COMPUTATIONAL OPTIMIZATION IN CLOUD COMPUTING

A Dissertation submitted in partial fulfillment of the requirement for the

Award of degree of

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By

DEEPAK CHAURASIA

(2K14/ISY/03)

Under the esteemed guidance of

Mr. MANOJ KUMAR
Associate Professor

2014-2016

Department of Computer Science & Engineering

Delhi Technological University

Delhi-110042

India

CERTIFICATE

This is to certify that the thesis entitled "**Computational Optimization In Cloud Computing**" submitted by **Deepak Chaurasia (2K14/ISY/03)** to the Delhi Technological University, Delhi for the award of the degree of **Master of Technology** is a bona-fide record of research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: DTU, Delhi

Date: _____

Mr. MANOJ KUMAR
Associate Professor
Department of Computer Science & Engineering
Delhi Technological University, Delhi

ACKNOWLEDGEMENT

First I would like to express my gratitude towards my supervisor **Mr. Manoj Kumar** *Associate Professor, Department of Computer Science* for his able guidance, support and motivation throughout the time. It would not have been possible without the kind support and help of many individuals and **Delhi Technological University**. I would like to extend my sincere thanks to all of them. I would like to express my gratitude and thanks to **Dr. O.P Verma (*Head of Dept.)*** for giving me such an opportunity to work on the project. I would like to express my gratitude towards **my parents & staff** of Delhi Technological University for their kind co-operation and encouragement which helped me in completion of this project. My thanks and appreciations also go to my **friends and colleagues** in developing the project and people who have willingly helped me out with their abilities.

Deepak Chaurasia
Roll No.: 2K14/ISY/03
Department of Computer Science & Engineering
Delhi Technological University

ABSTRACT

The arrival of cloud computing in recent years has attracted an interest from various users, organizations and institutes to take benefits of services and applications. Cloud computing offers on-demand availability of resources with high scalability. But as the number of internet users is increasing day by day, it gets quite difficult to handle the requests coming from millions of user. In such cases it is very common that the performance and availability of system goes down. This presented work discusses the design and implementation of system which does neural network based cloudlet classification and schedules them based on fuzzy based priority scheduling algorithm. This paper demonstrates that with this fuzzy based priority scheduling algorithm we can reduce the average waiting time and average turnaround time of cloudlets. The results are compared with conventional Priority based scheduling and First Come First Serve scheduling algorithms.

TABLE OF CONTENT

LIST OF FIGURES

SCREEN SHOTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Cloud Computing Evolution

In previous decades we have been given the idea of information processing, in a more efficient and better manner .With the emergence of cloud computing[20] headache of processing and accessing and storing, the data through internet at larger scale is gone now. The network based computing idea has led to the evolution of Grid computing in early 1990s and since 2005, to utility computing which ultimately brought to the development of cloud computing. From very long time researchers and scientists are trying to give utilities as services to its user. User can demand for software, platform and hardware resources from a cloud provider through internet and charged on the respective usage basis .So cloud computing is a path to utility computing by IT giants like Microsoft, IBM, Hp, Amazon, Google etc. Within very short span of time this technology has scattered over the globe. The Cloud technology has a sensational rise in the industry that can be seen in Figure 2.1.

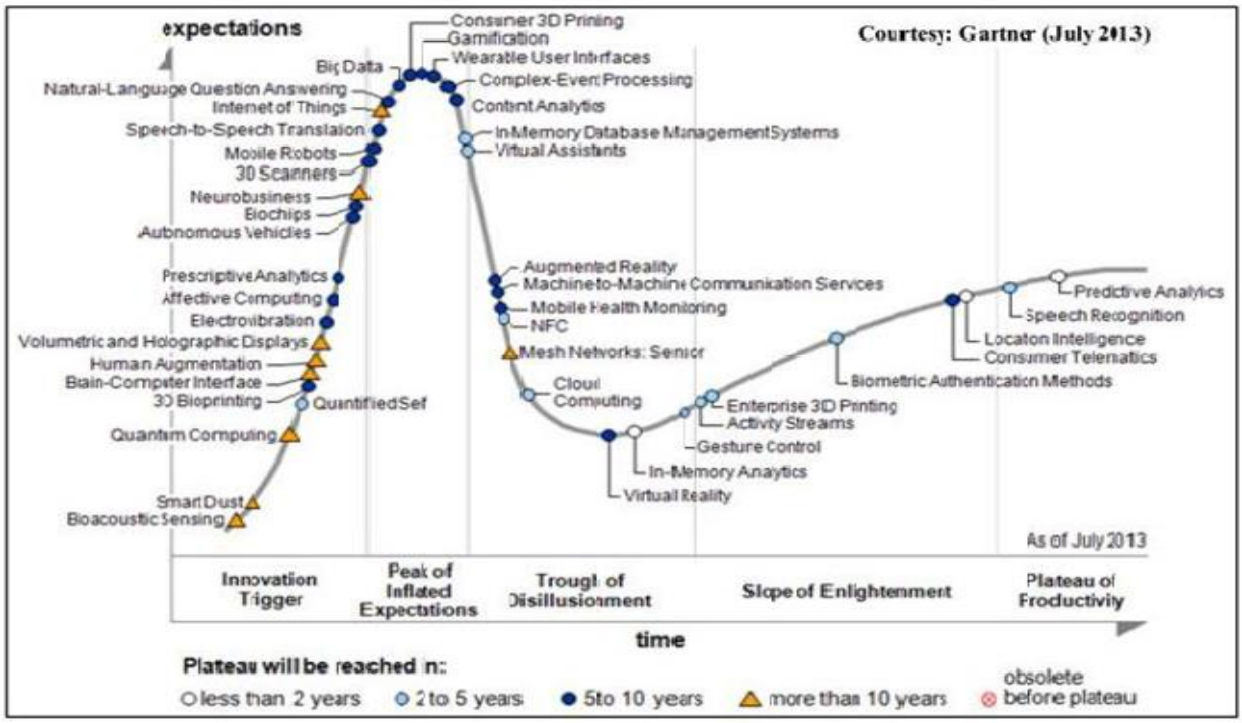


Figure.2.1: Hype Cycle for Cloud Computing. 2013

1.2 Cloud Computing Definitions

Since 2007, term cloud has got more popularity in IT industry [21] .There are vast number of definitions are given by the various scientists and researchers. Cloud computing has been defined according to different- different application. But for cloud computing there is no any standard and common definition is available. There are some of very good definitions that are chosen among all definition for cloud computing, they are as following-

- NIST: "Cloud computing is a model to enable ubiquitous, convenient on demand network access to a pool of shared resource that is network, server, storage and application. These can be rapidly provisioned and released with minimum management effort or service provider interaction".[1]

- FOSTER: "A large -scale of distributed paradigm that is driven by the economics of scale ,in that a pool of abstracted virtualized, dynamically-scalable, managed storage, computing power ,platforms and services are delivered on demand too intersected external customer over internet".[2]

A specific and goal oriented definition of cloud computing is provided [22] by U.S. National Institute of Standards and Technology. With cloud service delivery and deployment models it also provides specifications for the characteristics of cloud computing. Foster's definition can be said to be little bit different as an educational representative he mainly focuses on various methodological features that differentiate cloud computing models from other distributed computing paradigm. For example, computing entities are virtualized and delivered as services, and these services are dynamically driven by economies of scale.

The term "Cloudonomics" has been given by Mr. Joe Weinman which defines cloud computing's economical perspective, which is discussed below:

1. **C**ommon Infrastructure: It is a common and standard pool of the resources that is made available to all the cloud users charged on the usage basis.

2. **L**ocation Independence**:** User can access cloud resources and services from anywhere around the globe. That leads the better performance and gives better response of system in time.

3. **O**nline connectivity: There is need of maintain consistent internet connection to access the cloud resource over the cloud.

4. **U**tility Pricing: Cloud services are being provide on pay-per-use pricing and benefits the users as per their demand.

5. On-**D**emand Resources: Scalable elastic resources are allocated and de-allocated without delay or costs associated with change.

1.3 Cloud Components

As far as the topological aspect of cloud is concerned, there are various elements we do have like clients (or users), the datacenter and the distributed servers. These components integrate the cloud as one single unit. In Figure 1.2, these components are shown:

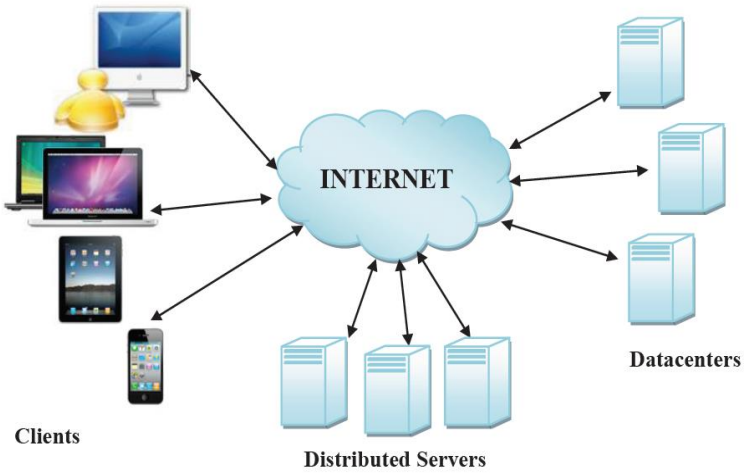


Figure.1.2: The Components Of Cloud Computing

Each component has specific role assigned to it in delivering services asked by the cloud users. These components communicate with each other over the entire network, as per the requirements configuration and connectivity. The detail information about these components are discussed below [3]

1.3.1 Clients

These are generally, the computing devices which we use in our daily work. It can be in format of a tablet computer, laptop, a PDA and a mobile phone. All these devices should be able to access the cloud computing interface through internet. With these devices end users communicate with the cloud interface and thereafter can use the service or applications as per their choice.

1.3.2 Datacenters

A datacenter is one of the most core elements of cloud computing model. A datacenter consists of several nodes that have the capacity of occupying in a large room. Cloud service providers configure these data centers. Various factors like cloud deployment model and the cloud service model decides this configuration.

1.3.3 Distributed Servers

These servers, also called as nodes. They need not be deployed at the same location but may be distributed geographically as per the convenience and comfort of the service provider. From users prospect these servers looks like to working together, without knowing the actual location of the server. Distributed servers increase the fault tolerance of the network.

1.4 Cloud Computing Services

Once a cloud is being deployed by cloud provider and it is ready to be used by its cloud users. Now It is time to be decided that how this service will be offered by cloud provider. We can also

refer to the figure 1.3 for the brief description of the services provided by cloud computing .There are basically three levels in which the cloud services can be offered they can be named as software as a service , platform as a service and infrastructure as a service respectively. The details of these levels can be given as following-

1.4.1 Software As A Service(SAAS)

In Traditional way when we wanted to use some software application then there was a need of purchasing it and installing it onto our computer. But with this cloud computing model users need not to purchase and install software onto their system . A variety of software applications in the of cloud services can be used by user by just registering on cloud services provider website. These application can be accessed via internet such as web based services. Cloud consumers need not to worry about required hardware, network administration, programmer and developer to deploy their applications. Users need also not to worry about configuration and maintenance of software's and hardware. It is totally the responsibility of third party whom users are registered with. It works on "pay per use" subscription model .Companies are such as Facebook , youtube ,Google docs, NetSuite and Microsoft online are some of the famous example of this model.

1.4.2 Platform As A Service (PAAS)

Platform as a service (PaaS) is collection of development tools and software's which are housed on the cloud provider's server. The environment and development tools are provided by this model that allow application developers to build their applications according to their requirement. PaaS provides tools to cloud user to create their own version of customized application. It is middle-layer of service mode of cloud computing . The facility to cloud user to develop their own application using variety and required programming language, libraries, services and tools is provided by it. Users have full of control over the deployed application configuration in this model. There are some features which are provided to the cloud user at this layer are-

- Operating system for the application
- Server software
- Database management system

- Storage
- Network
- Tools for design and development

Service provider companies for example window Azure, Engine Yard and Google App Engine are the example of this model.

1.4.3 Infrastructure As A Service (IAAS)

Infrastructure as a service reduces the huge investment in computing hardware like processers, networking tools, and servers. We can say that this layer fulfills all the hardware requirements by user to develop their to own version of application. So in IaaS, clients have own programmer, developer, and networker administrator which are responsible for configuration and working of network. A user can use and run any operating system like windows, Linux or Solaris and other software like Matlab, code block, eclipse, Net beans etc. Here headache of managing the cloud infrastructure which was required to run their application is not there for the cloud users . Cloud users are just supposed to manage the operating system and deployed applications that will be running on the hardware infrastructure provided by this model. IaaS Services are being provided by Companies such as Amazon, Go grid etc.

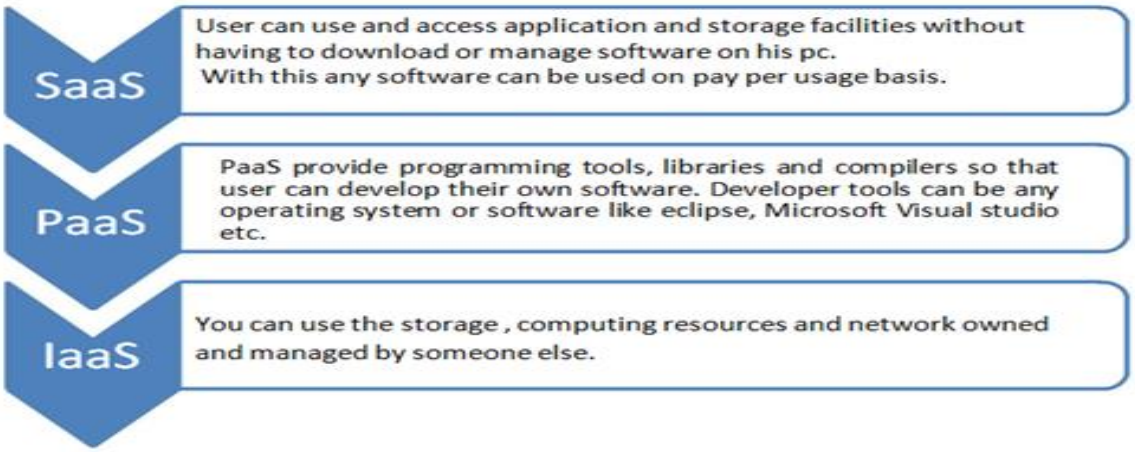


Figure.1.3 : Different services provided by cloud Computing

1.5 Cloud Deployment Model

A cloud deployment model represents a particular type of cloud environment. We can distinguish different type of cloud environment them by their ownership, size, and accessibility.

1.5.1 Public Cloud

Public cloud are developed to serve the requirements of general public. A Cloud Service Provider (CSP) serves the service requirement like if there is any storage requirement of user, or if there is any application is demanded on cloud . public clouds are managed ,developed , operated and owned by third party like government, academic, or business organization. Examples of public clouds are Amazon's Elastic Compute Cloud (EC2), IBM's Blue Cloud, Google's App Engine, Sun Cloud and Windows Azure Services Platform.

1.5.2 Private Cloud

Private cloud ,offers the cloud based services for private use only. An organization can develop its own private cloud infrastructure or it can request to a third party . Private cloud permits only the authorized users , hence gives the organization bigger and better control over their data. Big organizations like Microsoft, Google they have their own cloud infrastructure, which are supposed to be rather secure than the public clouds facility. An organization can also serve many other users and organizations by their own private cloud for e.g., Google is serving its users by providing them services like Gmail, Drive, Google App engine and many more.

1.5.3 Community Cloud

Community cloud is a type of cloud hosting in which the whole system setup is mutually shared among many organizations and these organizations belong to a particular type of community, for example banks and trading firms. It is a kind of setup that is shared among several companies that belong to a specific group which has similar computing requirements. The group members generally have similar performance, privacy and security concerns. The main goal of these communities is to achieve their similar business related objectives. A community cloud can be managed by a third party or it can be internally managed . community cloud can be hosted externally or internally. Community cloud is having cost saving capacity because the cost of cloud is shared among organizations within the community. A community cloud is better for businesses

and organizations that work on joint tenders , research or ventures and that needs a centralized cloud computing ability for building managing, and implementing similar projects.

1.5.4 Hybrid Cloud

Hybrid cloud is a combination of two or more different types of Cloud deployment models. It may be a combination of a private cloud, a public cloud or a community cloud .Cloud infrastructures are provided in these clouds are entirely different way. Here the benefits of the multiple deployment models are available with this hybrid cloud hosting.

