A

PROJECT REPORT

on

"A SECURE CLOUD STORAGE MECHANISM BASED ON FAULT AND LOAD BALANCING."

Submitted in partial fulfillment of requirements for the award

Bachelor of Technology

COMPUTER SCIENCE & ENGINEERING

BY

Mr. Omkar K. Khadke

Under the Guidance of

Prof. Lowelesh N. Yadav

Assistant Professor

Department of Computer Science and Engineering

SHRI SAI COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E. New Delhi DTE Mumbai & Affiliated to DBAT University)

Nagpur Road, Bhadrawati, Dist.Chandrapur (M.S.)

www.sscet.in

Session 2021-22

“A SECURE CLOUD STORAGE MECHANISM BASED ON FAULT AND LOAD BALANCING.”

Submitted in partial fulfillment of requirements for the award bachelor’s degree of Technology

In

COMPUTER SCIENCE & ENGINEERING

BY

Mr. Omkar K. Khadke

Department of Computer Science and Engineering

SHRI SAI COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E. New Delhi DTE Mumbai & Affiliated to DBAT University)

Nagpur Road, Bhadrawati, Dist.Chandrapur (M.S.)

www.sscet.in

Session 2021-22

SHRI SAI COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E. New Delhi DTE Mumbai & Affiliated to DBAT University)

Nagpur Road, Bhadrawati, Dist.Chandrapur (M.S.)

www.sscet.in

Department of Computer Science and Engineering

CERTIFICATE FROM ACADEMIC ADVISOR

This is to certify that the project report entitled **"A Secure Cloud Storage Mechanism Based on Fault & Load Balancing"** submitted by **Mr. Omkar K. Khadke** in the partial fulfillment of the requirement for the the award **Bachelors degree of Technology (Computer Science Engineering)** is a bonafide work carried by him/her**.**

The results of the investigations enclosed in this report have been verified and found satisfactory.

EXTERNAL EXAMINAR

(Signature of Guide)
Prof. Lowelesh N. Yadav,
(Assistant Professor)
Department of CSE,
SSCET,Bhadrawati

(Signature of Mentor)
Prof. Neehal B. Jiwane
(Assistant Professor)
Department of CSE,
SSCET,Bhadrawati

SHRI SAI COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E. New Delhi DTE Mumbai & Affiliated to DBAT University)

Nagpur Road, Bhadrawati, Dist.Chandrapur (M.S.)

www.sscet.in

Department of Computer Science and Engineering

DECLARATION BY THE CANDIDATE

I, **Mr. Omkar K. Khadke** hereby certify that the project report entitled **"A Secure Cloud Storage Mechanism Based on Fault & Load Balancing"** is submitted in the partial fulfillment of the requirement for the award of the degree of **Bachelors of Technology (Computer Science Engineering)**

This Record is a bonafide work carried out by me under the guidance of **Prof. Lowelesh N. Yadav**, Assistant Professor, Shri Sai College Of Engineering and Technology, Bhadrawati, Chandrapur dist. The results embodied in this Project report have not been reproduced/copied from any source. The results embodied in this Project report have not been submitted to any other university or institute for the award of any other degree.

(Signature)

Mr. Omkar K. Khadke
(1941901242048)
Dept. of CSE,
SSCET, Bhadrawati

SHRI SAI COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E. New Delhi DTE Mumbai & Affiliated to DBAT University)

Nagpur Road, Bhadrawati, Dist.Chandrapur (M.S.)

www.sscet.in

Department of Computer Science and Engineering

CERTIFICATE FROM PRINCIPAL AND HOD

This is to certify that the project report entitled **"A Secure Cloud Storage Mechanism Based on Fault & Load Balancing"** submitted by **Mr. Omkar K. Khadke** bearing Uni Roll No. **1941901242048** in the partial fulfillment of the requirement for the award the degree of **Bachelors of Technology (Computer Science Engineering)** is a bonafide work carried by him/her**.**

The results of the investigations enclosed in this report have been verified and found satisfactory

Asst. Prof. L. N. Yadav
Head of the Department,
Department of CSE,
SSCET, Bhadrawati

Dr. V. S. Gorantiwar
Principal
SSCET, Bhadrawati

ACKNOWLEDEGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible. I take this opportunity to express my gratitude to all those who helped me in doing this project.

I express my earnest and deep sense of gratitude to guide **Prof. Neehal B. Jiwane,** Asst. Prof, Department of Computer Science & Engineering. It was a great pleasure and honor to work under him/her.

I express my earnest and deep sense of gratitude to **Asst. Prof. L. N. Yadav,** Head of the Department, Computer Science & Engineering without whose valuable suggestions and guidance, creation of this the project report wouldn't have been possible.

I also owe the gratitude due towards **Dr. V. S. Gorantiwar,** Professor, PRINCIPAL of SSCET Bhadrawati, for his suggestions and encouragement given to me in completing the Technical Seminar work.

I would also like to thanks **Faculty Members** and **Management of SSCET**.

I am grateful to my parents, brother & friends for the way they co-operated and expected me to achieve success that has stirred my ambition always to do the best.

Mr. Omkar K. Khadke

TITLE of project

“A SECURE CLOUD STORAGE MECHANISM BASED ON FAULT AND LOAD BALANCING.”

ABSTRACT

Key generation and key authentication provide a security facility for storage a data over cloud network. The data storage over cloud network process involved multiple servers and required multiple key authentication process. The multiple key authentication process degraded the performance of cloud network. And the process of key generation and key authentication take more time and invites man in middle attack and cloud data threat by third party. Security in Cloud computing can be addressed in many facets viz. authentication, integrity, confidentiality etc. Data integrity or correctness, one of most challenging aspects in Cloud computing, is the issue related to Cloud user's worry about alteration/deletion in his data stored remotely.

Many cloud storage providers claim that they provide a very solid security to their users, but we should know that every broken security system was thought once to be unbreakable. As some examples we can mention Google's Gmail collapse in Europe in February 2009, a phishing attack on Salesforce.com in November 2007 and a serious security glitch on Dropbox in June 2011. If we look a bit deeper in the structure of cloud computing systems, we may feel even more insecure, because they make use of multi-tenancy. Many cloud computing providers work with third parties, so users lose even more trust, especially when they do not know these third parties well. In such a situation users may not dare use the cloud storage system to store their private data. Apart from this, until now there has not been made any standardization for the Security in the cloud. Any software update could lead to a security breach if care is not taken

In this dissertation proposed a key generation and key distribution technique for the user and server for authentication of data. The key generation technique adopts the mechanism of binary digit shift with the operation of XOR and much more arithmetic operation. The process of key distribute in the form of session along with authentication. The named of algorithm is cyclic shift key generation because the process of key path automatically shift in the form of next key value.

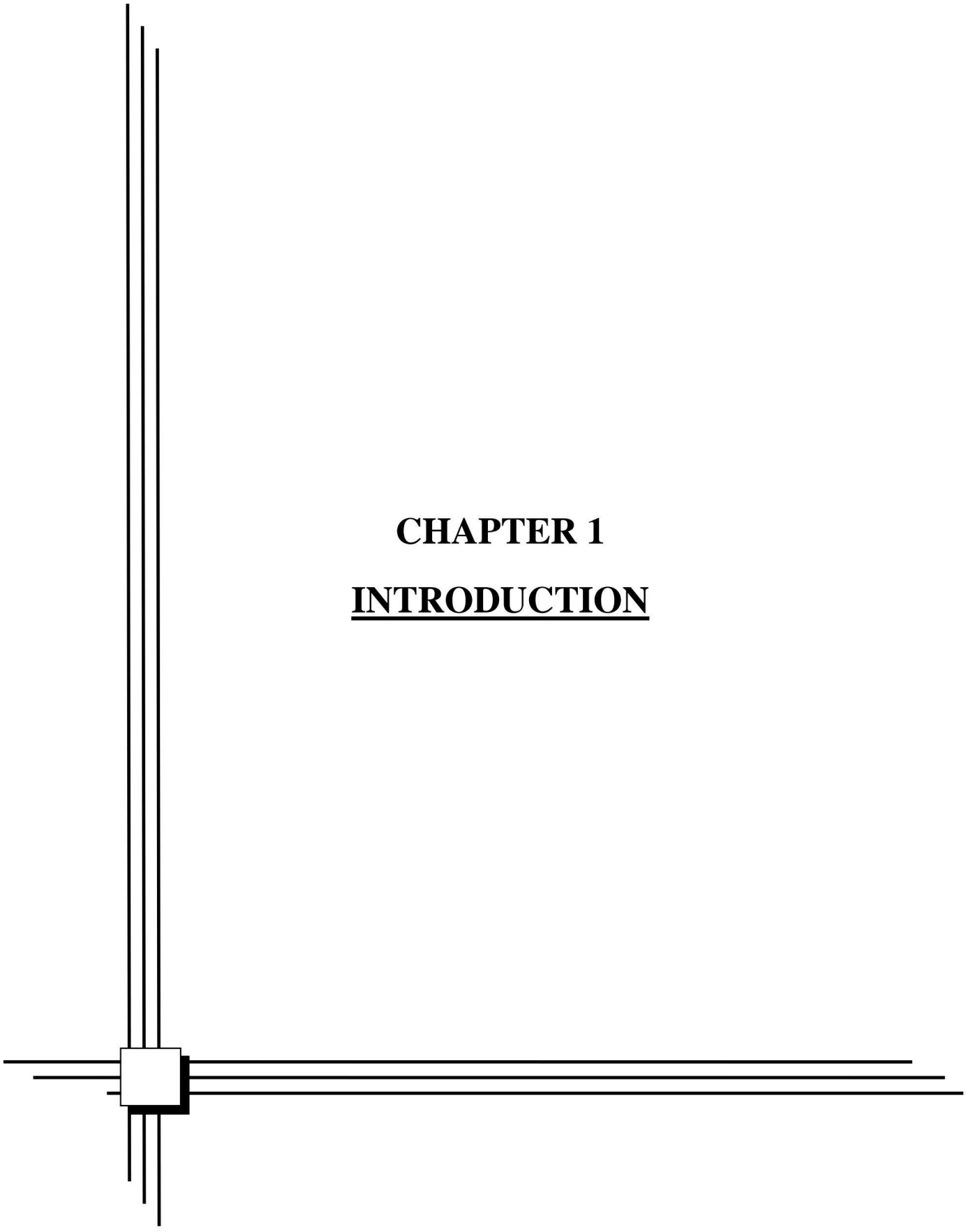
For the validation of proposed algorithm used MATLAB software and used different size of file for the process of encryption and decryption. Our execution time is reduces instead of AES, Blowfish and cryptography algorithm.

General Terms or Keywords : AES, DSS, RSA, ECA, KMS, TEK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES



CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Cloud computing is a most widespread and popular form of computing, promising high reliability for customers and providers both at the same point of time from many fields of sciences or industry. Clients from the different field are served by datacenters in cloud environment geographically spread over the world. Cloud serves a large number of requests coming from various sources over datacenter with high power consumption. However, to provide such a large computing power required a huge power, leading to high power consumption and cost. Request types in cloud system also affect the services which are public and private requests whose proportion is random in nature. A survey in 2006 over the performance of cloud environment in the USA shows datacenter consumed 4.5 billion kWh units of power, which is 1.5% of total power consumed in the USA and this power requirement is increasing 18% every year [1]. In general, cloud computing deals with various issues live poor resource utilization and load balancing and many more. Some of the issues are discoursed as follows: 1) as cloud computing tools are used by industry and they have issues with the rapidly growing request and a number of servers deployed, increasing the power consumption. 2) Task allocation of request among datacenter without having knowledge of QoS provided by servers. 3) Current task allocation algorithms only focus on balancing the request and improve unitization of the system but not the failure probability of system. 4) High loaded data centers have high failure probability and due to high load, this may lead to slow down of datacenter and poor QoS (Quality of service) to the client and client provider. 5) While few of the servers are overloaded and some of them are idle or under loaded. 6) Some request needs to be computed with QoS but due to high load and fault rate they may the QoS promised which is not appropriate to the user and will be a critical issue. 7) As per recent study [2-5], utilization of data centers is a major problem because 60% data centers are idle and most of 20% data centers are utilized and waste of the resources.

