Arrival Based Deadline Aware Job Scheduling in Cloud

A Thesis

submitted in partial fulfillment of the requirements for the award of the degree of

Master of Engineering

in

Computer Science and Engineering Department

by

Swati Gupta

Reg No: 801532055

Under the supervision of

Dr. Rajesh Kumar

Thapar University

Patiala-147001, Punjab, India

July 2017

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "Arrival Based Deadline Aware Job Scheduling in Cloud", in partial fulfilment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Rajesh Kumar and refers other researchers work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Swati

(Swati Gupta)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Rajesh Kumar)

Professor

Computer Science and Engineering Department

Acknowledgement

First, I would like to acknowledge the kindness of omnipresent God to gave me strength and courage to overcome all the hurdles. I would like to express my deep gratitude to my supervisor **Dr. Rajesh Kumar** for his invaluable advice and encouragement at every step of my ME program. Without his unfailing support and belief in me, this thesis would not have been possible. His contribution to this thesis goes well beyond their role as an academic supervisor and includes constant support on a personal level without which this journey may never have been completed. And for this, I am truly grateful. He is a great mentor for my life as well.

I would like to express my gratitude to a PhD scholar **Mr. Ashok Khunger**. He was always there for any of my problem to be resolved and constantly motivated me to complete my work. I also thank my fellow ME colleague Ms. Simran Setia for her help in reading and revising the thesis. I am grately thankful to **Dr. Maninder Singh** Head, CSED and **Dr. Ashutosh Mishra**, PG Coordinator for making labs available with almost all facilities. I would be failing in gratitude if I do not express my acknowledgement to **Dr. S. S. Bhatia**, Dean of Academic Affairs for their constant support and motivation.

Finally, I would like to express my sincere and deep gratitude to my parents and family member for their love, encouragement, care and whole hearted cooperation in helping to complete the thesis.

Swati

Swati Gupta

Abstract

Cloud Computing has been gaining much attention in recent years due to the fact that many real world applications have become more complex and dynamic. Cloud Computing provides its users with facilities where a user need not to pay for the whole license of a software or to buy a huge storage, a user can only pay according to his demand.

Cloud Computing is the delivery of services over the Internet according to user's demands. It can basically be defined as anytime, anywhere, through any device accessing of various services through the Internet. Due to advances in Cloud Computing, the user base of cloud computing is increasing hugely and therefore there is need of scheduling algorithms in order to schedule all those resources provided by cloud server.

This study contains a brief review of a scheduling algorithms for the resources in cloud environment and discuss their pros and cons. To make the algorithms more applicable to real time scenario in order to schedule the resources according to user defined parameters and to make it more beneficial, a new variant of these algorithm is proposed which aims to assign jobs in clouds in order to finish more and more jobs within the deadline constraint provided by its users. Scheduling of resources according to deadline is an important measure as there may be jobs which needs to be done before time and have no effect after the deadline is over so an algorithm which resolve this need is to be developed in order to make cloud computing services more effective.

The objective is to propose the algorithm for deadline constrained jobs and also define its benefits over other algorithms. Further, it defines the different parameters that can be used to compare an algorithm and uses those parameters to compare the algorithms. The algorithm developed here is named as Arrival based Deadline First Job Scheduler(ADSF) and is compared to First Come First Serve Algorithm(FCFS) in terms of waiting time, the delayed number of jobs and the total delay time. The results show a significant improvement in the former as compared to latter.

Table of Contents

List of Figures

List of Tables

Table No. **Title** **Page No.**

List of Abbreviations

NIST	U.S National Institute of Standards and Technology
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
AWS	Amazon Web Services
CSP	Cloud Service Providers
ADSF	Arrival based Deadline First Scheduler
FCFS	First Come First Serve
RR	Round Robin
BMHA	Batch Mode Heuristic scheduling Algorithms
OMHA	Online Mode Heuristic scheduling Algorithms

Chapter 1

Introduction

Cloud is used as a metaphor to internet. In most client-server pictorial representation, one draws cloud to represent as Internet as can be seen in Figure 1.1 and thereby the name Cloud Computing. As organisations on a large scale feels actually very difficult to handle the data and storage so, it is very important for Cloud Computing to take over in a really good manner and also provide good facilities which it ought to feature.

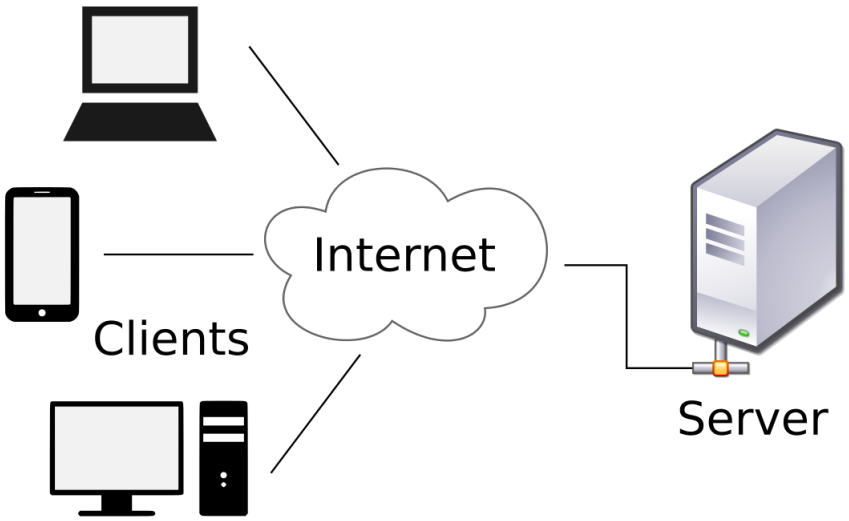

Figure 1.1: Client Server Architecture

1.1 Definition

According to U.S National Institute of Standards and Technology(NIST), Cloud Computing is a model for enabling easy to use, on-demand network access to a group of resources which can be shared and be fastly released with minimum requirement of human manager at the broker side.[1]. Cloud Computing could be seen as a technology and a platform that provides storage and hosting services to

its users over the internet. Also, the users need not to have the infrastructure, rather it could be accessed from anywhere through any device with the help of internet. So, this is device and location independent. Many companies are now moving to cloud by investing in their public cloud as in Amazon, Microsoft, IBM etc. Their cloud services are namely, AWS(Amazon Web Services), Microsoft Azure, Cloud IBM and IBM Bluemix, respectively. The architecture of the Cloud Computing has been well depicted by the Figure 1.2 as described by NIST [1]. Cloud Computing has the following 5 essential characteristics:

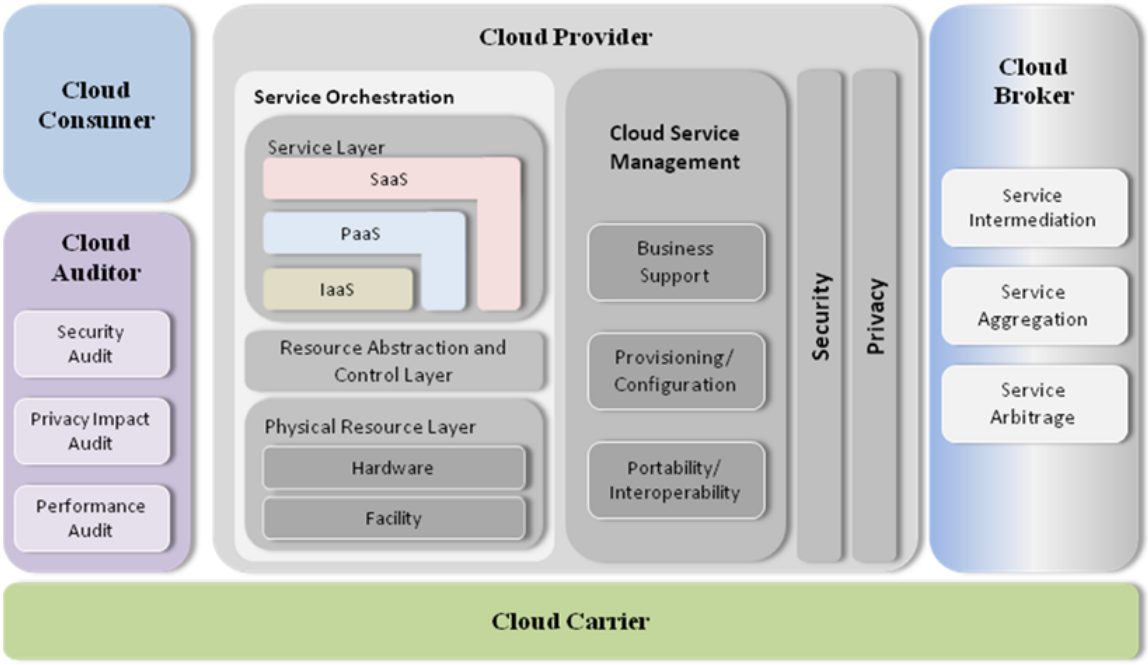

Figure 1.2: Cloud Computing Architecture [1]

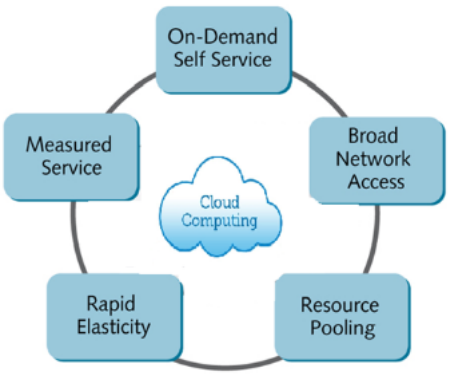

Figure 1.3: Essential Characteristics of Cloud Computing

- *On-demand self-service*: As the name suggest, this characteristic of Cloud Computing makes it a user demand service i.e. whenever a user wants to access something over the internet, he can do that by himself, without any human interaction.
- *Broader Network Access*: This means providing it ti a broader access i.e. on tablets, phones, laptops etc and with higher reachability.
- *Resource Pooling*: It should be a multi-tenant environment and with location independence.
- *Rapid Elasticity*: It should have the ability to grow or shrink according to the policy between the customer and the provider.
- *Measured Service*: A user has to only pay for the service he uses and not for the whole software, platform, storage or any other service which makes it a measured service.