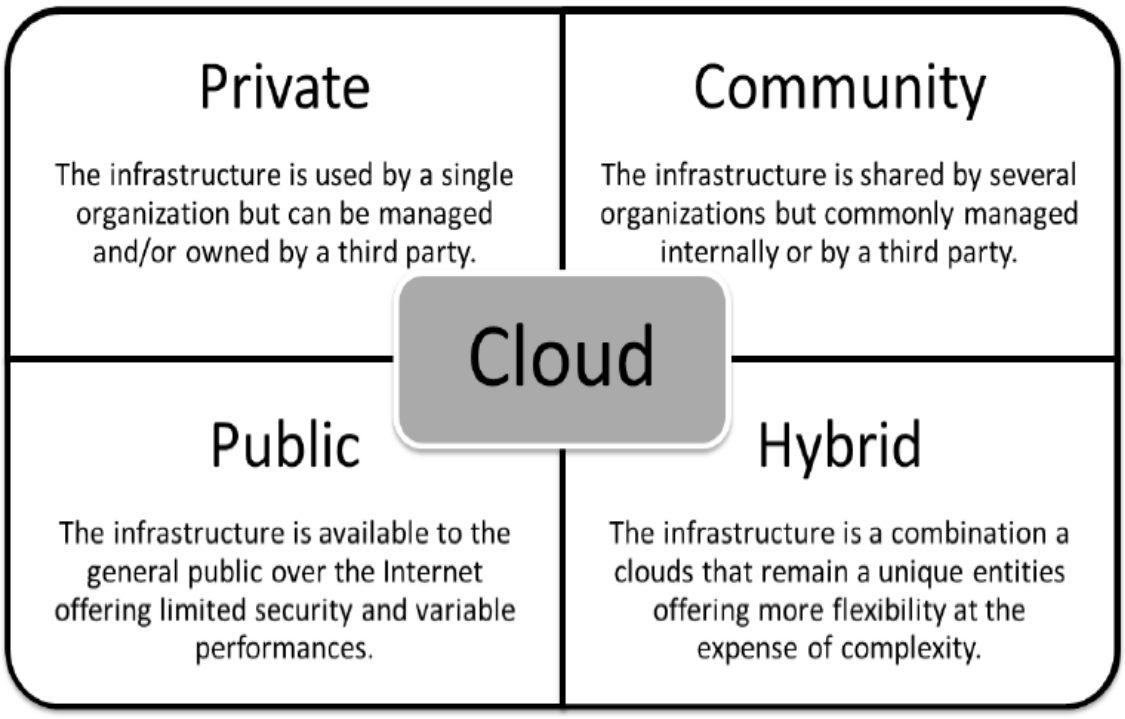


Figure 1.4 : The Type Of Cloud Deployment Model

1.6 Issues In The Cloud Computing

Cloud is a gathering of thousands of computers that look like a single huge computer that is running as a personal computer, cell phone or any other device which is capable of using Internet and which can share resources to provide services on demand basis to the end users. The services

can be storage or computation or any other . Cloud is capable of providing 1GB or 2 GB storage capacity to the cloud users that can be shrinking or expanding on demand basis. Besides cloud, end user cannot get such type of facility as it cannot separate 2GB from a 80 GB hard disk. Highly scalable, reliable , flexible and sustainable services are being provided by the cloud . There are various issues related to the cloud services which are as following-

1.6.1 Task Scheduling

The jobs in a proper manner should be scheduled for the better utilization of computing , storage and network resources. By this way we can enhance the efficiency of cloud environment.

1.6.2 Load Balancing

It is the appropriate distribution of cloudlets on the virtual resources so that least resources are in under loaded condition and simultaneously least resources are over loaded condition.

1.6.3 Scalability

It is the capacity of cloud based system which decides how well the system performs when there is scaling up and scaling down of cloud workload.

1.6.4 Elasticity

It is the property that decides when and how the system would have been scaled up and down in order to satisfy the requirement of the cloud users.

1.6.5 Availability Of Services

Cloud services should be available on time and the resources should be allocated as per demand by the users.

1.6.6 Performance Uncertainty Due To Virtualization

Virtual machines are not easily deployable beforehand as the requirement is not very clear early from the user side.

1.6.7 Interoperability

Cloud infrastructure uses the shared concept to carry out the things. Same tools and application are used across multiple cloud platforms. Hence it’s required a need of migrating in or out and switching to clouds.

1.6.8 Portability

It ensures that the work on one cloud platform can be portable to another cloud platform.

1.6.9 License Management

Cloud computing gives facility to the user that he can use the services on the pay per usage . For this ,there is an agreement between the cloud user and the cloud providers before the delivery of the services to the client. The client gets login details and he is charged for the service he uses.

1.6.10 Security In The Cloud

Security is one of the most basic issue of cloud computing . For using this technology beneficiaries are surrendering all their company's sensitive information to a third-party cloud service provider. This information will be used by the clients using cloud. This can also create a great risk to the company . So businessman are supposed to choose reliable cloud provider.

1.7 Fuzzy logic

The concept of fuzzy logic[10] was first conceived by Dr. Lotfi Zadeha. In classical (a.k.a crisp or binary) logic the variable can have only 2 values as its membership that is either 1(true) or 0(false) so it is also called as 2-valued logic (or Bivalent logic). It means variable is either the part of a given set or it is not the part of that set. While in fuzzy logic the variable (also called as linguistic variable) can have any real membership value in the range of 0(false) and 1(true). It means in fuzzy logic variable is a part of a given set to what extent the is important .In nut shell the Fuzzy logic is a superset of crisp logic. Fuzzy logic replaces Binary values with the degree of truth. These degree of truth is often used to make good decisions in the environment of uncertainty and imprecision.

1.7.1 Fuzzy Inference System

A fuzzy inference system[11] tries to find out answers from a knowledge base by using its fuzzy inference engine component. This inference engine is considered to be the brain of the fuzzy based system . Figure 1.5 shows a typical fuzzy inference system. The inputs are converted into fuzzy

sets using its fuzzification module. In a fuzzy inference system, each input and output variable is represented by its membership function. The fuzzy Inference system uses a set of fuzzy rules defined in its Fuzzy rule base component. . Fuzzy rules are defined in form of "IF-THEN" rules. By using these rules an input space is transformed to an output space . Now The defuzzification component converts the fuzzy set generated by inference engine into a crisp value by aggregation algorithms.

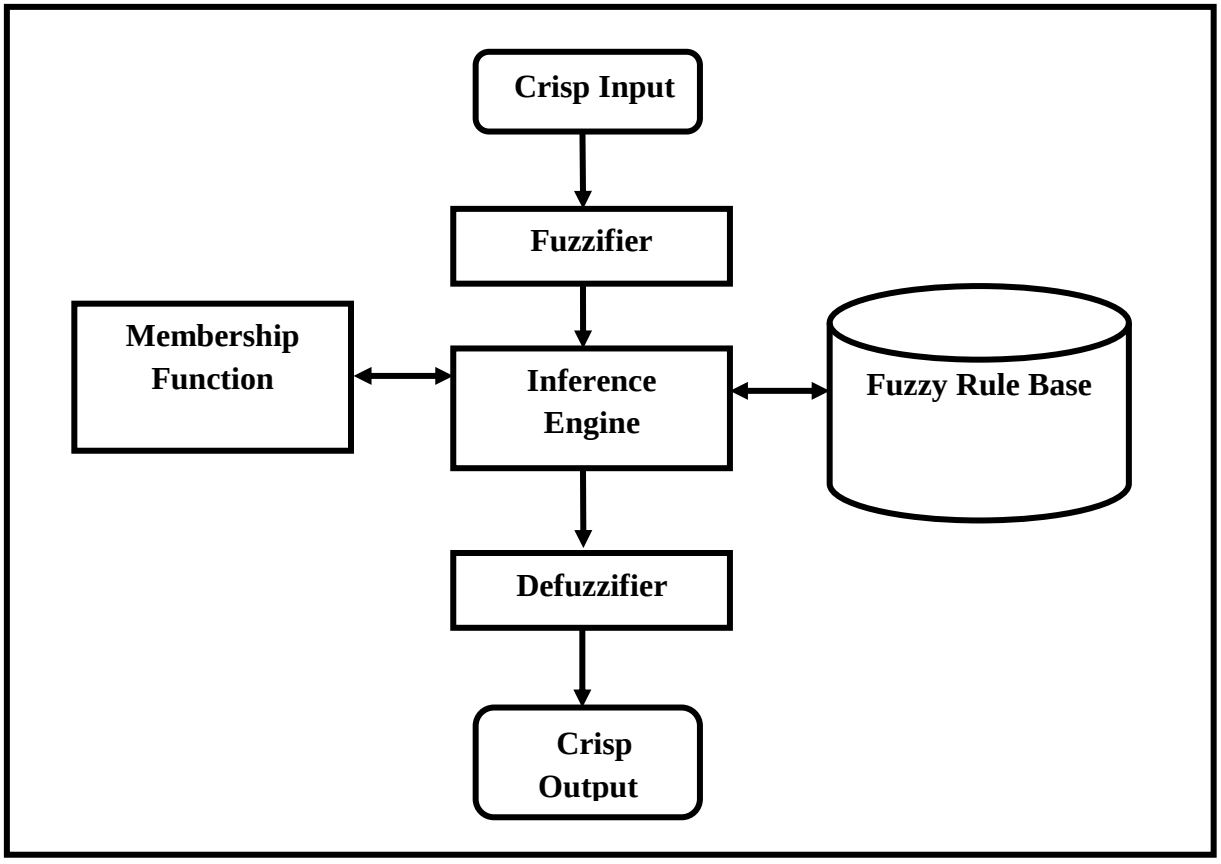


Figure 1.5: The Block Diagram Of Fuzzy Inference System

1.7.2 Membership Function

The membership function for a fuzzy set[12] corresponds to as an indicator function of the classical sets. It can be represented in the form of a curve(or graph) that defines how each point in

the given input space is mapped to a membership value between 0 and 1 or a degree of truth . Some of the common shape of a membership function are triangular , trapezoidal and bell curves . The input space is also referred to as the universe of discourse.

1.7.3 Crisp Set and Fuzzy Set

In a crisp set, an element is either a member of the set or not. For example, a jelly bean belongs in the class of food known as candy. Mashed potatoes do not. Fuzzy sets, on the other hand, allow elements to be partially in a set. Each element is given a degree of membership in a set.

1.8 Neural Network

Artificial Neural Networks (ANNs)[13] as shown in figure are inspired by the working of human brain .This biologically inspired system is the combination of neurons and synaptic connections among them ,which are capable of transferring the data via many layers. Here neuron is the most basic part of network which receives the inputs and after processing that input it generates the output. Because of its property of learning it has the adaptive nature it means that depending on the information running through it this complex system can change its internal structure .Change in internal structure is brought by adjusting the weights of the connections. As shown in the diagram below, every line between two neurons represents a connection and shows the path for the flow of information. Each connection is associated with weight. This weight controls the communication between the two connected neurons. If this network generates a expected output at its output layer then there is no any need for adjusting the weights. But if the network does generate erroneous output then the system adapts by modifying the weights in order to better the subsequent results.

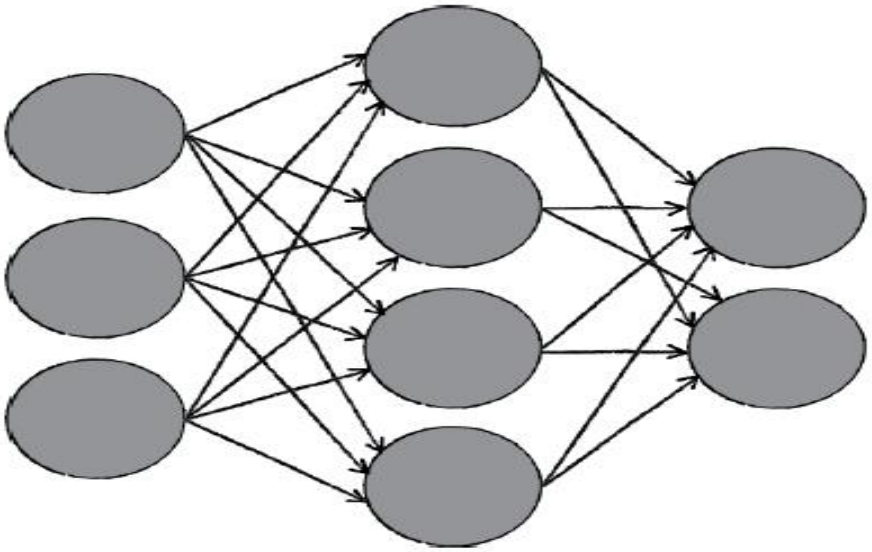


Figure 1.6: The architecture of artificial Neural network

1.8.1 Learning's In Neural Network

There are many type of strategies for learning of network, some of them are as following-

1.8.1.1 Supervised Learning

For this kind of learning[14] we have a dataset of input/output pairs, which is also known as the training set or master. In this learning process for every input of the network output is generated and this output is compared with consists in updating the weights at training set the network output is generated and this generated output is compared with known desired output and depending on the error we adjust the connections of the network.

1.8.1.2 Unsupervised Learning

This type of learning is used when there is no availability of training set or master. So unlike supervised learning there is no feedback is provided for the correctness of the mapping. Here In this Unsupervised learning the network should be able to detect by itself any common behavior and feature presented in the data set. Networks are stronger enough to cluster the data based on the pattern of the data. Here clustering, means dividing data set of elements into various groups according to some unknown pattern.

.

1.8.1.3 Reinforcement Learning

Here unlike supervised learning the value of training set input/output pairs is not a metrics of the difference between the generated and the desired value, but rather an evaluation of the result as 'wrong' or 'right' result.

1.9 Architecture Of Neural Networks

An ANN is a data processing system that consists of large number of highly Interconnected artificial neuron(processing elements) by connecting links having weights associated to .

1.9.1 Single Layer Feed-Forward Perceptron

In this single layer feed-forward network the inputs are directly connected to the outputs , via a series of weights. In this way it can be considered the simplest kind of feed-forward network.

1.9.2 Multi Layer Feed-Forward Perceptron

This type of network consists of multiple layers of processing units, which are interconnected in a feed-forward way. Every neuron in one layer is directed connected to the neurons of the subsequent layer. Along with input and output layers this very architecture also consists of one or more hidden layers in its network.

1.9.3 Recurrent Network

It is a feed forward neural network which has having one or more hidden layers but it also has at least one feedback.

1.4 Organization Of Thesis

I have organized the thesis in the following order

Chapter 1 provides an overview of Cloud Computing and the terminologies associated with it. This chapter explains the Cloud technology in an elaborated way, its implementation details, major challenges and benefits of the technology to the IT industry. Various important aspects of the Cloud

Computing are discussed here. After reading this chapter you can have a better idea about this technology.

Chapter 2 provides an overview of Scheduling and the terminologies associated with it. It also gives information regarding various scheduling algorithms that are used in cloud computing systems, to balance the overall workload among the nodes in the system. Discussion on various performance metrics for each of the scheduling algorithm in cloud computing environment is done to have better idea about these algorithms.

Chapter 3 describes the proposed approach that is being implemented the in cloud computing systems. This Chapter explains the proposed classification and cloudlet scheduling technique which should be adopted to have better performance of the system.

Chapter 4 Shows the results that are obtained by implementing the proposed approach and by making certain observations. This chapter justifies the proposed approach in contrast to existing algorithms.

Finally, **Chapter 5** concludes the thesis by discussing the overall contribution of the research in the context of related work in the area.

CHAPTER 2

TASK SCHEDULING

2.1 Introduction

Task scheduling is one of the crucial issues in cloud computing environment where various tasks are competing together for getting executed. For this purpose various tasks scheduling algorithms[17] are designed so that each task gets its fair chance for execution. Resources are distributed among many hosts in the distributed environment of Cloud computing . Hence , Efficient utilization of resources, higher throughput, and lower power are the major factors that must be kept in mind while we are designing a task scheduling algorithm. In the computer science field Job scheduling is a combinatorial problem in nature where ideal jobs are allocated to available resource at a particular moment of time **.** In cloud computing environment, a task scheduling algorithm can is applicable on different levels. Firstly it is applied at level of broker where a broker has to decide which cloudlet should be given to which virtual machine and secondly it is be applied at virtual machine in which a virtual machine decides on which host a task will run .In our scheduling algorithm, policy is applied at broker level i.e. between tasks and virtual machines.