This shows the poor utilization of resources but this shows the importance of a new approach that has sufficient strategy to minimize wastes of resources and increasing reliability by allocating task over resources which in the case of Cloud is VM with

low failure probability to provide high QoS to users. The existing algorithms only take into consideration cloud as non-faulty in nature and fail to provide specific QoS when a fault occurs. So to overcome these issues and improve the performance of the system, we have proposed approaches for resource load balancing and allocation. Figure 1.1 shows cloud computing features, type and various other properties [6]

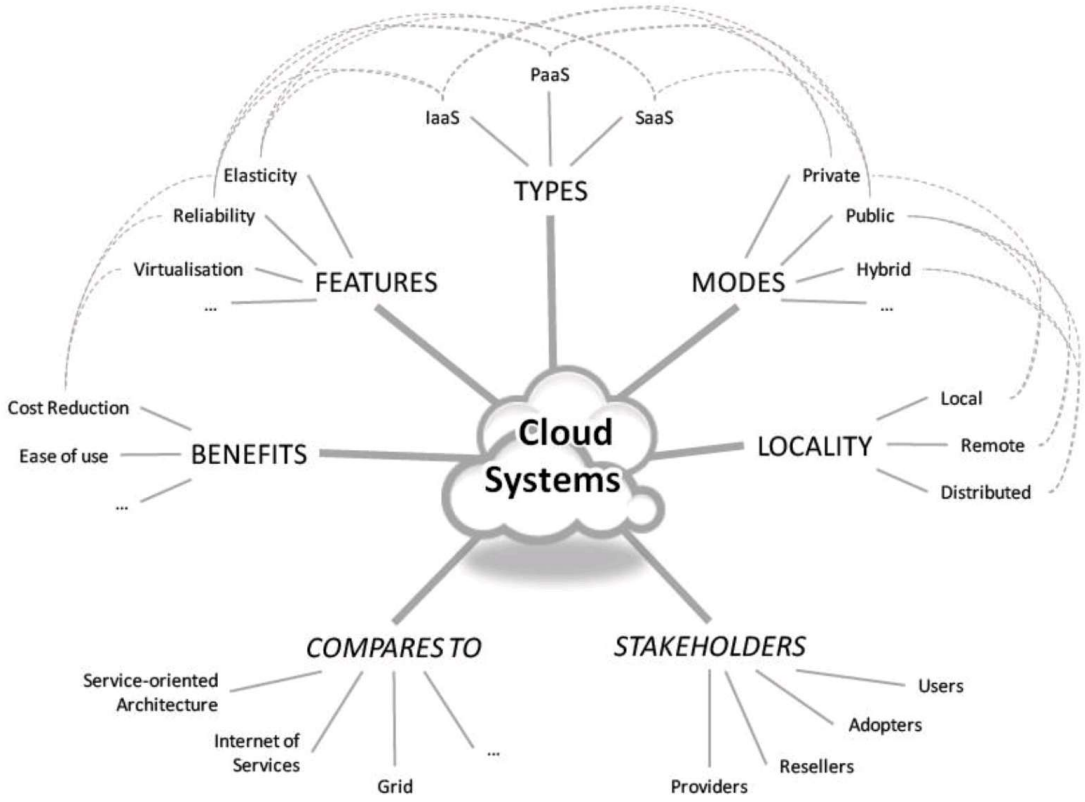


Figure 1.1 Cloud system characteristics and properties

1.1. Characteristics of Cloud

Cloud is a distributed environment, where the servers are placed at various geographical locations but seems to a user as a single entity. Cloud computing provides better performance than any other distributed system line Grid computing or cluster computing and many more. There are various characteristics of cloud computing which makes is superior than any other system which are as follows [6, 17];

a) **High Availability**

One of the most important features of the cloud is all time availability of resources in form of storage, computational capability and high network resources. This property also states that the resources are available in overloading conditions also.

b) **Pay per use model**

This feature made cloud computing popular in the industry due to affordable nature of cloud by an industry with high infrastructure or a business holder with the small requirement can easily manage and have its own infrastructure and high computing system at a low cost. Cloud computing allows a user to pay for only those resources, which are used by him for that specific period of time rather than purchasing a complete server or private infrastructure.

c) **Elasticity**

Cloud is said to be flexible and scalable at the same time. This feature allows the cloud to scale its resources up or down based on the user or business needs for a period of time. This allows the cloud to have high availability under overloaded condition also and provided uninterrupted services to the user without failure and high quality of service. d) Reliability

Cloud computing ensures to provide high reliable computing services and resources to the user which means that the user will be provided with uninterrupted services with the quality of services as assured to the client.

1.2. Business Models

Cloud computing provides various service driven business models to provide a different level of computation to the users. Cloud computing provides 3 type of service models listed as: Software as a Service (SaaS), platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Our work focuses on improving the performance of cloud infrastructure as a service in a faulty cloud environment [17].

The fault is a behavior of every distributed system because fault may occur any time that may be due to system failure, network failure or disk failure.

1.3. Issues in Cloud Computing

Cloud computing deals with various issues to maintain above discoursed characteristics and quality of serves assured to the user by cloud providers in term of high resource availability, computational capability [7-8]. Some or the issues dealing with resource management, resource scheduling, and managing system performance are discoursed below.

- Resource allocation
- Load balancing
- Migration
- Power efficient resource allocation and load balancing algorithms
- Cost efficient resource allocation and load balancing algorithms
- Fault tolerant algorithms
- Behavior-based algorithms
- Trust management

1.3.1. Resource Allocation

Resource Allocation strategy (RAS) in the cloud is all about the scheduling of tasks or requests by cloud provider in such a manner to balance the load over all the servers and provide high Quality of Service to clients. It also includes the time required to allocate the resources and the resources available. The main aim is to improve the utilization of resources and complete all the request within the deadline and with least execution time [9].

An optimal RAS should avoid the following criteria as follows:

a) *Resource contention* situation arises when two applications try to access the same resource at the same time.

b) *The scarcity of resources* arises when there are limited resources.

c) *Resource fragmentation* situation arises when the resources are isolated

d) *Over-provisioning* of resources arises when the application gets surplus resources than the demanded one.

e) *Under-provisioning* of resources occurs when the application is assigned with fewer numbers of resources than the demand.

Resource allocation algorithm can be categorized into three subcategories as from the literature review conducted over existing proposed algorithms.

Categorization is as follows:

1) Static

2) Dynamic

3) Learning-based.

Static scheduling algorithms are referred to algorithms which are not affected by system and behavior of cloud some of the algorithms line SJF, FCFS, Round robin etc [11]. On the other hand, dynamic algorithms are those whose objective function depends on the system parameters line deadline, available resources, resource utilization of host and many more example of these algorithms is a deadline-based algorithm, cost-based algorithm, utilization based algorithm [5-10]. The problem with these algorithms is that they do not take into consideration the previous performance of host and system as a whole. Moreover, the past faulty nature of the system is not taken into consideration and leads to large request failure. Dynamic algorithms deal with the issue of local minima these algorithms are not able to find a global best solution and stuck in local best solution.

1.3.2. Load Balancing

Load balancing aims to distribute load across multiple resources, such as server, a server cluster, central processing. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource

Goal of Load Balancing [12] are as follows:

1) To improve the performance substantially.
2) To improve system stability.
3) To have scalability in the system.
4) To improve the system condition under high load or request rate.

Types of Load balancing algorithms [13]

Sender-Initiated: When load balancing algorithm is triggered by the sender.

Receiver Initiated: When load balancing algorithm is triggered by the receiver.

Symmetric: It is the combination of both sender initiated and receiver initiated.

Load balancing is also used to manage the average utilization of the system as a whole to avoid creation of hot spots i.e. the request should not be clustered on a single datacenter rather should be spread over the servers. So it aims it find an underloaded server and move the requests to that selected server. This makes a requirement of a load balancing algorithm to fulfill these requirements taking into consideration system utilization and quality of service without failure.

1.3.3. Migration

Migration in cloud infrastructure plays an important role in cloud Infrastructure under system overloading condition. In cloud infrastructure when the server gets overloaded i.e., the utilization is beyond a threshold is considered to be overloaded, in such condition we need to migrate a virtual machine from overloaded server to an under loaded or neutral server [14]. This help to balance the load and

prevent the server from any failure. So there is a requirement of an intelligent and efficient migration algorithm or balance the condition and improve the performance of the system.

1.3.4. Power efficient resource allocation and load balancing algorithms

The power efficiency of a cloud environment is an important issue for a green cloud environment. As 53% of the total expense of a datacenter is spend on cooling i.e. power consumption [15].

In a survey in 2006 on datacenters established U.S consumed more than 1.4% of total power generated during the year [16]. Therefore we require improving the power efficiency of infrastructure. The problem can be solved in various ways and various proposal are been made to solve and improve the performance. So to do this we need to design power-aware resource allocation and load balancing algorithm to improve the total power consumption of the system and any such algorithm will result in a reduction of overall power consumption.

1.3.5. Cost efficient resource allocation and load balancing algorithms

Cloud computing uses pay-per-use model to ensure least cost and payment only for the resources used. To maintain this feature cloud controller algorithms like resource allocation migration and load balancing are responsible for maintaining this characteristic by offering the resources which can complete the client request on time and within the budget of client and have least cost that can be offered. So we require cost aware algorithm which are cost efficient and can provide the best system performance by improving utilization and power consumption all at the same time [10, 15]. These type of algorithm are referred to as multi-objective algorithms, there are many proposals made for improving the performance of the system but they only take into consideration either power or cost, so cannot guarantee the best performance.

1.3.6. Fault tolerant algorithms

Cloud computing environment is a type of distributed environment like grid computing and cluster computing. Existing algorithms consider cloud as nonfaulty

but faults are a part of distributed environment which may be due to hardware or software failure at any point of time [93, 99, 100]. There are many fault aware and fault prediction algorithms been proposed for grid environment to improve the reliability of the system. So similarly we require fault aware algorithms to make system fault aware reduce the failure probability of the system and increase the reliability of the system.

1.3.7. Behavior-based algorithms

Most of the resource allocation and load balancing algorithm proposed for cloud infrastructure are dynamic algorithms like min-min, max min and many more. These algorithms take into consideration only the current behavior \ status or the server and system for selection of server. The problem with these algorithms is that they do not take into consideration the previous performance of the system for prediction of the better solution rather than stuck in local minima. Behavior-based algorithm lists genetic algorithm, ant colony, particle swan optimization, monkey search and many more. So there is a need of algorithms taking into consideration the previous and present performance of the system for decision making.

1.3.8. Trust management

Trust models are been used in all form of distributed environments ranging from MANETS (Mobile ad hoc network), Sensor network and Grid computing to validate the reliability of nodes over distributed network. In grid computing, various trust models are been proposed to ensure trust in term of security and reliability of the server or the node. Trust models are to resolve the problem of reliability in any heterogeneous environment, which contributed of nodes having different configuration spread over a network. There are many models being proposed in a cloud computing environment.

What is Trust?

Trust can be defined as an entity based on reliability and firm belief based on an attribute of the entity. Trust is the firm belief in the competence of an entity to act

as expected, such that this firm belief is not a fixed value associated with the entity, but rather it is subjected to the entity's behavior and applies only within a specific context at given time [18]. The definition simply means that trust is a variable changing believe, based on both static and dynamic parameters.

Trust can also be defined as "the subjective probability by which an individual expects that another individual performs a given action on which its welfare depends" [19-20]. Trust can be categorized into three major classifications which are as follows [21]:

a) *Blind trust:* This is the default trust before any event in the system, and which would include an agent to initiate a relationship with unknown entities.
b) *Conditional trust:* This is a classic state of trust during the life of the agent. This condition trust is likely to evolve, and can be subject to some sets of constraints or condition.
c) *Unconditional trust:* Such a trust is the probability be configured directly by an administrator, and would not be sensitive to successful/unsuccessful interaction and external recommendation of any other sources of evolution of the conditional trust.

1.4. Problem statement

The aim of this work is to make system fault tolerant and more reliable computing system with improved performance in cloud infrastructure environment. A number of algorithms have been worked out for long period of time but they assumed cloud as non faulty. So in our work, we have proposed various fault tolerant algorithms to resolve various issues as follows:

1) To design a fault and deadline aware load balancing algorithms for private and hybrid cloud, which aim to improve QoS of load balancing algorithm and minimize the faults, resource utilization, minimize response time and avoiding overloading of any single resource in cloud.

2) To design learning based fault aware resource allocation algorithms, to provide a global best schedule with least scheduling time complexity.

3) Designing fault aware and power-efficient scheduling algorithms for improving power efficiency and request failure count in the cloud.

1.5. Parameters Used

Fault rate: defined as the total count of request failed over a period of time T

Failure Probability: as the probability of request to fail on a specific host or system.

Reliability of a system: This feature of a system can derived from the failure probability of system which can be defined as:

$$Reliability\% = 100 - Failure\_Probability\% \quad (1.1)$$

Power Efficiency: The ratio of the output power over the input power i.e. the percentage power consumed over a period of time.

Utilization: this is the capability of the host to be used out of total available resources. *Average Resource Utilization:* This is an average of utilization of resources over the whole system i.e. all available hosts.

Average start time: Average waiting time of request before been scheduled or allocated.

Average Finish time: Average of finishing time of all the request executed by system.

Scheduling Delay: Total time to find a suitable resource for a set of tasks.

Make Span: Total execution time of system including scheduling delay for set of requests /Task

1.6. Performance Parameters

To study the performance of proposed algorithm over existing algorithm we require to compare these parameters listed below:

Average utilization: Average utilization is the average percentage of time during which the server is busy processing jobs during a simulation

Power Utilization: Power utilization can be defined as the power consumed in kWh during the simulation.

Average queue length: This is the average size of the queue of a server during a simulation.

Request failure count: Total count of requests failed during a simulation.

Request completion count: Total count of requests Completed during a simulation.