1.2 Evolution of Cloud Computing

Cloud computing has emerged as a result of the evolution of some of the existing technologies. The era of cloud computing started way back in 1999 with the advent of Salesforce.com. But cloud computing gained popularity with Amazon Web Services in 2002. Amazon's Elastic Compute Cloud (EC2) provides secure and scalable cloud computing environment for masses. The integration of Elastic Compute Cloud with Simple Storage Service (S3) makes it even more promising. Simple Storage Service allows the user to store any amount of information and retrieve it at any point of time. A series of developments took place before this era of cloud computing began. Some of those developments are:

- *Mainframe Computing*- Multiple users were able to avail the Centralised computer through various terminals which ought to perform in order to make the availability of main frame computers to the whole organisation. But an organisation will not spend money for each of its employ to provide it with the computer that is where the main frame computing came into be-

ing that provided an economically stable piece of technology which allowed the employees to share the resources among themselves thereby helping the organisation economy and also fulfilling the needs of the employees for developing any new application.

- *Grid Computing*: Grid Computing is termed as a distributed computing in which various computer resources at different locations work together to achieve common goal. It is a special type of parallel computing comprising of network of loosely coupled clusters. Both cloud computing and grid computing are believed to be scalable. Grid computing is used to manage a number of short-term jobs while cloud computing is used to manage long term jobs. Cloud computing works on pay-per use basis while there's no such thing in grid computing.
- *Utility Computing*: Utility computing is metered services in which every customer is charged according to the specific usage of available computer resources. Cloud and Utility Computing work on pay-per use basis.
- *Autonomous Computing*: Autonomic Computing comprises of self-management characteristics of available networked computing resources. Self-management characteristics make it even more robust to unpredictable changes. Cloud computing uses autonomic systems in order to achieve robust organisation providing its customers economies of scale.
- *SaaS Computing*: SaaS computing provided the users with the subscription of applications with the help of internet.

The summary of evolution of Cloud Computing can be seen in Figure 1.4.

1.3 Service Models of Cloud

Cloud Computing has 3 different service models, (i) SaaS, (ii) PaaS and (iii) IaaS as shown in Figure 1.5

- *IaaS*: Infrastructure as a Service provides the user with infrastructure which includes hardware, software, storage and other infrastructure components.

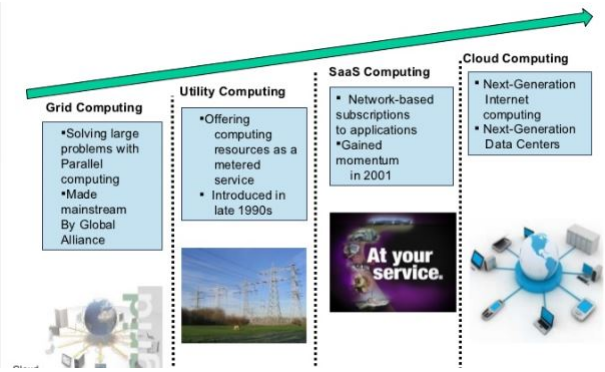

Figure 1.4: Evolution of Cloud Computing[2]

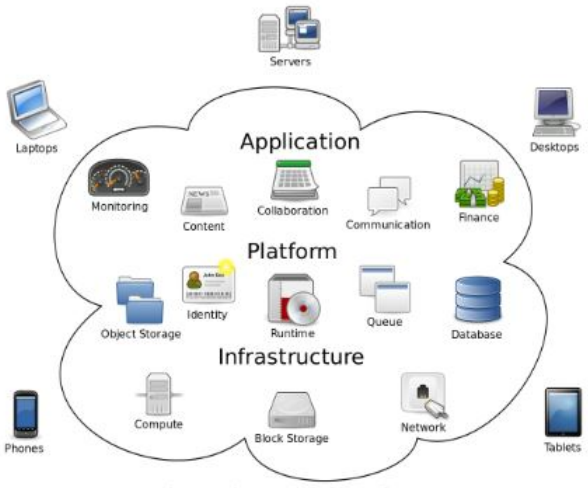

Figure 1.5: Service models of Cloud Computing

- *PaaS*: Platform as a Service provides the user with platform for helping customers to develop, applications without the difficulty of maintaining other facilities.
- *Saas*: Software as a Service provides software on demand to users on a subscription basis.

1.3.1 IaaS

Infrastructure as a Service(IaaS) delivers hardware that is managed by the broker itself. This works with all Cloud computing providers. It gives access to registering applications in a virtualized domain called "Cloud". With IaaS, in any case, the customer is offered access to virtualized infrastructure keeping in mind the major offering to handle their own stages. The registering of the user on an infrastructure

is particularly that of virtualized kind of use. The definition incorporates such providing as virtual server feature, organize association, transmitting of data, IP addressing and the feature that balance the load of data. Various advantages of IaaS are:

- *Scalability*: the services are offered after one sends the requirements, thus there is not any delay in increasing capability or there is no wastage of any capability that is unused.
- *No Hardware Capital Expenditure*: the hardware that supports IaaS is build up and managed by a cloud supplier, saving the user with his time and value
- *On Demand and Pay Per Use*: the facilities are provided on demand and user just pay cost for the resource that he uses.
- *Location Independence:* The cloud can be accessed from anywhere provided with internet facilities independent of the device or the place.
- *Security*: services provided there in a public cloud, or private clouds are encrypted by the providers to strengthen the security.
- *No reason for unavailability*: There are multiple sources at a particular cloud provider so even if one fails then also he may divert the request to another and can continue with the requests.

1.3.2 PaaS

Platform as a Service, also known as PaaS, is a layer of Cloud Computing that delivers a platform to allow the users to make programming applications through the internet. PaaS services are provided by the CSPs and accessed by users merely with the help of their application program. Software developers, web developers and businesses will all require PaaS early or lately for their own benefit.

Software developers will make the most of a PaaS resolution to make an application that they are going to provide. Web developers will use singly-dedicated PaaS facilities at each stage of the method to program, check and deploy their

websites. Businesses will develop their own organisational programs, notably to make distinct required development and testing applications. The working of a PaaS environment could be described as[7]:

- *Manufacturing of Package Services:* PaaS is often used by the programmers for creating of the applications and testing of it.
- *Pay Per Use*: In general, procured on a subscription basis, shoppers solely acquire what they are seeking for. Sharing of required services ends up in paying according to the usage.
- *Huge Number of Choices*: The users will opt for the options they need whereas not considering others. They will then opt for a facility to fulfill their wants.
- *Managing and Supporting*: Platforms , infrastructure and other services are managed for patrons and any assistance needed is also there.
- *Self Upgradation:* Features are automatically updated by adding on the extra features.

1.3.3 SaaS

Software as a service (or SaaS) is an approach of providing applications with the help of internet facilities. Rather than buying the complete software, user just use it over the internet, liberating themselves from managing complicated services and infrastructure.

SaaS applications are typically known as Web-based code, user demanded software, or deployed online software system. SaaS applications run on a SaaS provider's own servers. The supplier handles everytime there is a request for the launching of the software, together with security, availability and capability[8]. The characteristics of a SaaS environment could be summarised as follows:

- *Multitenant Design*: A multitenant outline, inside which all clients and applications share one, basic framework and code base that is midway kept up. Because of SaaS merchandiser customers are all on an equal foundation and

code base, merchants will start a considerable measure of rapidly and spare the improvement time prior spent on keeping up fluctuated adaptations of old code.

- *Easy Customisation:* The ability of the clients to simply avail the services that they want while not accessing any other service or applications, SaaS is used. This facility is purely based on the type of user one is i.e. if an organisation is availing the facility or an individual. These are continually backed up by the upgrades which in turn makes it less risky and highly proficient in any case.

- *User Friendly Availability:* Much better availability of information from any networked device thereby making it simple to manage authorisation and authentication, test information use, and assure that everyone sees equal data at any point of time irrespective of the type of user or of location.

- *SaaS Uses Client Internet*: Anyone accustomed to Amazon.com or My Yahoo! are accustomed to the internet structure of typical SaaS services. With the SaaS model, a user can built any application with the point-and-click ease, and thereby making the work of a user easy for the development of an application using a software service.

A brief summary of these models could be well understood by Figure 1.6

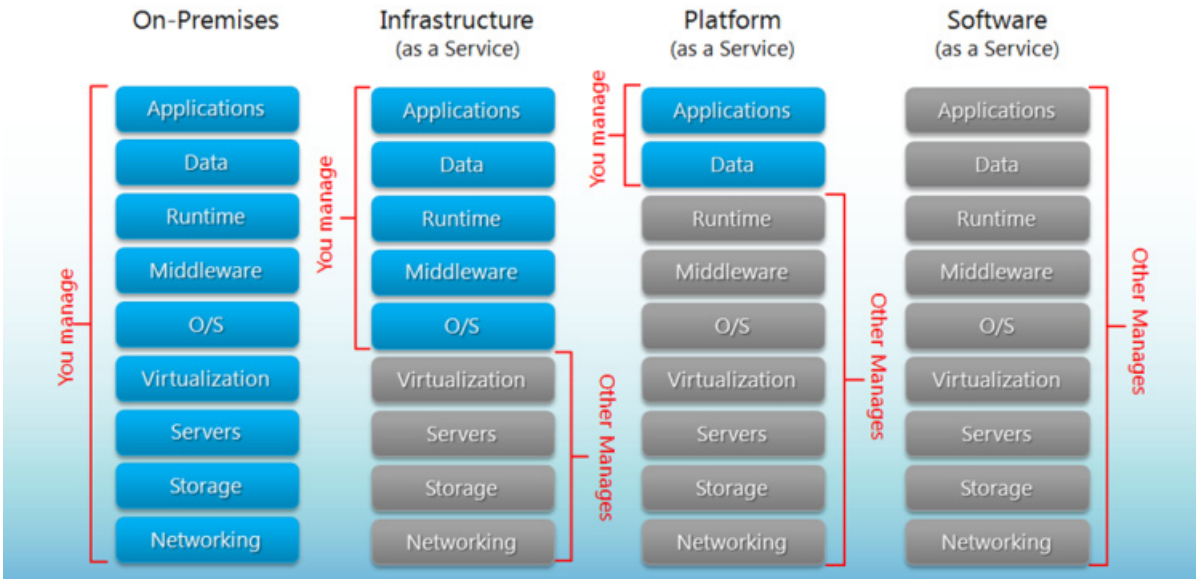

Figure 1.6: Summary of Service models

1.4 Cloud Deployment Models

There are basically four four cloud deployment models[9]:

- *Public cloud:* Public Cloud provides the services through the internet where general public can use it and that service can be free or may be pay per use basis depending upon the provider. A user can simply access the resources available online and pay accordingly which are available to all the public accessing the internet. Amazon Elastic-Compute-Cloud, IBM's Blue-Cloud, Sun Cloud, Google AppEngine are some of the examples of public cloud.
- *Private cloud:* A private cloud is that where only the owner of the cloud has the access to the subscribed resources and none other than the authorised personnel or organisation can access it. It is generally costlier as compared to public cloud but is the most securely available deployed cloud services provided by the CSPs such as Amazon, IBM, Microsoft.
- *Hybrid cloud:* Hybrid is generally a term used for the combination of more than one thing where in this case it is a collection of private and public cloud characteristics, or community or any other cloud.
- *Community cloud:* A community cloud is established in the scenario where two or more organisations have common basic requirements and are thereby agreeing to work together to access the cloud facilities. Although it limits the number of people accessing it to a particular number thereby increasing the cost of the cloud but provides with more security features making it more generic to use. An example of community cloud is Google's "Gov Cloud".