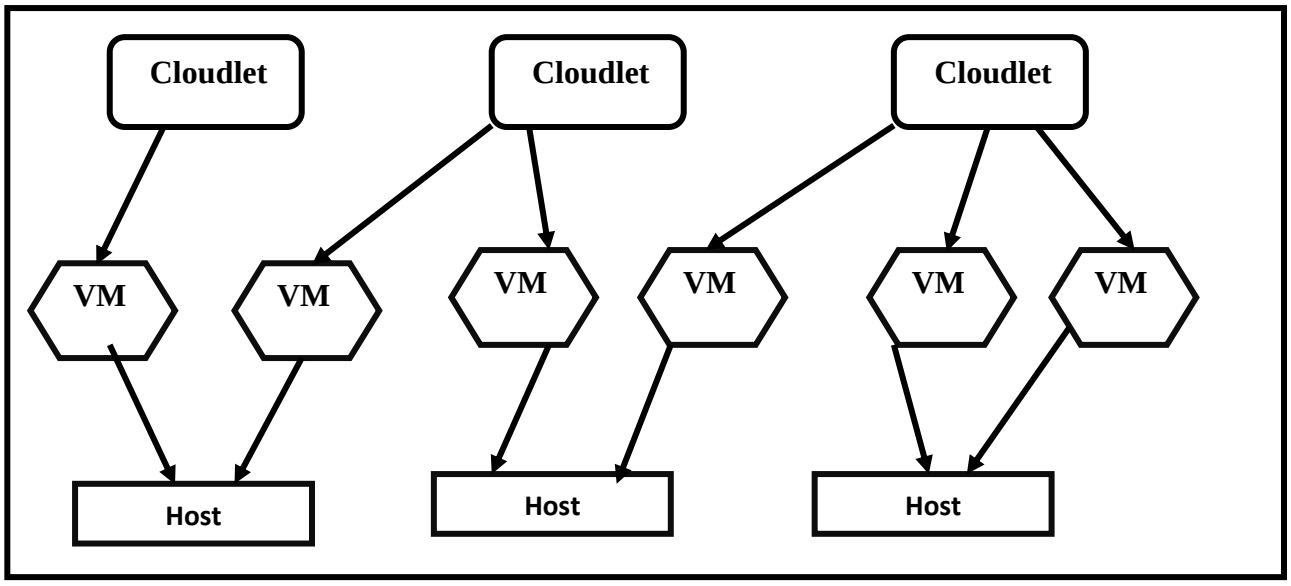


Figure 1.6: Levels of scheduling in cloud Computing Scenario

2.2 Classification Of Various Scheduling Techniques

2.2.1 Preemptive/ Non-preemptive Techniques

Here the scheduler can stop or postpone execution of any job in case of preemptive scheduling algorithm. But in of non-preemptive algorithms it is not possible till job completion or running job voluntarily give up resources.

2.2.2 Static Scheduling v/s Dynamic Scheduling

In static scheduling algorithm information about status of the task and resources is advanced and then schedule the task to the resources. In dynamic scheduling algorithm in which task is allocated to runtime.

2.2.3 Online v/s Batch Mode Scheduling

In online scheduling algorithm when job is arriving it is executed. In batch mode scheduling is called offline scheduling in which task is scheduled in batch mode means task is executed in specific time interval.

2.3 Process Of Scheduling

The Process of scheduling involves following phases:

2.3.1 Discovery And Filtering Of Resources

Every datacenter is registered with a datacenter broker and datacenter transfers all the resource related information to this datacenter broker. On arrival of jobs to the broker site, broker first checks the current status of the resources that are requested from the datacenter and then filters out the resource that is free and in active state.

2.3.2 Resource Selection

Resource availability and resource scheduling policy decide how the resource are selected from the resource pool and assigned to the tasks.

2.3.3 Submission Of Tasks

After mapping of jobs ,they are sent to their respective allocated resource.

2.4 Various Types Of Scheduling Algorithms

There are various types of cloudlets scheduling algorithms [5, 18] which are used and they are explained as following one by one:

2.4.1 First Come First Serve

As the name implies, the task which arrives first gets the resource first. Initially, all the tasks are collected in a buffer. Their arrival times are also stored and as the resource becomes available they are submitted to it based on who arrives first. Advantage of this algorithm is that it is simple to implement and fast result. Main drawback of this algorithm is urgent tasks has to wait for a longer period of time and also less utilization of resources.

2.4.2 Round Robin Scheduling Algorithm

In this algorithm, time is sliced out into fixed intervals known as time-slice or quantum. All the jobs are stored in a buffer and are served in a FIFO order but for a specific amount of time. If the time quantum expires then the current job is preempted and store in a waiting buffer and the resource is allocated to next task. Advantage of round robin is that response time of the system decreases.

2.4.3 Priority Based Scheduling

In it priority is first assigned to the tasks. Priority determination is based on some pre determined parameters. Arrived tasks are stored in a priority queue along with their priorities. Task with highest priority has been taken out from the queue and submitted to the resource. Starvation is the major disadvantage of priority based scheduling in which a lower priority task may never get a chance for the resource.

2.4.4 Min-Min Scheduling Algorithm

In this algorithm, a task with minimum amount of time required of a resource, mainly processor is scheduled to a processor with minimum processing power. In this way smaller jobs are scheduled first, therefore there is a possibility for starvation for longer jobs.

2.4.5 Max-Min Scheduling Algorithm

In this algorithm, a task with max amount of time required of a resource (CPU) is scheduled to a processor with minimum amount of processing power. In this way longer jobs are scheduled first, therefore there is a possibility for starvation for smaller jobs.

2.4.6 Most Fit Task Scheduling Algorithm

In this algorithm, a task with high fitness value is selected from the task pool and submitted to the required resource. Experimentally it has been shown that make-span is high in this case.

2.4.7 Cloud Task Scheduling Based on Ant Colony Optimization

In this paper [6], cloud task scheduling is viewed as an NP complete problem and an ant colony optimization is used for scheduling tasks to virtual machines. The main goal of the algorithms is minimizing the make span of a given tasks set. Experimental results showed that the ant colony optimization outperformed FCFS and round robin algorithms.

2.4.8 Scheduling Using Improved Genetic Algorithm in Cloud Computing for Independent Tasks

In this paper [7], a modification to a genetic algorithm is proposed for scheduling tasks to Virtual Machines. The algorithm was compared against the traditional genetic algorithm and performed better in terms of the make span for resource utilization.

2.4.9 Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual machine

In this paper, a novel VM assign algorithm is presented which allocates incoming jobs to available virtual machines. Here the virtual machine assigned depending on its load i.e. VM with least request is found and then new request is allotted. With this algorithm underutilization of the virtual machine is improved significantly and later it is compared with existing ActiveVM algorithm.

2.4.10 Particle Swarm Optimization Algorithm

Particle swarm optimization is used to minimize the total cost of execution of application workflows on Cloud computing environments. It calculates the total cost of execution by varying the communication cost between resources and the execution cost of computing resources.

2.5 Various Criteria For Selecting Scheduling Algorithms

Various criteria for scheduling are max CPU utilization, Max throughput, Min turnaround time, Min waiting time and Min response time. Throughput is the number of processes that complete their execution per time unit. Turnaround time is the amount of time to execute a particular process, which is the interval of time of submission of a process to the time of completion. Waiting time is the sum of the periods spent waiting in the ready queue. Response time is the time from the submission of a request until the first response is produced . Scheduling parameters considered for the comparison of existing techniques is briefly explained as follows:

2.5.1 Makespan Time

It can be defined as the time spent from the beginning of the first task in a job to the end of the last task of the job. Makespan is the one of the most popular measurements of scheduling algorithms and few papers given in the following discussion adopt it.

2.5.2 Fairness

Fairness measures or metrics are used in network engineering to determine whether users or applications are receiving a fair share of system resources. Few authors consider fairness as the parameter for job scheduling. Fair resource allocation plays a significant role in scheduling of jobs in order to achieve user's satisfaction and get higher benefits.

2.5.3 Optimization

It is defined as finding an alternative with the most cost effective or highest achievable performance under the given constraints by maximizing desired factors and minimizing the undesired ones. Few papers which consider optimization focus on maximizing the resource utilization as well as system throughput while minimizing the cost.

2.5.4 Energy Consumption

IT is the amount of energy used by a process or system. Few authors have discussed about energy consumption. Green cloud computing has emerged as a new paradigm to reduce energy consumption and thereby increase the efficiency. Energy efficient data centers, also known as green data centers are the main focus of most researchers in current day scenario.

2.5.5 Load balancing

It can be defined as a computer networking method for distributing workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. It aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of anyone of the resources. It also reduces energy consumption by migrating the jobs between under-utilized and over utilized machines.

2.5.6 Deadline

In general, deadline can be defined as the latest time by which something has to be completed. In context to cloud computing, meeting the user's requirements, QoS and also SLA within the specified time comes under deadline. Few papers discuss about deadline aware scheduling algorithms.

2.5.7 Execution time

The execution time of a given task is defined as the time spent by the system executing that task. In cloud computing scenario, it can be considered as the time taken to process user's request and give services. Considerable amount of work has been done by many authors to reduce execution time of the jobs.

CHAPTER 3

PROPOSED METHOD

3.1 Introduction

The fast growth in cloud computing model has been forcing the need of enhancing the existing cloudlets scheduling, management and execution algorithms in cloud data centers as well as proposing new ones. As We saw in the previous chapter there are a big number of scheduling techniques have been suggested by various researchers, each having its own pros and cons .Some algorithms emphasize on resource utilization , some on having the minimum execution time , some on having the minimum waiting time and some emphasizing on setting a trade-off between different parameters. So considering all the factors with this presented work the main aim is to let the cloudlet to be run on its best fit virtual machine and while scheduling those cloudlets on virtual machines cloudlet should not unnecessarily wait .The proposed system has been depicted in following comprehensive block diagram shown in the Figure 2.1.

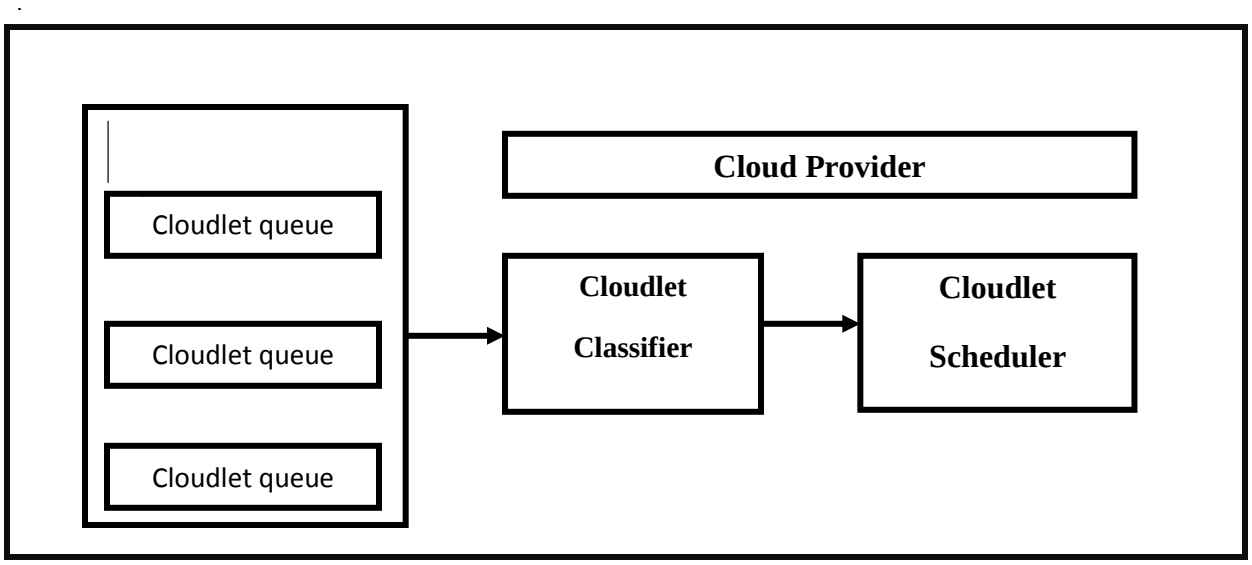


Figure 3.1: The Block diagram of the proposed system

As shown in the above figure , Cloud users will submit the list of cloudlets to the system with the objective of getting them executed . Cloud provider has two Segments as *Cloudlet Classifier* and *Cloudlet Scheduler*. In first segment there is a classifier[19] which is based on neural network model. It classifies and predict the appropriate class for the submitted cloudlets. Second segment consists of fuzzy based cloudlet scheduler which has the responsibility of scheduling the classified cloudlets on the virtual machines based on the dynamic fuzzy priorities.

3.2 Segment 1

This model is suggested to classify the assigned cloudlets into predefined categories depending on their common behavior and pattern. In this model we have a neural network based system that identifies common patterns of the jobs submitted to the cloud .Cloudlet classification is done when they are submitted to the cloud for processing .

3.2.1 Decided Categories For Cloudlets

This classification is done by keeping an objective of higher intra class cloudlets similarity and lower inter class cloudlets similarity. It means cloudlets of same class will have similar properties and resource requirements. This neuron based model not only used for classifying a batch of cloudlets into above 5 categories but it can also be used as a predictor too using that we can make many predictions like- a particular submitted cloudlet is of which type and what type of virtual machine should be assigned to this cloudlet . It means by this system we can predict the best fit virtual machine for cloudlets. .This strategy of classifying the jobs leads to less resource wastage and higher energy efficiency .This neural based classifier figures out the properties of the cloudlet submitted and then it categories the cloudlets in to 5 classes namely-

- Class 1: Foremost Task
- Class 2: Processing Concentrated (PC) Task
- Class 3: Data Concentrated (DC) Task
- Class 4: PC-DC Task
- Class 5: Normal Task

Priority of cloudlets is assumed to be on the scale of 10. *Foremost task* class consists of critical cloudlets it means the priorities of these cloudlets are relatively quite higher (let us say [9, 10]) and they are not supposed to be waited for long and they should be executed before any other cloudlets.

Processing Concentrated (PC) cloudlets are those cloudlets which demands higher computing resources such as fast processers and fast ram. *Data Concentrated (DC)* cloudlets are those cloudlets which demands higher bandwidth and secondary storage resources. *Processing Concentrated- Data Concentrated (PC-DC)* class has those cloudlets which demand both higher computing resources and higher bandwidth and secondary storage resources. *Normal class* contains those cloudlets which have demand for average capacity virtual machines.

Here the cloudlets are assumed to be independent of each other means there is no any kind of dependency between the cloudlets .It is also assumed there is no any restriction or the limitation on the number and type of resources availability given by cloud provider.

3.2.2 Neural Net Classifier Architecture

The Neural network based classifier model is shown in the below Figure 2.2 .The cloud users submit the cloudlets to the cloud provider for the purpose of getting them executed. The cloudlets are typically submitted with some attributes. Here each submitted cloudlets will have some characteristics associated with them.

The cloudlets will have attributes as Priority (p_i), Instruction count(IC_i) , Deadline(DL_i), Bandwidth(BI_i), and Storage(ST_i) . Here subscript 'i' denotes the i^{th} cloudlet among all the submitted cloudlets. The cloud Provider receives the list of cloudlets ,and executes them at a specific data center, and then sends the results back to the cloud users.

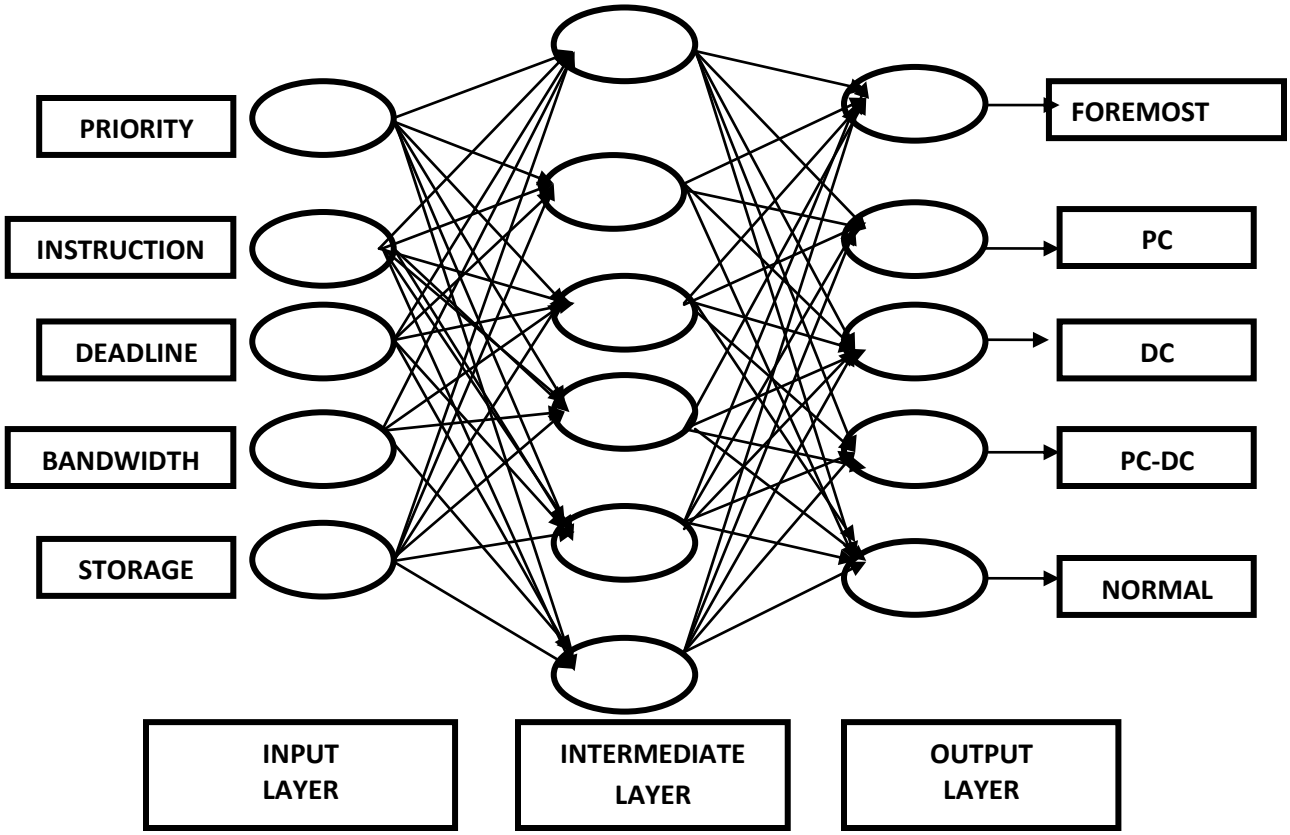


Figure 3.2 : The model of neural network used in proposed work

Here this artificial neural network (ANN) based system is able to learn how to categorize the cloudlets into above 5 classes . Here the ANN has the capacity to learn relationship among various variables of the system based on this learning of relationship it is possible for this network to classify the cloudlets .As shown in the figure the ANN receives the list of cloudlets with above 5 inputs and based on the learning it classify them into 5 categories as shown in output .