Average start time: This the average of the start time of all the requests generated during the simulation.

Average finish time: This the average of finish time of all the requests generated during the simulation.

Scheduling delay: The times taken to find a suitable server for a set of given requests.

Make span: The time taken to complete all requests over a given cloud environment.

Failure probability: the probability of failure of each request is a given system.

CHAPTER 2

LITERATURE SURVEY

2. 1 LITERATURE SURVEY

2.1. Introduction

In this section, we have discoursed various existing approaches from the field of resource allocation, load balancing, green computing and trust management. This section aims to identify the research gaps and focus on the current state of artwork in the field of resource allocation, load balancing, cost efficiency and green computing.

2.2. Resource Allocation

Many researchers have done research and introduce us some beneficial and optimal scheduling algorithm. [22] Proposed a modified Min-Min algorithm, this chooses the task with least completion time and schedule to serve accordingly. Author has proposed load balancing Min-Min algorithm which having basic properties of Min-Min algorithm and consider minimizing completion of all request. In this proposal three level of service models are used.

1. Request manager- To take request and forward to Service managers.
2. Service manager- various manger works or task and dispatch them to respective service node.
3. Service Node- Service node provide service to request which came to request mode

They have merged two approaches (OLB Opportunistic load balancing and load balance minimum) scheduling algorithms in this model. The main focus of combined approaches is to distribute the request or dispatched task basis of their completion time to suitable service node via an agent. This approach not saying about main system, suppose if request are somehow moving or scheduled in the same server and due to lots of load sever need more power to complete these

request and more physical heat will generate and to stop heating system need an external cooling system which also lead to extra power source and one more important thing is due to overheating system performance slow down The same way [23] proposed and another algorithm for task scheduling, this paper proposed VM resource allocation basis on genetic algorithm to avoid dynamic VM migration to completion of request. They have proposed a strategy to share or allow resource equally to VM so it can work fast and minimize response time to subscribe. They also proposed hotspot memory (virtual memory) assignment and dispose that after completion of request via remapping of VM migration. Here VMware distribution tool is used to schedule computation work in a virtual environment. As genetic algorithm characteristics is to find best, fittest VM in terms of Cloud computation.

This paper checks fitness of each VM and schedule task accordingly. When creating a VM a process executes to create that and increase process work that also lead to more process and increase energy consumption. Hu, Jinhua et al.[24] Proposed another scheduling algorithm, this paper proposed an approach for collective collaborative computing on trust model. The trust value taking as a factor for task scheduling, trust value mutually took from consumers as well service provider, which make it fail free execution environment. Here they have proposed a mathematical equation to calculate the Reputation point which enhances the reputation of VM in terms of fast execution and type of task. If the reputation of VM is high them more task allocation will be happening to that VM. To calculate Reputation many factors have to consider which also reflect QoS of cloud computing. This paper also proposed a way to serve a request reliability, as well trust management with a reputation of VM factor which are lead to trustworthy. Trust has calculated by a mathematical equation and schedule accordingly.

Hu, Jinhua et al. [25] proposed a live VM migration algorithm, this paper proposed a method for VM live migration with various resource reservation system. VM migration is taking place on the basis of source machine load, if the load is high then it can wear, during execution of the request it migrates the VM to another server or data centers to complete the task without interruption for better

performance. Resource reservation done both sides, i.e., Source machine and target machine as well will in such manner CAP (maximum availability of CPU) allocate them and adjust memory resource dynamically. At the end of target machine, they properties time bound program which will keep monitoring for cup resource utilization. Memory Reservation done by allocating crating certain number of VM and when the migration process comes into existence these VM got shut down to evacuate the space to migrate VM. Sometime it may be possible that target machine not having enough space to migrate in such condition that physical machine should remove from candidate machine for migration and which physical machine having the capability or enough space will lead to migrate VM. This paper implemented and simulated using Xen Virtualization.

Barroso et al.[26] This paper proposed an algorithm, dynamic and integrated resource scheduling algorithm for cloud data center which balance load between servers in overall run time of request, here they are migrating an application from one data center to another without interruption. Here they are introducing some measurement to ensure load balancing. They have given a mathematical reputation to calculate imbalance load to calculate average utilization to its threshold value to balance load. To implement DAIRS they have used physical server with physical cluster and Virtual servers with virtual cluster. Application migration saves time instead of migrating whole VM data. Zhanjie Wang [27] proposed an dynamic algorithm for resource allocation in cloud using fuzzy logic and pattern recognition based on power and storage parameters. The propose algorithm is derived from Fast Bid algorithm. The algorithm tries to improve the network traffic and communication load over the system. The algorithm shows better result than Min-Min algorithm in term of make span and network load

Parvathy S. Pillai [28] et al. proposed a novel resource allocation algorithm derived from game theory for resource allocation in cloud. In this work author has used uncertainty principle of game theory for allocation of virtual machines in cloud. This work improves the communication cost and resource wastage over the system. Abdullah Yousafzai [29] et al. surveyed and reviewed resource allocation algorithm in cloud. This work contributed an review and comparative study or

current state of art cloud resource scheduling and allocation algorithms for cloud. Moreover this article proposes an taxonomy for resource allocation in cloud environment, which shows various ways to solve the issue of resource allocation and different aspects of resource allocation. Figure given below shows the taxonomy.

Many other resource allocation algorithm are been proposed [88 -97] using various dynamic techniques to improve the performance of the system are been studied.

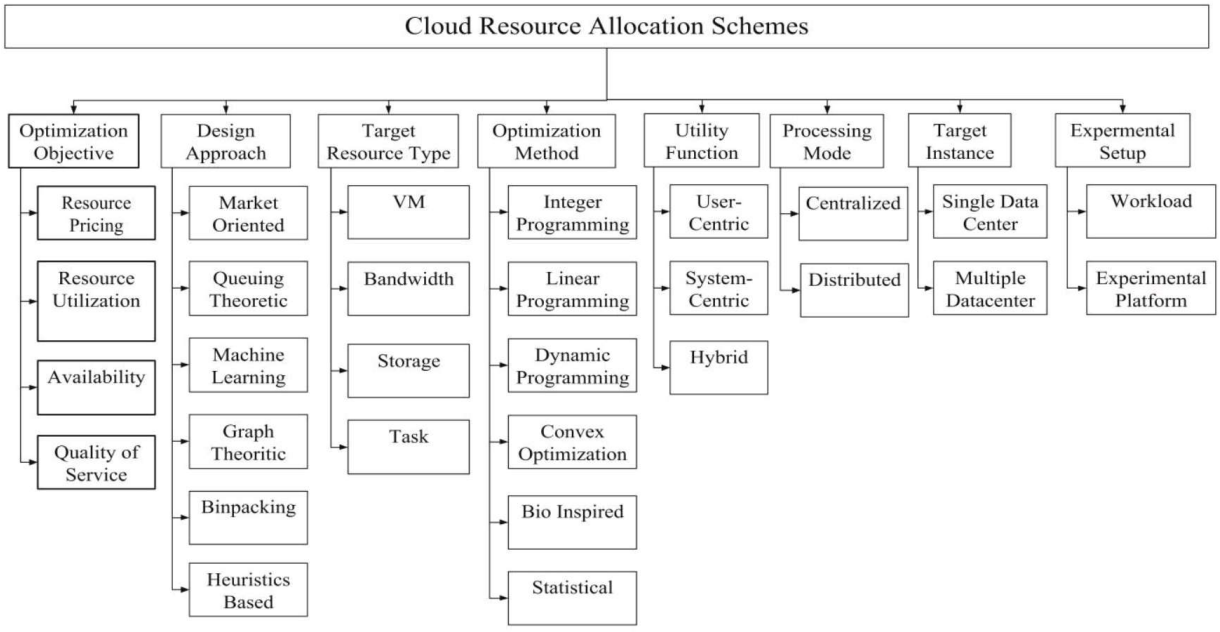


Figure 2.1 Taxonomy for resource allocation in cloud.

2.3. Load Balancing Algorithms

Various load balancing tactics have been proposed till now which can also be classified into static and dynamic in nature [30]. Yamamoto et al. [31] proposed a disbursed strategy to balance the load using replication of data. Authors have proposed two replication approaches 1) in the route random replication procedure, replicas saved within the peers along the trail of soliciting for to look. 2) In the course adaptive replication procedure replicas saved most effective within the peers in step with their likelihood of replication. This paper does not use the knowledge about the ability of servers for selection of server.

In [32] Rao et al. have offered a framework for load balancing in distributed environment, named as HiGLOB. Right here, authors have used two principal add-

ons 1) histogram supervisor - generates a histogram to preserve a global information regarding the distribution of the load within the system, and 2) load-balancing manager - reallocates the load at any time when the node becomes overloaded or under loaded. Nevertheless, there's overhead associated at the same time setting up and preserving the histograms.

Zeng et al. [33] have proposed a load re-balancing algorithm to work out the crisis of load balancing in distributed environment. They have additionally ensured the reliability of the process where one chunk of a file and two duplicate copies are allocated in three exceptional different servers at a time. In this algorithm author uses the master server periodically for checking of chunk servers and to differentiate which chunk server is over- loaded and which is not. Nonetheless, this master server turns into a single-point failure for the load balancer. Fan et al. [34] have proposed an adaptive load optimization algorithm (AFLBA) for the Hadoop distributed file procedure which uses two modes: 1) disk utilization expense system and 2) carrier blocking off rate system. The proposed algorithm uses the storage average utilization of each data node and probability of blocking consumer request of each knowledge node. Since this algorithm isn't disbursed so it creates a performance bottleneck node within the HDFS.

Hasio et al. [35] and Chung et al [36] have proposed an improved load balancing algorithm for distributed file system to overcome the issue of bottleneck and improve the performance of system. They have proposed to use CHORD protocol for creation the node server.

Many other load balancing approaches are been proposed to avoid the condition of over loading [101, 102] using max-min, min-min and dynamic strategy.

2.4. Power Efficient Algorithms

Several researchers have introduced various models and methods to conserve energy. Some of them are discoursed below.

Louis Rilling et al. [37] proposed a virtual infrastructure optimization solution using the ant colony optimization algorithm for finding better paths through graphs. The most common approach while performing workload consolidation is that the

workload is allotted to a physical machine (e.g. CPU) and those resources which require excessive provisioning are converted into a lower power state.

Osvaldo Adilson de Carvalho Junior et al. proposes the use of a function that can ensure the most appropriate behaviour to the principles of Green IT but not the quality of service. For this he proposes the use of GreenMACC (Meta-scheduling Green Architecture) and its module LRAM (Local Resource Allocation Manager) to automate the execution of all scheduling policies implemented in the Scheduling Policies Module so as to provide Quality of Service in Cloud Computing and determine its flexibility. [38] Task consolidation is an efficient method which is used to reduce power consumption by increasing the resource utilization but due to task consolidation resources may still draw power while being in the idle state. Young Choon Lee et al. has introduced two algorithms to maximize the utilization of resources of the cloud. The two algorithms are ECTC and MaxUtil. ECTC works on the premise of calculating the energy which is being used by a particular task when there are simultaneous tasks running parallel with it, and then it is compared with the optimal energy which is required. MaxUtil focuses more on the mean usage of a particular task when it is being processed. [39]

Dzmitry Kliazovich et al. presented a simulation environment for data centers to improve their utilization of resources. Apart from working on the distribution of the tasks, it also focuses on the energy used by the data center components. The simulation outcomes are obtained for various architectures of data centers. In [40] Robert Basmadjian et al. proposed the use of proper optimization policies reducing the power usage and increasing the resource utilization without sacrificing the SLAs. He developed a model which worked on incrementing the capability of the processor to process tasks. [41] Zhou Zhou et al proposes a Three Threshold Energy Saving Algorithm [TESA] which has three thresholds to divide hosts between heavy load, light load & middling load. Then based on TESA 5 VM migration policies are suggested which significantly improves energy efficiency. [42].

Dung H Plan et al. proposed GreenMonster protocol which improves renewable energy consumption while maintaining performance by dynamically moving services across IDCs. GreenMonster uses Evolutionary Multi-objective Optimization Protocol [EMOA] to make service placement and migration

decisions. [43]. Liang Liu et.al. proposed a new VM architecture which has capabilities of Live Virtual Machine Migration, VM placement optimization and online VM Monitoring. This architecture gives us a considerable energy saving. [44]. Aman Kansal et al. proposes a power metering solution for virtual machines. The proposed solution has a very small runtime overhead and provides accurate and practical information for power capping to improve the energy efficiency of the datacenters. [45].

Abbas Horri et al. [46] proposed a novel approach to improve the power efficiency of system for cloud infrastructure based on the resource utilization history of virtual machines in cloud.