1.5 Virtualization

Virtualization is making of virtual (not real) things like hardware, software, platform, storage or a network device[10]. During a virtualized surroundings IT enterprise must handle alot of changes because the changes occur more quickly in virtual surroundings as compared to a physical surroundings. Due to virtualiza-

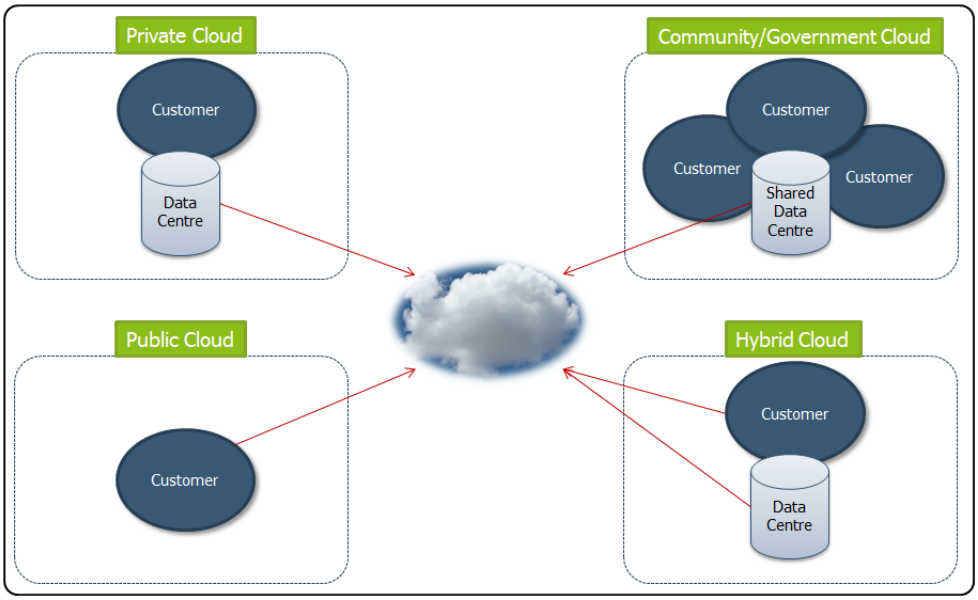

Figure 1.7: Deployment models of Cloud Computing

tion, cloud computing came into existence which is scalable too. Virtualization has the ability to share a single physical available type of a resource between many organizations. It is done by giving a logical name to the present storage and then enabling a pointer to that source whenever required.

Types of Virtualization:

- *Hardware Virtualization:* When the Virtual Machine Manager(VMM) is installed directly on machine without any further requirement of any other service then it is called Hardware Virtualization.
- *Operating System Virtualization*: Operating System Virtualization is when there are more than one OS running on a machine enabling the use of both the operating system on a machine with its features and services concurrently efficiently as shown in Figure 1.8.
- *Server Virtualization*: Server Virtualization creates a different environment for different OS environments which are logically different from the hosted server allowing a huge amount of resource usage (hardware, utilities, space) alongwith security and separability of various environments.
- *Storage Virtualization:* This enables different storage devices to be collectively represented as a single device as shown in Figure 1.9.

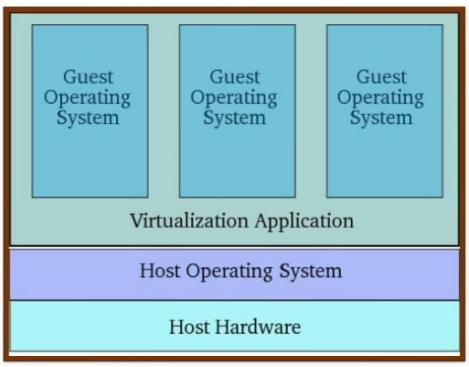

Figure 1.8: Operating system Virtualization[3]

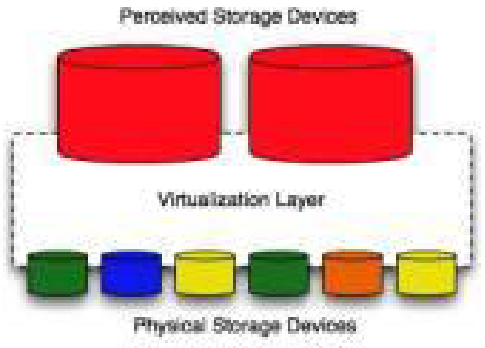

Figure 1.9: Storage Virtualization[3]

Hypervisor: A program that is required to provide the access of resources provided by a system to a virtual machine can be called as a Hypervisor or a Virtual Machine Manager(VMM). Following are the types of hypervisor:

- Native Hypervisor: Native(or Type 1) Hypervisors are present directly on the hardware platform and no further requirement of any hardware is there. These are good for a single user.

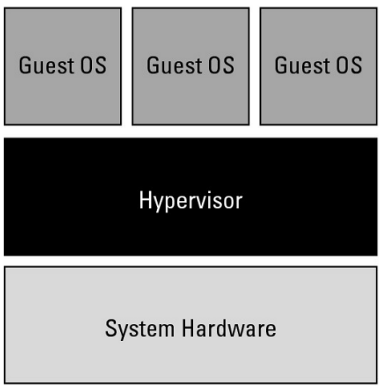

Figure 1.10: Native Hypervisor

- Embedded Hypervisor: These are integrated in a processor or an isolated chip and operates from there. Due to these hypervisors the network providers have better performance as compared to previous era.

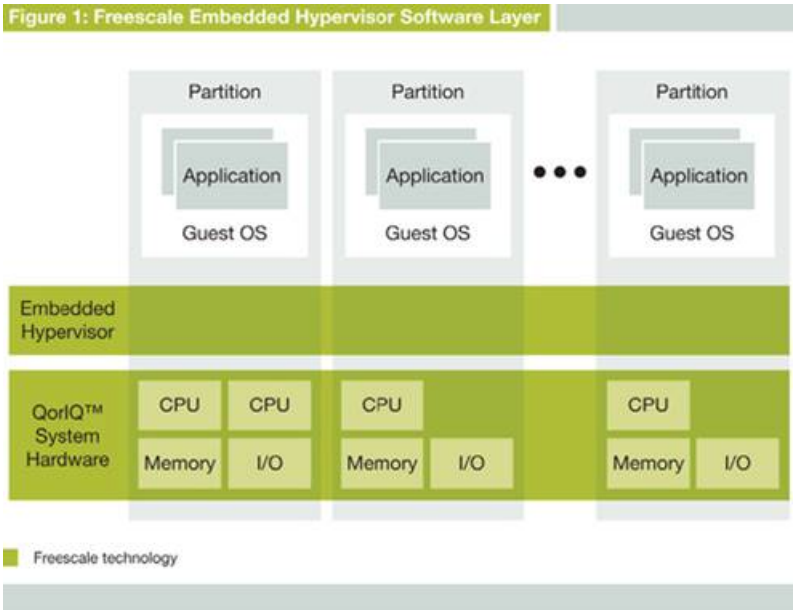

Figure 1.11: Embedded Hypervisor

- Hosted Hypervisor: Hosted(or Type 2) Hypervisors are those which present above both the hardware and the OS level in a machine and are very useful in areas of public and private cloud where performance enhancement is done by this VMM.

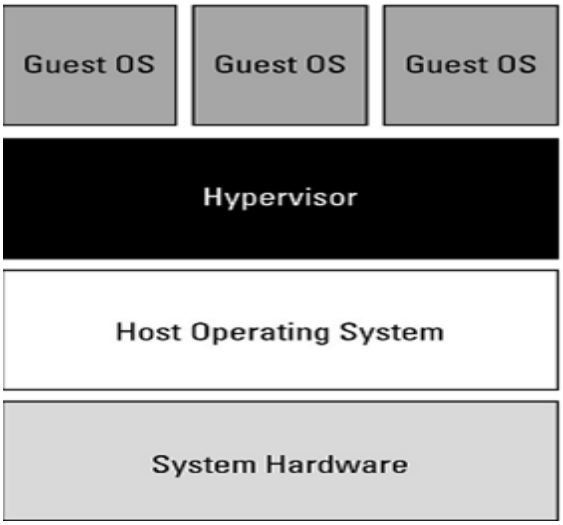

Figure 1.12: Hosted Hypervisor

1.6 CloudSim

CloudSim is a generalized, simulation environment to enable cloud computing environment with services like IaaS, PaaS and SaaS.[4]. Using CloudSim developers and programmers can check the performance of a newly programmed algorithm in a very easy to implement and use software and that too is available for free. The main benefits of using CloudSim for basic testing of the developed programs for cloud environment are:

- **Less time:** it needs very small amount of effort and time to implement Cloud-based application.
- **Flexibility and deployability:** Programmers can test the working of their code written in variety of Cloud provider environments with very little deployment effort.

