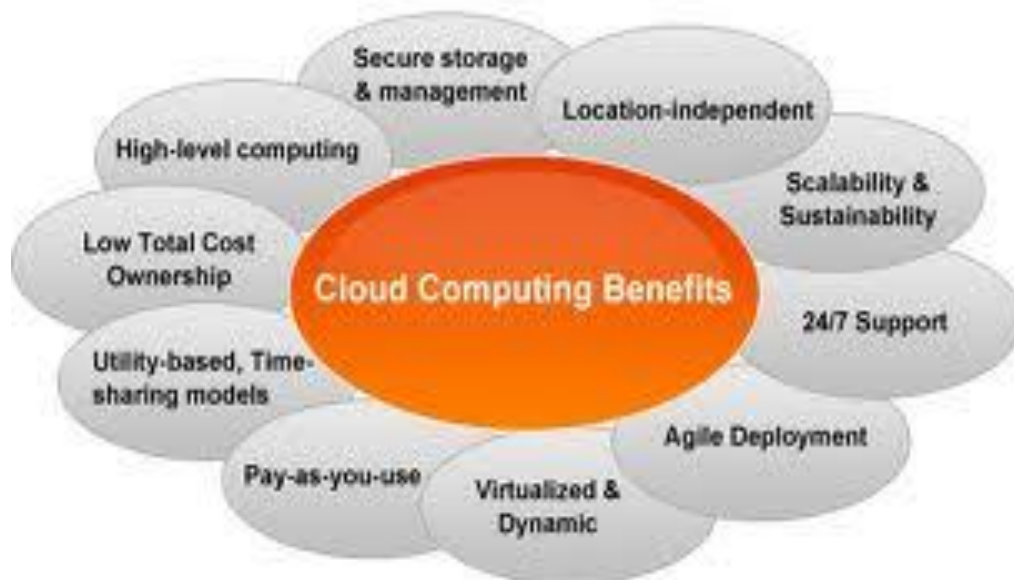


Fig 1.12: Cloud Computing Benefits

1.9 CLOUDSIM

Cloudsim is a new, generalized and extensible simulation toolkit that is basically designed to simulate cloud computing environment. In cloudsim each entity such as datacenter, host, virtual machine etc are represented in the form of class of java.

By using cloudsim Cloud customers can test their services in repeatable and controllable environment, so that they can apply in real world cloud computing environment. We use eclipse as a platform for running our java classes.It support for user-defined policies for allocation of hosts to virtual machines and policies for allocation of host resources to virtual machines

1.10 ARCHITECTURE OF CLOUDSIM

According to cloudsim architecture, there is intermediate broker between the user and the cloud environment. So user must first submit its job to the broker and then broker after applying some algorithm send it to the virtual machine. Virtual machine is created in hosts and hosts are created in datacenter. A datacenter can contain more than one host and also a host can contain more than one virtual machine and also same thing is between cloudlet and virtual machines. Energy conservation is an important issue for data centers. [22] We have to consolidate the computation workloads of virtual machine into the less number of physical servers, so that energy is conserved. To reduce energy consumption of data centers one must consolidate the computation workloads of virtual machines onto a minimum subset of physical servers.

We can either use timeshared or spaceshard environment in our policies.In case of space shared environment only one task is utilizes the computing power of that resource while in case of timeshared environment they are executed parallely that is at the same time they can use same resources.

Figure 1.13 has taken from Google that explains the detail architecture of cloudsim.

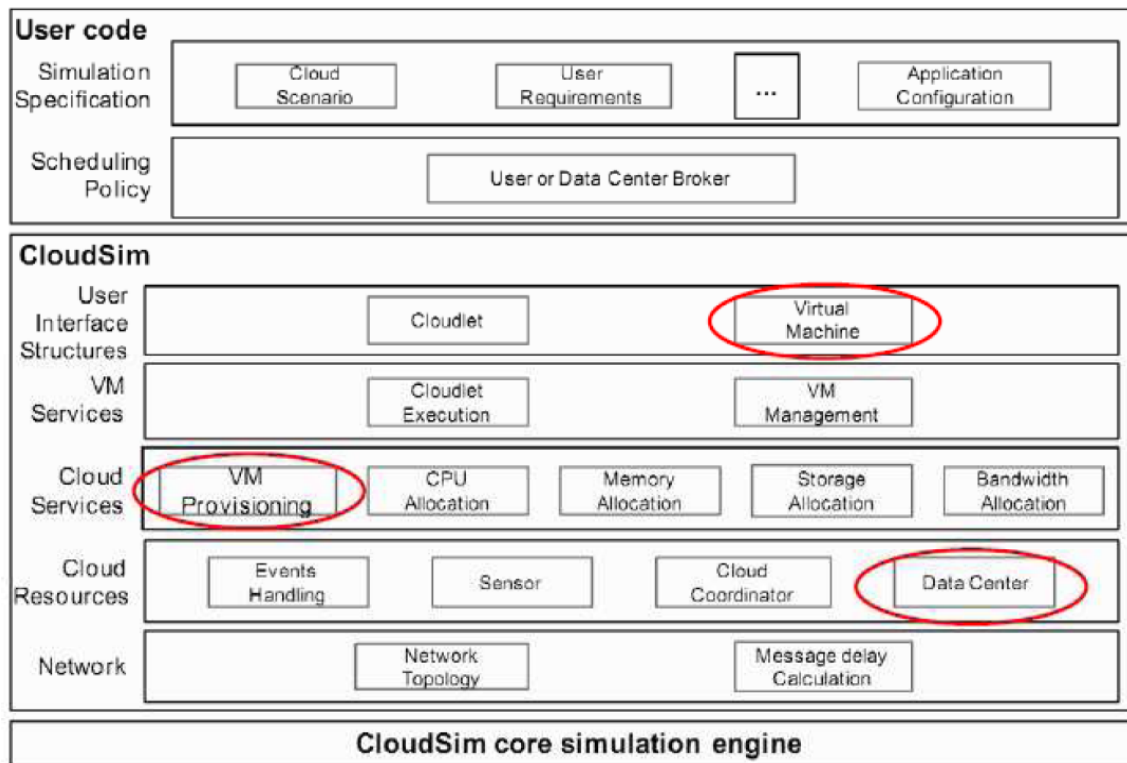


Figure 1.13: Cloudsim Architecture

Figure 1.14 has taken from Google that explains the detailed architecture of cloudsims. So we choose either of them according to our requirement. We have chosen timeshared environment in virtual machine .for some cloudlet we have chosen spaceshared and for some timeshared.

Figure 1.15 explains the class diagram of cloudsims.