The first work in large-scale virtualized datacenters has been proposed by Nathuji and Schwan [47]. In their proposed method, the resource management is split into local and global managers. Local manager coordinates power management methods of VMs in each host because the authors assumed that VM guests have a power aware OS. Global manager monitors the performance of multiple hosts and selects the appropriate host for requested VM migration. However, in situation that the guest OS is non-power-aware, this power management method may be inefficient [47]. Salimi and Sharifi in [48] proposed an approach to schedule a set of VMs on a shared PM. The goal of the scheduling algorithm was to minimize the execution times (Makspan) of batch applications running on VMs based on considering the interferences of concurrent VMs. To identify the interference, they first presented an interference model in terms of number of concurrent VMs, processing utilizations of VMs and also the network latency. Nasrin Akhter & Mohamed Othman [49] surveyed and reviewed energy aware resource allocation algorithm in cloud. This paper reviews lasted proposal made for improving the energy efficiency of system. Major contribution this work is the broad study and classification of various ways to improve power consumption in cloud environment. The figure below shows the taxonomy proposed

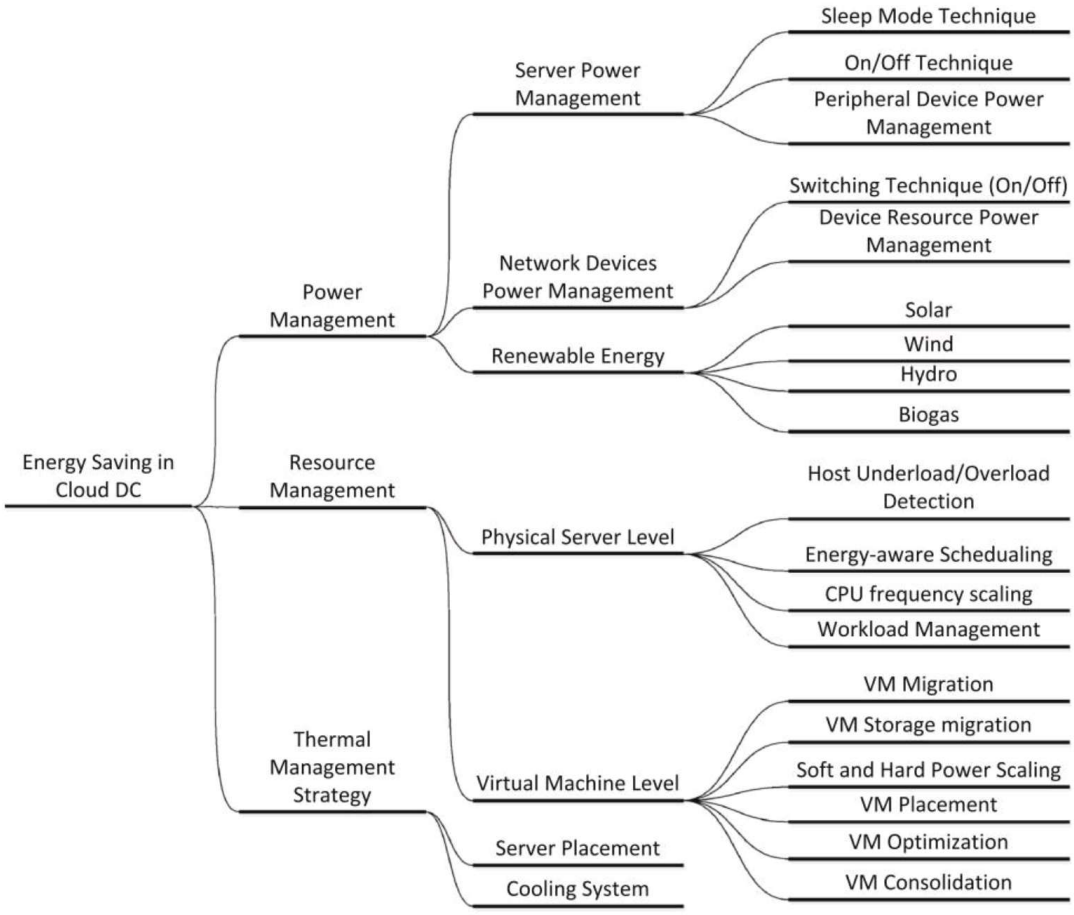


Figure 2.2 Energy aware allocation taxonomy

2.5. Cost Efficient Algorithms

Li Chunlin et al. [50] proposed and cost and energy aware resource provisioning algorithm for cloud. This paper presents the cost and energy aware service provisioning scheme for mobile client in mobile cloud. Proposed work proves to be cost optimal and energy efficient as compares to simply cost aware allocation algorithms. Ehsan Ahvar et al. [51] has proposed an network aware cost optimal algorithm. This algorithm takes into consideration network performance and cost for resource allocation and selection of best server, using artificial algorithm to perform better than typical greedy heuristics. Khaled Metwally et al. [52] proposed a Mathematical modeling based on Integer Linear Programming (ILP) technique to solve optimally the resource allocation problem. However, ILP technique is knows for solving well known problem of scheduling in operating system. Author has proposed a model to use linear programming for selection of appropriate resource.

Balaji Palanisamy et al. [53] proposed a cost aware allocation algorithm for MapReduce in cloud. This article presents a new MapReduce service model for cloud named Cura. Cura is cost efficient MapReduce model and cloud service to select the resource at run time for distributed problem with least cost and most efficient resource. Cura is also responsible for creation and selecting of cluster for dealing with workload. It also includes, VM-aware scheduling and online virtual machine reconfiguration, for better management and reconfiguration resources.

2.6. Behavior based algorithm

Behavior are the algorithm which are inspired from behavior of nature and behavior of animals and other living organism around us and their hierarchical evolvement over the decades. These behaviors inspires us to make decisions based on previous behaviors or the environment for making better decision that may be prediction or forecasting. Some of these algorithms proposed in the field of cloud computing are discoursed below.

Bei Wang &Jun Li [54] proposed a genetic algorithm based load balancing algorithm. In order to boost the search efficiency, the min-min and max-min algorithm are used for the population initialization. But these may stuck in local minima and to find best solution genetic algorithm is proposed. Proposed algorithm proves to provide better solution but the scheduling delay to find best solution is much higher than min-min and max-min algorithms. Keke Gai [55] proposed and cost efficient data / storage allocation algorithm using genetic algorithm for video and metadata storage over cloud. This algorithm aims to provide heterogeneous memory storage space over cloud with least cost using genetic programming to select cheapest service provider. Output proves that the proposed algorithm proves to provide improved communication costs, data move operating costs and energy performance.

Lizhen Cui [56] proposed a genetic algorithm based replica management algorithm for cloud. Author has proposed a tripartite graph based model to formulate the data replica placement problem and propose a genetic algorithm based data replica placement strategy for scientific applications to reduce data transmissions. The proposal provides better performance that random selection policy in Hadoop Distributed File System. Jasraj Meena [57] proposed a cost efficient genetic algorithm to optimize the cost for work flow schedule rather than for single tasks.

The proposed algorithm proves to execute the workflow with least cost. The algorithm is been tested over popular workflow like Montage, LIGO, Cyber Shake, and Epigenomics. Anjuli Garg [58] proposed a honey bee life cycle based task scheduling strategy for cloud. Author has taken into care utilization and task size to schedule the task and select the server which can execute with least execution time. Anqi Xu [59] proposed an Particle Swarm Optimization for task scheduling for cloud infrastructure to improve the Quality of Service of system. The author has taken into consideration multi objective to improve Make span and cost. The algorithm proves to perform better than ACO and min-min algorithm. Bohrer et al. [60] proposed a most known base scheduling algorithm ACO (ant colony optimization) they proposed ant colony optimization algorithm to load balance by distributing request in a cloud computing environment. This paper proposed LBACO with dynamic load balancing strategy to distribute load among the node. The problem with traditional ACO in cloud is that it's a schedule task to most frequent (high pheromone intensity) node, if what if node is bearing heavy load in such situation may create a problem of overhead. This paper proposed and LBACO algorithm to reduce such problem. In this algorithm decrease the time of computation and monitor load on each VM with tracking previous scheduling. Xiaobo et al. [61] proposed and Real-time VM provisioning model, which is based on energy models which follow a Min-Price RT-VM Provisioning to allocate VM.Suraj, S. Rin et al. [62] proposed a genetic algorithm for task allocation in cloud environment with least execution time and maximum resource utilization.

Many other proposal made [82-86] using ACO, genetic algorithm and other learning based algorithm are been studied. Jaradat [87] proposed a Big Bang-Big Crunch optimization algorithm to solve the problem of scheduling classed for a timetable. This algorithm has proved to perform better than existing GA based algorithm.

2.7. Trust Models

Numerous trust models have been proposed in cloud. MohdaIzua Mohd Saad proposed a novel data provenance trusted model to provide secured access to data provenance via a secured communication channel [63]. This model also proposes

consolidation storage with logging for virtual storage at physical layer in cloud environment. As shown in figure 1.

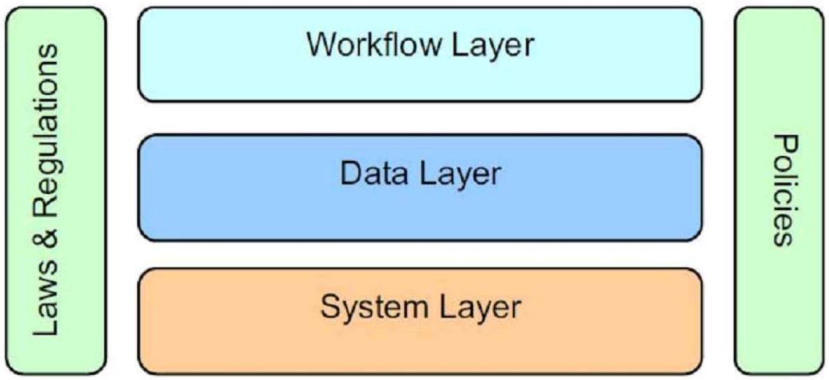


Figure 2.3. The trust cloud framework.

WenAn Tan proposed a trust service-oriented workflow scheduling algorithm [64]. The scheduling algorithm uses a trust metric that is combination of direct trust and recommendation trust. Proposed model also provided balancing policy to balance user requests, based on time, cost, and trust. Rizwana A.R. Shaikh proposes a trust based solution in terms of a trust model that can be used to calculate the security strength of a particular cloud service [365]. Proposed algorithm uses trust value for selecting a trusted cloud service.

[66] Xiaodong Sun introduces a trust management model based on fuzzy set theory and named TMFC including direct trust measurement and computing, connecting, and trust chain incorporating where the issue of recommended trust has been addressed to find the miss behavior of intermediate middle nodes. And this proposed model is designed for the cloud users to make decision on whether to use the services of some cloud computing providers by using trust value sets about providers and then finding trust relationships among them. QiangGuo introduced a definition of trust in cloud systems and the properties of trust are analyzed [67]. Based on the properties of trust of a server, a trust evaluation model called ETEC is proposed. Proposed trust model includes a time based comprehensive evaluation method for calculation of direct trust and a space evaluation method for calculating

recommendation trust of server. For computing the trust in cloud, an algorithm based on the trust model is given. Experimental analysis shows that the proposed model can calculate the trust vale of server effectively and reasonably in cloud computing environments. Xiaoqiong Yang also proposed A Statistical User-Behavior Trust Evaluation Algorithm Based on Cloud Model for statistic behaviors. Proposed algorithm used threshold for each type of behaviors and each user's performance and its membership status in cloud [68]. Then the membership degree and the behavior weight will be used to calculate the user's trust using a simple normalization function. Proposed algorithm uses the evaluated domain trust and recommendation trust, behavior trust for users' further dynamic authorization of access control and request load balancing. Junfeng Tian proposed a Trusted Control Model of Cloud Storage with access control (TCMCS) to handle all the interactions between a client and cloud storage to ensure the secure user access and data manipulation. The proposed trust model is responsible for managing different cloud storage and manages security and integrity of user data over the cloud. Since users only need to care about their own business logic and the development of application program is greatly simplified [69]. Proposed model can be specified as shown in figure 2.

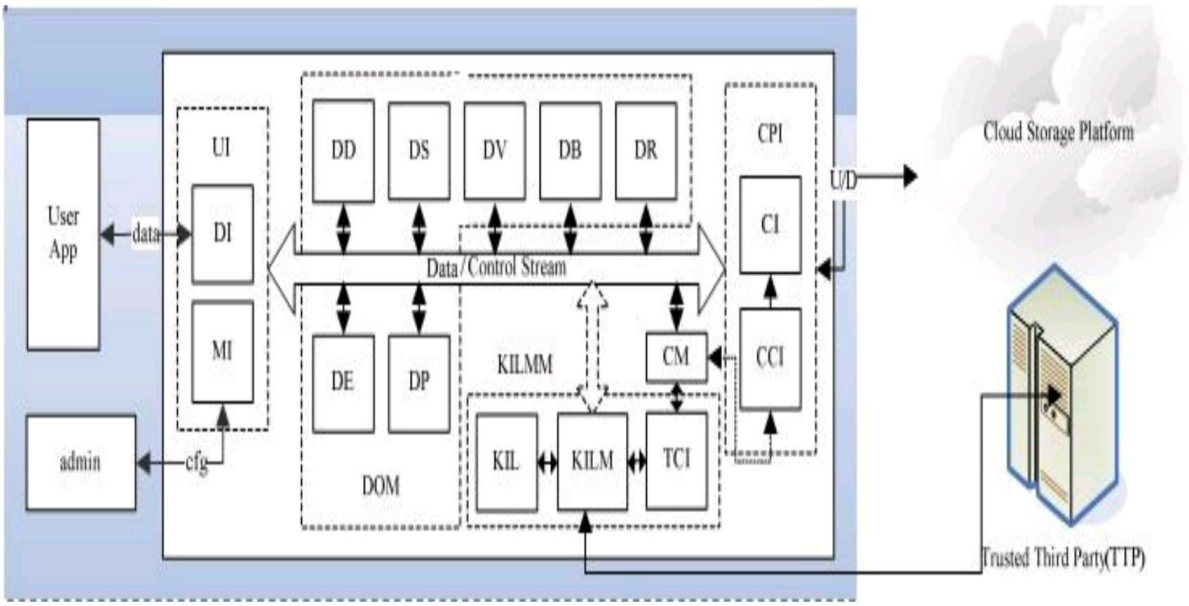


Figure 2.4 The logical structure diagram of TCMCS.