For the network development MultiLayer Perceptron (MLP) is chosen .It uses hyperbolic tangent activation function in the hidden layer and linear function in the output layer , plus bias in these two layers .For the training purpose the back propagation algorithm is taken. For this back propagation step we need to know the actual correct output for a given input so that we can calculate the output error using that we can train the system.

A training file is made using that a series of experiment is conducted to train the above neural system and by this training it learns the behavior of the system and once it is trained the system is used for actual testing purposes.

The classifier acts as an interface between users and cloud infrastructure. It analyses the resource requirements of the submitted job and decides which category the cloudlet belongs to. We can also say that It indirectly gives the hint about the best suited virtual machine for running that cloudlet so that energy consumption can be reduced. The result of this classification is a Log File (LF) which is maintained for each user and this file is used in next scheduling phase . LF is considered as a list of records of the form-

Cloudlet list (**Cloudlet_id, Cloudlet_type, Date, Userid**)

3.2.3 Construction Of Data Set:

For building the data set for the training of the neural based system various threshold values are decided by the system for the attributes of the cloudlets. Based on these threshold values the decision about the class of cloudlets is taken .These Thresholds are selected by examining real time workload. The indications of the threshold values for various attributes of cloudlet are as following-

pTh: Threshold value for the priority of cloudlets
dlTh: Threshold value for deadline
mipsTh: Threshold value for cloudlet mips requirement
biTh: Threshold value for bandwidth
stTh: Threshold value for storage

Rule for the classification are shown in table 3.1 .Here "Yes" value of an attribute indicates that value of that attribute is greater than value of Threshold decided for that attribute. "No" value of an attribute indicates that value of that attribute is lesser than value of Threshold decided for that attribute . Likewise "Don't care" value of an attribute indicates the value of attribute doesn't matter whether it is greater or lesser than threshold. Here "1" indicates depending on the attribute value the cloudlet will belong to that class only exclusively.

Here by the Rule table it is visible that Larger the Million Instruction Execution Per Second(MIPS) requirement of cloudlet and lesser the Deadline time of cloudlet , this will make cloudlet Processing concentrated(PC) hence speedy and fast physical machine will be needed in this scenario. The Mips requirement of cloudlet is calculated by its counting its line of code and expected execution time. We can also observe that higher Bandwidth and Storage requirement will make cloudlet Data concentrated(DC).Hence for DC cloudlets we will require virtual machines which have more bandwidth and storage capacity and there is no constrained over its processing power. There will certainly be some cloudlet which will have highest priorities and we don't want them to be waiting so we make a new class for them and have given name to this class as Foremost. The classifier is trained using 500 row based dataset .

PRIORITY	MIPS REQUIREMENT	DEADLINE	BANDWIDTH	STORAGE	PC	DC	PC-DC	NORMAL	FOREMOST
No	Yes	Yes	Yes	Yes			1		
No	Yes	Yes	Yes	No			1		
No	No	Yes	Yes	Yes			1		
No	Yes	No	Yes	No			1		
No	No	Yes	No	Yes			1		
No	Yes	Yes	No	No	1				
No	Yes	Don't care	No	No	1				
No	Don't care	Yes	No	NO	1				
No	No	No	Yes	Yes		1			
No	No	No	Don't care	Yes		1			
No	No	No	Yes	Don't care		1			
No	No	Yes	Yes	No			1		
No	Yes	No	No	Yes			1		
No	No	No	No	No				1	
Yes	Don't care	Don't care	Don't care	Don't care					1

Table 3.1 : Rule Table For Classification Of Cloudlets

3.3 Segment 2

In this segment we will be scheduling the classified cloudlets(resulted from 1st segment) onto their best fit virtual machines using the new fuzzy priority based algorithm . The fuzzifier module(as shown in the Figure 2.3) of this segment takes cloudlet size and its user defined priority as input and then it converts into new fuzzy based priority which will be further used for scheduling the cloudlets onto the virtual machines.

3.3.1 Fuzzy Priority Based Decision

After application of fuzzy rules (that are defined by us in knowledge base) we shall have a new priority of the cloudlets and that new priority will decide the execution sequence of cloudlets. The block diagram of model is as following figure. For the sake of understanding we have shown the working of the fuzzifier by a simple example in table. The Fuzzy rules are deciding the new priority .These Fuzzy rules are being shown in detail the next upcoming chapter.

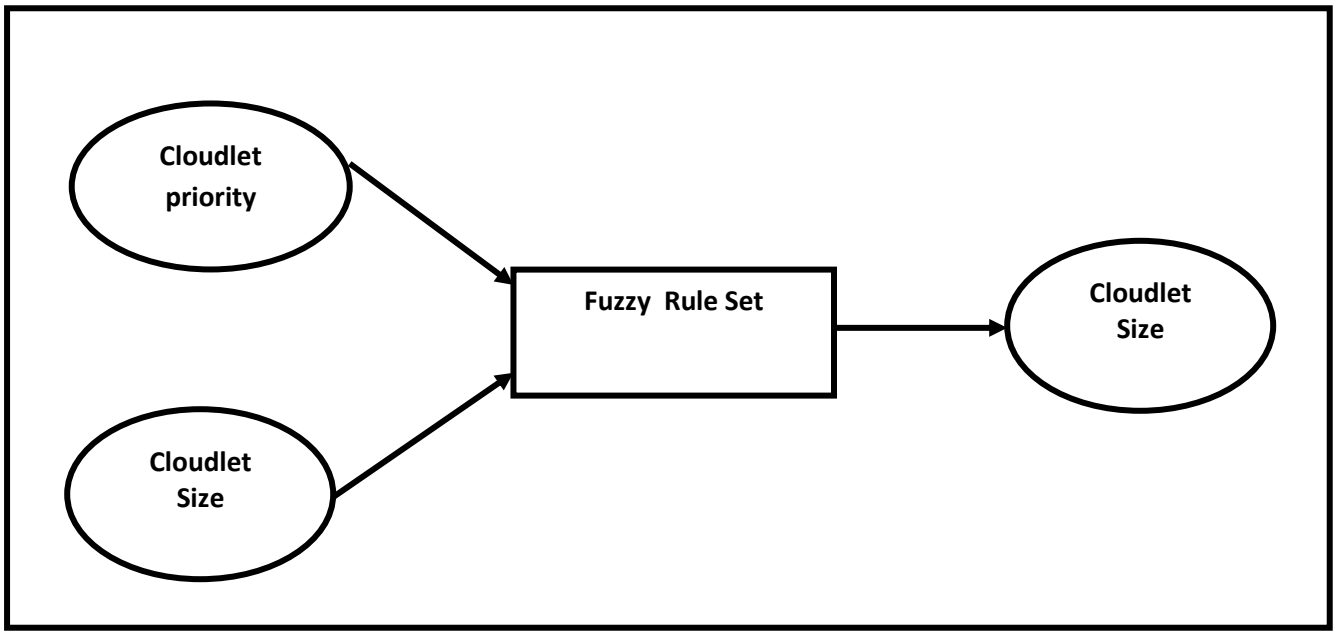


Figure 3.3 : Block Diagram Of Fuzzifier Used In The System

After running the above fuzzifier module we get fuzzified priorities (as shown in table 3.2). we have eight cloudlets [C1...C8] . For each cloudlets pre assigned priorities and size are shown in the second row and third row of the table respectively. After the application of fuzzifier the we get the new fuzzy based priorities as shown in fourth row. This table demonstrates that if the cloudlet has considerably high length and moderate priority then that cloudlet must wait a little so that shorter cloudlets don't have to wait much hence average waiting time goes down.

Cloudlets	C1	C2	C3	C4	C5	C6	C7	C8
pre assigned priority	2.2	4.5	5.5	2.5	8.6	2.5	1.6	4.8
Size(Line of Code)	5200	2543	4987	7123	9987	11342	7623	1287
Fuzzy Rule Priority	3.60	7.37	6.00	1.90	2.02	0.88	1.99	7.86

Table 3.2 : Application Of Fuzzifier Module

3.3.2 The Algorithm For Scheduling Sub Segment Of The System

1> For each type of virtual machine in the system ,For each cloudlet in the cloudlet list the fuzzifier fetch the parameter like cloudlet size, priority and arrival time, waiting time etc.

2>The fuzzifier module will take the cloudlet size and cloudlet pre_assigned priority and its waiting time as an inputs and will generate the fuzzy based new priority for that cloudlet as an output.

3>Sorting module will sort the cloudlet list based on their new fuzzy priority.

4>Resource allocation module will allocate the cloudlets to the virtual machine based on their new priorities.

5>If a fresh cloudlet arrives go to step 1.

7>Calculate the average waiting time and average turnaround of cloudlets.

3.4 Flow Diagram Of The Proposed System

Here figure 2.3 shows the comprehensive flow diagram for the proposed system. It depicts all the component of the system and how they interact with each other.

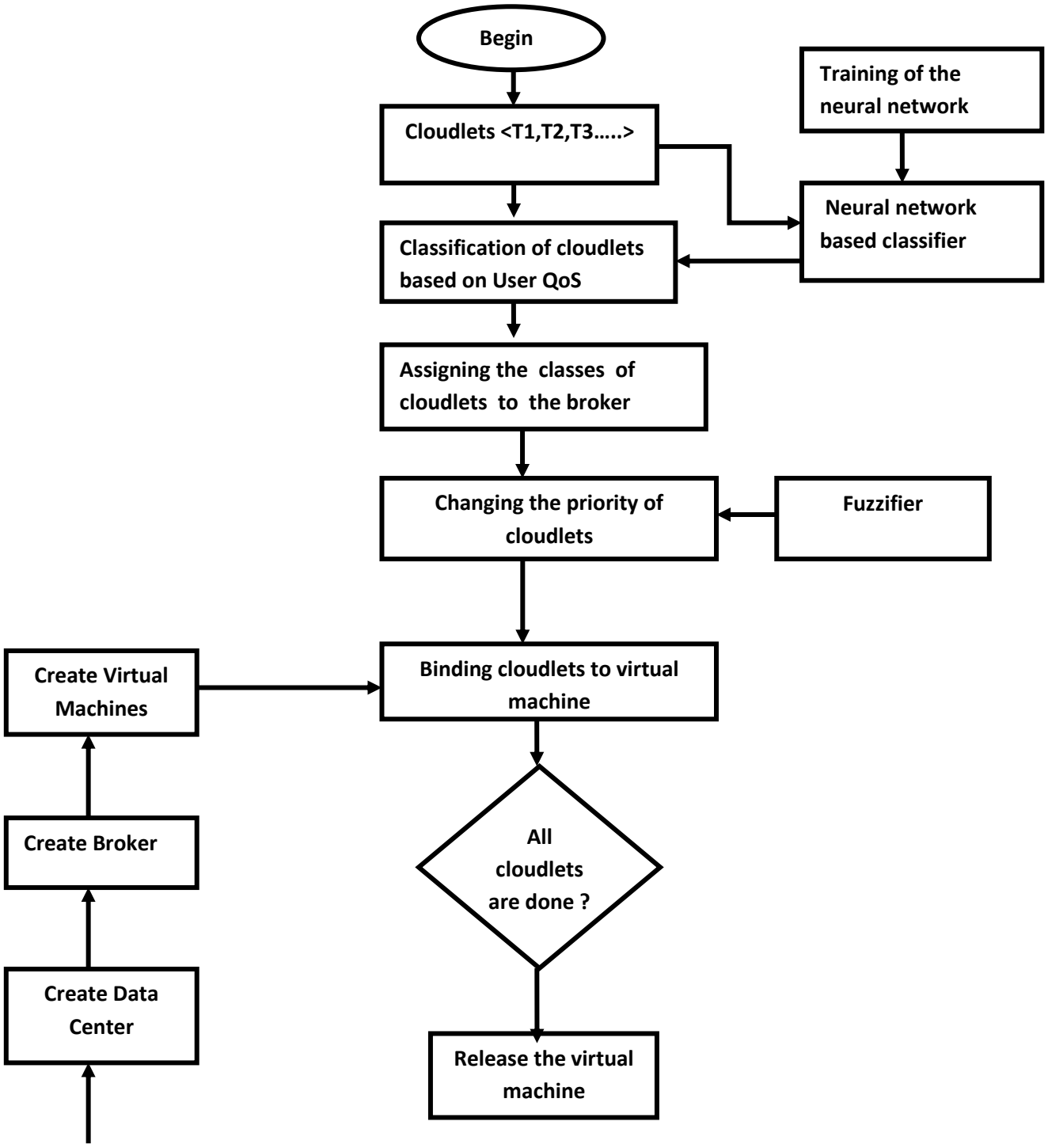


Figure 3.4 : Flow Diagram of Proposed Algorithm

CHAPTER 4

RESULTS AND IMPLEMENTATION

4.1 Introduction

As we discussed in previous chapters, choosing a right scheduling technique for a system is one of the major tasks for developers. With this proposed methodology Our the main aim is to make the system efficient by making improvement in selection of virtual machine for the cloudlet execution and reducing the average waiting the of cloudlets.

In this research work our aim is to better the performance of cloud computing system. For the evaluation of performance of our proposed method we are using the average waiting time of cloudlets .We did comparison of this present work with some existing algorithms.

4.2 Used Technologies

4.2.1 Java

Java [8] has many attractive properties which made it so popular among its users. These are various features that help developers to deploy and develop their many applications and services in Cloud computing environment. In cloud computing a platform is provided with the a help of that a user can interact with the cloud services and further using as per the needs. This platform should be accessible from any device whether it is a laptop, a mobile, a tablet, or any other device. Java's great ability to compile once and run on any platform (that has java virtual machine installed in it) makes this possible. Java's byte code makes it more secure and portable for being used. Many computational tasks performed in cloud network require fast and smooth networking services. Java has many networking features in it which are used in cloud computing network. One another popular features of Java is applets which have revolutionized the web and the internet technology .While implementing the proposed methodology java is been used intensively in the project.

4.2.2 Cloudsim

Cloudsim [4] is a simulation toolkit and it is designed to simulate the cloud computing scenario .This is a generalized framework which enables seamless simulation, modeling and experimentation of new cloud computing application and infrastructure services . Cloudsim 3.03 is the latest release and it is built in JAVA programming language. I used eclipse a JAVA based Integration Development Environment (IDE) for this experiment . Cloudsim provides repeatable controllable and environment for the simulation of cloud. It has various entities such datacenter, hosts, processing units, broker, virtual machines, cloudlet etc which are used to emulate real cloud entities.

4.2.2.1 Cloudsim Architecture

The architecture is shown in the following figure 4.1 . The detail of each section is given as following:

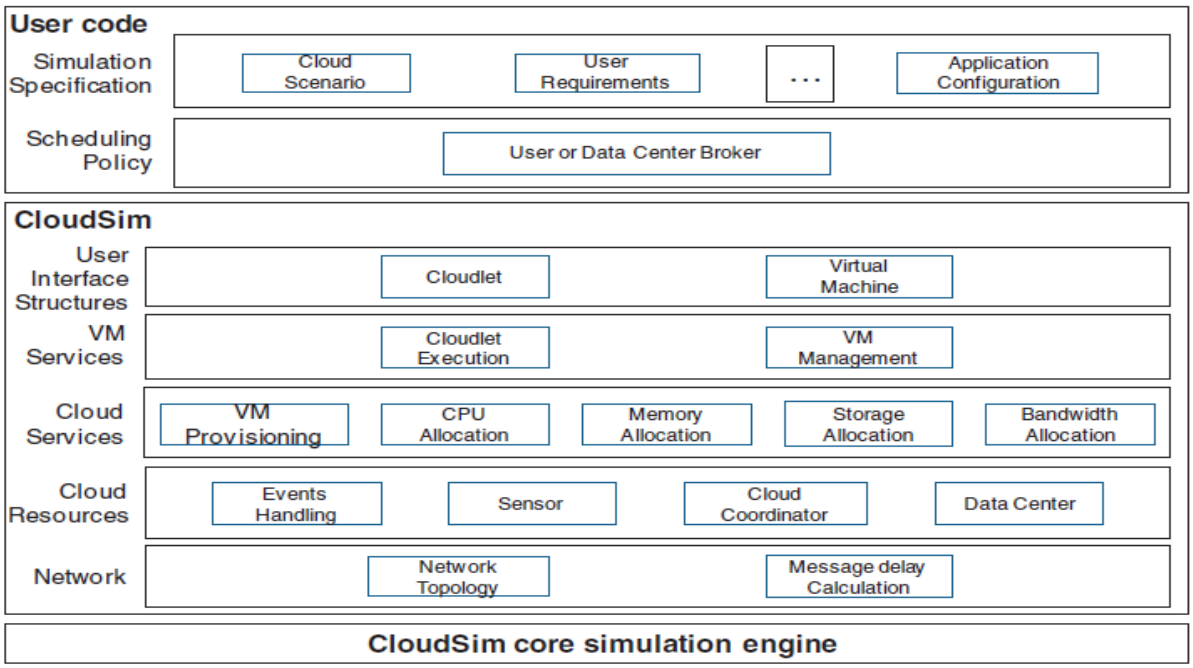


Figure 4.1 : The Architecture Of Cloudsim Tool

4.2.2.1.1 User Interface Structure

4.2.2.1.1.1 Cloudlet

This class provides the cloud based application services (such as content delivery, social networking & business workflow). Every cloudlet has a pre-assigned instruction length and data transfer (both input and output fetches) which it needs to undertake during its life cycle.