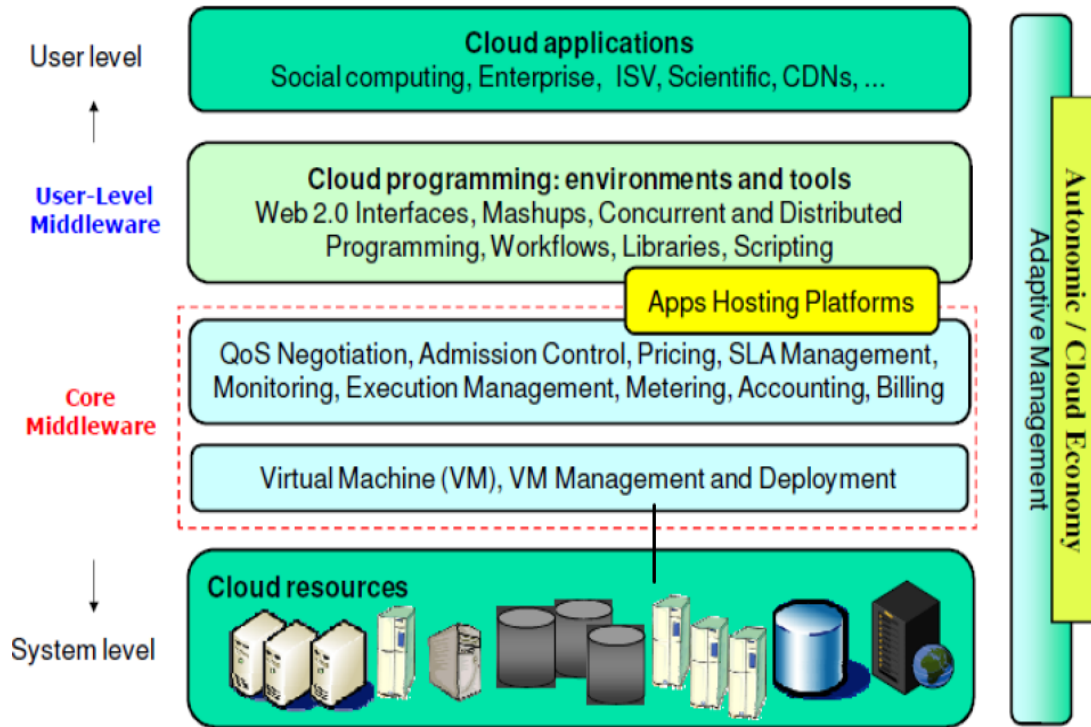


Figure 1.13: Detail Architecture of Cloudsim

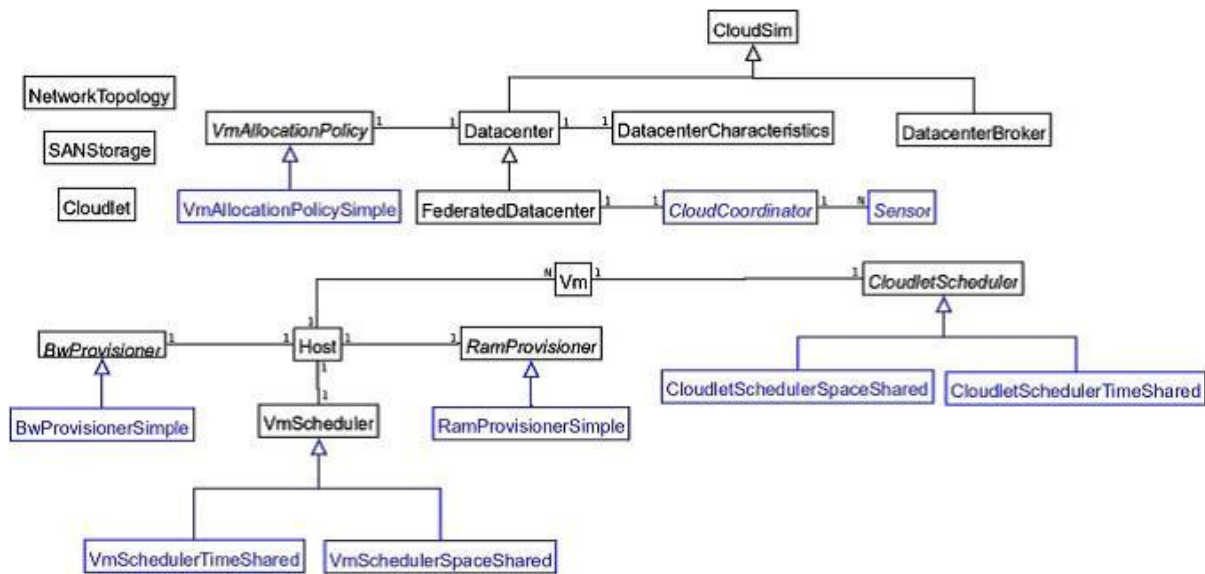


Figure 1.14: Class Diagram Of Cloudsim

1.11 SUMMARY

In this chapter, we have discussed why cloud computing comes into picture and what are the previous technologies due to which the cloud computing arises.also we discussed the concept of virtualization and types of cloud exist in the real world.we also tried to give brief overview of various types of services that are provided by the cloud computing.we tried to explained the detailed and layered architecture of cloud computing.and in last we discuss several benefits of using cloud computing and various issues that are associated with the cloud computing.

CHAPTER 2

SCHEDULING

2.1 INTRODUCTION

Scheduling is one of the major issue that is always considered while running various jobs in cloud computing. The goal of scheduling algorithms in distributed systems is properly utilizing the processing units and reduces the execution time of job. The main purpose is to schedule jobs to the adaptable resources in accordance with adaptable time. By considering various transaction logic constraints we find out a proper sequence of jobs for their execution. [19]Job scheduling is a combinatorial optimization problem in the fields of computer science where the ideal jobs are assigned to required resource at a particular instant of time. Makespan or completion time is the total time taken to process a set of jobs for its complete execution.

Scheduling algorithm can be classified into two groups.

1) Static scheduling algorithm. This algorithm uses the concept of pipelining. It first prefaces the required data and applies pipelining on different stages of task execution. This Scheduling imposes less runtime overhead.

2) Dynamic scheduling algorithm. In this algorithm, we does not have any information about the job that comes for execution. We do not know the execution time of the job may so allocation of job done randomly. Both algorithms having their own benefits and drawbacks. The performance of dynamic algorithm is higher than static algorithm but complexity is higher.[4] In distributed system, varieties types of scheduling algorithm have been used. We can also use that algorithm in environment of cloud by making certain changes and also verify them in our environment. Throughput and performance of the computation is the main criteria for the judgement of any scheduling algorithm. If both parameters are high then it is good. For providing job scheduling algorithm in cloud we cannot use them directly.

Figure 2.1 shows the sequence and relation of components in cloud environment. It is clear with the diagram that we can apply scheduling algorithm in either between host and virtual machine or between virtual machine and Job.

We have developed our scheduling algorithm in between the virtual machine and job.

i.e. Allocation of jobs to the virtual machines are done through our scheduling algorithm.

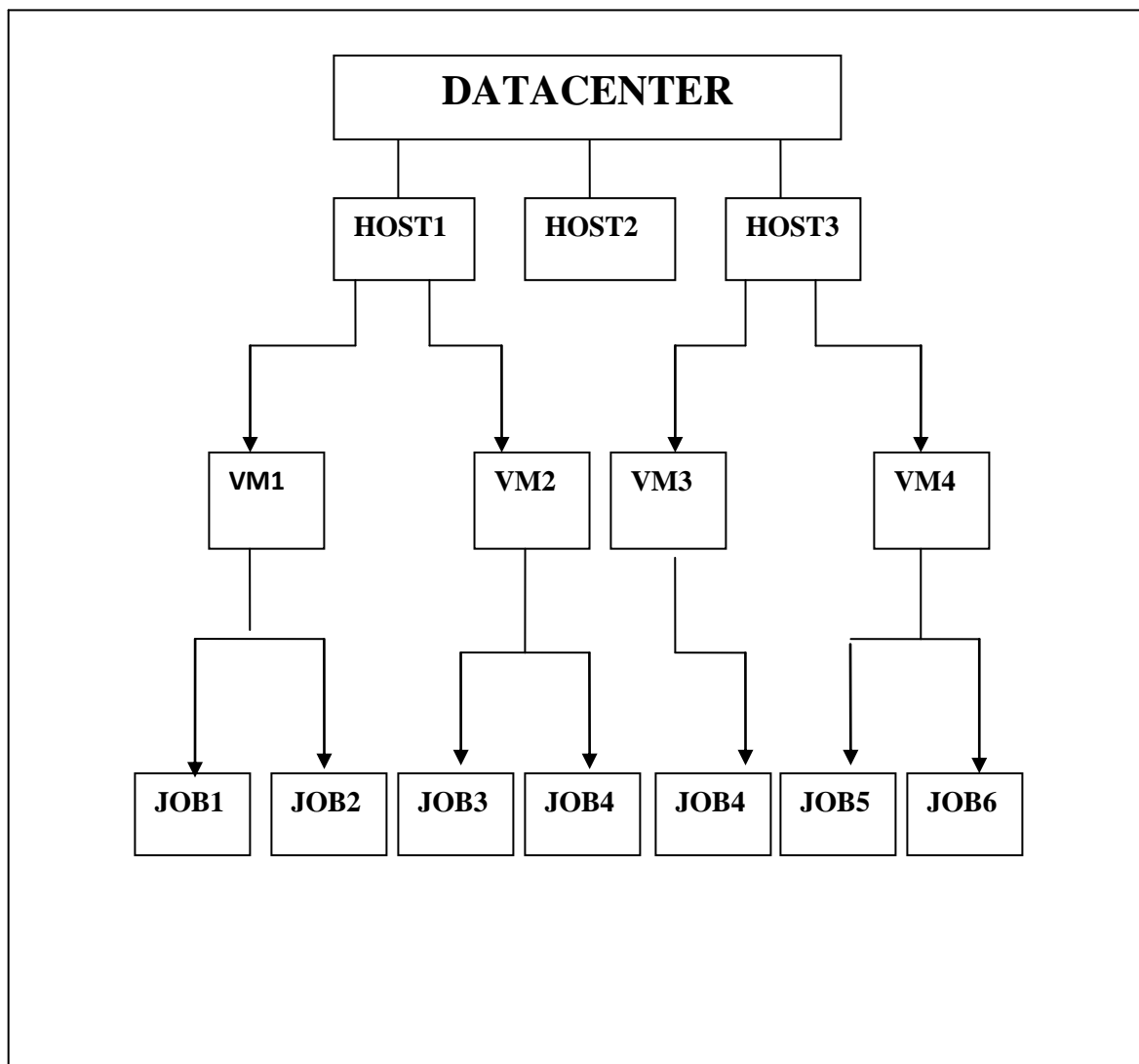


FIG 2.1: SEQUENCE OF COMPONENTS

2.2 TYPES OF SCHEDULING

Job scheduling algorithm can simply classified into two groups:

2.2.1 BATCH MODE HEURISTICS SCHEDULING ALGORITHMS (BMHA)

In BMHA, first we make a queue of jobs and then grouped into a batch when they arrive into the system. So after a certain period of time, scheduling algorithm starts. Min–Min algorithm and Max–Min algorithm First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR) are the examples of BMHA.