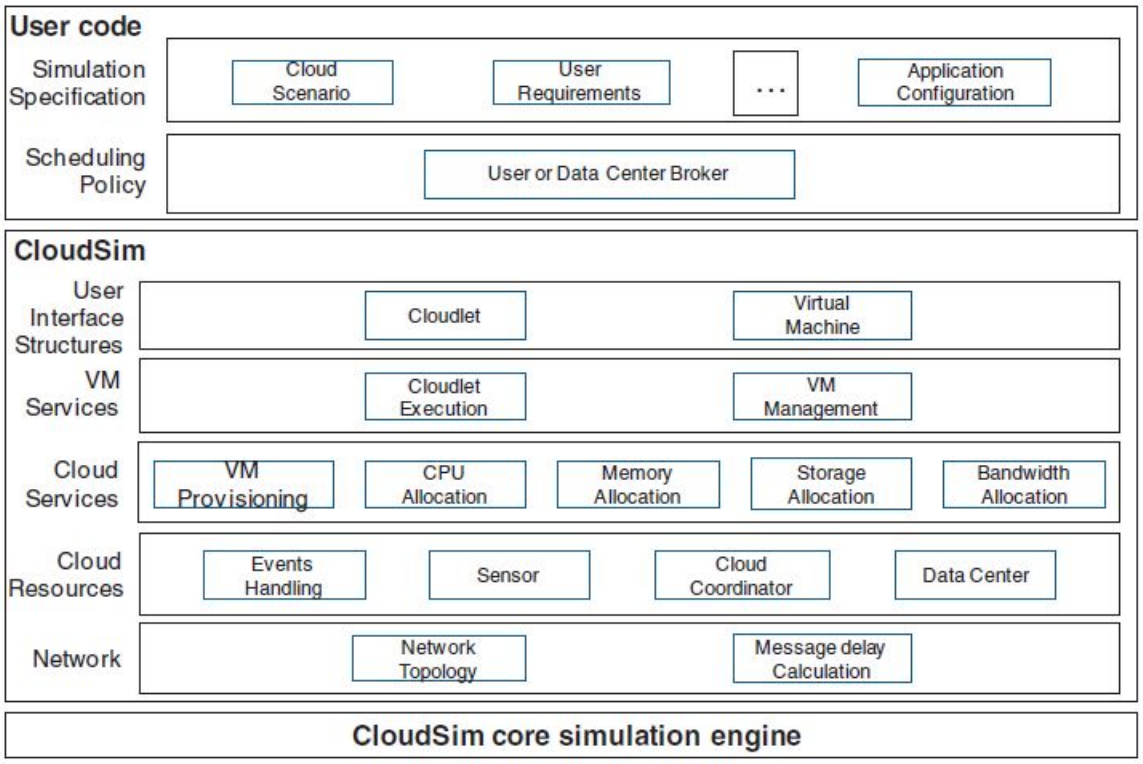

Figure 1.13: CloudSim Architecture[4]

1.6.1 CloudSim Architecture:

The architecture of CloudSim[4] could be well understood by the figure 1.13. The CloudSim layer provides support for modelling and simulation of cloud environments which has data centers, VMs, hosts, broker and cloudlets. It conjointly allocates hosts to VMs. A CSP will implement custom-made methods to review the potency of various policies in VM provisioning.

The user code layer uses specifications like the number of machines, their specifications, etc. in order to implement the algorithms. The main elements of the CloudSim framework are:

- *Regions:* These are basically the 6 continents in the world where people live and are called as regions from where a cloud provider could provide us with cloud facilities.
- *Data Centres:* This has the infrastructure services of a cloud that are provided by various CSPs. It has a collection of hosts that may be either same or different in nature according to their hardware configuration.
- *Data Centre Characteristics:* It is data relating to data centre resource configurations called as its characteristics.
- *Hosts:* It is actually the physical resource where a user is allocated a resource.
- *The user base:* It is the users that are sending there requests to the CSPs and a collection of all of them along with their details and requirements makes the user base.
- *Cloudlet:* It specifies the user requests and tasks. It has an ID, name of user base, along with input and output files. With the help of cloudlet, a user demand is presented to a broker with its specifications and that cloudlet is then assigned to the host. This has various specifications such as storage, length, etc.
- *Service Broker:* The broker is the main manager of all the requests that decides where will the user be allocated resource. It is the broker that

decides which process should be applied for allocation of resources.

- *VMM Allocation Policy:* It decides method for assigning VMs to hosts.
- *VM Scheduler:* It is a scheduler for scheduling various Virtual Machine over a data center.

1.6.2 CloudSim Implementation

The CloudSim Class Diagram[4] can be well understood by the Figure 1.14 The

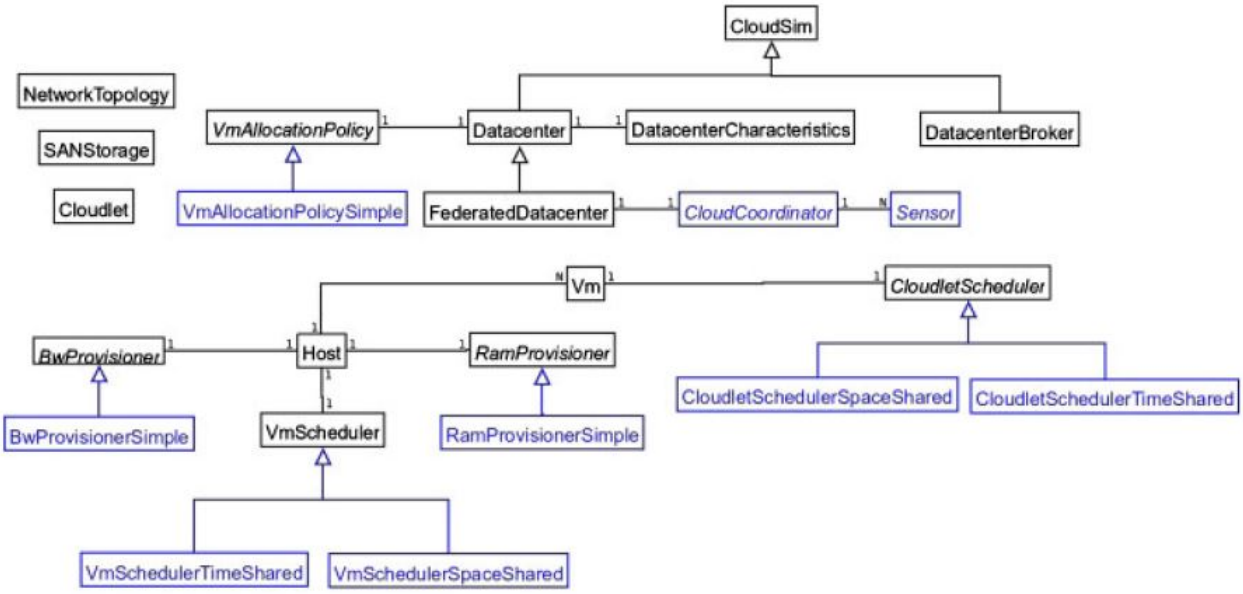

Figure 1.14: CloudSim Implementation and Class Diagram[4]

CloudSim has following classes:

- *DataCenter*: This class provides us with the basic hardware services that are delivered by CSPs. It has a collection of hosts that may be either same or different in nature according to hardware configuration. Every DataCenter has some policies for allocating various resources like bandwidth, memory, and storage devices to the users.
- *DataCenterBroker*: It is like a mediator between SaaS services and CSPs. The broker submits requests on behalf of the users and discover suitable Cloud service providers that can very well provide the user with its requirements. It is responsible for the QoS fulfilment of a demand, The program-

mers need to modify this class according to their required implementation in order to apply a broker policy in deciding the hosts and VMs. The significant difference between the broker and the CloudCoordinator is that the broker thinks from client point of view whereas the CloudCoordinator wants to benefit the data center. So the former works in favour of client whereas the latter tries to enhance the performance of the providers.

- *DataCenterCharacteristics*: It contains characteristics of data center.
- *Cloudlet*: Cloudlet is actually the jobs that needs to be executed in a CloudSim having the requirements of the user and also the specifications and that is made to be presented in a list and submitted to the broker for further scheduling and thereby a cloudlet is the most basic and foremost requirement of the cloud scheduling algorithms as the tasks defined there are nothing but cloudlets.
- *CloudletScheduler*: This is the policy used for scheduling the cloudlets which can be either Time shared or Space shared. Time shared means they will run in parallel and Space shared means they will share the space allocated to them.
- *Host*: This is a physical resource on which VM are allocated and further the cloudlets are sent in order to satusfy the user requirements. These cloudlets are having all specific characteristics such as dual core or quad core as in normal computers.
- *Network Topology*: It stores the topology data which enables a server to determine which particular strategy it should follow in this particular kind of assignment.
- *RAM Provisioner*: This is used for determining the policy for assigning RAM to the Virtual Machines. The allocation to the user is possible only if the RAM Provisioner approves that and the needs are sufficiently fulfilled in the best way. It has no limitation on the request but, if it is more than the available RAM, it is simply discarded and another request is entertained.

Chapter 2

Literature Review

Cloud Computing is useful in every scenario due to increasing demands of resources. A cloud application is managed by third party service provider. Cloud Applications provide a number of benefits to its users. Cloud Applications are believed to be scalable. Third party service provider can scale up or down the services according to user requirements. Cloud applications do not need a physical server by virtue of virtualization. User can download the application from anywhere. This means a cloud application does not consume any space in the physical harddrive. In a cloud, a datacenter can be shared by a number of cloud applications leading to low costs. keeping in mind the above stated benefits, cloud application has become a major breakthrough in this technological era. Job scheduling algorithms are necessary as they provide a provider with the facility of determining which resource should be allocated to a user. Many scheduling algorithms exists for cloud environment. Some of them are discussed in this section.