4.2.2.1.1.2 Virtual machine

This class provides a virtual machine, which is managed by cloud component host . The virtual machine has specifications as accessible memory, processor, storage size, and the VM's internal provisioning policy is from an abstract component called the CloudletScheduler.

4.2.2.1.2 VM Services

4.2.2.1.2 .1 Cloudlet Execution

This is Used to support modelling of other composition and performance metrics for applications such as transactions in data-base oriented applications.

4.2.2.1.2.2 VM Management

The VM management is used for the policies related to Virtual Machine life cycle such as allocation of a host to VM, VM creation, VM Migration and VM destruction .

4.2.2.1.3 Cloud Services

4.2.2.1.3.1 Vm Provisioning

The basic functionality of this component is to select an available host in the data center that meets the memory, storage, and availability requirement for Virtual machine and allocate to it.

4.2.2.1.3.2 Cpu allocation

It refers to the allocation of cpu to each of the services.

4.2.2.1.3.3 Memory allocation

It refers to the management of primary memory of the host.

4.2.2.1.3.4 Storage allocation

It refers to the management of secondary memory of the host.

4.2.2.1.3.5 Bandwidth allocation

This is class that models the policy for allocation and provisioning of bandwidth to VMs.

4.2.2.1.4 Cloud Resources

4.2.2.1.4.1 Events Handling

This is responsible for managing and processing event queues and controlling step-by-step (sequential) execution of simulation events..

4.2.2.1.4.2 Cloud Coordinator

It is responsible for periodically monitoring the internal state of data centre resources and based on that it undertakes dynamic load-shedding decisions.

4.2.2.1.4.3 Sensor

This interface must be implemented to instantiate a sensor component that can be used by a Cloud Coordinator for monitoring specific performance parameters such as energy-consumption, resource utilization.

4.2.2.1.4 .4 Data centre

This class models provides the core infrastructure level services (hardware) that are actually offered by Cloud providers . It manages a set of computing hosts that can
either be homogeneous or heterogeneous that in turn manage virtual machines.

4.2.2.1.5 Network

4.2.2.1.5.1Network Topology

This class has the information for bringing network behaviour (latencies) in the simulation experiment.

4.2.2.1.5.2 Message delay calculation

This class contains the procedures that calculate the delay that occurred in the delivery of the message

4.2.3 Classes In The Cloudsim Tool

In this section, we are providing the finer details related to the fundamental classes that are used in CloudSim, which are also known as building blocks of the simulator. The overall Class design diagram for CloudSim is shown in Figure 4.2.

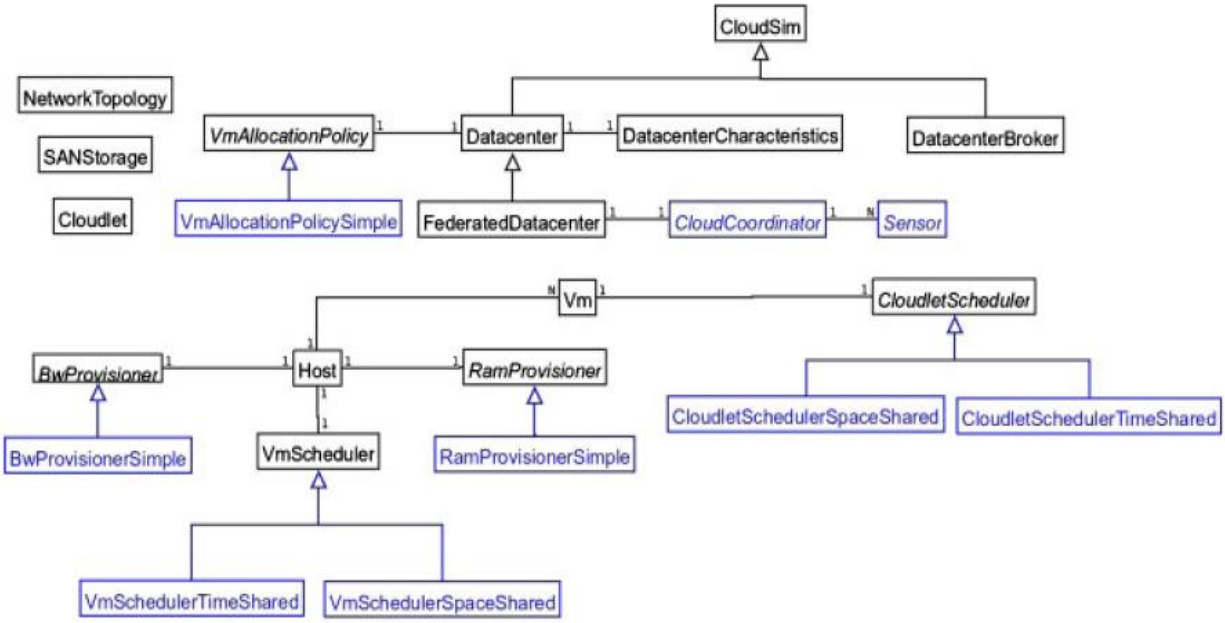


Figure 4.2 : The Class Diagram Of Cloudsim Tool

4.2.2.1 BwProvisioner

This class has the policy for provisioning of bandwidth to VMs. The BwProvisioningSimple allows a VM to reserve as much bandwidth as required by it; Although it is constrained by the total available bandwidth of the host.

4.2.2.2 CloudletScheduler

This class is extended by the implementation of different policies that determine how to allocate and schedule various available cloudlets to the virtual machine. By default 2 types of provisioning policies are offered as

- Space-shared (CloudetSchedulerSpaceShared)
- Time-shared (CloudletSchedulerTimeShared).

4.2.2.3 Datacenter Broker or Cloud Broker

This class provides a broker, which is responsible for mediating negotiations between Cloud user and Cloud providers. It discovers suitable Cloud service providers (CSP) by querying the CIS and undertakes online negotiations for allocation of resources/services that can meet the application's QoS needs

4.2.2.4 Datacenter Characteristics

This class contains configuration related information of data center resources.

4.2.2.5 Host

physical resource such as a computing resource or storage server are being provided by this class. Many information such as the amount of ram and storage, a list and type of processing cores (to represent a multi-core machine), an allocation of policy for sharing the processing power among virtual machines, and policies for allocating memory and bandwidth to the Virtual machines are encapsulated by this class.

4.2.2.6 RamProvisioner

The allocation policy of primary memory (RAM) to the virtual machine is represented by this class. The deployment and execution of virtual machine on a host possible only if the RamProvisioner component does approve that the host has the asked amount of free Ram memory.

.

4.2.2.7 SanStorage

This class provides a storage area network which is commonly ambient in Cloud-based data centers for storing huge chunks of data (such as Amazon S3, Azure blob storage). It does implement an interface that can be used to emulate storage and retrieval of any amount of data which is subject to the availability of network bandwidth.

4.2.2.8 Vm

This virtual machine component is hosted and managed by a Cloud host component. Every VM has some specification related to it like accessible memory, processor, storage size, and the VM's

internal allocation policy which is extended from an abstract component called as CloudletScheduler.

4.2.2.9 VmmAllocationPolicy

VM Monitor utilizes this abstract class for the allocation policies of VMs to hosts. One the basic functionality of the VmmAllocationPolicy is to choose the available host in a data center that fulfills the memory, storage, and availability requirement for a VM deployment.

4.2.2.10 VmScheduler

This is a class which is implemented by Host component of data center that provides the policies(space-shared, time-shared) required for allocating processor cores to virtual machine. To build the application-specific processor sharing policies the functionalities of this class can easily be overridden.

In the simulation of the proposed algorithm the implementation work and further comparison is done with the traditional approach of scheduling . Using CloudSim toolkit the algorithm is written in java and is run on Eclipse IDE platform.

4.2.2 JFUZZYLOGIC

JFuzzyLogic[9] is an open source software package. It is basically java based library for the fuzzy system development. It implements a complete Fuzzy inference system (FIS) in Fuzzy Control Language (FCL). JFuzzyLogic provides a fully functional and complete implementation of a Fuzzy Inference System (FIS), It offers a programming interface (API) using that we can code fuzzy control logic and use in our application . This library has the benefits of open source software and fuzzy system community which has several advantages. JFuzzyLogic's chief goal is to facilitate and accelerate the building of fuzzy systems. We get the above goal achieved by:

- Using standard programming language (FCL) that reduces the learning curves,
- Providing a fully functional and complete implementation of fuzzy inference system(FIS) ,
- Creating an API that developers could use and extend in their applications,

- Making the software independent to the platform,
- Making the software open source.

By all these benefits we can accelerate and smoothly develop and test fuzzy systems in both industrial and academic environments.

4.2.4 Neuroph

Neuroph[15][16] is a general Java neural network framework which provides a very easy manner to use neural networks in our Java programs thanks to its facility of Java class library along with an comfortable to use neural network API. A graphical user interface based editor for creating and training of neural networks is also come with.

4.3 Demonstration

We have demonstrated our proposed technique by the 2 experiments we conducted . The Experiment 1 is conducted using 20 cloudlets and Experiment 2is conducted using 100 cloudlets. The results are generated and compared with the priority based scheduling**.**

4.3.1 Experiment 1

In this experiment we are taking the real time input from the user in a file which consists of 20 cloudlets . Each cloudlet is associated with its QoS which includes its 5 attributes based on that its classification is done**.**

4.3.2.1 Input dataset Used

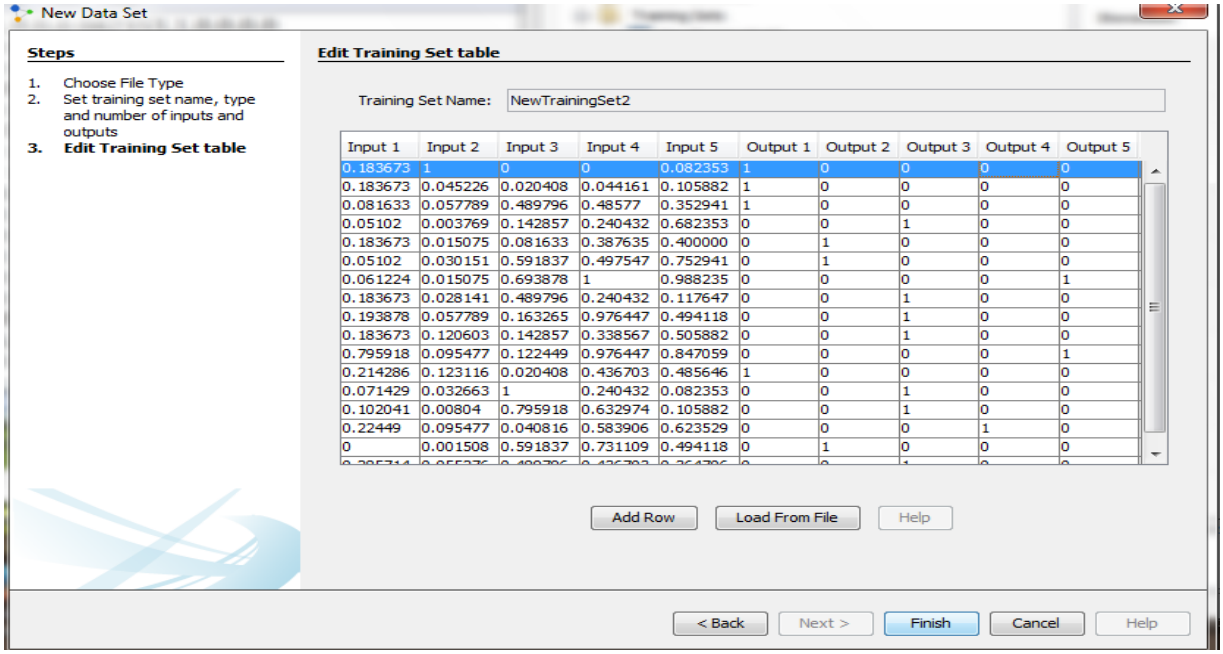
For the sake of better understanding the experiment 1 is conducted on a 20 cloudlet list. The input(not normalized) is provided to the system using the Comma Separated Values (CSV) file. The snapshot of this input file is shown in Screen Shot 4.1 as shown below.

```
@attribute instruction(mi) numeric
@attribute exe_time numeric
@attribute bandwidth numeric
@attribute storage numeric
@attribute priority numeric
20000,200,100,10,3.1
20000,4,500,800,4.0
4000,10,500,1200,5.6
15000,6,1500,900,5.7
2000,1.3,3000,900,8.8
1600,8,3000,100,2.5
10000,5,1800,800,7.0
12000,3,1600,990,7.5
32000,16,1200,1800,8.7
10800,6,2200,900,8.7
8000,4,3500,2048,2.3
8000,6,2300,900,1.2
18000,6,2000,800,3.1
5000,2.4,2300,900,9.7
4440,5,1800,500,2.4
39150,8.7,1500,2000,1.2
6000,2,1200,1000,2.4
10000,3.5,1500,1100,4.5
12000,4,5000,950,7.5
7000,3.9,2400,700,9.0
```

Screen Shot 4.1 : Input File For The Demonstration In CSV Format

4.3.1.1 Conducting The Experiment

The training of neural network based classifier is being done using approximately 500 row real time data set. The screen shot 4.2 below shows the application of input data set(normalized) for the objective of training of system . Here 5 inputs are the 5 attributes of cloudlets and 5 outputs are the classes in which the cloudlets are to be categorized. Now after this training the testing is done using our 20 cloudlets file. This cloudlet file is fed up into the trained classifier segment of the system..The result of testing data set of classifier component is applied to the cloudlet scheduler component of the system.



Input 1	Input 2	Input 3	Input 4	Input 5	Output 1	Output 2	Output 3	Output 4	Output 5
0.183673	1	0	0	0.082353	1	0	0	0	0
0.183673	0.045226	0.020408	0.044161	0.105882	1	0	0	0	0
0.081633	0.057789	0.489796	0.48577	0.352941	1	0	0	0	0
0.05102	0.003769	0.142857	0.240432	0.682353	0	0	1	0	0
0.183673	0.015075	0.081633	0.387635	0.400000	0	1	0	0	0
0.05102	0.030151	0.591837	0.497547	0.752941	0	1	0	0	0
0.061224	0.015075	0.693878	1	0.988235	0	0	0	0	1
0.183673	0.028141	0.489796	0.240432	0.117647	0	0	1	0	0
0.193878	0.057789	0.163265	0.976447	0.494118	0	0	1	0	0
0.183673	0.120603	0.142857	0.338567	0.505882	0	0	1	0	0
0.795918	0.095477	0.122449	0.976447	0.847059	0	0	0	0	1
0.214286	0.123116	0.020408	0.436703	0.485646	1	0	0	0	0
0.071429	0.032663	1	0.240432	0.082353	0	0	1	0	0
0.102041	0.00804	0.795918	0.632974	0.105882	0	0	1	0	0
0.22449	0.095477	0.040816	0.583906	0.623529	0	0	0	1	0
0	0.001508	0.591837	0.731109	0.494118	0	1	0	0	0

Screen Shot 4.2: Training Dataset Used By The Classifier

The fuzzy logic concept is being applied in our program using JFuzzyLogic java based library . for that we need to build a fuzzy control language(FCL) file. This file consists of the definition of membership and the fuzzy rules . Here the membership function is defined for each linguistic variable and the fuzzy rules data base decide the output when the input is provided.