2.2.2 ONLINE MODE HEURISTIC SCHEDULING ALGORITHMS (OMHA)

In On-line mode heuristic scheduling algorithm, scheduling of jobs is done when they arrive the system. So, on-line mode heuristic scheduling algorithms are better to apply in cloud environment. Example of On-line mode heuristic scheduling algorithm is Most fit task scheduling algorithm (MFTF)

2.3 PROCESS OF SCHEDULING

Process of scheduling in cloud is divided into three steps:-

Resource discovering and filtering:-resources are discovered by the datacenter broker present in the network information about the status of each resource is also collected by the datacenter broker.

Resource:—On the basis of certain parameters resources and tasks are selected. This is the stage that comes before submission of the job.

Task submission:-Task is given to the selected resource.

2.4 VARIOUS TYPES OF SCHEDULING ALGORITHM

FIRST COME FIRST SERVE ALGORITHM:

In this algorithm first they make a queue of jobs. Then they form a batch of jobs from where they select jobs for execution on the basis that which jobs comes first. It is simple and fast. But disadvantage of this algorithm is that if a job has higher priority then it should wait for his turn which may be sometime not feasible solution.

ROUND ROBIN ALGORITHM:

In this algorithm, processes are executed in FIFO manner but a fix amount of cpu time is given to all process when their turn comes, that time is called a time-slice or a quantum. If a job does not execute before its time-quantum expires, then for the next process that is present in the waiting queue, cpu is preempted. Then the process from which the cpu is taken is placed at the end of the ready list.

MIN-MIN ALGORITHM:

In this algorithm, those processes are chooses first which requires small time for execution. It means it gives higher priority to small jobs. But in this algorithm starvation for longer jobs may be possible.

MAX-MIN ALGORITHM:

In this algorithm, those processes are chosen first which require longer time for execution. It means it gives higher priority to longer jobs. But in this algorithm starvation for smaller jobs may be possible.

MOST FIT TASK SCHEDULING ALGORITHM:

In this algorithm, one task is selected on the basis of fitness value from the queue. Failure ratio of this algorithm is high compared to other algorithms.

PRIORITY SCHEDULING ALGORITHM:

In this algorithm, a priority is assigned to each job either internally or externally on the basis of that priority jobs are selected for execution. Shortest job first (SJF) is one of the examples of priority scheduling algorithms.

2.4 SUMMARY

In this chapter we have discussed the concept of scheduling between jobs and virtual machines. We tried to give a brief introduction of scheduling in cloud computing and also discuss the sequence of components and their relation to each other. Then we discussed various types of scheduling. After that we have discussed already applied scheduling algorithms and then made some comparison between them.

CHAPTER 3

RELATED

WORK

Initially, scheduling algorithms were being implemented in grids. Due to the reduced performance faced in grids, now there is a need to implement scheduling in cloud. The primary benefit of moving to Clouds is application scalability. Unlike Grids, scalability of Cloud resources allows real-time provisioning of resources to meet application requirements. This enables workflow management systems to readily meet Quality of- Service (QoS) requirements of applications, as opposed to the traditional approach that required advance reservation of resources in global multi-user Grid environments. Cloud services like compute, storage and bandwidth resources are available at substantially lower costs. Cloud applications often require very complex execution environments. These environments are difficult to create on grid resources. In addition, each grid site has a different configuration, which results in extra effort each time an application needs to be ported to a new site. Virtual machines allow the application developer to create a fully customized, portable execution environment configured specifically for their application. Traditional way for scheduling in cloud computing tended to use the direct tasks of users as the overhead application base. The problem is that there may be no relationship between the overhead application base and the way that different tasks cause overhead costs of resources in cloud systems. For large number of simple tasks this increases the cost and the cost is decreased if we have small number of complex tasks.

Some of the scheduling algorithm are already applied in cloud computing which also consider same parameters that we have considered in our algorithm. [9] RASA is a combination of two scheduling algorithms; Max-min and Min-min. it mainly uses the benefits of Max-min and Min-min algorithms and overcome their disadvantages. This algorithm doesn't considers deadline of each job, arrival rate of the jobs, cost associated to the execution of job on resources. RASA gives better results than other scheduling algorithm in large scale distributed systems.

In RSDC algorithm[8] main job is divided to sub jobs. Request and acknowledge time of each job is calculated separately for balancing the jobs. The time of request and acknowledgement in the form of a shared job is calculated then scheduling of that job is

done, due to which efficiency of the system is increased.[10] An Optimal Model for Priority based Service Scheduling is a new scheduling algorithm based on priority. A control scheme is to be applied for admission. In this algorithm priority is assigned to each admitted queue. For the admission of each queue tolerable delay and service cost is to be calculated. The main advantage of this algorithm is that it provides guaranteed quality of service and it also achieve very high service completion rate. This algorithm provides main preference to those users which paid highest cost for cloud service.

Extended Max-Min Scheduling Using Petri Net and Load Balancing algorithm[12] based on impact of RASA algorithm.selection criteria of this algorithm is different from max-min algorithm. It uses expected execution time as a selection criterion while max min uses completion time as a selection criterion. In this algorithm, for the work done by cloud user in cloud computing environment they used non-preemptive priority queuing model. For example:-for uploading and downloading the file they require efficient job scheduling algorithm, for that purpose they create only one web application. This algorithm provides high quality of service requirements and maximum benefits to the cloud computing service providers.

Improved Cost-Based Algorithm for Task Scheduling algorithm[14] provides efficient mapping of jobs to available resources in cloud. Three separate lists are created by dividing jobs of user on the basis of priority. This scheduling algorithm calculates both cost of each resources used and performance of computation and also improves the ratio of computation to communication. In this gang scheduling algorithms job migration is possible and also handles the starvation of jobs. Grid and Cluster computing includes parallel scheduling of jobs. Creation of virtual machine is dynamic i.e. according to the the demand of user virtual machine is created or destroyed.

The Static scheduling algorithms make all decisions before Scheduling begins. The simplest is the Round Robin (RR) algorithm. The ready queue in the RR algorithm is a logical ring, and each task is assigned for execution in turn. If a task cannot be completed in its time quantum, it will return to stay in the queue and wait for the next turn. The major advantage of

the RR algorithm is that tasks are executed in turn so that they do not need to wait for the previous largest one to complete. However, it has the problem that if the queue is fully loaded and workload is heavy, tasks might have to wait in the queue for a very long time. In the extreme case, starvation might occur, and more context switching, hence consumes more processing node time.

The Fastest Processor to Largest Task First (FPLTF) algorithm schedules tasks according to the workload in the grid system. The algorithm needs the information of CPU speed and task workload. FPLTF works in two steps: first, the task scheduler sorts tasks and resources to reduce task and CPU searching time. Second, the scheduler allocates the largest task in the queue to the fastest available resource in the grid environment. Dynamic Fastest Processor to Largest Task First (DFPLTF) is an extension of FPLTF. Work Queue with Replication (WQR) is an extension of the Work Queue (WQ) algorithm. It is a knowledge free scheduling algorithm. It does not need any information for scheduling. WQR has the characteristic that a faster processor will be assigned more tasks than a slower processor. Moreover, it does not use any prediction algorithm for CPU speeds and task lengths prediction. WQR uses FCFS and random transfer task to allocate resource node.