In [70] gupta has proposed a QoS Based Trust Management Model for Cloud IaaS that is suitable for trust value management for the cloud IaaS parameters. Proposed

a scheduling algorithm based on trust value is done for better resources allocation and enhance the QoS provided to the users. In this paper, an approach for managing trust in Cloud IaaS is proposed.

Various other trust models are also been proposed [103 -106] to improve the system performance and reliability of reliable computing.

CHAPTER 3

EXISTING METHODOLOGY

3. PROBLEM ANALYSIS

FAULT AND DEADLINE AWARE LOAD BALANCING

With the rapid growth in technology, there is a huge proliferation of data requests in cyberspace. Distributed system/servers play a crucial role in the management of request in cloud which are distributed among the various geographical zones. Many of time the system gets over loaded due to few of servers with high number of request and some of servers being idle. This leads to degradation of performance of over loaded servers and failure of requests. On these over loaded servers average response time of server increases. So there is a requirement to design a load balancing algorithm to optimize resource utilization, response time and avoid overload on any single resource.

The management of data in cloud storage requires a special type of file system known as distributed file system (DFS), which had functionality of conventional file systems as well as provide degrees of transparency to the user, and the system such as access transparency, location transparency, failure transparency, heterogeneity, and replication transparency [71]. DFS provides the virtual abstraction to all clients that all the data located closest to him. Generally, DFS consists of master-slave architecture in which master server maintains the global directory and all metadata information of all the slave servers. Whereas, slave represents a storage server that stores the data connected to master server and other storage servers as well. This storage server handles the thousands of client requests concurrently, in DFS. The load distribution of requests on these storage servers is uneven and lead to performance degradation overall. Resources are not exploited adequately, because some server gets too many requests and some remain idle. In a distributed storage system, load can be either in terms of requests handled by a server or storage capacity of that server or both.

In this section, we have proposed a set of approaches that balances the load of servers and effectively utilizes the server capabilities and resources. The main contribution of this work is to improve the average resource utilization of system and removing hot spots and cold spots in the system i.e. the unbalancing of requests over the system should be removed.

3.1. Approach 1: Fault and Load Aware Load Balancing in Cloud Storage

In this approach, we have proposed a Fault and Load based Load balancing algorithm (FLBLBA) that can balance a load of servers dynamically by considering its parallel processing capability, processing time and its request queuing capacity. Proposed algorithm aims to improve the performance cloud storage system by reducing request failure count, Average queue length, average utilization, total execution time. The work is divided into various sections, where section 1 focusses on basic description of problem. Section 2 & 3 describes the proposed approach and algorithm. In section 4 we have presented experimental results and comparative study of proposed algorithm.

3.1.1. Problem Statement

Distributed file systems provide a common virtual file system interface to all users as in DFS storage servers are distributed geographically and because of this load dis- tribution of client's requests to these servers become uneven. This problem can be illustrated clearly through Figure 3.1. Here, we have taken five storage servers S1, S2, S3, S4 and S5 with their respective service rate (S_r) present in the system. Service rate of a server signifies the number of requests processed by a server in a given time. Initially at time t=0, we assume that each server receives an approximately equal amount of requests as shown in Figure 3.1(a). We have taken total 8 requests to illustrate the scenario of our problem statement. In the second case as shown in Figure 1(b) after time t=2, each server process the client's requests as per its service rate and server S1 requests gets over much earlier than other servers and S1 becomes idle. Server S3 and S5 are fully loaded and takes their time to process all requests. From this scenario, we can say that distributed file system does not utilize each server

efficiently. In real- world situation, these requests are too large as compare to server service rate.

So in order to increase the system performance some requests which are in queue must be migrated to the idle servers or least loaded server and completes the request without failure. Our aim is to avoid queue like situations, utilizing the capability of each server efficiently and fulfill maximum request without failure.

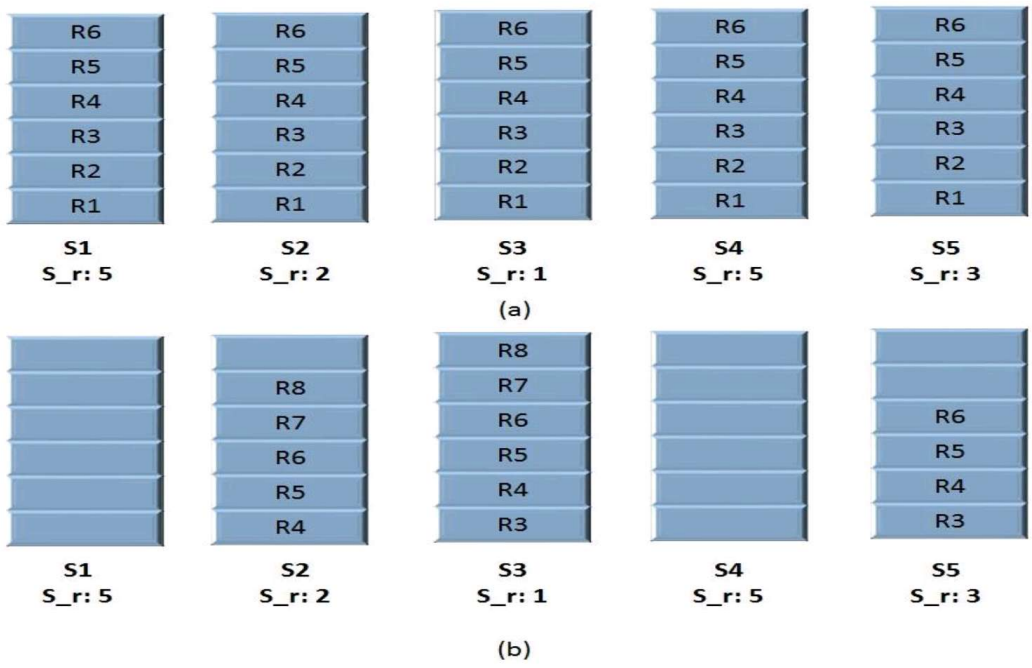


Figure 3.1 Problem statement for load balancing (a) at time t=0, servers receive equal amount of client requests. (b) at time t=2, scenario of servers after processing the receive requests.

3.1.2. Problem Approach

Here, we have proposed a Fault and Load based Load balancing algorithm (FLBLBA) that can balance the load of servers dynamically by considering its parallel processing capability, processing time and its request queuing capacity. Proposed approach takes four main parameters of a server 1) Server request queue size - buffer space to store the client requests to be handled by the server. 2) Server service rate (λ) - the number of CPUs available for processing the client request in a server. 3) Processing time (S_T) – time takes to process a request which differs from server to server.4) Fault rate. Modern servers are equipped with many features like multiple CPUs, large storage, high I/O capability etc. We have chosen the multiple CPUs feature as a main parameter for load balancing of our proposed approach.

Following are the few assumptions that we have considered for our proposed approach:

- It is assumed that all the servers belong to same organization which can be geographically apart from each other. So each server maintains the replica of every server data.
- It is also assumed that all servers are strongly connected with each other through high bandwidth medium.
- Each server maintains global view which contains the information of its neighbors through master server.

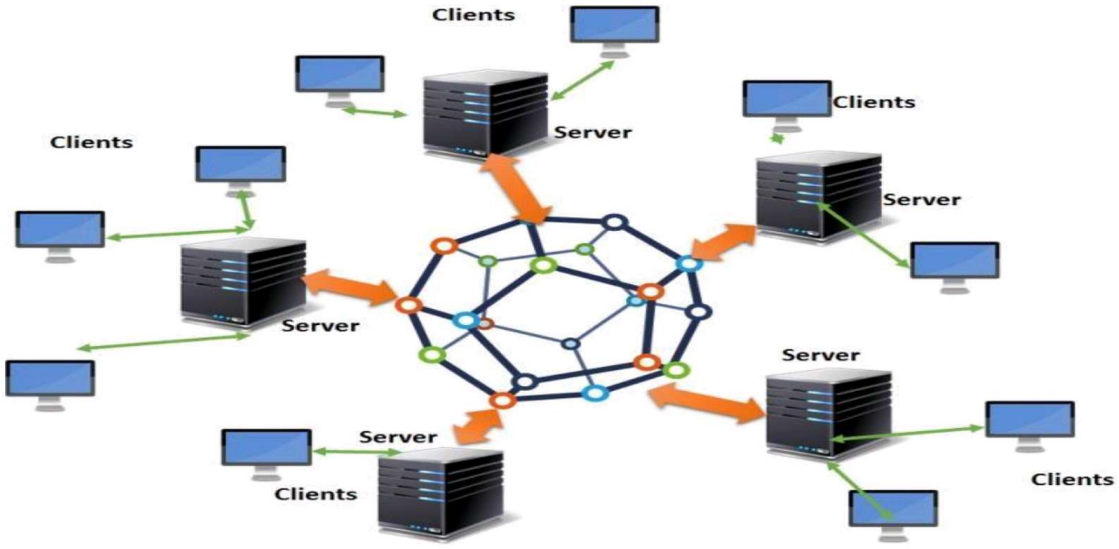


Figure 3.2 Organization of distribute storage servers.

Figure 3.2 shows the general scenario of distributed storage servers. In figure 2, there could be N connected servers where N €{1,2,3 ….. n-1}, in the system. Each server has following properties such as request queue, number of CPUs, storage capacity. Clients send their requests to the respective server. Many times the incoming request rate (ρ) increases exponentially to a particular server. This is because of the series of client's requests to that data that is stored within the server. In case, when a server gets too many requests than server buffers them in their request queue and the size of request queue gets increases dynamically only upto its predefined threshold limit. Once, the request queue breaches the threshold limit than server is considered as overloaded server and triggers the load balancer. Load balancer classifies the least loaded server on the basis of their request queue and processing capacity. As soon as

the least loaded server gets classified than overloaded server migrate its load to that server and balances the load. Various notations are used in the proposed approach and represented as follows: ρ - Current queue size of server.

λ_i - Service rate that is number of request processed simultaneously on a server.

S_T -Service time is the time taken by server to process the request

$Q\_L_{Current}$ -Current queue length of server

$Q\_L_{Threshold}$ - Threshold limit of server request queue.

ΔLi - additional load on server i.

Wi - Waiting time for a request at server i.

FT_i - Count of request failed.

FR_i -Fault rate that is the numbe of request failed due to system failure over time t.

F_j - Fitness value of neighbors of server i. (j €{1,2,3 ….. n-1}

We have considered the real world scenario where the server request queue size and service rate changes with respect to time t dynamically and represented as $\delta\rho$ and $\delta\lambda$ respectively

$$\delta\rho = \frac{\rho}{\delta t} \; and \; \delta\lambda = \frac{\lambda}{\delta t} \qquad (3.1)$$

Fault rate of a server can be given as:

$$FR_i = FT_i / time \qquad (3.2)$$

Storage server is said to be overloaded if:

$$\delta\rho > Q\_L_{threshold} \qquad (3.3)$$

When server *i* where *i* €{1,2,3 ….. n-1} is overloaded then it calculates the amount of extra load ΔL i on that server which can be calculated as follow:

$$\Delta L_i = Q\_L_{current} - Q\_L_{threshold} \qquad (3.4)$$

The condition when a load balancer module gets triggered on the overloaded server *i* is given below:

$$T(i) = \begin{cases} 1, & \Delta L > 0 \\ 0, & otherwise \end{cases}$$

(3.5)

$T(i)$ = Triggering function.