2.1 Categorisation of Algorithms

Job Scheduling Algorithms can be broadly classified into two categories:[11]:

- Batch Mode Heuristic Algorithms (BMHA)
- Online Mode Heuristic Algorithms (OMHA)

2.1.1 Batch Mode Heuristic Scheduling Algorithms

In BHMA, jobs when arrive are batched together and the scheduling starts after a period of time. Some of the algorithms under BHMA are[11]:

- *First Come First Serve:* In this, the jobs are submitted as soon as they arrive which means that the job which comes first is served first. This algorithm is similar to the Operating System Scheduling Algorithm and hence named the same. But this algorithm has drawback as no priority is set.
- *Round Robin*: In this, a time quantum is allocated and a job is run for that particular time in a first come first serve manner and then switched with other job when the time slice is over. The time quantum plays a very vital role. If it chosen as very large, it will behave like First Come First Serve and if it is chosen as very small, then there will be more number of switching.
- *Min-Min Algorithm*: In this, the concept of parallel computing is used and the shorter jobs are run in parallel and then the longer jobs are scheduled and run.
- *Max-Min Algorithm*: This is the opposite of previous algorithm in which the jobs that have larger request are assigned first and run which may lead to delay in shorter jobs and also lead to more delayed cloudlets.
- *Priority Scheduling Algorithm*: In this, all the jobs are assigned a priority and according to the particular priority these jobs are allocated which may in turn lead to larger waiting time for the jobs with lower priority.

2.1.2 Online Mode Heuristic Scheduling Algorithms

Most Fit Task Scheduling is one of the examples of OHMA.

- *Most Fit Task Scheduling Algorithm*: There are some parameters on the basis of which the jobs are fitted best in queue and then allocated according to the requirements it specifies.

2.2 Deadline and Budget Distribution Based Cost-Time Optimization Workflow Scheduling Algorithm

Deadline and Budget Distribution based Cost-Time Optimization Workflow Scheduling Algorithm [12] proposed by A. Verma and S. Kaushal scheduled job using Genetic algorithms. They divided the tasks into two categories: i) Synchronisation task and ii) Simple tasks.
The algorithm is:

Step 1. Discover accessible services and request QoS parameters of services for each task.
Step 2. a) Cluster the tasks into task partitions.
b) Estimates the minimum execution time and price for each task from the accessible set of services.
c) Calculate the finish time and price.
d) If values calculated in c) meet the deadline and budget provided by the client, then the process will be further executed otherwise not.
Step 3. a) The deadline and budget is partitioned over task partitions calculated in step 2(a).
b) Sort all available lists in descending order of their price.
Step 4. Opt for a resource to execute specific task from the list which leads to minimal processing cost and execution time.
Step 5. Repeat the step 4 till all tasks among all partitions have been scheduled.

2.3 Shortest Job Scheduling

Devi, Poonam and Trilok Gaba proposed Shortest Job Scheduling for the tasks[13]. The authors assigned a priority to tasks according to memory requirements. Resources are allocated according to priority assigned. This approach worked for

public as well as private clouds.

2.4 Priority based Job Scheduling Algorithm

In Priority based Job Scheduling Algorithm in Cloud Computing, proposed by Shamsollah Ghanbaria and Mohamed Othmana, priority is assigned according to AHP i.e. Analytical Hierarchy Process[14]. AHP is a Multi-Criteria Decision-Making (MCDM) and Multi-Attribute Decision-Making (MCDM) model. The architecture of AHP is made up of three levels: i)objective level, ii)attributes level and iii)alternatives level. In this, a vector of weights are calculated for comparing the priority. The resource is allocated after assigning a level to the job according to priority. This algorithm has more finish time than other algorithms.

2.5 Pre-emptable Shortest Job Next Scheduling

There is another algorithm entitled Pre-emptable Shortest Job Next Scheduling in private Cloud Computing in which the authors Sanghani, Khimani, Sutaria and Vasani combined the traditional round robin and shortest process next algorithm[5]. This algorithm is pre-emptable so the time to process a request is less as compared to other algorithms. The proposed architecture can be seen in Figure 2.1.

2.6 Location based Minimum Migration in Cloud (LBMMC)

Deadline Based Task Scheduling in Cloud with Effective Provisioning Cost using Location based Minimum Migration in Cloud (LBMMC) proposed by K. Sathya and S. Rajalakshmi, monitors all the virtual machines in all cloud locations centre[15]. It identifies the state of virtual machines i.e. it is in sleep, idle, or running state. It also checks number of tasks running and variety of tasks

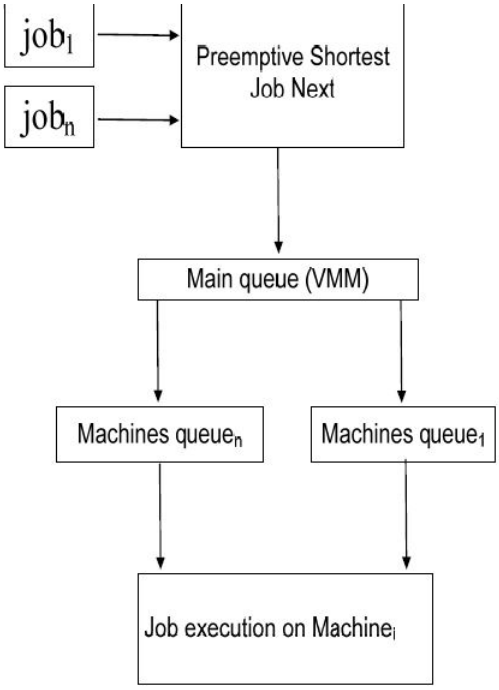

Figure 2.1: Architecture of PSJN[5]

it will be able to run. Once a virtual machine is found to be overloaded, migration to 1 or more VMs from the closest location cloud centers is done. If virtual machine is found to be under loaded, migration from 1 or more VMs (which are already running to perform some jobs) from the closest location cloud centers is done. Location based VM choice algorithm is applied to carry out the selection method.

2.7 Energy Efficient Scheduling for Virtualised Environments

Du, He and Meng proposed an algorithm named, Energy-Efficient Scheduling for Tasks with Deadline in Virtualized Environments[6] which can be seen in Figure 2.2. In this, there is an efficient utilisation of energy in virtualized environment. Here, 'r' denotes the resource which needs to be allocated, 's' denotes the speed of VM and 'M' denotes the makespan of a resource. The tasks are sorted in decreasing order in a waiting list. For each task, makespan is calculated and VM with least speed is selected for the job completion. If the completion time is more

```
{
    Sort all the tasks in the decreasing order of r_i in a waiting list
    Let i = 1
    while the waiting list is not null
        Compute M(i) for task i
        if M(i) is not null then
            Choose the VM j that has the minimum s_j from M(i)
            Assign task i to VM j
            Update the completion date of VM j
            if the completion date of VM j is bigger than the given threshold then
                Mark VM j as non reusable
            end if
        else
            return no solution
    end while
    return the generated schedule
}
```

Figure 2.2: Energy Efficient Scheduling for Virtualised Environments[6]

than given threshold, VM selected is marked non reusable.

2.8 Deadline Aware Resource Allocation for Cloud Computing Jobs (DCloud)

Dan Li, Congjie Chen, Junjie Guan, Ying Zhang, Jing Zhu and Ruozhou Yu proposed an algorithm named Deadline-Aware Resource Allocation for Cloud Computing Jobs[16]. It contains three major modules:

- *Resource Request:* They took the request in the form of a tuple which has 4 values; i)number of VMs a user wants, ii)Bandwidth required for the requested VM, iii)The time for which a job is run and iv)Time difference between finish time and expected last time.
- *Resource Allocation:* Two mechanisms which were considered are time sliding and bandwidth scaling. By time sliding, they tend to enable a lag between the job finish time and also the actual arrival time. Bandwidth scaling allowed dynamic adjustments of the bandwidth allocated to VMs.
- *Charging:* Suppose a selfish user always declares the job deadline as small as the running time of the job or purposely requests a really less bandwidth so as to pay less cost, the balance between VM and Bandwidth demand could

be affected. So the authors tried to implement it in such a way that the client truthfully declares the values.

DCloud considered only homogeneous bandwidth. The main focus in DCloud's request abstraction is the last 2 values: the execution time of job p, and the job deadline d. A user will expect the least time for the job to end, after that the result might not be usable. DCloud manages to handle the resource requests in both temporal dimension and spatial dimension to improve the usage of resource more efficiently.

The job arrival rate in DCloud is Poisson Process. The authors compared the algorithm results with Virtual Cluster Algorithm and Baseline Allocation Algorithm.

Chapter 3

Research Problem

3.1 Problem Statement

Cloud Computing is a growing technology in today's era as all the companies are moving towards cloud. Some of the examples are:

- *Microsoft*:Microsoft is one of the amazing providers in every field, either its of physical devices or the cloud one. It has 3 cores: Azure, Windows Server and Microsoft system Center. Its cloud is growing because it offers flexibility and take the existing products into considerations with their benefits. The Windows Azure covers most of the services such as IaaS, DBaaS, and PaaS. Microsoft retailers are also working on to provide the user with other services also.

- *Amazon*: Amazon's Amazon web Services (AWS) is the most used public cloud as it provides with almost all the facilities with its wide services such as EC2, S3 etc. The various facilities it provides makes it one of the top competitors in the cloud market. Other Amazon partners give backup and personal storage, knowledge integration, security and configuration management.

- *VMware*: VMware is a hybrid cloud provider that works for the benefits of both public and private cloud. It is historically the earliest of all. Its approach is similar to that of AWS. The private cloud facilities are provided by VSphere. The hybrid facilities are provided by VCloud Air.

- *Google*: As Google is a well framed name in the field of IT, Google find it easier to establish the user base among the already set users. With its history in knowledge, applications like BigQuery are really helpful in big data analysis. Due to the fact that Google has same partners that AWS

uses in its hybrid cloud, users can expect a similar behavioural performance of Google Cloud too.

- *IBM*: Bluemix is a hybrid cloud developed by IBM. It has open design and also focuses on developer needs and services requirements. Using a service known as Relay, IBM is ready to make sure that public and private cloud appear invariably equivalent by adding the transparency and serving the users with DevOps efforts.

Now, due to large number of user requests, scheduling of those requests has been a really difficult tasks for all the vendors. Scheduling algorithm used by a vendor should be favorable to both the user and the provider. Although there are many scheduling algorithms already present but none of them deals with the scenario where request from user may or may not come at a particular time or at a regular interval. Job scheduling refers to the different algorithms that CSPs use to process user requirements for infrastructure, platforms or software in a cloud environment. Users do not actually own a resource but they are allocated with the resources they need at the time of scheduling. It also improves efficiency and minimize the variation during resource demand. Earlier job scheduling algorithms in Cloud computing, solely take into account a way to fulfil the QoS needs for the resources users. They rarely take into account a way to build the utmost profits for the resource providers. While a CSP assumes, it will gain utmost profits by providing the resources irrespective of user's QoS needs. To minimise the difference between these two necessities, the job scheduling system should take efficient and economic measures. A cloud scheduling algorithm that adopts a scenario where jobs could be scheduled at different times by considering both the arrival time and the deadline needs to be proposed.