In the min-min algorithm, the minimum completion time for each task is computed with respect to all the resource node in a grid. The task with overall minimum completion time is selected first and assigned to the corresponding resource node in a grid. The newly mapped task is removed, and the process repeats until all tasks are mapped. The max-min algorithm is very similar to the min-min algorithm. The collection of minimum completion time is calculated for every task. The task with overall maximum completion time from the group is selected and assigned to the corresponding machine. Once the machine that provides the earliest completion time is found for every task, the task that has minimum completion time is determined and then assigned to the corresponding machine. The max-min algorithm has the same complexity as the min-min algorithm.

The dynamic scheduling algorithms, such as DFPLTF and Suffrage-C, make some or all decisions during scheduling a task on grid. In a grid environment, the load of each processor varies quickly, and dynamic scheduling algorithms are more appropriate than static ones. The DFPLTF algorithm is based on FPLTF with a good ability to adapt to the grid environment. DFPLTF needs three types of information to schedule: Task_size, Recourse_speed and Resource_load. Recourse_speed is the speed of the Recourse and the value of it is relative. Resource_load is the amount of the resource time which is unavailable to the application. Task_size is the time required for the resource to complete the task execution. Initially, the Resource Time to Become Available (RTBA) of each Recourse is initialized to 0. The scheduler will assign tasks to resources according to the best Task completion time (TCT).

CHAPTER 4

PROPOSED ALGORITHM

4.1 MOTIVATION

Scheduling is one of the major issues in cloud computing. With the help of scheduling algorithm we can properly utilize the resources of virtual machines for the execution of jobs. Many algorithms apply before submitting the jobs to the virtual machine. Scheduling algorithm is applied according to the requirement scenario in the cloud. For example, round robin algorithm is used for achieving highest response time. If we want some most important jobs is to be executed first then we applied priority based scheduling algorithm. This algorithm implemented in cloudsim is properly utilizing the resources of virtual machine and allowing a large number of jobs for execution. It increases the scalability and also the response time. Also the complexity of our algorithm is less because it requires less computation. [2] There are an increasing number of Cloud services available in the Internet. Cloud Services provide varieties of services. Different cloud servers are available to provide the relative cloud services to the users so that fit different requirements of different clients are to be fulfilled. To achieve this, an efficient process allocation algorithm is required. To provide the effective allocation without delay, several clouds are grouped together to provide common environment. [6] Job-scheduling algorithms are used for accomplishing several goals like expected outcome, efficiency of utilization of resources, low makespan, and high throughput and also certain other extra parameters.

4.2 PROPOSED ALGORITHM

In this algorithm each virtual machine has its own resources. A condition is apply only on the number of processing elements available in a virtual machine. More than one resources can also be taken in this algorithm. Here virtual machine along with its number of CPU is represented in the form of a table.

This algorithm is explained with the help of scenarios. This scenario contains four virtual machines and each have its own no of CPU as given below in the table.

Table 4.1: Number of cpu available in virtual machine

Virtual machines	No of CPU
Vm1	1
Vm2	2
Vm3	3
Vm4	4

Now the jobs comes for execution in virtual machines. Requirement of CPU by jobs are given below in table.

Table 4.2: Number of cpu required by jobs

Jobs	No of CPU
Job1	1
Job2	1
Job3	2
Job4	1
Job5	4
Job6	1

Here two cases arise:

Case 1:

In this case jobs are allocated to the virtual machines sequentially, such as Job 1 allocates to the first virtual machine. Job 2 allocate to second virtual machine. And similarly job 3 allocates for the third virtual machine job 4 to four virtual machines. But no virtual machines have sufficient number of resources for job5. So job 5 and job 6 are not executed.hence total number of jobs executed are 4.Now comes to case2.

Case 2:

If jobs allocates to the virtual machine by applying some conditions.

For job 1, First the condition is checked that no of CPU available in a virtual machine is greater than zero If it is greater than zero then we find the difference between the no of CPU available in the first virtual machine and no of CPU required by job1. If this difference is greater than zero then we set (min=no of CPU in the first virtual machine-no of CPU required by job1).

This procedure is repeated for each virtual machine. If new min is less than previous min then we reset the value of min.

In our example

For job1 difference is calculated from first virtual machine such as $1-1=0$, Set min=0;

From second virtual machine $(2-1)=1$ hence new min is greater than previous min so we cannot change its value. From third virtual machine it gives $(3-1)=2$ here also new min is greater than previous min and the same thing is happen for fourth virtual machine $(4-1)=3$.

Hence minimum value comes from first virtual machine so job1 is allocated to the first virtual machine.

Now update the table of VM.

Table 4.3: Updated table after execution of job1

Virtual machines	No of CPU
Vm1	0
Vm2	2
Vm3	3
Vm4	4

Now job 2 comes

Again the condition is to be checked that whether the no of cpu available in virtual machine is greater than zero.

Now here first virtual machine is rejected.

Now comes to second virtual machine it gives $(2-1)=1$ hence min is set to 1.

From third virtual machine it gives $(3-1) = 2$ hence it is greater than previous min so updation of min is not required. Similarly from fourth virtual machine it gives $(4-1) = 3$.

Finally job 2 is allocated to second virtual machine.

Now updated table is

Table 4.4: Updated table after execution of job2

Virtual machine	No of CPU
Vm1	0
Vm2	1
Vm3	3
Vm4	4

Now job 3 comes

Now we again the condition is to be checked that no of cpu available in virtual machine is greater than zero or not. Hence first virtual machine is rejected.

From second virtual machine $(1-2) < 0$.difference is less than zero therefore it is also rejected.

Now comes to the third virtual machine hence min is $(3-2)$ i.e.1 so set $\text{min}=1$.

From fourth virtual machine it gives $(4-2) = 2$.hence updation of min is not required and finally third virtual machine is selected for job 3.

Now updated table is

Table 4.5: Updated table after execution of job3

Virtual machine	No of CPU
Vm1	0
Vm2	1
Vm3	1
Vm4	4

Now job 4 comes similar procedure applies as we applied for previous jobs job 4 is allocated

to third virtual machine. Now updated table is

Table 4.6: Updated table after execution of job4

Virtual machine	No of CPU
Vm1	0
Vm2	1
Vm3	0
Vm4	4

Now job5 comes which is allocated to fourth virtual machine. Now updated table is

Table 4.7: Updated table after execution of job5

Virtual machine	No of CPU
Vm1	0
Vm2	1
Vm3	0
Vm4	0

Now job6 comes which is allocated to second virtual machine. Finally all jobs are executed.

Table 4.8: Updated table after execution of job6

Virtual machine	No of CPU
Vm1	0
Vm2	0
Vm3	0
Vm4	0

it has two advantages first is that all jobs are executed properly and second is that utilization of resources is maximum.

Flowchart of execution of one job is given below similarly other jobs are executed.