We have built a FCL file dedicated to our experiments. The FCL file is inspired by the real world input. The content of this FCL file is as following:

```
FUNCTION_BLOCK result_occuppier

VAR_INPUT
  CLOUDLET_LENGTH : REAL;
  PRIORITY : REAL;
END_VAR
```

```
VAR_OUTPUT
  NEW_PRIORITY: REAL;
END_VAR

FUZZIFY CLOUDLET_LENGTH
TERM very_small := (0, 1) (4000, 0);
TERM small := (2000, 0) (4000,1) (6000,0);
TERM medium := (5000, 0) (7000,1) (9000,1) (10000,0);
TERM long := (9000, 0) (11000, 1)(13000, 1)(15000, 0);
TERM very_long := (14000, 0) (30000,1);
END_FUZZIFY

FUZZIFY PRIORITY
TERM very_low := (0,0) (1,1) (1.5,0);
TERM low := (1,0) (2,1) (3,0);
TERM medium :=(2.5,0) (4,1) (5,0);
TERM high := (5,0) (6,1) (7,0);
TERM very_high := (6.5,0) (10,1);
END_FUZZIFY

DEFUZZIFY NEW_PRIORITY
TERM very_low := (0,0) (1,1) (1.5,0);
TERM low := (1,0) (2,1) (3,0);
TERM medium :=(2.5,0) (4,1) (5,0);
TERM high := (5,0) (6,1) (7,0);
TERM very_high := (6.5,0)  (10,1);
METHOD : COA;
DEFAULT := 0;
END_DEFUZZIFY
```

RULEBLOCK No1

AND : MIN;

ACT : MIN;

ACCU : MAX;

RULE 1 : IF CLOUDLET_LENGTH IS very_small and PRIORITY IS very_low THEN NEW_PRIORITY IS very_high;

RULE 2 : IF CLOUDLET_LENGTH IS very_small and PRIORITY IS low THEN NEW_PRIORITY IS very_high;

RULE 3 : IF CLOUDLET_LENGTH IS very_small and PRIORITY IS medium THEN NEW_PRIORITY IS very_high;

RULE 4 : IF CLOUDLET_LENGTH IS very_small and PRIORITY IS high THEN NEW_PRIORITY IS very_high;

RULE 5 : IF CLOUDLET_LENGTH IS very_small and PRIORITY IS very_high THEN NEW_PRIORITY IS very_high;

RULE 6 : IF CLOUDLET_LENGTH IS small and PRIORITY IS very_low THEN NEW_PRIORITY IS medium;

RULE 7 : IF CLOUDLET_LENGTH IS small and PRIORITY IS low THEN NEW_PRIORITY IS medium;

RULE 8 : IF CLOUDLET_LENGTH IS small and PRIORITY IS medium THEN NEW_PRIORITY IS high;

RULE 9 : IF CLOUDLET_LENGTH IS small and PRIORITY IS high THEN NEW_PRIORITY IS high;

RULE 10 : IF CLOUDLET_LENGTH IS small and PRIORITY IS very_high THEN NEW_PRIORITY IS very_high;

RULE 11 : IF CLOUDLET_LENGTH IS medium and PRIORITY IS very_low THEN NEW_PRIORITY IS very_low;

RULE 12: IF CLOUDLET_LENGTH IS medium and PRIORITY IS low THEN NEW_PRIORITY IS low;

RULE 13 : IF CLOUDLET_LENGTH IS medium and PRIORITY IS medium THEN NEW_PRIORITY IS medium;

RULE 14 : IF CLOUDLET_LENGTH IS medium and PRIORITY IS high THEN NEW_PRIORITY IS medium;

RULE 15 : IF CLOUDLET_LENGTH IS medium and PRIORITY IS very_high THEN NEW_PRIORITY IS medium;

RULE 16 : IF CLOUDLET_LENGTH IS long and PRIORITY IS very_low THEN NEW_PRIORITY IS very_low;

RULE 17 : IF CLOUDLET_LENGTH IS long and PRIORITY IS low THEN NEW_PRIORITY IS very_low;

RULE 18 : IF CLOUDLET_LENGTH IS long and PRIORITY IS medium THEN NEW_PRIORITY IS low;

RULE 19 : IF CLOUDLET_LENGTH IS long and PRIORITY IS high THEN NEW_PRIORITY IS low;

RULE 20: IF CLOUDLET_LENGTH IS long and PRIORITY IS very_high THEN NEW_PRIORITY IS low;

RULE 21 : IF CLOUDLET_LENGTH IS very_long and PRIORITY IS very_low THEN NEW_PRIORITY IS very_low;

RULE 22 : IF CLOUDLET_LENGTH IS very_long and PRIORITY IS low THEN NEW_PRIORITY IS very_low;

RULE 23 : IF CLOUDLET_LENGTH IS very_long and PRIORITY IS medium THEN NEW_PRIORITY IS very_low;

RULE 24: IF CLOUDLET_LENGTH IS very_long and PRIORITY IS high THEN NEW_PRIORITY IS low;

RULE 25: IF CLOUDLET_LENGTH IS very_long and PRIORITY IS very_high THEN NEW_PRIORITY IS low;

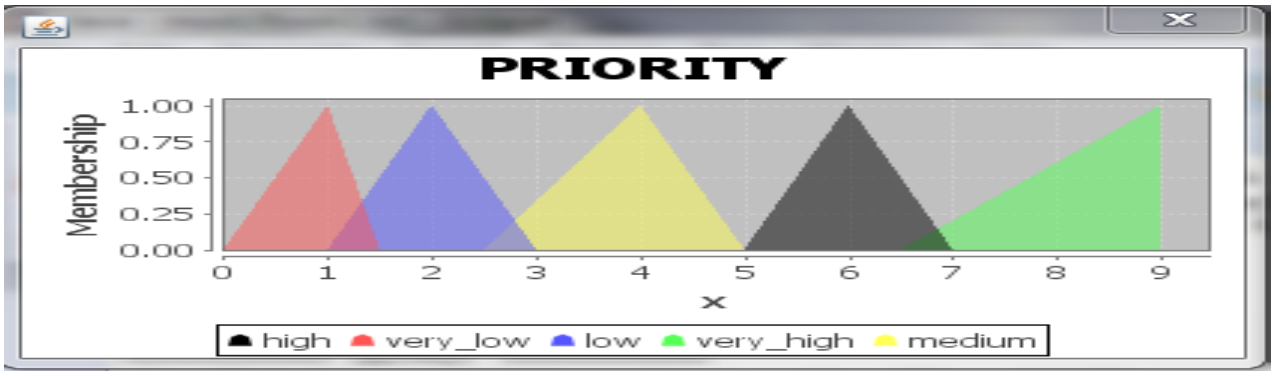
END_RULEBLOCK

END_FUNCTION_BLOCK

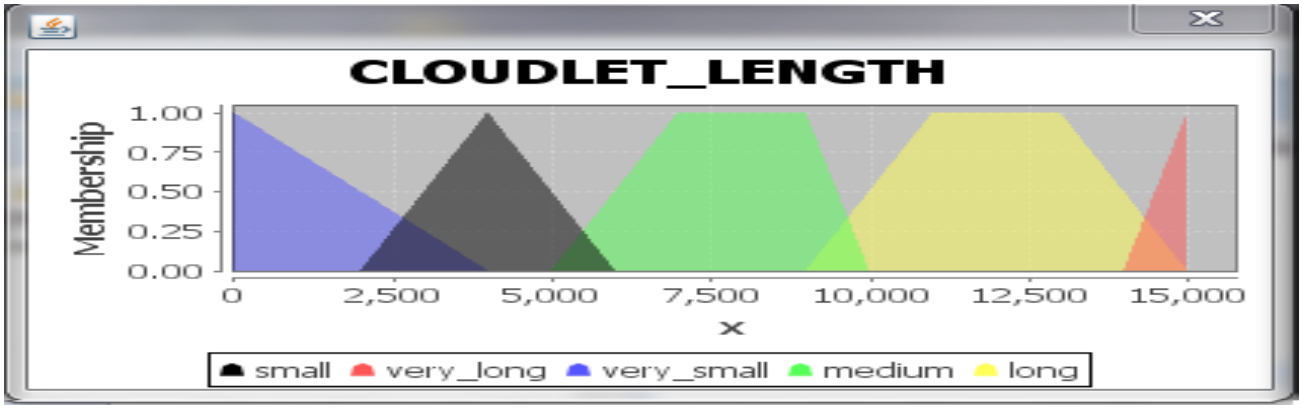
As shown in the FCL file there is membership function definition and 25 fuzzy rules are written .Using the rule set of this file the fuzzy related decision are taken . The CLOUDLET_LENGTH

and PRIORITY are input and NEW_PRIORITY is output linguistic variable. Priority is defined on the scale of 10. CLOUDLET_LENGTH can have linguistic values as “very small”, ”small”, ”medium”, ”long”, ”very long”. These values are corresponding to the number of lines of code of cloudlets. priority can have linguistic values as “very low”, ”low”, ”medium”, ”high”, ”very high”. These values represent the corresponding user defined importance of cloudlets. the output will be represented by fuzzy priority and it will have linguistic values as “very low”, ”low”, ”medium”, ”high”, ”very high”. These fuzzy rules are self explanatory .the first rule says that “IF CLOUDLET_LENGTH IS very_small and PRIORITY IS very_low THEN NEW_PRIORITY IS very_high”. This is applied because this will let smaller cloudlet comparatively wait less and by this average waiting time will get reduced.

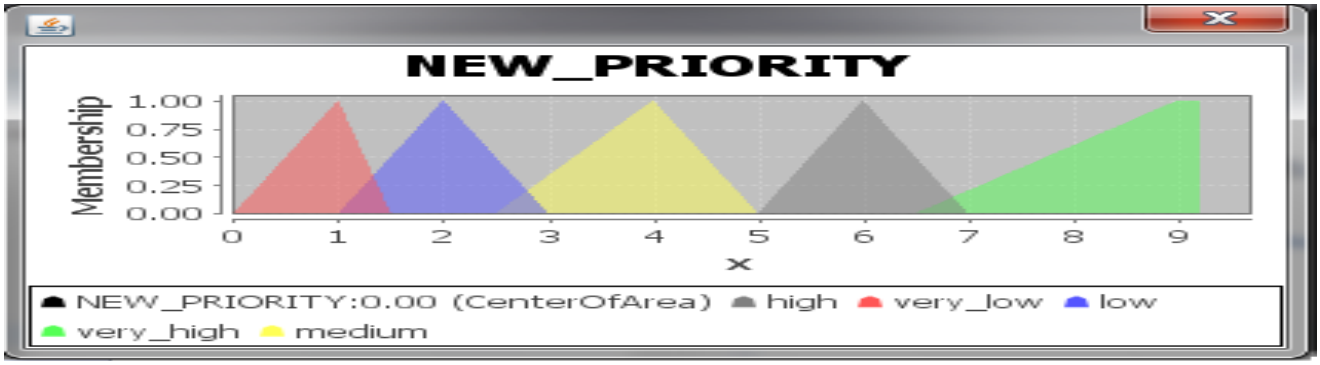
The following screen shots 4.3,4.4 and 4.5 respectively represent the graphical versions of membership function for PRIORITY , CLOUDLET_LENGTH and FUZZY PRIORITY attributes . They are built using java buttons. They show the variation of membership values for the linguistic variables. These graphical versions of membership function gets popped as we execute our java based application.



Screen Shot 4.3 : Membership Function For The PRIORITY



Screen Shot 4.4 : Membership Function For The CLOUDLET_LENGTH



Screen Shot 4.5 : membership function for the NEW _PRIORITY

The Output of the Classifier is taken by the scheduler segment of the system. The output of the classifier are the various classes of the cloudlets which need to be scheduled by fuzzy based priority method over virtual machines by this scheduler segment.

The Table 4.1 shows the description of the virtual machine which we used for the execution of the classes of cloudlets. Each virtual machine will be mapped to only one class of cloudlets which depends on the requirement of the cloudlets of that class .For example virtual machine with id 0 will be used by the FORMOST class of cloudlet, virtual machine with id 1 will be used by the PC

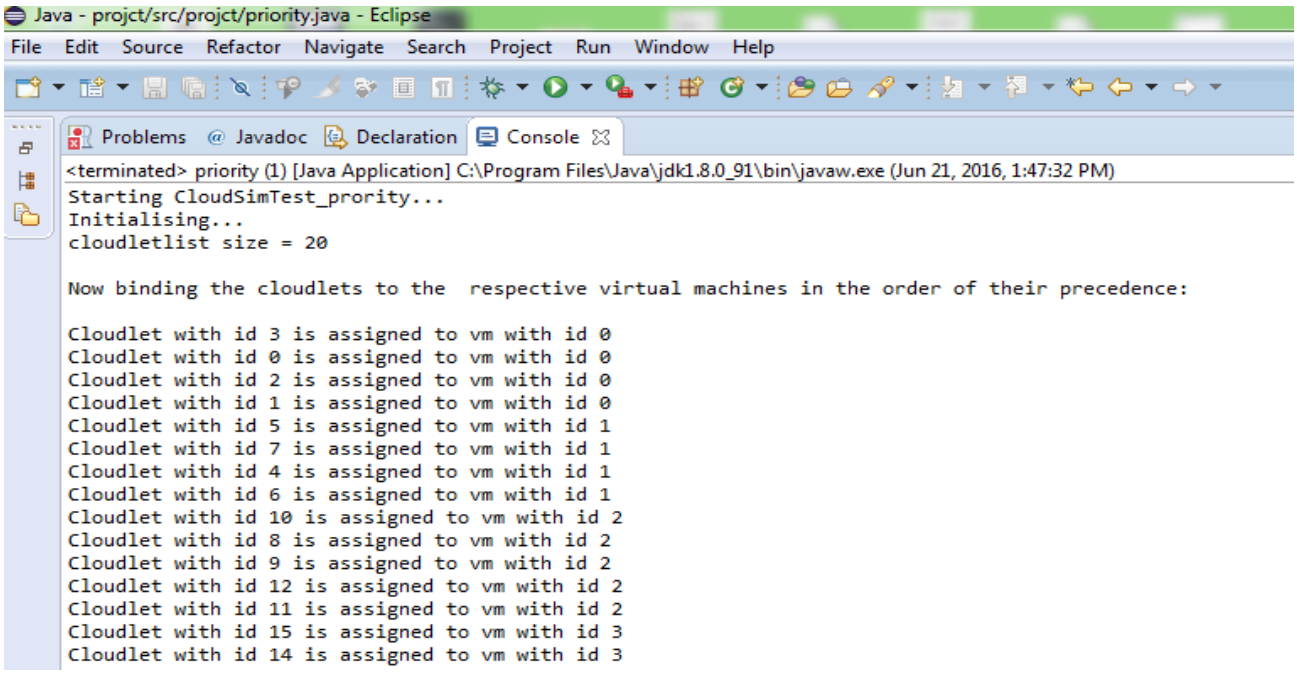
class, virtual machine with id 2 will be used by the DC class , virtual machine with id 3 will be used by the NORMAL class, virtual machine with id 4 will be used by the PC-DC class.