3.2 Research Motivation

Scheduling is necessary in every aspect whether its physical or virtual world. None of the algorithm dealt with the scheduling of jobs at different arrival time of the job requests being made by users. They only considered probabilistic approach

in order to generate the jobs. This encouraged me to prepare a job scheduling algorithm which will not only focus on QoS needs of users' requests at different arrival time but also the deadline.

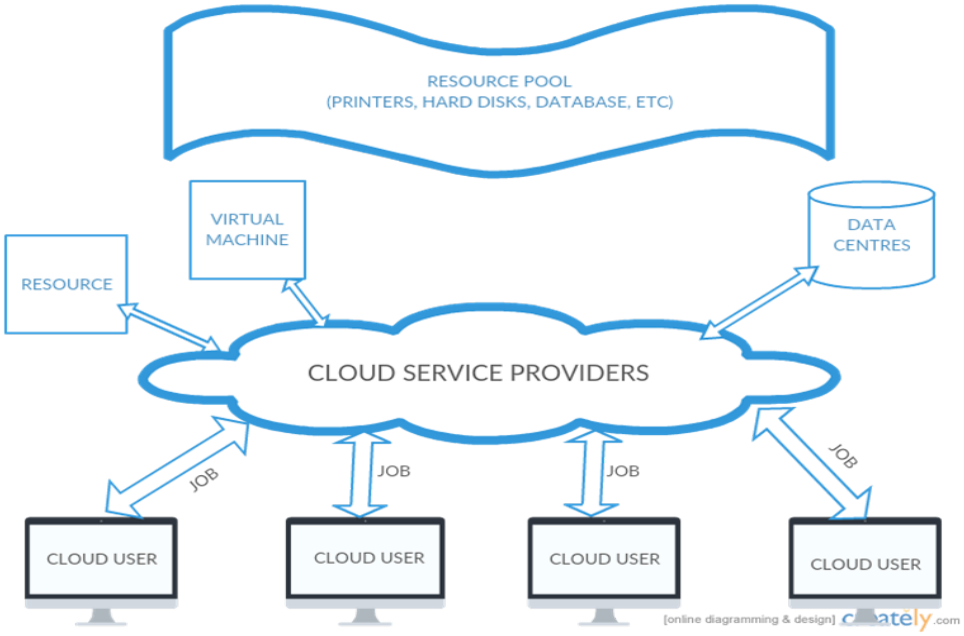

Figure 3.1: Job Scheduling in Cloud Computing

3.3 Research Gaps

Considering all the factors discussed above the following objectives could be formulated:

- To study about the different Cloud Service Providers.
- To study various algorithms and techniques for scheduling.
- To develop an algorithm to schedule the jobs with a particular arrival time and within a given deadline constraint.
- Execution of proposed algorithm and comparing result with traditional algorithm.

Chapter 4

Implementation

The following section discusses the implementation details of the project and the algorithm used with backend functions.

4.1 Extracting of Data from Excel File Provided by User

The request of the user in this research work is taken with the help of excel file. It has all the values which makes a CSP aware of the specifications of the resources. The parameters which are provided in the excel file has been shown in Figure 4.1 are:

- *Arrival Time*-The time(in ms) at which job has arrived.
- *Main Memory Required*-Main memory(in MB) required to accomplish the task.
- *Storage*-The storage(in MB) required by each cloudlet to store data and execute successfully.
- *Deadline*-The time(in seconds) in which cloudlet should complete its execution.

The data extraction is done by using XSSFworkbook class of Apache POI. The steps can be summed up as:

1. Include poi-3.12.jar in Java program's classpath.
2. Create an object of XSSFWorkBook which can be made by opening of the file with the help of FileInputStream.

	A	B	C	D
1	Submission Time (ms)	Main memory (mb)	Min Storage	Deadline
2	284290	125691	33	69
3	99880	775985	89	42
4	160530	624812	57	81
5	396489	977441	8	51
6	125711	393314	56	34
7	58708	119904	13	58
8	707453	133949	2	38
9	757182	414678	62	23
10	645825	306903	25	85
11	497770	442595	3	62
12	350652	451243	25	31
13	445910	525344	86	11
14	666235	542536	17	70
15	848938	156069	19	67
16	65284	171735	84	24
17	310017	36016	71	56
18	19613	890091	95	60
19	734145	165710	87	61
20	702545	403812	80	55
21	123177	725166	43	48
22	3772	6689	13	35
23	27602	961853	14	78
24	526805	774456	90	26
25	220369	873909	11	30

Figure 4.1: Excel File Containing Details of the Jobs

3. Take a Sheet from the created workbook by calling getSheetat() method of XSSFSheet class.

4. Go to the particular Row of that sheet by using getRow() method of XSSFRow class.

5. Go to the Cell that has the data by using getCell() method by of XSSFCell class.

6. Depending upon Cell type, use getStringCellValue(), getNumericCellValue() or getDateCellValue() method to fetch the value.

7. Close() method is used to close the workbook after each value is fetched from the sheet.

4.2 Using CloudSim Tool

The steps required for using CloudSim tools are:

1. Initialise the CloudSim package and the library.
2. Create DataCenters, then brokers and the VM Lists.
3. Create Cloudlets according to specifications in workload file. Initialise cloudlet according to those specifications.
4. Add cloudlet to the cloudlet list and submit the list to the broker.
5. Start the simulation and print the final results.

The process of the CloudSim simulation environment could be seen in Figure 4.2.

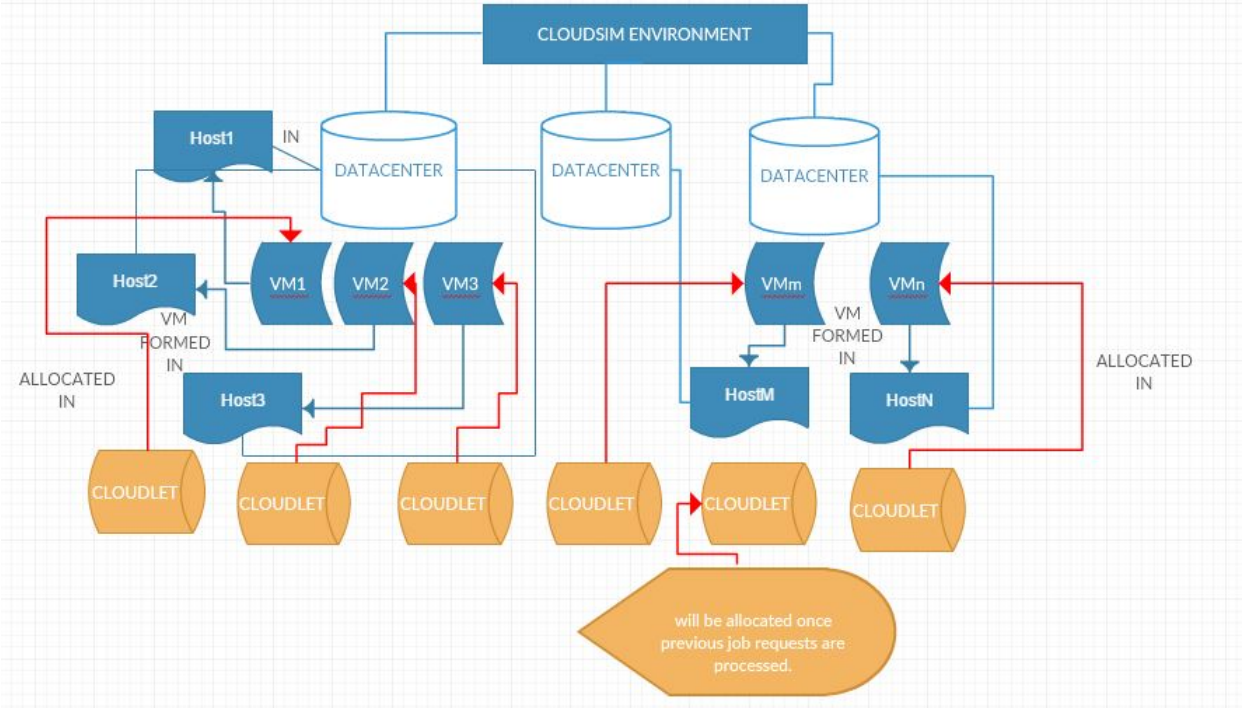

Figure 4.2: Working of CloudSim Simulation Environment

4.2.1 Creating DataCenter and Hosts

The data center is the actual hardware having several VMs. Following are the characteristics of the Data Center:

- System Architecture
- Operating System

- Virtual Machine Manager
- Host list
- Time zone of the resource located
- Cost of using processing in the resource
- Cost of using memory in the resource
- Cost of using network in the resource

In this research work, only one datacenter is considered with 400 Virtual Machines and 200 hosts.

4.2.2 Creating Cloudlets

In CloudSim, cloudlet is a job submitted to cloud. Cloudlet class in CloudSim defines following parameters in its constructor:

- *Cloudlet Id*: It is the unique identification number associated with each cloudlet.
- *Cloudlet Length*: It defines the length of cloudlet in terms of number of instructions to be executed.
- *Cloudlet File Size*: It represents size of input file of the job before its execution.
- *Cloudlet Output File Size*: It represents size of cloudlet file after its execution.
- *Number of processing elements*: Pe represents processing element. pesNumber represents the number of processing elements available for completing the task.

The additional parameters used are Arrival Time, Main Memory, Storage and Deadline as discussed in section 4.1.

4.3 Implementing ADSF in the CloudSim project

In this thesis, Arrival based Deadline First Scheduling Algorithm(ADSF) is proposed. In this algorithm, deadline is given the priority and user's requests are assigned according to that. The algorithm could be implemented by modifying the DataCenterBroker class.

4.3.1 Arrival based Deadline First Scheduling Algorithm

- The algorithm proposed in this work starts by creating a cloudlet list from the details submitted by the user and sorting them according to the arrival time.
- Once they are sorted, next step is to send them to the broker for further scheduling, according to the deadline first scenario.
- The next step involves sending the cloudlet to the host according to the Arrival Time using the Schedule() function and not all at the same time.
- At last, the cloudlet which has been submitted is removed from the list and the process continues till no more request is left.