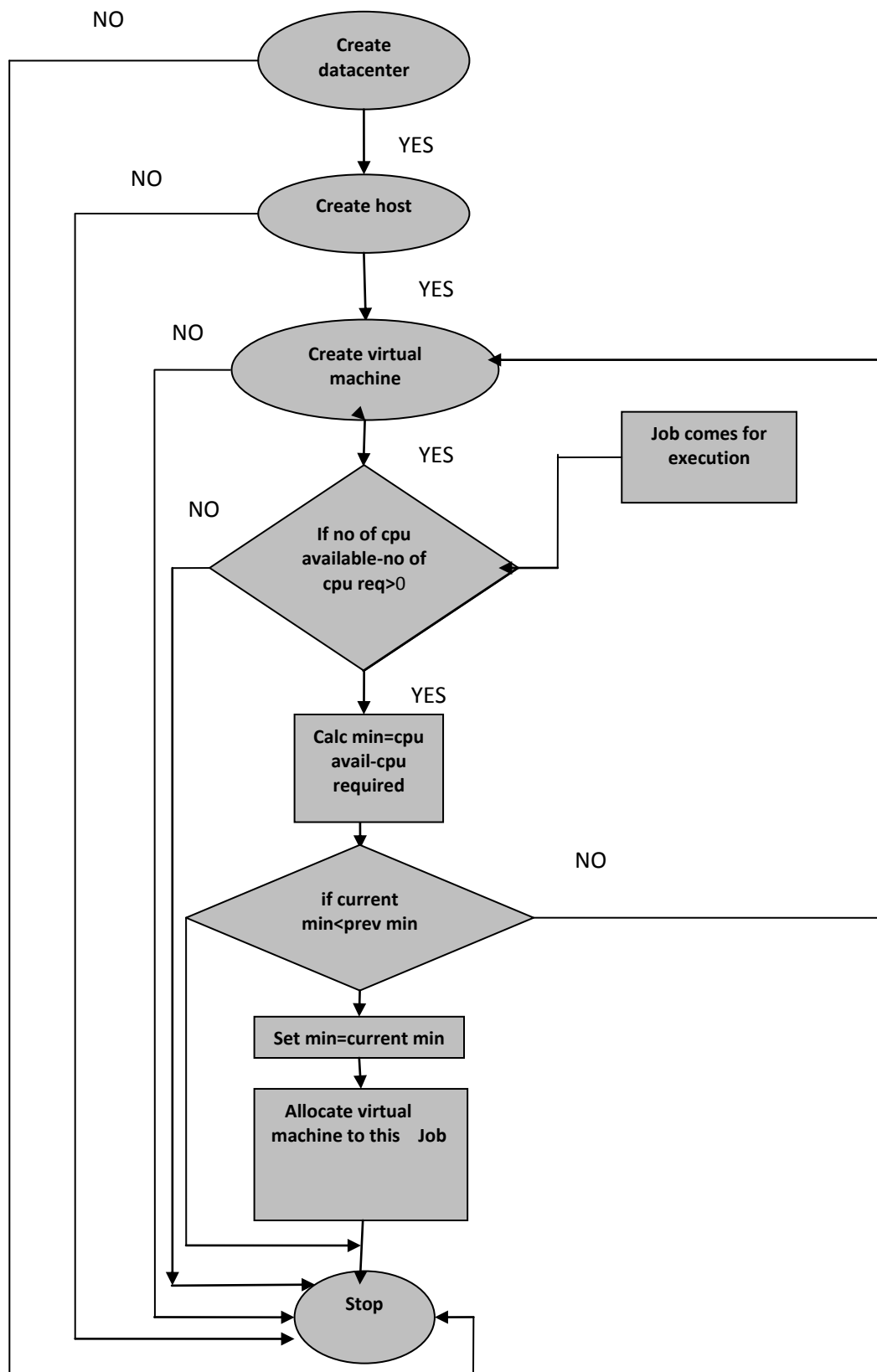


Figure 4.1: Flowchart of proposed algorithm

CHAPTER 5

EXPERIMENTAL RESULTS

This scheduling algorithm is implemented in Cloudsim. The Cloudsim is a new, generalized and extensible simulation toolkit[20] that is basically designed to simulate cloud computing environment. A class is created for implementing our policy. we used spaceshared environment in virtual machine and both timeshared and spaceshared environment in cloudlets.

Here we discuss certain scenarios for explaining our results

Table that represent number of cpu available in virtual machines.

Table 5.1: Number of CPU available in virtual machine

Vm	No of cpu
vm1	1
vm2	2
vm3	3

Case 1:

INPUT

Table that represent number of cpu required by jobs

Table 5.2: Number of CPU required by jobs in case1

Jobs	No of cpu
Job1	1
Job2	1
Job3	2
Job4	1
Job5	2
Job6	2

Output after running the program in cloudsim

PROGRAM STARTED

Initialising...

NUMBER OF PES ALLOCATED TO FIRST VIRTUAL MACHINE = 1.0

NUMBER OF PES ALLOCATED TO SECOND VIRTUAL MACHINE = 2.0

NUMBER OF PES ALLOCATED TO THIRD VIRTUAL MACHINE = 3.0

HOW MANY CLOUDLET YOU WANT TO RUN

6

ENTER THE NUMBER OF PES REQUIRED BY 1 cloudlet

1
ENTER THE NUMBER OF PES REQUIRED BY 2 cloudlet
1
ENTER THE NUMBER OF PES REQUIRED BY 3 cloudlet
2
ENTER THE NUMBER OF PES REQUIRED BY 4 cloudlet
1
ENTER THE NUMBER OF PES REQUIRED BY 5 cloudlet
2
ENTER THE NUMBER OF PES REQUIRED BY 6 cloudlet
2
FIRST VIRTUAL MACHINE SELECTED FOR 1 CLOUDLET
REMAINING NO OF PES IN FIRST VIRTUAL MACHINE 0.0
SECOND VIRTUAL MACHINE SELECTED FOR 2 CLOUDLET
REMAINING NO OF PES IN SECOND VIRTUAL MACHINE 1.0
THIRD VIRTUAL MACHINE SELECTED FOR 3 CLOUDLET
REMAINING NO OF PES IN SECOND VIRTUAL MACHINE 1.0
SECOND VIRTUAL MACHINE SELECTED FOR 4 CLOUDLET
REMAINING NO OF PES IN SECOND VIRTUAL MACHINE 0.0

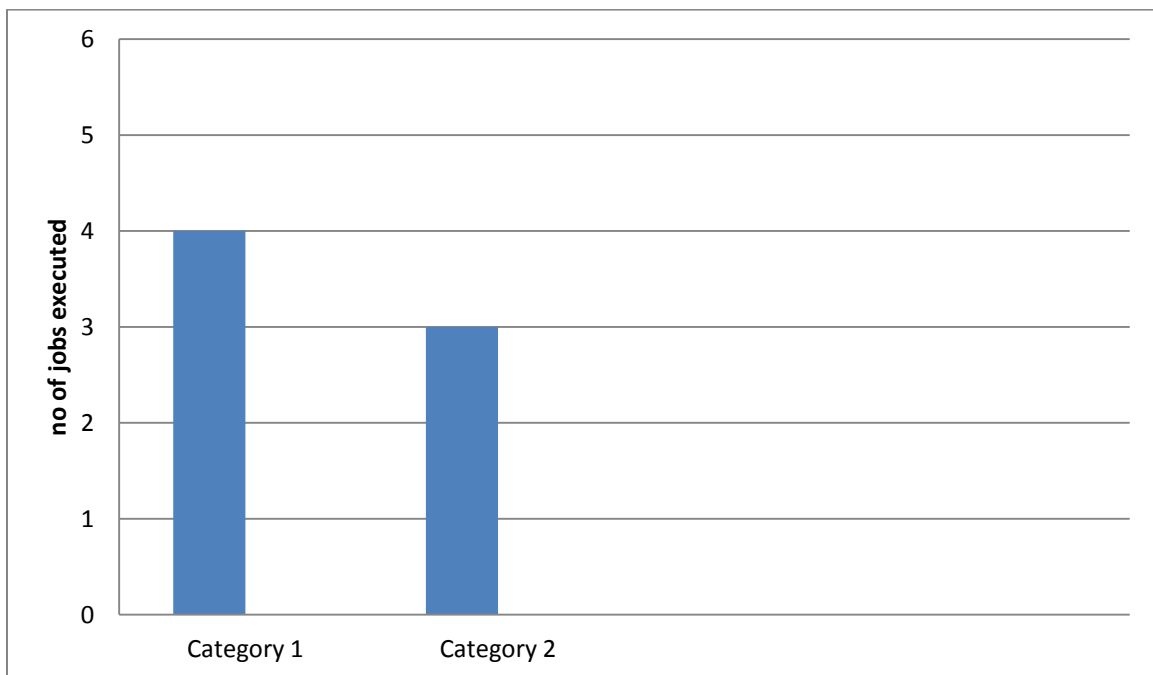
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
0.1: Broker: Sending cloudlet 2 to VM #2
0.1: Broker: Sending cloudlet 3 to VM #1
0.1: Broker: Sending cloudlet 4 to VM #1
0.1: Broker: Sending cloudlet 5 to VM #2
2500.1: Broker: Cloudlet 0 received
2500.1: Broker: Cloudlet 1 received
2500.1: Broker: Cloudlet 3 received
2500.1: Broker: Cloudlet 2 received
5000.1: Broker: Cloudlet 4 received
5000.1: Broker: Cloudlet 5 received
5000.1: Broker: All Cloudlets executed. Finishing...
5000.1: Broker: Destroying VM #0
5000.1: Broker: Destroying VM #1

5000.1: Broker: Destroying VM #2
 Broker is shutting down...
 Simulation: No more future events
 CloudInformationService: Notify all CloudSim entities for shutting down.
 Datacenter_0 is shutting down...
 Broker is shutting down...
 Simulation completed.
 Simulation completed.
 ONLY4 CLOUDLET ARE ABLE TO RUN DUE TO LESS NUMBER OF VM

===== OUTPUT =====

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	2500	0.1	2500.1
1	SUCCESS	2	1	2500	0.1	2500.1
3	SUCCESS	2	1	2500	0.1	2500.1
2	SUCCESS	2	2	2500	0.1	2500.1

PROGRAM FINISHED!