VMID	TYPE	MIPS	NO. OF PES	RAM	BW	SIZE	VMM
0	FORMOST	3500	1	1028	2500	10000	Xen
1	PC	3500	3	1028	1200	10000	Xen
2	DC	1000	1	512	3000	10000	Xen
3	NORMAL	2000	1	512	1500	10000	Xen
4	BOTH	3500	3	1028	3000	10000	Xen

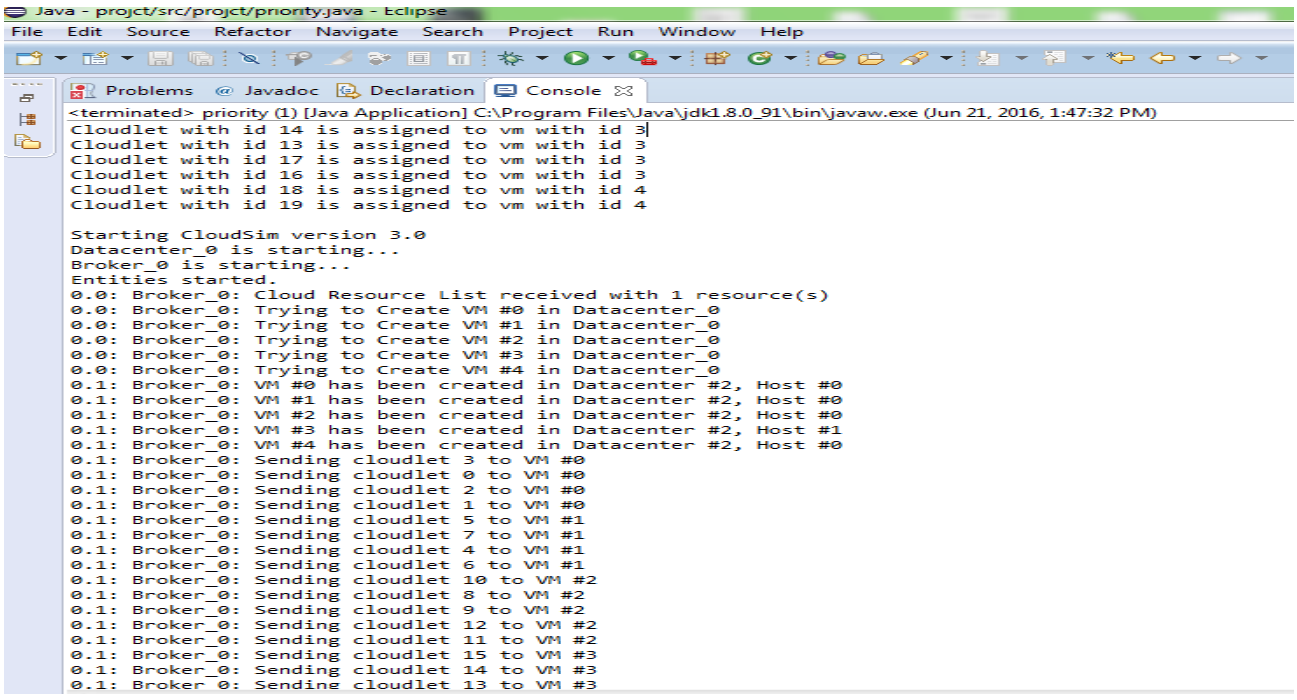
Table 4.1: The Specifications Of The Virtual Machines

4.3.1.2 Output Of Priority Scheduling

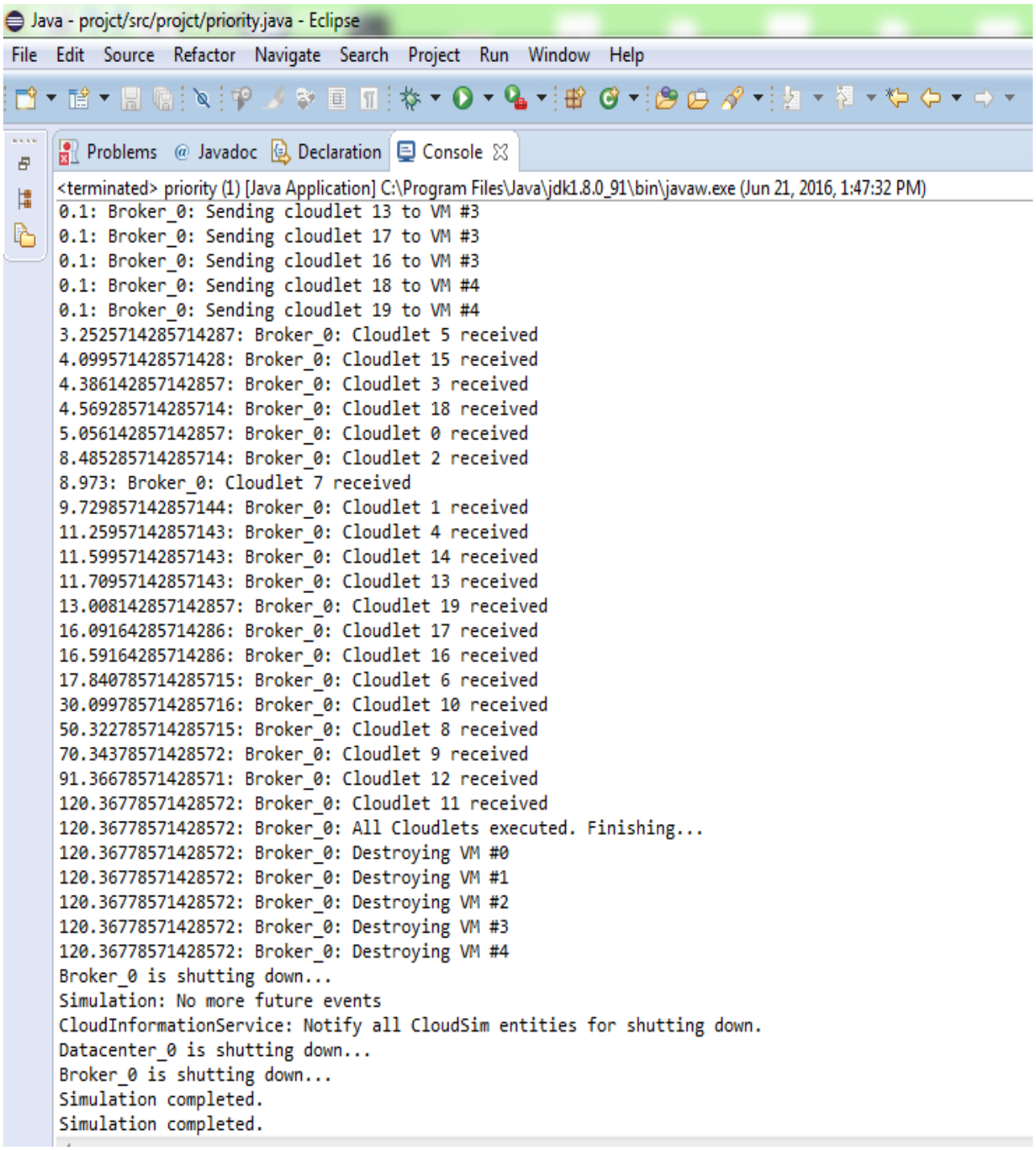
We have implemented Priority based scheduling in cloudsim tool for the purpose of comparing the results with our proposed algorithm. Here the same classified cloudlets are being used for results generation. Screen shot 4.6, 4.7, 4.8 and 4.9 show the output of running this priority scheduling algorithm in above stated scenario.



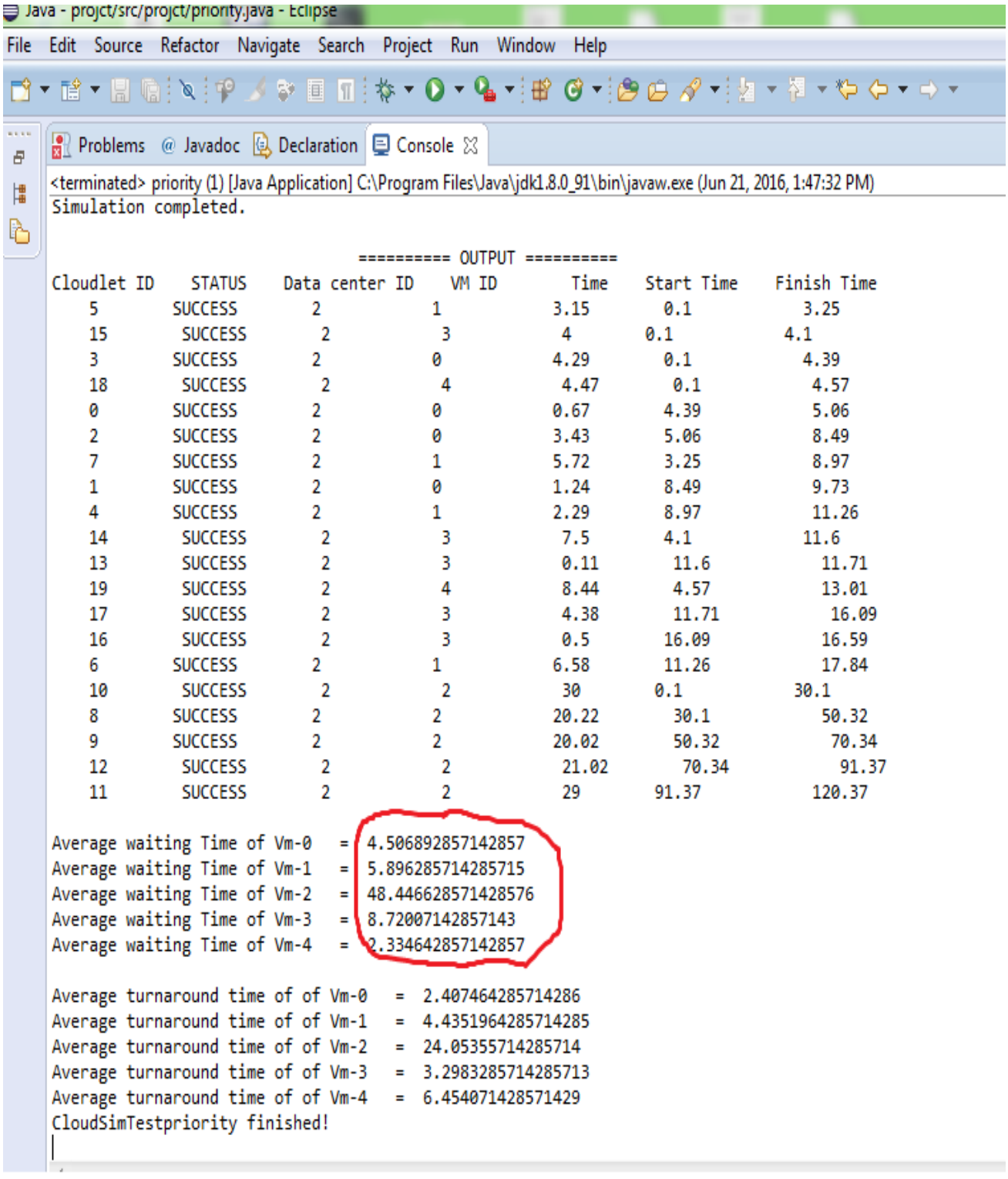
Screen Shot 4.6: Output 1 Of 4 For Priority Based Scheduling



Screen Shot 4.7: Output 2 Of 4 For Priority Based Scheduling



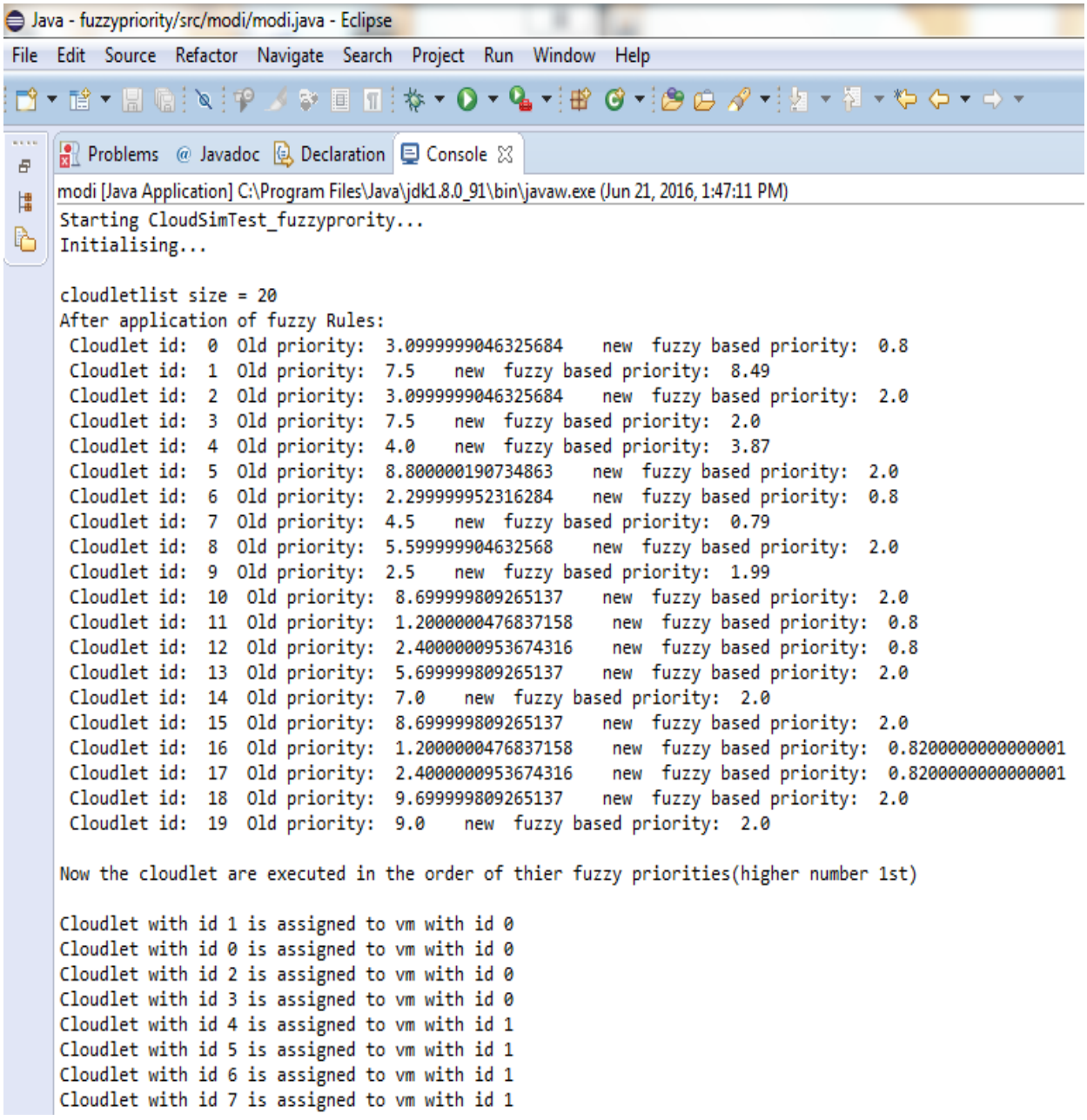
Screen Shot 4.8 : Output 3 Of 4 For Priority Based Scheduling



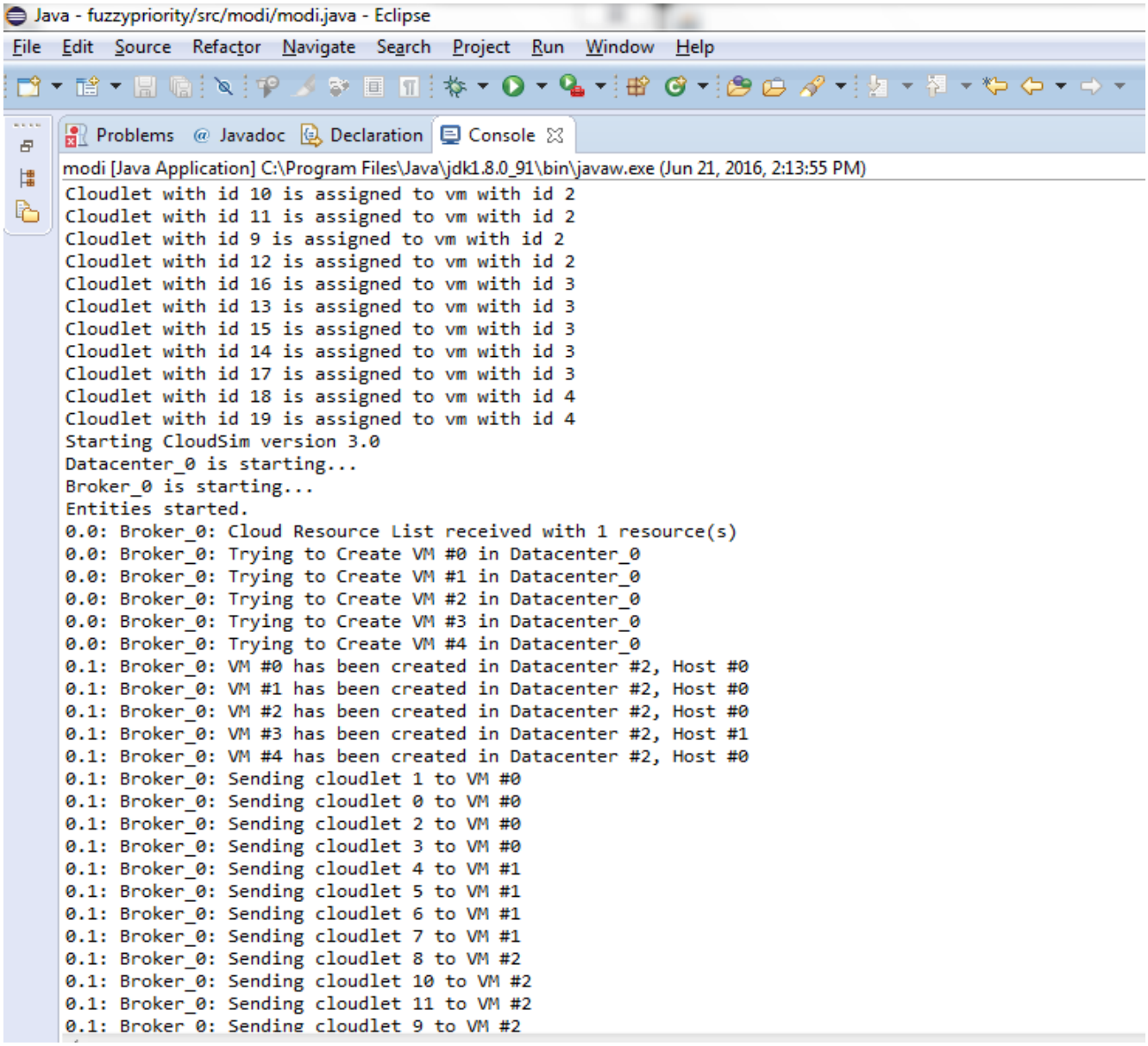
Screen Shot 4.9: Output 4 Of 4 For Priority Based Scheduling