Once, the load balancer module is triggered, server i find the least loaded or idle server that can accommodate its load and adequately process the service requests without failure. For this load balancer calculates the fitness value Fj that can be calculated using the following fitness function:

$$\Delta M_j = Q\_L_{threshold} - Q\_L_{current} \qquad (3.6)$$

Here, ΔMj is free request queue of server j. If ΔMj is negative, then server j request queue is overloaded otherwise it is least loaded.

$$F_j = \alpha_1 .\Delta Mj + \alpha_2.\lambda_j + \alpha_3.(\frac{1}{FR_j}) + \alpha_4 .(\frac{1}{W_j}) \qquad (3.7)$$

Here, α1 and α2 are constants and may vary according to scenario such that

$$\alpha 1 + \alpha 2 + \alpha 3 + \alpha 4 = 1 \qquad (3.8)$$

For our proposed scenario, we have considered the value of ***α1*** and ***α2*** is 0.5 it is because both the parameters play the equal role in load balancing. In this way, load balancer calculates the fitness value for each neighbors of server *i.* and select that server which has maximum fitness *Fj* value, i.e. fault rate of server less than migrating server and migrate the Δ*Mj* amount of load to server *j*. Selecting the server with maximum fitness value in turn decreases the failure probability of request and completes the request as soon as possible with least waiting time.

3.1.3. Proposed Model

Proposed algorithms have been designed to balance the client requests over the servers and distribute the load over the system uniformly. Here, load balancer as shown in figure 3.3.(a) regularly exams for the request queue dimension of server and tries to restrict the problem of overloading of any server with the aid of migrating the extra request to other idle or least loaded and least faulty neighboring server in cloud. Proposed load balancing algorithm is divided into two stages. In first stage list of idle servers is created, and in second stage the server with highest fitness value and which can fulfill the request with least failure probability.

Stage I

Algorithms checks and calculate the fitness value for the neighbor server to store them in a list shown in figure 3.3(b). Load balancer utilizes this list to select the server that has highest fitness value. Load balancer calculates the waiting time over each server from above list which can be given as:

$$W_k = \frac{Q\_L_{current\_k}}{\lambda_k} \times S\_T_k \quad (3.9)$$

Equation 1 shows the W_k waiting time of i^{th} request at server 'k'.

Stage II

In second stage load balancer then finds the server with least waiting time, least fault rate and highest service rate i.e. highest fitness value from the list.

A proposed algorithm also tries to improve the server response time by selecting the server having least CPU utilization. In this way, proposed algorithm utilizes the idle or underutilized server to increase the overall performance of the system and reduce requests failure over the system by reducing the probability of request failure.

Algorithm : Load balancing

1. **FLBLBA** (Server s, $Q\_L_{current}$,λ_k,S_T_k, FR_i)

Input: Server s, Queue length $Q\_L_{current}$, service rate λ_k, service time S_T_k, fault rate FR_i

2. s← server
3. $Q\_L_{current}$ ← current queue size
4. λ_k ← Service rate of server k
5. S_T_k← Service time of server k
6. FR_i ← fault rate of server k
7. Compute W_k
8. **if** ($\delta p > Q\_L_{Threshold}$) **then**
9. check server queue status.
10. Add request to queue;
11. Process_request();
12. **else**
13. server is overloaded
14. S ← Find_server(server_neighbour_list L)
15. find under loaded server.
16. S ← migrate request
17. Goto step 7

Output: Load balances the request.

Figure 3.3 (a) Proposed load balancing algorithm.

Algorithm : To find suitable server

1. **Find_server**(server_neighbour_list L)

Input: server_neighbour_list L

2. **for** k=1 to L.size()
3. S1←L.get()
4. F_k= **α1** ΔMj + **α2**λ + **α3** $(1/FR_j)$+ **α4** $(1/W_j)$
5. temp_list t← F_k
6. **end for**
7. L2 = Sort(t)
8. S2 ←min (L2)
9. return S2

Output: The server with minimum fitness value.

Figure 3.3 (b) Find a neighbor server algorithm.

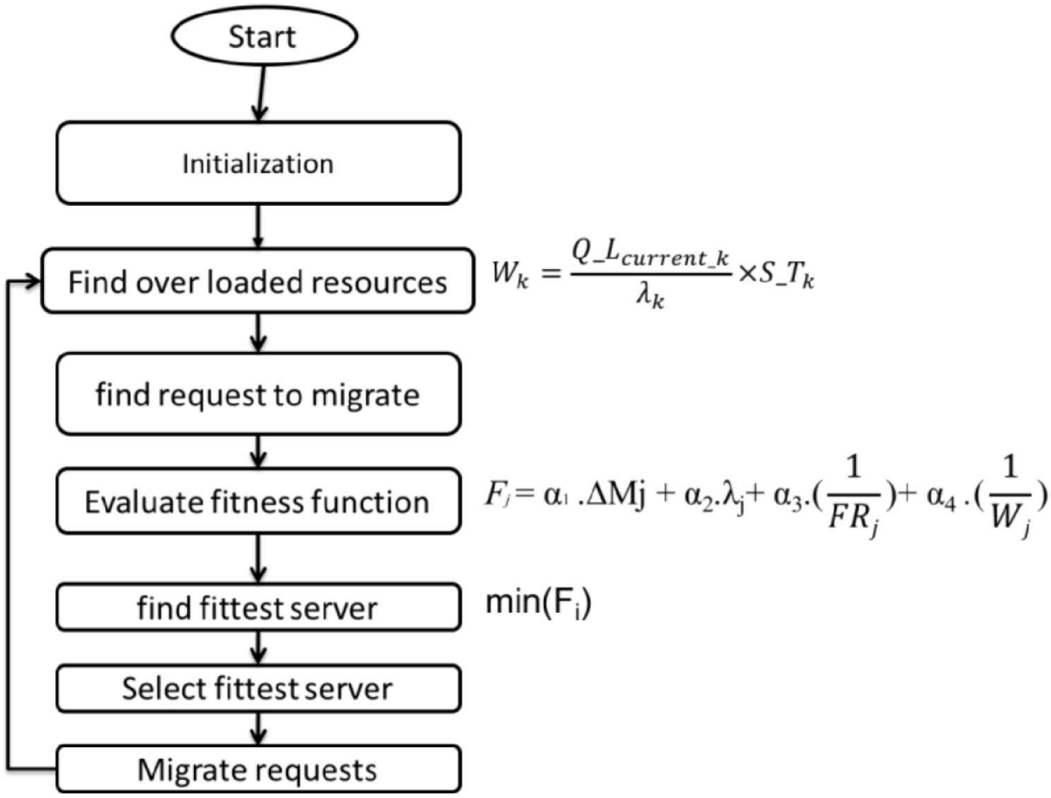


Figure 3.4 Proposed algorithms flow diagram

Figure 3.4 shows the flow of the algorithm with various phases of algorithm and interaction among them to find the fittest server for each request.

3.1.4. Experiment and Result

Performance analysis of proposed FLBLBA algorithm is finished using CloudSim [11] simulator where we now have thousands of requests to be completed by 12 storage servers. All of the servers work concurrently with constant quantity of CPU cores to process the client request rapidly. Each server has a request queue to buffer the incoming client requests, storage ability to store the data and fulfill the client requests.

For the given problem statement in section 3 where the load is unbalanced, it is assumed that half of storage servers get client requests and others remain idle. Our motive is to equally distribute the received client requests among the servers to avoid the scenario of overloading. In the simulation scenario numbers of storage servers are kept fixed with varying number of requests handling. We have also compared the

obtained results with the least load balancing algorithm. Following table depicts the configuration parameter for our simulation environment.

Table 3.1: Experimental parameters used for simulation environment

No. of client requests	No. of Servers	No. of CPU cores available per server	Storage capacity of servers (GB)	Server queue length
800	12	7	500	20
1000	12	7	500	20
1200	12	7	500	25
1800	12	7	500	25
2400	12	7	500	25

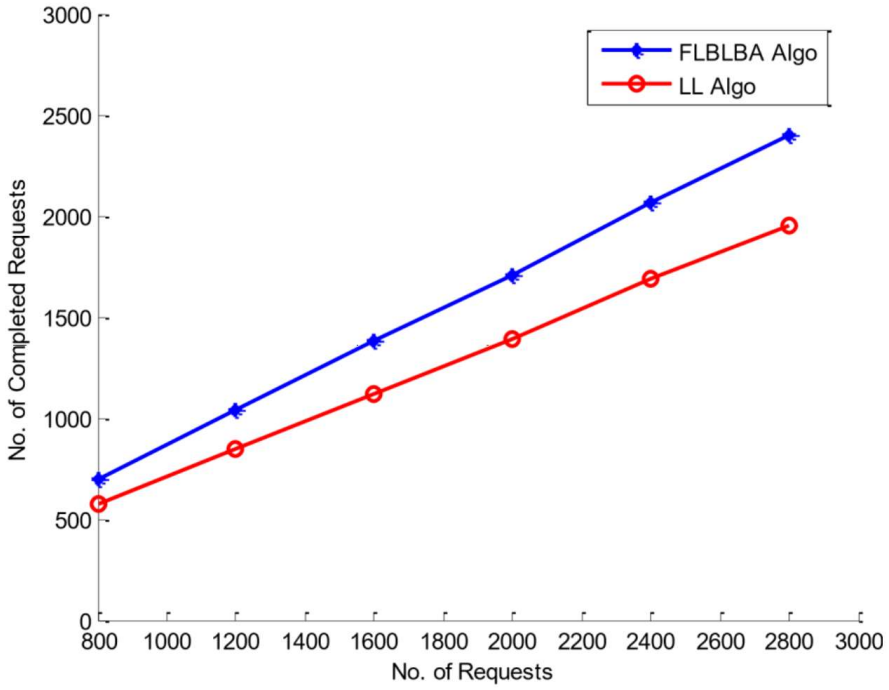


Figure 3.5 Number of request completed

Figure 3.5 shows the number of processed client requests by server in a given time. Here, Figure 3.5 represents the graph between numbers of sent requests vs. numbers of completed request whereas Figure 3.6 represents the graph between no. of sent

requests vs. no. of failed requests for the proposed and least load algorithms. In least loaded algorithm when any server get overloaded then load balancer selects the server of which request queue is least loaded without considering the CPU parameter. For the pro- posed algorithm we have considered the CPU parameter and from obtained results as shown in Figure 3.5, Figure 3.6 and Figure 3.7 that the proposed algorithm perform much better over the least load algorithm. Figure 3.8 shows that the proposed FLBLBA algorithm improves the average utilization of the system drastically over increasing requests due to improvement in total request completed. In all set of client requests, proposed algorithm process more number of client's request with better overall response time as shown in Figure 3.7.

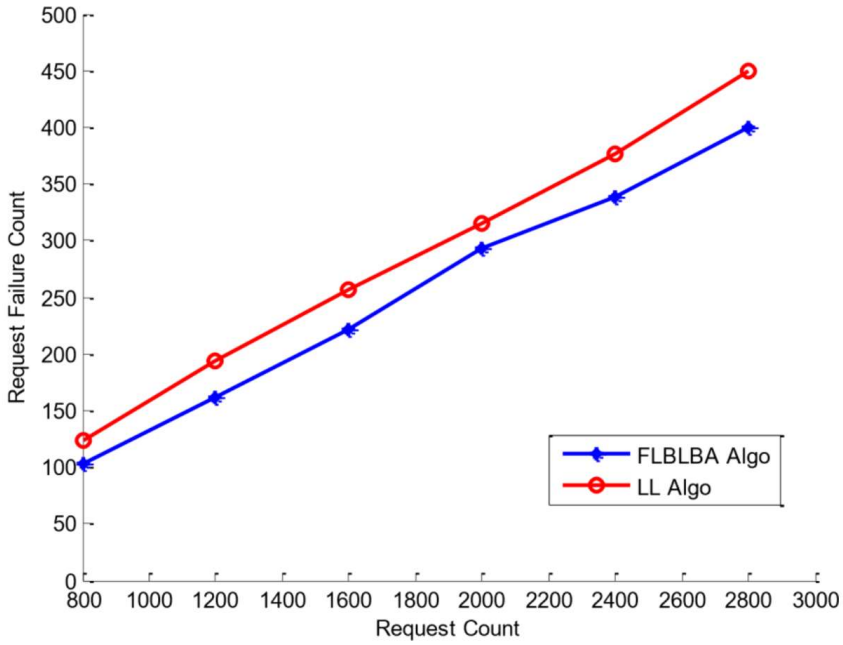


Figure 3.6 Number of sent requests vs. no. of failed requests.

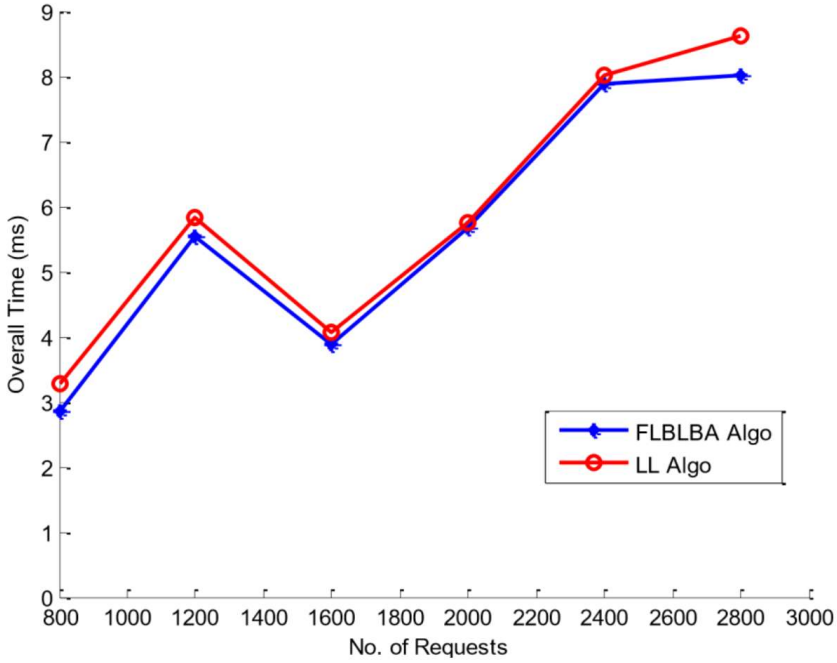


Figure 3.7 Overall response time.