Category 1 represents no of jobs executes applying our algorithm.

Category 2 represents no of jobs executes by sequentially allocation of jobs to virtual machine.

Case 2:**Table 5.3: Number of CPU required by jobs in case2**

Jobs	No of cpu
Job1	2
Job2	2
Job3	2
Job4	1
Job5	2
Job6	2

PROGRAM STARTED

Initialising...

NUMBER OF CPU ALLOCATED TO FIRST VIRTUAL MACHINE = 1.0

NUMBER OF CPU ALLOCATED TO SECOND VIRTUAL MACHINE = 2.0

NUMBER OF CPU ALLOCATED TO THIRD VIRTUAL MACHINE = 3.0

HOW MANY CLOUDLET YOU WANT TO RUN

6

ENTER THE NUMBER OF CPU REQUIRED BY 1 cloudlet

2

ENTER THE NUMBER OF CPU REQUIRED BY 2 cloudlet

2

ENTER THE NUMBER OF CPU REQUIRED BY 3 cloudlet

2

ENTER THE NUMBER OF CPU REQUIRED BY 4 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 5 cloudlet

2

ENTER THE NUMBER OF CPU REQUIRED BY 6 cloudlet

2

SECOND VIRTUAL MACHINE SELECTED FOR 01 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 0.0

THIRD VIRTUAL MACHINE SELECTED FOR 11 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 1.0

FIRST VIRTUAL MACHINE SELECTED FOR 31 CLOUDLET

REMAINING NO OF CPU IN FIRST VIRTUAL MACHINE 0.0

Starting CloudSim version 3.0

Datacenter_0 is starting...

Broker is starting...

Entities started.

```

0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #1
0.1: Broker: Sending cloudlet 1 to VM #2
0.1: Broker: Sending cloudlet 2 to VM #2
0.1: Broker: Sending cloudlet 3 to VM #0
0.1: Broker: Sending cloudlet 4 to VM #1
0.1: Broker: Sending cloudlet 5 to VM #2
2500.1: Broker: Cloudlet 3 received
2500.1: Broker: Cloudlet 0 received
2500.1: Broker: Cloudlet 1 received
5000.1: Broker: Cloudlet 4 received
5000.1: Broker: Cloudlet 2 received
7500.1: Broker: Cloudlet 5 received
7500.1: Broker: All Cloudlets executed. Finishing...
7500.1: Broker: Destroying VM #0
7500.1: Broker: Destroying VM #1
7500.1: Broker: Destroying VM #2
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
ONLY3 CLOUDLET ARE ABLE TO RUN DUE TO LESS NUMBER OF VM

```

```

===== OUTPUT =====

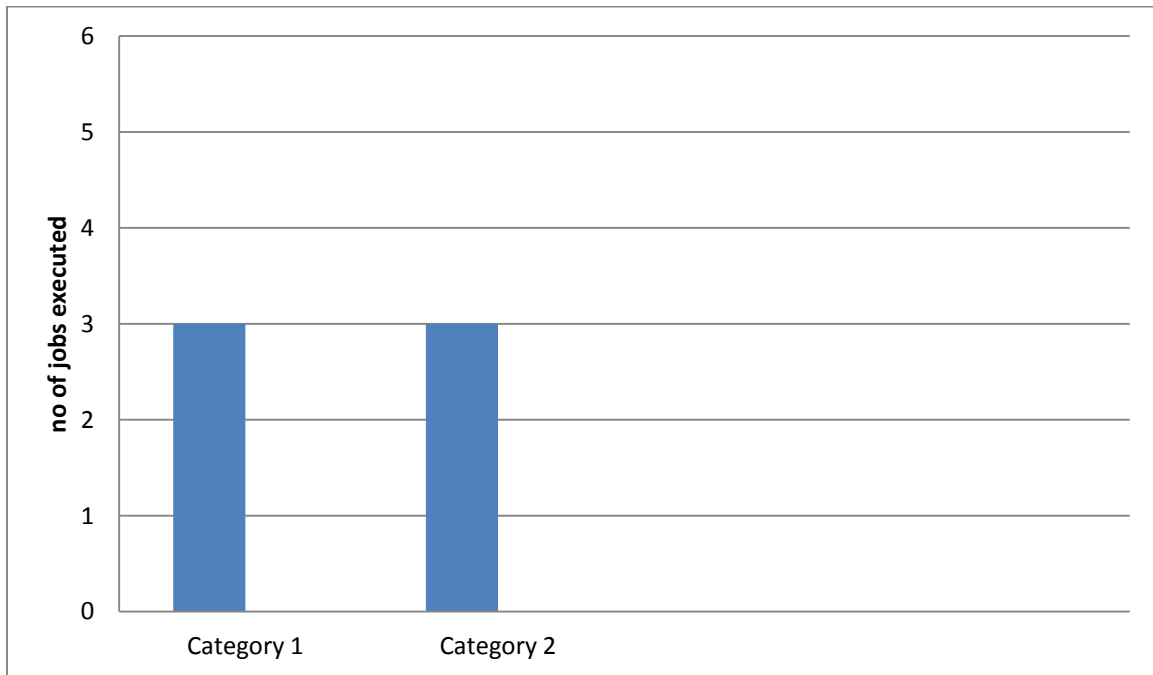
```

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
3	SUCCESS	2	0	2500	0.1	2500.1
0	SUCCESS	2	1	2500	0.1	2500.1
1	SUCCESS	2	2	2500	0.1	2500.1

```

PROGRAM FINISHED!

```

**Case 3:****Table 5.4: Number of CPU required by jobs in case3**

Jobs	No of cpu
Job1	1
Job2	1
Job3	1
Job4	1
Job5	1
Job6	1
Job7	1
Job8	1

PROGRAM STARTED

Initialising...

NUMBER OF CPU ALLOCATED TO FIRST VIRTUAL MACHINE = 1.0

NUMBER OF CPU ALLOCATED TO SECOND VIRTUAL MACHINE = 2.0

NUMBER OF CPU ALLOCATED TO THIRD VIRTUAL MACHINE = 3.0

HOW MANY CLOUDLET YOU WANT TO RUN

8

ENTER THE NUMBER OF CPU REQUIRED BY 1 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 2 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 3 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 4 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 5 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 6 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 7 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 8 cloudlet

1

FIRST VIRTUAL MACHINE SELECTED FOR 01 CLOUDLET

REMAINING NO OF CPU IN FIRST VIRTUAL MACHINE 0.0

SECOND VIRTUAL MACHINE SELECTED FOR 11 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 1.0

SECOND VIRTUAL MACHINE SELECTED FOR 21 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 0.0

THIRD VIRTUAL MACHINE SELECTED FOR 31 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 2.0

THIRD VIRTUAL MACHINE SELECTED FOR 41 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 1.0

THIRD VIRTUAL MACHINE SELECTED FOR 51 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 0.0

Starting CloudSim version 3.0

Datacenter_0 is starting...

Broker is starting...

Entities started.

0.0: Broker: Cloud Resource List received with 1 resource(s)

0.0: Broker: Trying to Create VM #0 in Datacenter_0

0.0: Broker: Trying to Create VM #1 in Datacenter_0

0.0: Broker: Trying to Create VM #2 in Datacenter_0

0.1: Broker: VM #0 has been created in Datacenter #2, Host #0

0.1: Broker: VM #1 has been created in Datacenter #2, Host #0

0.1: Broker: VM #2 has been created in Datacenter #2, Host #0

0.1: Broker: Sending cloudlet 0 to VM #0

0.1: Broker: Sending cloudlet 1 to VM #1

```

0.1: Broker: Sending cloudlet 2 to VM #1
0.1: Broker: Sending cloudlet 3 to VM #2
0.1: Broker: Sending cloudlet 4 to VM #2
0.1: Broker: Sending cloudlet 5 to VM #2
0.1: Broker: Sending cloudlet 6 to VM #0
0.1: Broker: Sending cloudlet 7 to VM #1
2500.1: Broker: Cloudlet 0 received
2500.1: Broker: Cloudlet 1 received
2500.1: Broker: Cloudlet 2 received
2500.1: Broker: Cloudlet 3 received
2500.1: Broker: Cloudlet 4 received
2500.1: Broker: Cloudlet 5 received
5000.1: Broker: Cloudlet 6 received
5000.1: Broker: Cloudlet 7 received
5000.1: Broker: All Cloudlets executed. Finishing...
5000.1: Broker: Destroying VM #0
5000.1: Broker: Destroying VM #1
5000.1: Broker: Destroying VM #2
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
ONLY6 CLOUDLET ARE ABLE TO RUN DUE TO LESS NUMBER OF VM