The implementation of the algorithm can be summed up diagramatically as shown in Figure 4.3

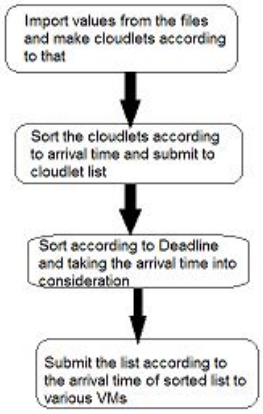

Figure 4.3: Implementation of ADSF

4.3.2 Algorithm

The algorithm is implemented in the DataCenterBroker class by using TimeShared scenario. This class is solely responsible for calling all the necessary methods and classes for further implementation of any algorithm and thereby helps a CSP to assign the jobs. The specification which were implemented in my research work is that the Hosts were 200 with 4 cores of 10000mips each, 32GB of RAM and 10TB of storage.

Algorithm:

Data:
Cloudlet: the cloudlet list containing various cloudlets
Deadline: the cloudlet deadline
ArrivalTime: the cloudlet arrival time
VMList: the VM list
Broker: the data center broker
I : Iteration variable

Procedure:
Sort Cloudlet according to ArrivalTime.
Submit the list to Broker.
For V < VMList do
    Sort the newList according to deadline.
    For C < newList.Length do
        Submit Cloudlet[C] to VM[V] according to Arrivaltime using schedule()
        Remove Cloudlet[C] from List
    End for
End for

Chapter 5

Results

The proposed ADFS algorithm has been implemented in CloudSim. Performance of the job is evaluated by gradually increasing the number of cloudlets and changing deadline patterns. Finally, this has been compred with FCFS.
So, the implementation consists of three parts. First, to import the values from the file and make cloudlets with them. The number of cloudlets are gradually increased. The arrival time and deadline are varied randomly. Second, to implement the algorithm in the DataCenterBroker class which was modified in order to sort the cloudlets according to the arrival time and deadline. Sorted cloudlets are then send to the host. Thirdly, it requires the evaluation of various parameters for both ADSF and FCFS by varying the number of cloudlets. The performance was evaluated by following parameters:

- Total cloudlets that missed the deadline
- Total Delay for the cloudlets
- Total Waiting time for the cloudlets

5.0.1 Cloudlets That Missed the Deadline

These are the cloudlets or the jobs which could not complete within the deadline provided by user in the excel file.

5.0.2 Total Delay for the Cloudlets

It is the total delay by which the cloudlets that missed the deadline are delayed. It is total sum of the delayed cloudlet's time.

5.0.3 Total Waiting Time

It is the time taken by a cloudlet to be scheduled on a resource from its arrival time i.e. the time taken to wait.

5.1 Results

The implementation of this algorithm is done in a simulation environment which is provided by CloudSim tool. CloudSim flow of execution starts by creating the datacenters, broker and VMLists. The cloudlets are created and submitted to broker. The broker binds them to various VMs. The hosts used in this work have 4 cores (10000 mips each), 32GB of RAM and 10TB of storage. The number of cloudlets are gradually increased as 100, 200, 500, 1000, 2000, 3000, 4000, 5000, 10000, 20000, 30000, 40000 and 50000. Figure 5.1 shows creation of VM in the datacenter.

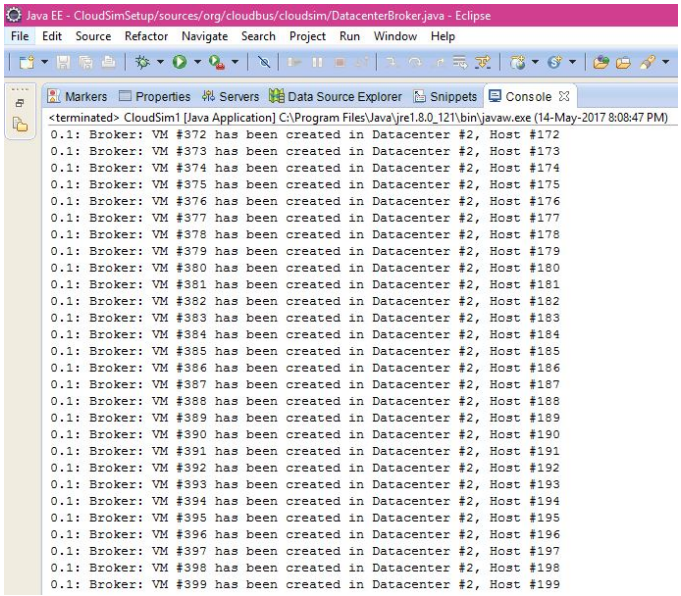

Figure 5.1: Creation of VM in the DataCenter

The basic algorithm which is implemented in DataCenterBroker(DCBroker) class sends all cloudlets at same time. Some modifications are made in DCBroker class for sending the cloudlets according to arrival time. Figure 5.2 shows the scheduling of cloudlets according to modified FCFS.

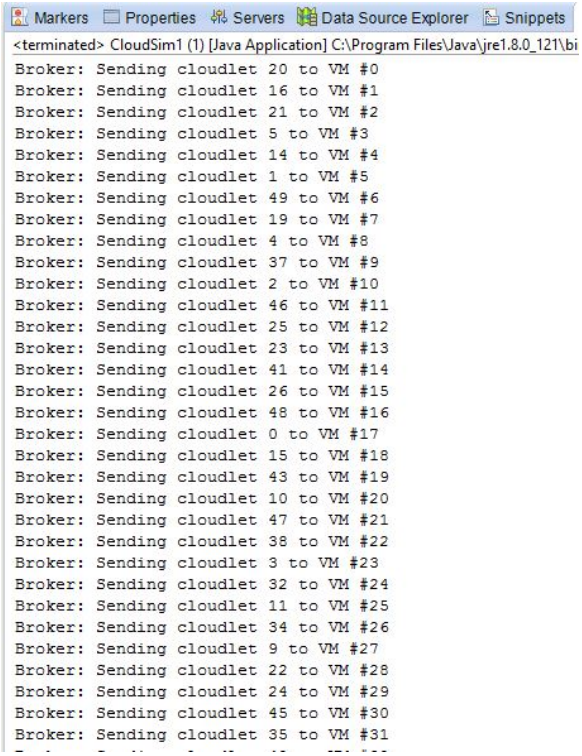

Figure 5.2: Scheduling of Cloudlets According to DCBroker

The ADSF broker sends the cloudlets to the VM according to the deadline and the arrival time both which can be seen in Figure 5.3. The cloudlet number 14 and cloudlet number 5 have almost similar arrival time. The deadline of Cloudlet 14 is early but in DCBroker it is scheduled late as compared to ADSF.

5.1.1 Delayed Cloudlets

The delayed cloudlets are those that could not complete their job in the mentioned deadline given in the file. The number of delayed cloudlets are highly affected by the type of algorithm. ADSF Algorithm could resolve this to a greater extent as compared to DCBroker. Although, the delayed cloudlets could be made 0 if we increase the number of hosts. The percentage of Delayed cloudlets for both ADSF broker and DCbroker are shown in Table 5.1:

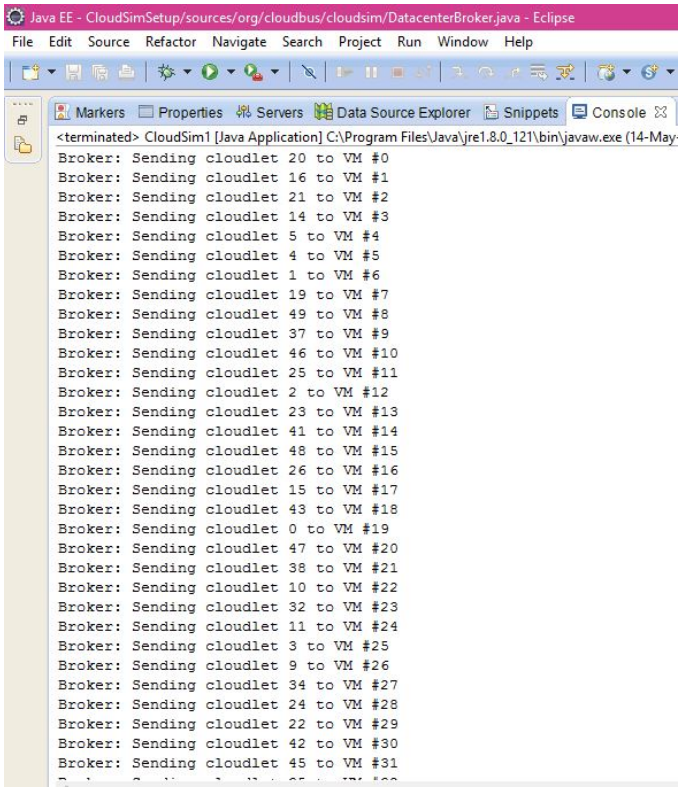

Figure 5.3: Scheduling of Cloudlets According to ADSF

Table 5.1: Delayed Cloudlets

No. of Cloudlets	DCBroker(%)	ADSF Broker(%)
100	9	3
200	7.5	2.5
500	7.4	1.6
1000	7.9	1.3
2000	7.9	1.1
5000	8.8	1.76
10000	9.8	2.61
20000	11.585	3.72
30000	13.44	5.92
40000	15.1575	7.81
5000	17.262	9.564

The Figure 5.4 shows the total cloudlets that were delayed due to different algorithms.