4.3.1.2 Output Of The Proposed Approach

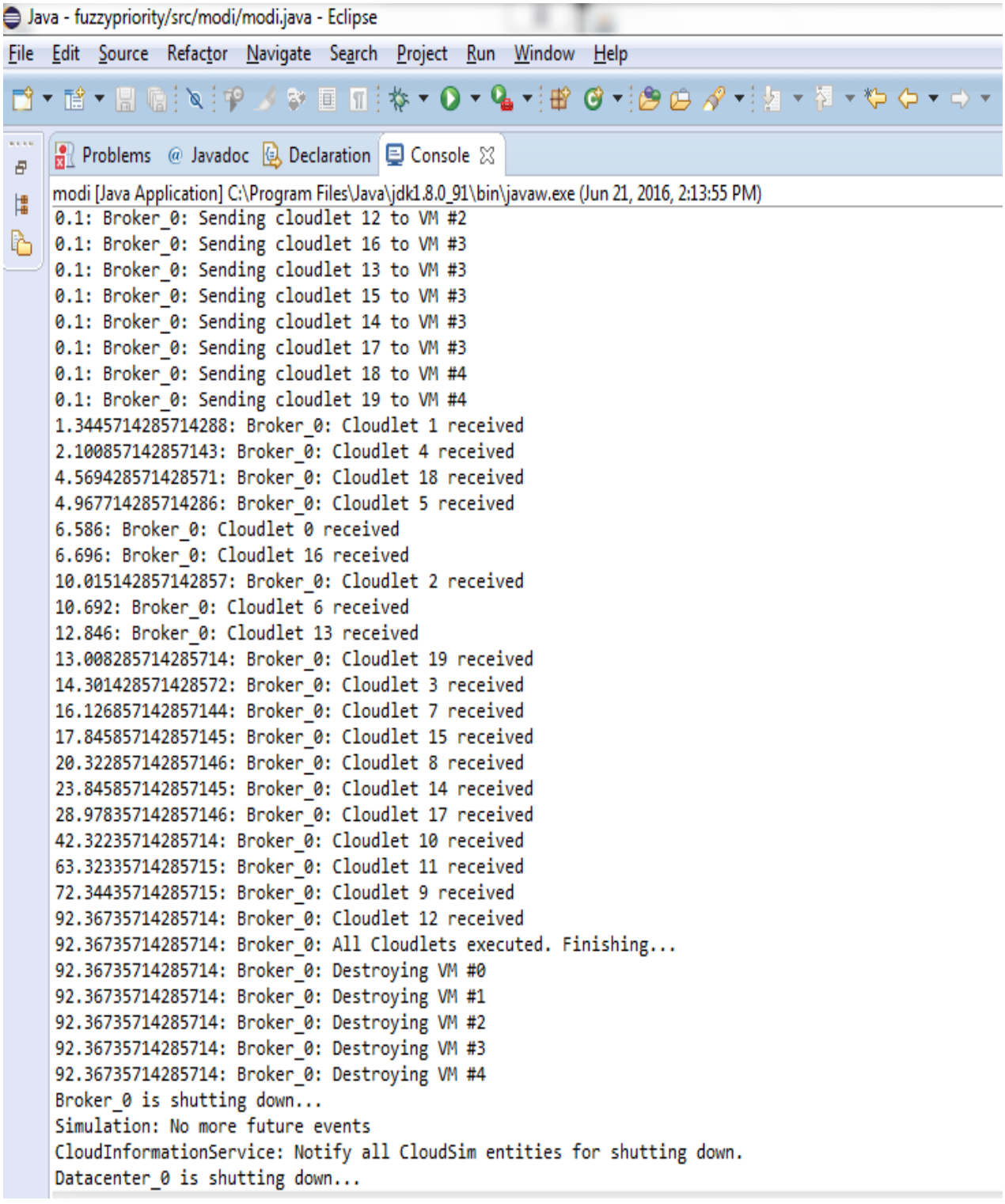
Now Its time to show the results of new fuzzy based proposed technique . Here The output screen shot 4.10,4.11 ,4.12 and 4.13 show the results when the experiment is done using cloudsim in the stated scenario with same input cloudlets.



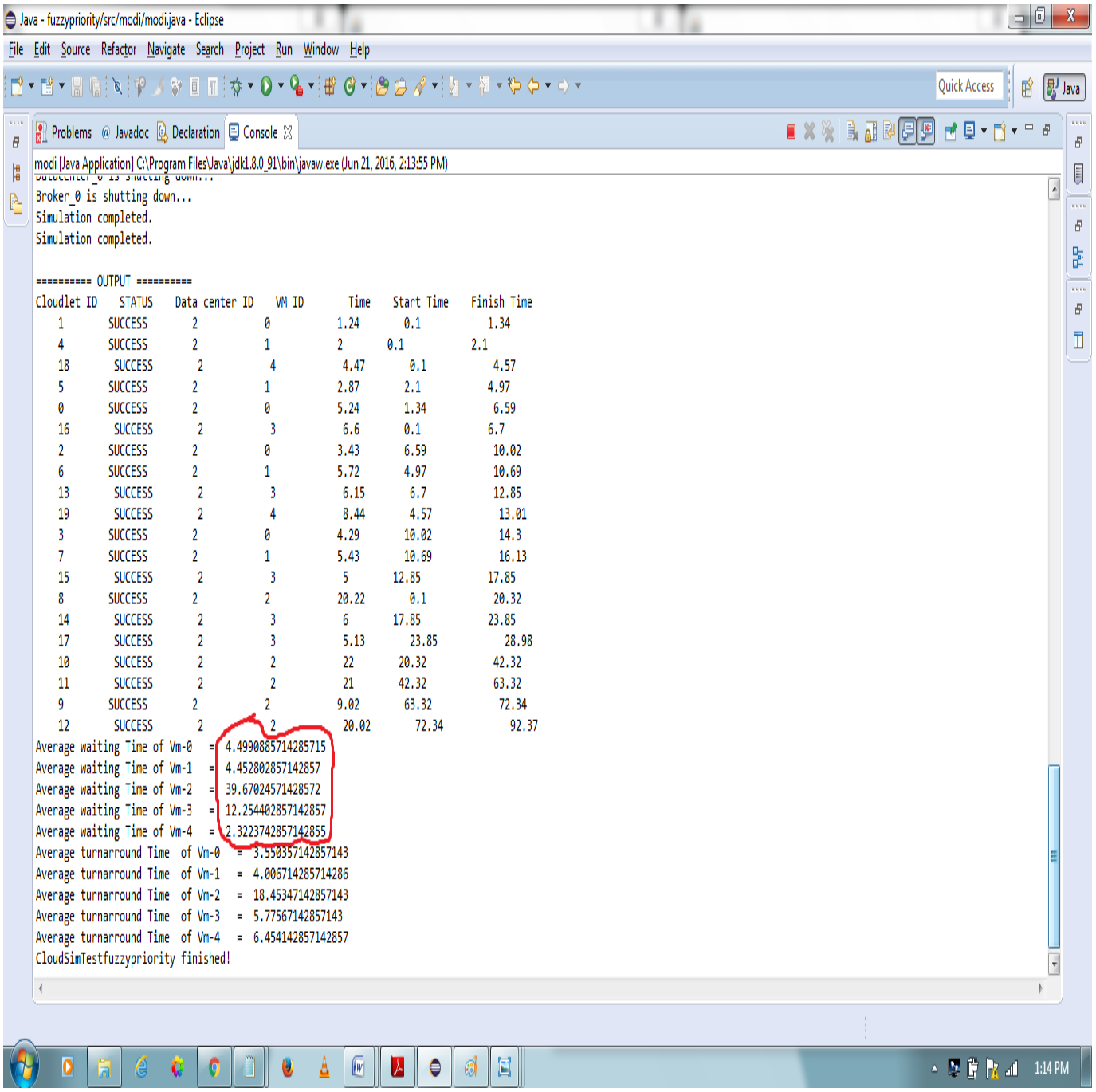
Screen Shot 4.10 : Output 1 Of 4 For New Proposed Approach



Screen Shot 4.11: Output 2 Of 4 For New Proposed Approach



Screen Shot 4.12: Output 3 Of 4 For New Proposed Approach



Screen Shot 4.13: Output 4 Of 4 For New Proposed Approach

4.3.2 Experiment 2

In this experiment a list of 100 random cloudlets is taken and the proposed algorithm is executed on them .As shown in the table the classifier segment (which was already trained) of the proposed system categorized these cloudlets into classes in different proportions like 8% in foremost cloudlet class, 25% in processing concentrated class,22% in data concentrated,34% in PC-DC ,11% in normal depending on the QoS specification of cloudlets.

For each class of cloudlets we build a virtual machine. In scheduler segment of the proposed system, datacenter broker which works on the fuzzy based scheduling technique bind the cloudlets of one class into its best fit virtual machine. The average waiting time for each class of cloudlet is calculated for the proposed method as well as when scheduled in traditional way (in first come first serve and priority based). As results shown in the table 4.2 we have lesser average waiting time for every class of cloudlet in the proposed method as compare to traditional methods of scheduling. The results are also depicted in the format of bar chart as shown in the figure 4.3 .

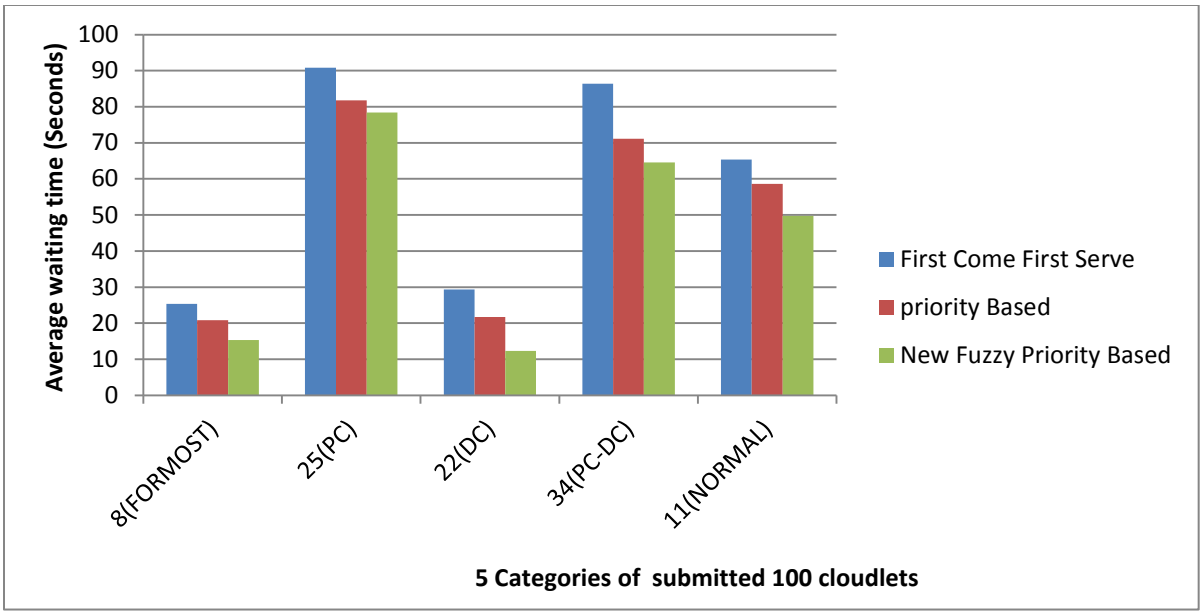


Figure 4.3:Comparison Of New Approach With FCFS And Priority Based Technique

<table>
<tr><th></th><th>FORMOST</th><th>PC</th><th>DC</th></tr>
<tr><td rowspan="4">100</td><td>8%
<table><tr><th>FCFS</th><th>PRIORITY</th><th>NEW</th></tr><tr><td>21.35</td><td>20.81</td><td>15.32</td></tr></table></td><td>25%
<table><tr><th>FCFS</th><th>PRIORITY</th><th>NEW</th></tr><tr><td>21.35</td><td>200.75</td><td>155.36</td></tr></table></td><td>22%
<table><tr><th>FCFS</th><th>PRIORITY</th><th>NEW</th></tr><tr><td>21.35</td><td>21.75</td><td>12.32</td></tr></table></td></tr>
<tr><td>PC-DC</td><td>NORMAL</td><td></td></tr>
<tr><td>34%
<table><tr><th>FCFS</th><th>PRIORITY</th><th>NEW</th></tr><tr><td>21.35</td><td>71.09</td><td>64.52</td></tr></table></td><td>11%
<table><tr><th>FCFS</th><th>PRIORITY</th><th>NEW</th></tr><tr><td>21.35</td><td>98.59</td><td>88.77</td></tr></table></td><td></td></tr>
</table>

Table 4.2 : Comparison table of results

CHAPTER 5

CONCLUSION AND FUTURE WORK

Considered the commercialization and the virtualization characteristics of cloud computing and Focusing on the problem of high efficiency and effectiveness in resource scheduling, in this presented work we have proposed a system which involves the prediction of the best virtual machine for the user submitted cloudlets as well as scheduling them in an efficient manner. The system includes neural network based classifier cum predictor, which has been trained to predict the appropriate class for the submitted cloudlets. This classifier is used to give answers of many questions like -a particular submitted cloudlet is of which type?? What type of virtual machine should be assigned to this cloudlet??. This strategy of classifying the jobs leads to less resource wastage and higher energy efficiency. The system has fuzzy priority based cloudlet scheduling. This algorithm is having benefits of shortest job first as well as Priority scheduling algorithm and is capable of removing the starvation problem that occurs in priority scheduling algorithm as well as reducing the average waiting time . Fuzzy priority scheduling algorithm gives better throughput also and lesser waiting time than traditional Priority Scheduling algorithm.

REFERENCES

1. P. Mell and T. Grance. The NIST Definition of Cloud Computing (Draft). National Institute of Standards and Technology, 53:7, 2010.

2. I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360- degree compared. In Proceedings of Grid Computing Environments Workshop,
pages 1–10, 2008.

3. M. Armbrust, A. Fox, and R. Griffith. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University
of California, Berkeley, Feb 2009.

4. R.N.Calheiros, R. Ranjan, A. Beloglazov, C. Rose, R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", in: Software: Practice and Experience (SPE), Volume 41, Number 1, ISSN: 0038-0644, Wiley Press, New York, USA., pp. 23–50, 2011.

5. Pinal Salot, "A Survey Of Various Scheduling Algorithm In Cloud Computing Environment", IJRET: International Journal of Research in Engineering and Technology ISSN: 2319-1163, Volume: 02 Issue: 02, pp. 131-135, 2013

6. S Banerjee, I Mukherjee and P.K. Mahanti "Cloud Computing Initiative using Modified Ant Colony Framework" World Academy of Science, Engineering and Technology, vol. 562009, pp. 221-224

7. Shaminder Kaur and Amandeep Verma "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment" I.J. Information Technology and Computer Science, vol. 10, pp. 74-79, 2012

8. Patrick Naughton and Herbert Schildt, Complete Reference Osborne/McGraw-Hill © 1999.

9. Pablo Cingolani ,Jesus Alcala-Fdez,(2013) "jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming"International Journal of Computational Intelligence Systems, Vol. 6, Supplement 1 (2013), 61-75.

10. L. Zadeh, "Fuzzy Sets", Information and Control, vol. 8, pp. 338–353, 1965.

11. D. Driankov, H. Hellendoom and M. Reinfrank, "An introduction to fuzzy control" ,Springer-Verlag, Berlin, New York, 1993.

12. Mamdani E.H., Assilian S., "An experiment in linguistic synthesis with a fuzzy logic controller "International Journal of Man-Machine Studies, Vol.7, No. 1, 1975.

13. Hornik, Kurt., Maxwell Stinchcombe. and Halbert White. “Multilayer feed forward networks are universal approximators” Neural networks 2.5, pp. 359-366, 1989

14. Martin T. and Mohammad B. Menhaj “Training feed forward networks with the Marquardt algorithm” pp. 989-993, 1994

15. Baptista D, Morgado-Dias F “A survey of artificial neural network training tools” 2013 Neural Comput Appl 23(3-4):609–615.

16.Dayana Benny, K.R soumya “Feed-forward neural network processing speed analysis and an experimental evaluation of Neural Network” Information Science and Engineering (ICISE), Second International Conference, 4-6 Dec.2010.

17. Shatha J. Kadhim , Kasim M. Aubidy “Design and Evaluation of a Fuzzy-Based CPU Scheduling Algorithm” Information Processing and Management Volume 70 of the series Communications in Computer and Information Science pp 45-52

18. Ru Jia , J. Keung, "An Empirical Investigation on the Simulation of Priority and Shortest Job First Scheduling for Cloud Based Software Systems", 22nd Australian Software Engineering Conference (ASWEC), pp.7887, 47 June 2013.

19. Auday Al-Dulaimy, Rached Zantout,Ahmed Zekri, Wassim Itani “Job Classification in Cloud Computing: The Classification Effects on Energy Efficiency ” IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), 2015.

20. R. Buyya , J. Broberg , A. Goscinski, "Cloud Computing Principles and Paradigms", 2011.

21. M.N.O. Sadiku, S.M. Musa and O.D. Momoh "Cloud Computing: Opportunities and Challenges" Potentials, IEEE, vol. 33, no. 1, pp. 34-36, 2014.

22. Avnish Kumar Yadav ,Maitreyee Dutta “A Novel Approach to Provide Broking Service in Cloud Computing”2014.