```

```

===== OUTPUT =====

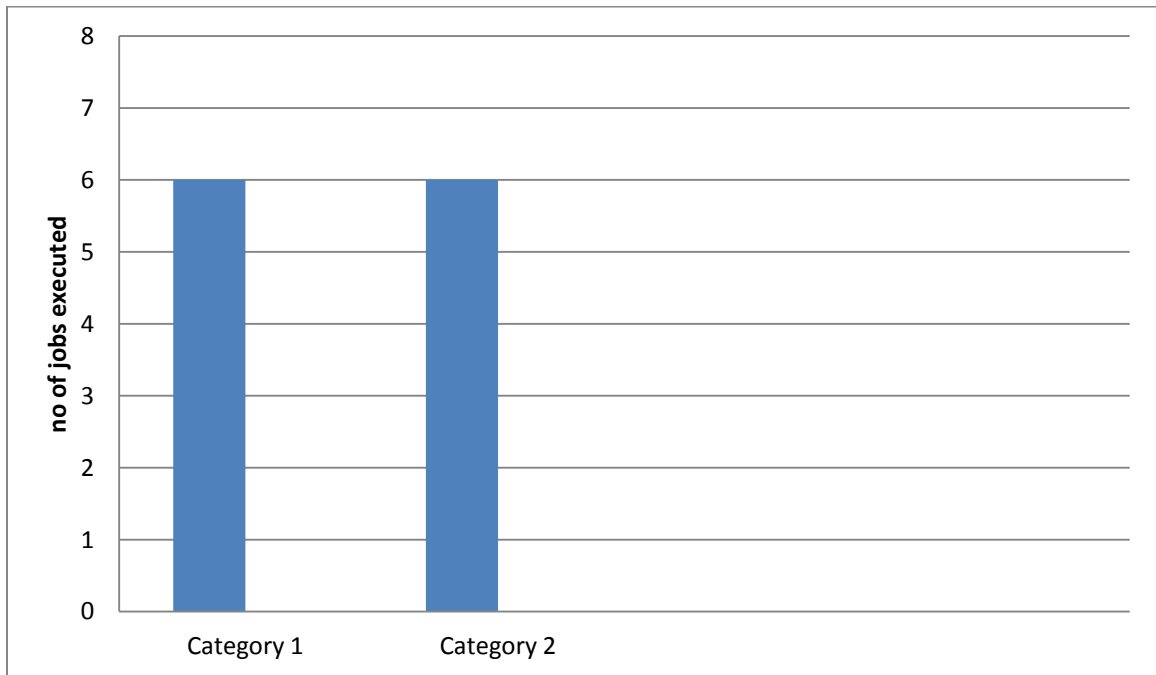
```

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	2500	0.1	2500.1
1	SUCCESS	2	1	2500	0.1	2500.1
2	SUCCESS	2	1	2500	0.1	2500.1
3	SUCCESS	2	2	2500	0.1	2500.1
4	SUCCESS	2	2	2500	0.1	2500.1
5	SUCCESS	2	2	2500	0.1	2500.1

```

PROGRAM FINISHED!

```

**Case 4:****Table 5.5: Number of CPU required by jobs in case 4**

Jobs	No of cpu
Job1	1
Job2	1
Job3	1
Job4	3
Job5	4
Job6	1
Job7	1

PROGRAM STARTED

Initialising...

NUMBER OF CPU ALLOCATED TO FIRST VIRTUAL MACHINE = 1.0

NUMBER OF CPU ALLOCATED TO SECOND VIRTUAL MACHINE = 2.0

NUMBER OF CPU ALLOCATED TO THIRD VIRTUAL MACHINE = 3.0

HOW MANY CLOUDLET YOU WANT TO RUN

7

```
ENTER THE NUMBER OF CPU REQUIRED BY 1 cloudlet
1
ENTER THE NUMBER OF CPU REQUIRED BY 2 cloudlet
1
ENTER THE NUMBER OF CPU REQUIRED BY 3 cloudlet
1
ENTER THE NUMBER OF CPU REQUIRED BY 4 cloudlet
3
ENTER THE NUMBER OF CPU REQUIRED BY 5 cloudlet
4
ENTER THE NUMBER OF CPU REQUIRED BY 6 cloudlet
1
ENTER THE NUMBER OF CPU REQUIRED BY 7 cloudlet
1
FIRST VIRTUAL MACHINE SELECTED FOR 01 CLOUDLET
REMAINING NO OF CPU IN FIRST VIRTUAL MACHINE 0.0
SECOND VIRTUAL MACHINE SELECTED FOR 11 CLOUDLET
REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 1.0
SECOND VIRTUAL MACHINE SELECTED FOR 21 CLOUDLET
REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 0.0
THIRD VIRTUAL MACHINE SELECTED FOR 31 CLOUDLET
REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 0.0
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
0.1: Broker: Sending cloudlet 2 to VM #1
0.1: Broker: Sending cloudlet 3 to VM #2
0.1: Broker: Sending cloudlet 4 to VM #1
0.1: Broker: Sending cloudlet 5 to VM #2
0.1: Broker: Sending cloudlet 6 to VM #0
2500.1: Broker: Cloudlet 0 received
2500.1: Broker: Cloudlet 1 received
2500.1: Broker: Cloudlet 2 received
2500.1: Broker: Cloudlet 3 received
5000.1: Broker: Cloudlet 6 received
5000.1: Broker: Cloudlet 5 received
```

Simulation: No more future events

CloudInformationService: Notify all CloudSim entities for shutting down.

Datacenter_0 is shutting down...

Broker is shutting down...

Simulation completed.

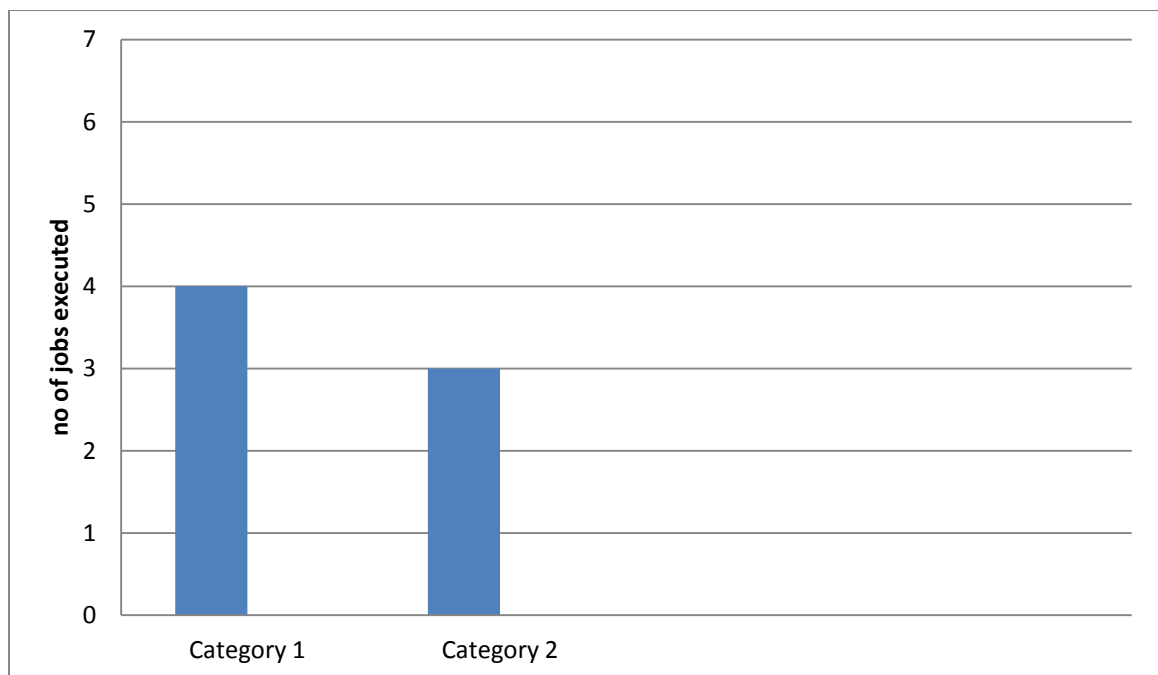
Simulation completed.