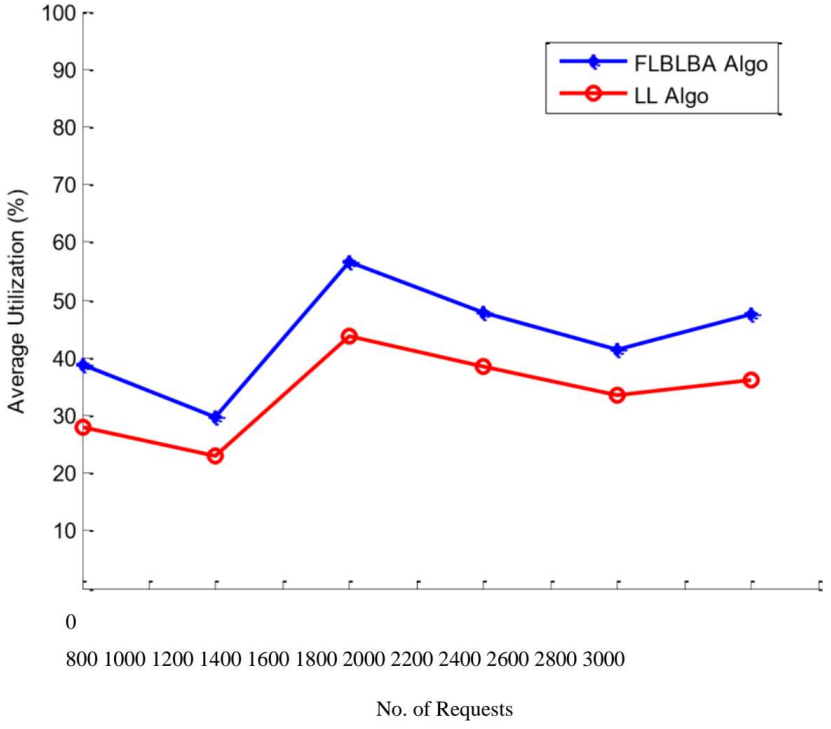


Figure 3.8 Average utilization of system

3.2. Approach 2: Deadline Aware Load Balancing of Distributed Servers in Distributed

In this approach, we have proposed a Deadline based Load balancing algorithm (DDLBA) that can balance the load of servers dynamically by considering its parallel processing capability, processing time and its request queuing capacity. Proposed algorithm aim to overcome the failures due to long waiting time rather than network or system failure. Algorithm takes care of request which have small deadline and need to be assigned to a server with high service rate that can fulfill the request within the deadline. Proposed algorithm improves the performance of cloud storage system by reducing request failure count, Average queue length, average utilization, total execution time. The work is divided into various sections, where section 1 & 2 describes the proposed approach and algorithm. In section 3 we have presented experimental results and comparative study of proposed algorithm.

3.2.1. Proposed Approach

Here, we have proposed a Deadline based Load balancing algorithm (DDLBA) that can balance the load of servers dynamically by considering its parallel processing capability, processing time and its request queuing capacity. Proposed approach takes three main parameters of a server 1) Server request queue size - buffer space to store the client requests to be handled by the server. 2) Server service rate (λ) - the number of CPUs available for processing the client request in a server. 3) Processing time (S_T) – time takes to process a request which differs from server to server. Modern servers are equipped with many features like multiple CPUs, large storage, high I/O capability etc. We have chosen the multiple CPUs feature as a main parameter for load balancing of our proposed approach.

Following are the few assumptions that we have considered for our proposed approach:

- It is assumed that all the servers belong to same organization which can be geographically apart from each other. So each server maintains the replica of every server data.

- It is also assumed that all servers are strongly connected with each other through high bandwidth medium.
- Each server maintains global view which contains the information of its neighbors through master server.

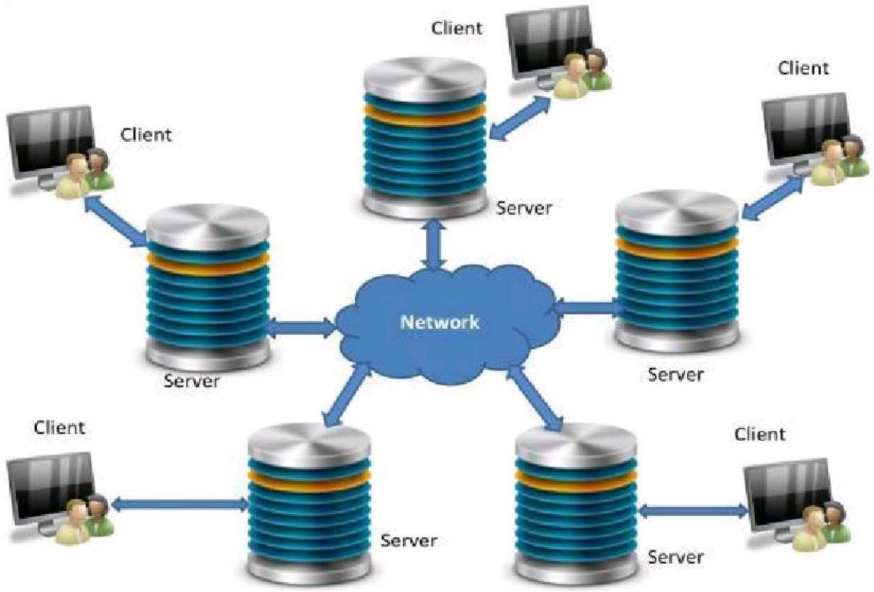


Figure 3.9 Organization of distribute data servers.

Figure 3.9 shows the general scenario of distributed storage servers. In figure 3.9, there could be N connected servers where N €{1,2,3 ….. n-1} , in the system. Each server has following properties such as request queue, number of CPUs, storage capacity. Clients send their requests to the respective server. Many times the incoming request rate (ρ) in- creases exponentially to a particular server. This is because of the series of client's requests to that data that is stored within the server. In case, when a server gets too many requests than server buffers them in their request queue and the size of request queue gets increases dynamically only upto its predefined threshold limit. Once, the request queue breaches the threshold limit than server is considered as overloaded server and triggers the load balancer. Load balancer classifies the least loaded server on the basis of their request queue and processing capacity. As soon as the least loaded server gets classified than overloaded server migrate its load to that server and balances the load. Various notations are used in the proposed approach and represented as follows:

ρ - Current queue size of server.

λ	- Service rate that is number of request processed simultaneously server.
S_T	-Service time is the time taken by server to process the request
Q_L threshold	- Threshold limit of server request queue.
Q_L Current	- current capacity of server request queue at time t.
ΔLi	- additional load on server i.
DLi	- Deadline time of request i
Fj	- Fitness value of neighbors of server i.(j €{1,2,3 ….. n-1}

We have considered the real world scenario where the server request queue size and service rate changes with respect to time t dynamically and represented as respectively

$$\delta\rho = \frac{\rho}{\delta t} \; and \; \delta\lambda = \frac{\lambda}{\delta t} \tag{3.10}$$

$$\delta\rho > \; Q\_L_{threshold} \tag{3.11}$$

When server i where i $\in \{1,2,3 \ldots . n\}$, is overloaded then it calculates the amount of extra load ΔL i on that server which can be calculated as follow:

$$\Delta L_i = Q\_L_{current} - Q\_L_{threshold} \tag{3.12}$$

The condition when a load balancer module gets triggered on the overloaded server i is given below:

$$T(i) = \begin{cases} 1, & \Delta L > 0 \\ 0, & otherwise \end{cases} \tag{3.13}$$

Once, the load balancer module is triggered, server i find the least loaded or idle server that can accommodate its load and adequately process the service requests without deadline failure. For this load balancer calculates the fitness value Fj that can be calculated using the following fitness function:

$$\Delta M_j = Q\_L_{threshold} - Q\_L_{current} \tag{3.14}$$

Here, ΔMj is free request queue of server j. If ΔMj is negative, then server j request queue is overloaded otherwise it is least loaded.

$$Fj = \alpha1\ \Delta Mj + \alpha2\lambda \qquad (3.15)$$

Here, α1 and α2 are constants and may vary according to scenario such that

$$\alpha1 + \alpha2 = 1 \qquad (3.16)$$

For our proposed scenario, we have considered the value of α1 and α2 is 0.5 it is because both the parameters play the equal role in load balancing. In this way, load balancer calculates the fitness value for each neighbors of server i.and select that server which has maximum fitness Fj value and waiting time of request less than deadline of request and migrate the ΔMj amount of load to server j.

3.2.2. Proposed Model

Proposed algorithms have been designed to balance the client requests over the servers and distribute the load over the system uniformly. Here, load balancer as shown in Figure 3.10(a) is responsible for consistently assessment of the request queue dimension of servers and tries to prevent the main issue of overloading of any server with the aid of migrating the request to other idle or least loaded neighboring servers which can also complete the request in deadline without failure in the system. Proposed load balancing algorithm is divided into two stages. In first stage list of idle servers is created, and in second stage the server with highest fitness value and which can fulfil the deadline in selected.

Stage I

Algorithms checks and calculate the fitness value for the neighbor server to store them in a list shown in figure 3.10 (b). Load balancer utilizes this list to select the server that has highest fitness value. Fitness value is evaluated based on deadline and the empty

Stage II

In second stage load balancer calculates the waiting time over each server from above list which can be given as:

$$W_k = \frac{Q\_L_{current\_k}}{\lambda_k} \times S\_T_k \qquad (3.17)$$

$$W_k > DL_i \qquad (3.18)$$

Equation 3.18 shows the waiting time of k^{th} request. DLi deadline of a request. Load balancer then finds the server with waiting time less than request deadline and highest fitness value from the list. Proposed algorithms also try to reduce the server response time by selecting the server with least CPU utilization. In this way, we utilize the server to increase the overall performance of the system and reduce request deadline failures over the system.

Algorithm : Load balancing

1. **DDLBA** (Server s, $Q\_L_{current}$, λ_k, , S_T_k, DLi)
Input: Server s, Queue length $Q\_L_{current}$, service rate λ_k, service time S_T_k, deadline DL_i
2. s← server
3. $Q\_L_{current}$ ← current queue size
4. λ_k ← Service rate of server k
5. S_T_k← Service time of server k
6. DL_i← Deadline time of request I
7. Compute W_k
8. **if** ($\delta\rho$> $Q\_L_{Threshold}$) **then**
9. check server queue status.
10. Add request to queue;
11. Process_request();
12. **else**
13. server is overloaded
14. S ← Find_server(server_neighbour_list L)
15. find under loaded server.
16. S ← migrate request
17. Goto step 7

Output: Load balances the request.

Figure 3.10 (a) Load balancing algorithm.

Algorithm : To find suitable server

1. **Find_server**(server_neighbour_list L)
   Input: server_neighbour_list L
2. **for** k=1 to L.size()
3. S1←L.get()
4. F_k= α1 ΔMj + α2λ
5. temp_list t← F_k
6. L2 = Sort(t)
7. **end for**
8. **for** j=1 to L2.size()
9. if(L2. W_k< DL_i) then
10. S2 ←L2
11. **end for**
12. select server which fits the deadline and have maximum fitness value.
13. return S2

Output: The server with minimum fitness value.

Figure 3.10 (b) Find neighboring server algorithm.