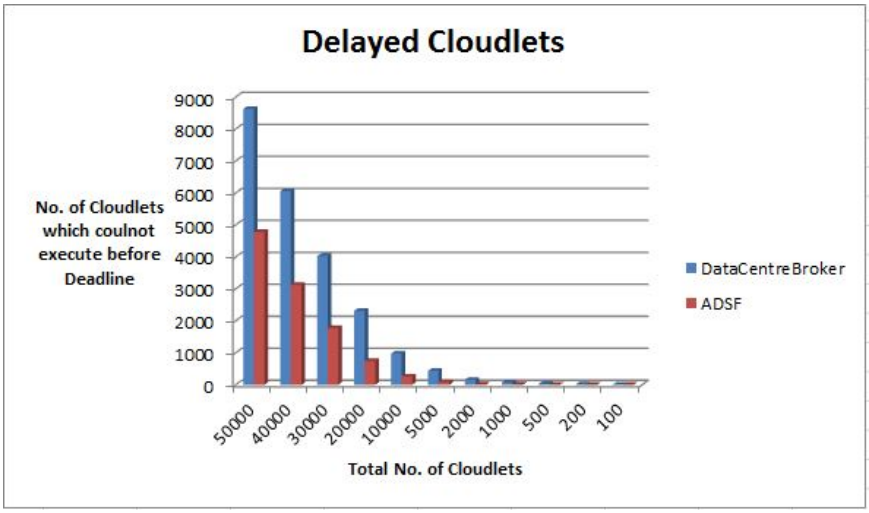

Figure 5.4: Number of Delayed Cloudlets

5.1.2 Waiting Time

The waiting time of the cloudlets are calculated. It can be clearly seen that the difference between waiting times of DCBroker and ADSF are not much. This shows ADSF was able to combat with the need of shorter waiting time. The average waiting time that both the algorithms exhibited are somewhat similar which can be seen in the Figure 5.5:

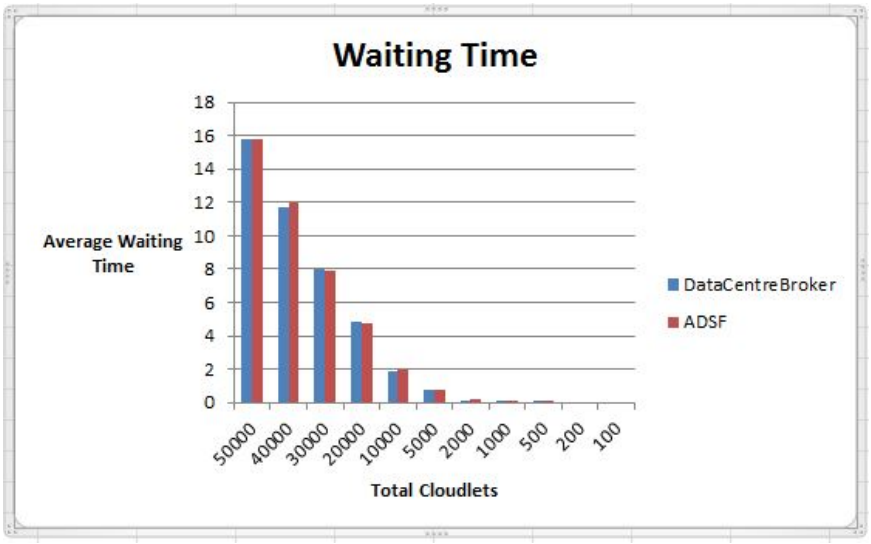

Figure 5.5: Average waiting Time of Cloudlets

5.1.3 Total Delay Time

The total delay of the delayed cloudlets is calculated. There is a huge difference between the two algorithms in case of Delay Time. The total delay that the cloudlets exhibited can be seen in the Table 5.2

Table 5.2: Total Delay in Cloudlets

Cloudlets	DCBroker	ADSF Broker
100	50.88	14.77
200	76.56	27.78
500	188.93	38.85
1000	390.38	32.64
2000	903.12	121.17
5000	4116.81	1770.81
10000	15288.41	10760.33
20000	69969.09	55310.20
30000	174698.06	145618.01
40000	327332.59	286817.48
5000	557608.31	486105.91

The total delay time of the cloudlets is shown graphically in Figure 5.6

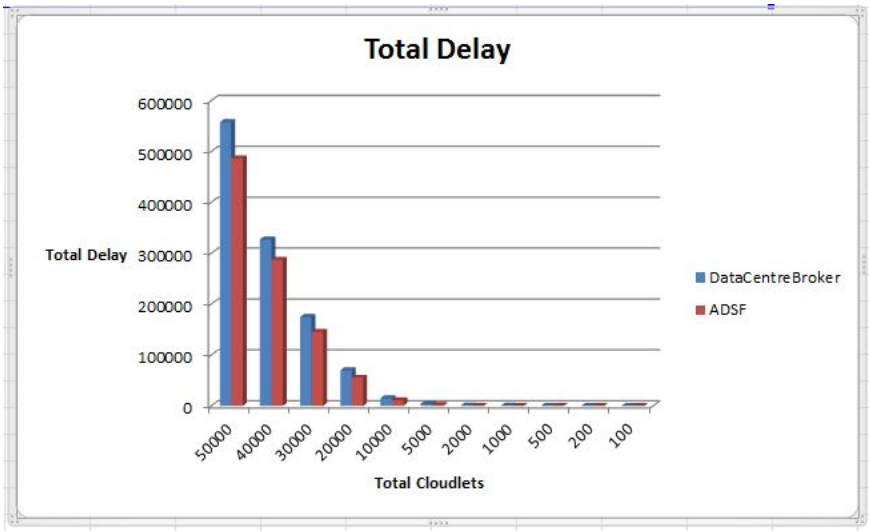

Figure 5.6: Total Delay Time of Cloudlets

As can be seen from the above statistics that ADSF is more efficient as compared to the modified FCFS algorithm. Total delay and total delayed cloudlets are much more in case of modified FCFS.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In today's era, Cloud Computing has been embraced by every single organisation, from start-ups to global organisations. Cloud Computing provides scalable solutions to every kind of business demand. For start-ups, as the business grows cloud services required also grow. Moreover, Cloud Computing serves the best environment for innovation as it makes deployment of new ideas very easy and at reasonable costs. Being able to connect business all over the world, it makes flow of innovative ideas very easy. In case of data loss, it provides data back-up recovery solutions. Cloud Computing has proved to be a robust technology in every way. So the services provided by Cloud Computing are ideal for every kind of business demands. But with increasing demands, there is a need of scheduling algorithms for resource allocation.

In this thesis, an effort has been made to develop an algorithm for different arrival time based scenario for Cloud Computing environment. The deadline and arrival time are present in the excel file provided by user. The proposed algorithm works much better than the DCBroker class which implemented modified FCFS. The basic FCFS algorithm works when there is no arrival time specified and all the cloudlets are submitted at same time. Modified FCFS helps in scheduling the cloudlets according to arrival time. Results of the ADSF are much better as compared to DCBroker class which implemented FCFS.

6.2 Future Works

- As a future work, this algorithm can extended further be to the extent where other factors such as energy, cost etc. can also be taken into consideration.
- It can further be extended to heterogeneous types of cloudlets.
- The Cloudlets that missed the deadline were about 5% when there were really large request, work can be done to further reduce it.

References

[1] P. M. Mell and T. Grance, "Sp 800-145. the nist definition of cloud computing," Gaithersburg, MD, United States, Tech. Rep., 2011.

[2] IBM, "Evolution," https://www.ibm.com/blogs/cloud-computing/2014/03/a-brief-history-of-cloud-computing-3/, 2017.

[3] JAVATPOINT, "Virtualization in cloud computing," https://www.javatpoint.com/virtualization-in-cloud-computing, 2017.

[4] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De 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, 2011.

[5] K. S. Nishant Sanghani, Khimani and P. Vasani, "Pre-emptable shortest job next scheduling in private cloud computing," *Journal of Information Knowledge and Research in Computer Engineering*, vol. 2, no. 2, pp. 385–388, 2012.

[6] G. Du, H. He, and Q. Meng, "Energy-efficient scheduling for tasks with deadline in virtualized environments," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[7] I. C. Limited, "Paas," https://www.interoute.com/what-paas, 2017.

[8] Salesforce.com, "Saas," https://www.salesforce.com/in/saas/, 2017.

[9] C. C. Rao, M. Leelarani, and Y. R. Kumar, "Cloud: Computing services and deployment models," *International Journal Of Engineering And Computer Science*, vol. 2, no. 12, pp. 3389–3390, 2013.

[10] T. Swathi, K. Srikanth, and S. R. Reddy, "Virtualization in cloud computing," *International Journal of Computer Science and Mobile Computing, ISSN*, pp. 540–546, 2014.

[11] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," *International Journal of Research in Engineering and Technology*, vol. 2, no. 2, pp. 131–135, 2013.

[12] A. Verma and S. Kaushal, "Deadline and budget distribution based cost-

time optimization workflow scheduling algorithm for cloud," *IJCA Proceedings on International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT 2012)*, vol. iRAFIT, no. 7, pp. 1–4, April 2012.

[13] P. Devi and T. Gaba, "Implementation of cloud computing by using short job scheduling," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 7, pp. 178–183, 2013.

[14] M. O. Shamsollah Ghanbari, "A priority based job scheduling algorithm in cloud computing," *International Conference on Advances Science and Contemporary Engineering*, 2012.

[15] M. K. Sathya and S. Rajalakshmi, "Deadline based task scheduling in cloud with effective provisioning cost using lbmmc algorithm," *International Journal On Engineering Technology and Sciences*, vol. 1, no. 7, 2014.

[16] D. Li, C. Chen, J. Guan, Y. Zhang, J. Zhu, and R. Yu, "Dcloud: Deadline-aware resource allocation for cloud computing jobs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 8, pp. 2248–2260, 2016.

List of Publications

IEEE Conference

1. Swati Gupta, Rajesh Kumar *"Arrival based Deadline aware Job Scheduling in Cloud"* in ISPCC 2017

Video Link

The video can be found on the link: https://www.youtube.com/watch?v=O69OmCyjV1o

Thesis

ORIGINALITY REPORT

%8	%5	%5	%2
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	%1
2	Li, Dan, Congjie Chen, Junjie Guan, Ying Zhang, Jing Zhu, and Ruozhou Yu. "DCloud: Deadline-aware Resource Allocation for Cloud Computing Jobs", IEEE Transactions on Parallel and Distributed Systems, 2015. Publication	%1
3	G. Kousalya, P. Balakrishnan, C. Pethuru Raj. "Automated Workflow Scheduling in Self-Adaptive Clouds", Springer Nature, 2017 Publication	%1
4	Dan Li, Congjie Chen, Junjie Guan, Ying Zhang, Jing Zhu, Ruozhou Yu. "DCloud: Deadline-Aware Resource Allocation for Cloud Computing Jobs", IEEE Transactions on Parallel and Distributed Systems, 2016 Publication	%1
5	docplayer.net Internet Source	<%1