ONLY4 CLOUDLET ARE ABLE TO RUN DUE TO LESS NUMBER OF VM

===== OUTPUT =====

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	2500	0.1	2500.1
1	SUCCESS	2	1	2500	0.1	2500.1
2	SUCCESS	2	1	2500	0.1	2500.1
3	SUCCESS	2	2	2500	0.1	2500.1

PROGRAM FINISHED!



Case 5:

Table 5.6: Number of CPU required by jobs in case5

Jobs	No of cpu
Job1	1
Job2	1
Job3	1

Job4	3
------	---

PROGRAM STARTED

Initialising...

NUMBER OF CPU ALLOCATED TO FIRST VIRTUAL MACHINE = 1.0

NUMBER OF CPU ALLOCATED TO SECOND VIRTUAL MACHINE = 2.0

NUMBER OF CPU ALLOCATED TO THIRD VIRTUAL MACHINE = 3.0

HOW MANY CLOUDLET YOU WANT TO RUN

4

ENTER THE NUMBER OF CPU REQUIRED BY 1 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 2 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 3 cloudlet

1

ENTER THE NUMBER OF CPU REQUIRED BY 4 cloudlet

3

FIRST VIRTUAL MACHINE SELECTED FOR 01 CLOUDLET

REMAINING NO OF CPU IN FIRST VIRTUAL MACHINE 0.0

SECOND VIRTUAL MACHINE SELECTED FOR 11 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 1.0

SECOND VIRTUAL MACHINE SELECTED FOR 21 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 0.0

THIRD VIRTUAL MACHINE SELECTED FOR 31 CLOUDLET

REMAINING NO OF CPU IN SECOND VIRTUAL MACHINE 0.0

Starting CloudSim version 3.0

Datacenter_0 is starting...

Broker is starting...

Entities started.

0.0: Broker: Cloud Resource List received with 1 resource(s)

0.0: Broker: Trying to Create VM #0 in Datacenter_0

0.0: Broker: Trying to Create VM #1 in Datacenter_0

0.0: Broker: Trying to Create VM #2 in Datacenter_0

0.1: Broker: VM #0 has been created in Datacenter #2, Host #0

0.1: Broker: VM #1 has been created in Datacenter #2, Host #0

0.1: Broker: VM #2 has been created in Datacenter #2, Host #0

0.1: Broker: Sending cloudlet 0 to VM #0

0.1: Broker: Sending cloudlet 1 to VM #1

0.1: Broker: Sending cloudlet 2 to VM #1

0.1: Broker: Sending cloudlet 3 to VM #2

2500.1: Broker: Cloudlet 0 received

2500.1: Broker: Cloudlet 1 received
2500.1: Broker: Cloudlet 2 received
2500.1: Broker: Cloudlet 3 received
2500.1: Broker: All Cloudlets executed. Finishing...
2500.1: Broker: Destroying VM #0
2500.1: Broker: Destroying VM #1
2500.1: Broker: Destroying VM #2
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
ALL CLOUDLET ARE EXECUTED

===== OUTPUT =====

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	2500	0.1	2500.1
1	SUCCESS	2	1	2500	0.1	2500.1
2	SUCCESS	2	1	2500	0.1	2500.1
3	SUCCESS	2	2	2500	0.1	2500.1

PROGRAM FINISHED!

CHAPTER 6

CONCLUSION AND FUTURE

WORK

6.1 CONCLUSION

Scheduling in cloud computing plays a very important role for establishing the cloud computing environment. The main aim of our scheduling algorithm is to manage the maximum number of jobs in very efficient manner. Reliability, scalability, throughput and timespan etc are the certain parameters that should be keep in mind while developing scheduling algorithm. With the help of this algorithm we can increase the resource utilization by executing more number of jobs and also efficiency of this algorithm does not vary if the number of resources increases. Also the time required for the computation is very less it just require the basic calculation which reduces the complexity of computation. In our algorithm, job is submitted to that virtual machine which is best fit for this job even though job has several options of running on other virtual machines. [3] Scheduling is one of the major issues in the management of execution of application in cloud environment.

In our algorithm we have taken only CPU as a main resource but anyone can also take other resource may be main resources. Number of host and virtual machine is fixed in our implementation but the number of jobs and their requirements of cpu varies according to the user. We can either make all jobs to be executed according to the timeshared policy or according to the spaceshared policy or may be the combination of both. In case of virtual machine we used spaceshared policy in our implementation but we can also used timeshared or combination of both at this level also. Cloudsim provides classes of all policies if we want to implement our own policy then we have to use concept of inheritance in java. We have implemented our own policy by inheriting simple policy class. If we want to take a group of resources in our selection criteria then that will can also happen but it requires more calculation. This idea of taking more than one resource in checking is explained in my future work. Job scheduling is a combinatorial optimization problem in the fields of computer science where the ideal jobs are assigned to required resource at a particular instant of time. Makespan or completion time is the total time taken to process a set of jobs for its complete execution.

6.2 FUTURE WORK

In this algorithm, only number of processing elements in a virtual machine is used as checking criteria but we can also add other parameters such as main memory size, no of softwares, hard disk etc. Checking condition can also be changed. Now our condition is to be applied on all the resources not only on single resources. For that we have to create matrix that represents number of resource availability of virtual machine. Then find sum of the differences of available resources by virtual machine to the required resources by job.

.

CHAPTER 7

REFERENCES

-
- [1] Pinal Salot M.E, Computer Engineering, Alpha College of Engineering, Gujarat, India (2013) "A survey of various scheduling algorithm in cloud computing environment".
- [2] Poonam Devi Department of CSE BITS College, BHIWANI, MDU, India (2013) BITS College."Implementation of Cloud Computing By Using Short Job Scheduling".
- [3] Sujit Tilak, 2Department of COMPUTER, PIIT, New Panvel, Maharashtra, India (2012). "A Survey of Various Scheduling Algorithms in Cloud Environment".
- [4] Yogita Chawla 1,2Pune University, G.H Rasoni College of Engg & Mgmt, Gate No.: 1200 Wagholi, Pune – 412207(2012) "A Study on Scheduling Methods in Cloud Computing".
- [5] Jairam Naik K. Research Scholar, Faculty of Computer Science & Engineering, JNT University, Hyderabad, India (2012) "Scheduling Tasks on Most Suitable Fault tolerant Resource for Execution in Computational Grid".
- [6] Swachil Patel (2013) "Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review".
- [7] Sonam rathore "efficient allocation of virtual machine in cloud computing environment".
- [8] Arash Ghorbannia Delavar, Mahdi Javanmard , Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi "RSDC (Reliable Scheduling Distributed in Cloud Computing)" in International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3, June 2012
- [9] Saeed Parsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009. Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.
- [10] [10] Dr. M. Dakshayini, Dr. H. S. Guruprasad "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment" International Journal of Computer Applications (0975 – 8887) Volume 32– No.9, October 2011

- [11] Shamsollah Ghanbari, Mohamed Othman “A Priority based Job Scheduling Algorithm in Cloud Computing” International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012).
- [12] El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy “Extended Max-Min Scheduling Using Petri Net and Load Balancing” International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012
- [13] Shalmali Ambike, Dipti Bhansali, Jae Kshirsagar, Juhi Bansiwala “ An Optimistic Differentiated Job Scheduling System for Cloud Computing” International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2, Mar-Apr 2012, pp.1212-1214
- [14] Mrs.S.Selvarani¹; Dr.G.Sudha Sadhasivam, “improved cost-based algorithm for task scheduling in Cloud computing”, IEEE 2010.
- [15] Swachil Patel, Upendra Bhoi “Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review” International Journal of scientific & Technology research volume 2, issue 11, November 2013 ISSN 2277-8616.
- [16] Jairam Naik K” Scheduling Tasks on Most Suitable Fault tolerant Resource for Execution in Computational Grid International Journal of Grid and Distributed Computing” Vol. 5, No. 3, September, 2012.
- [17] Salim Bitam Computer science department Mohamed Khider University of Biskra “Bees Life Algorithm for Job Scheduling in Cloud Computing”.
- [18] Tejinder Sharma, Vijay Kumar Banga “Efficient and Enhanced Algorithm in Cloud Computing”.
- [19] R.Raju “Minimizing the Makespan using Hybrid Algorithm for Cloud Computing”.
- [20] Rodrigo N. Calheiros¹, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms.”
- [21] AV.Karthick, “An Efficient Multi Queue Job Scheduling for Cloud Computing” 2014 World Congress on Computing and Communication

Technologies.

- [22] Yung-Ching Hsu , “Job Sequence Scheduling for Cloud Computing”
2011 International Conference on Cloud and Service Computing.
- [23] www.google.com