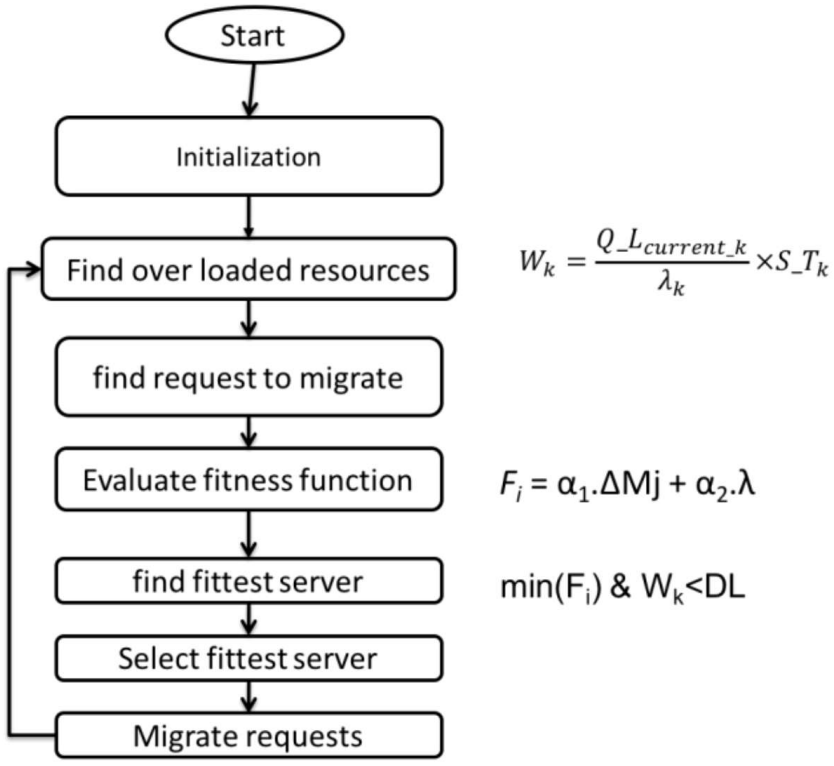


Figure 3.11.Proposed algorithms flow diagram

Figure 3.11 shows the flow of the algorithm with various phases of algorithm and interaction among them to find the fittest server for each request.

3.2.3. Experiment and Result

Efficiency evaluation of proposed DDLBA algorithm is completed utilizing simulations the place we've created hundreds of thousands of virtual machine requests to be served via 12 storage servers. All servers are deployed in distributed environment with unique count of CPU cores complete the requests. Each server has a fixed request queue size to buffer the incoming client requests and storage capacity. For the given main issue assertion in part 3 where the load is unbalanced, it is assumed that half of storage servers get client requests and others stay idle. Our aim is to equally distribute the received requests among the many servers to restrict the scenario of overloading. Within the simulation situation numbers of storage servers are kept fixed with various quantities of requests handling. We've got additionally compared the bought results with the least load balancing algorithm. Following table depicts the configuration parameter for our simulation atmosphere.

Table 3.2: Experimental parameters used for simulation environment

No. of client requests	No. of Servers	No. of CPU cores available per server	Storage capacity of servers (GB)	Server queue length
800	12	7	500	15
1000	12	8	500	15
1200	12	9	500	20
1800	12	10	500	20
2400	12	11	500	20

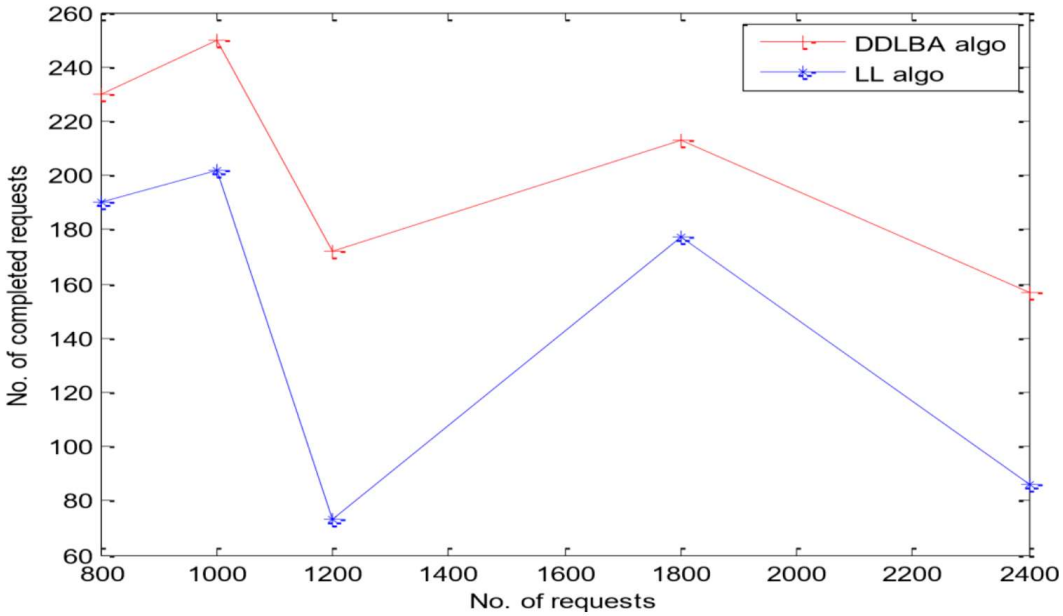


Figure 3.12 Comparison between no. of sent requests vs. no. of completed requests.

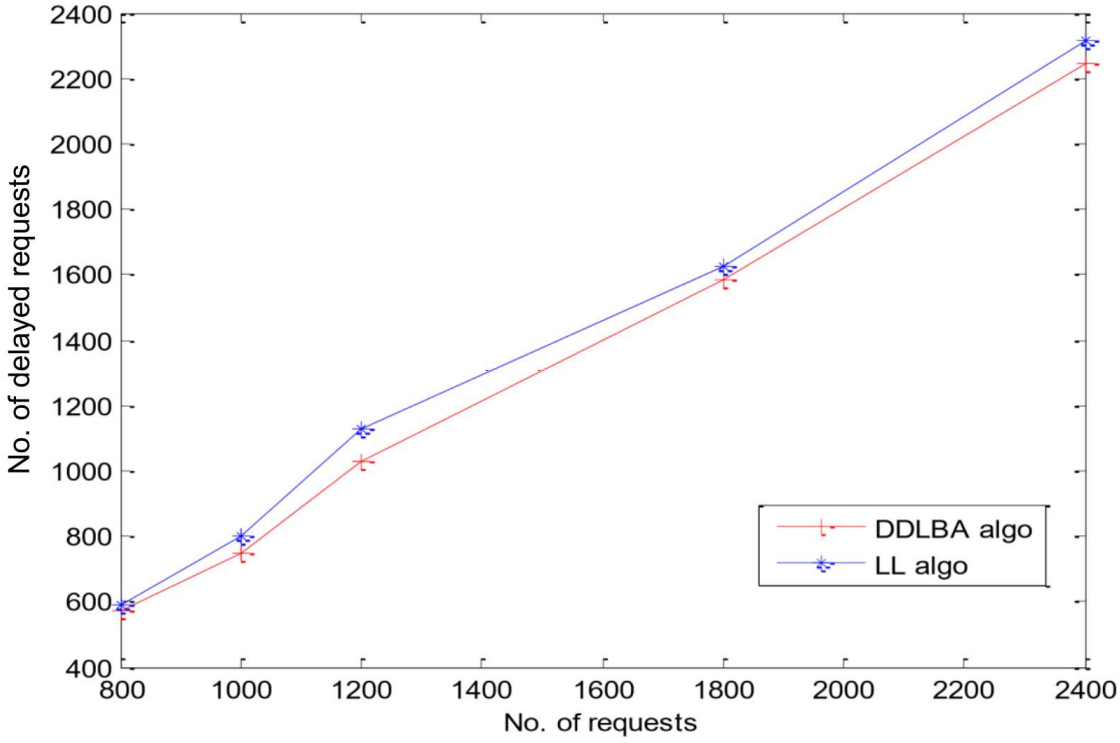


Figure 3.13 Comparison between no. of sent requests vs. no. of postponed requests.

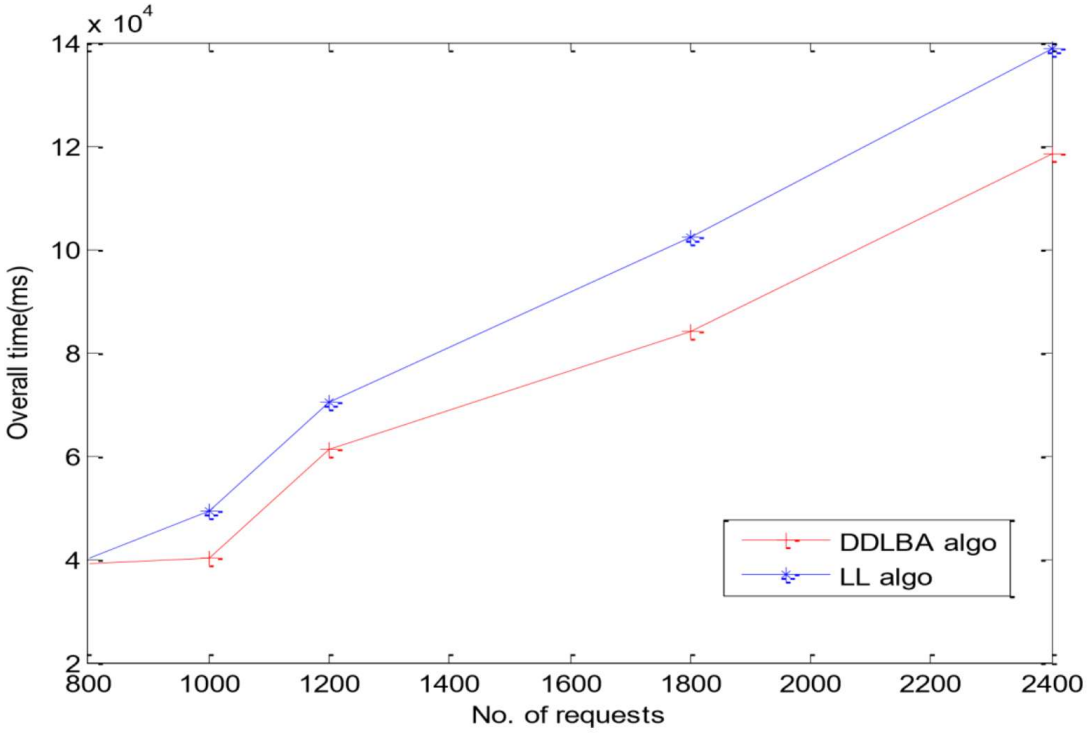


Figure 3.14 Overall response time.

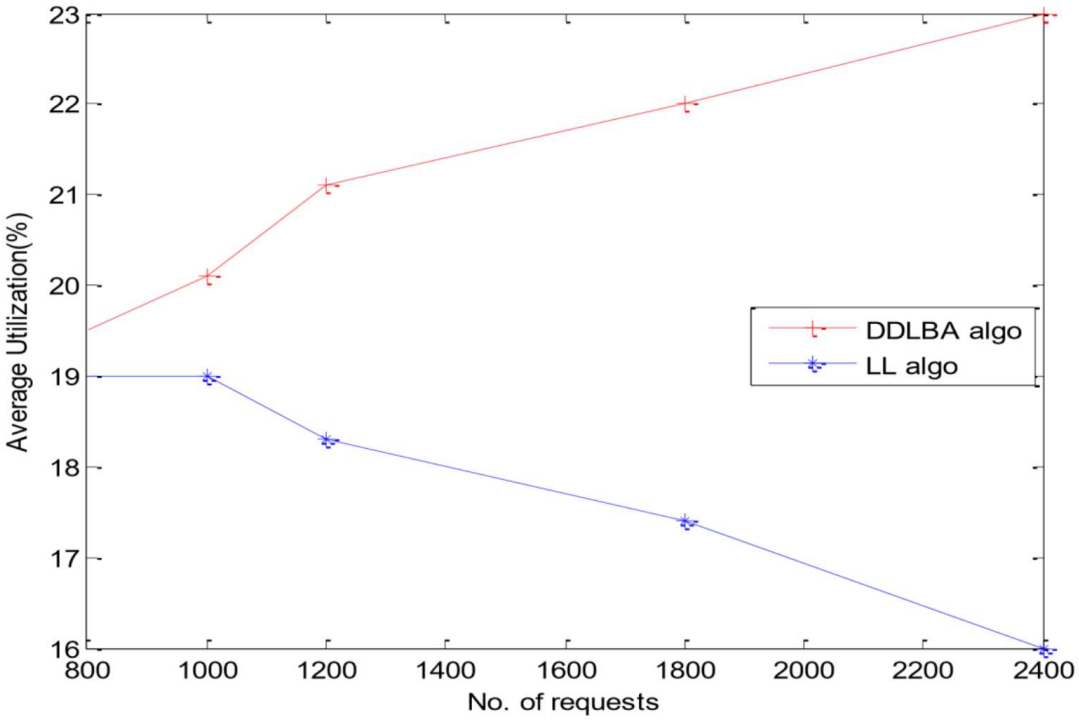


Figure 3.15 Average utilization of system

Figure 3.12 shows the number of processed client requests by server in a given time. Here, figure 3.12 represents the graph between numbers of sent requests vs. numbers of completed requests. Whereas figure 3.13 represents the graph between numbers of sent requests vs. numbers of postponed requests for the proposed and least load algorithms. In least loaded algorithm when any server get overloaded then load balancer selects the server of which request queue is least loaded without considering the CPU parameter. For the proposed algorithm we have considered the CPU parameter and from obtained results as shown in figure 3.12 & 3.14 that the proposed algorithm perform much better over the least load algorithm. Figure 3.15 shows that the proposed DDLBA algorithm improves the average utilization of the system drastically over increasing requests due to improvement in total request completed. In all set of client requests, proposed algorithm process more number of client's request with better overall response time as shown in figure 3.14.

3.3. Approach 3: Load Balancing Algorithm for Hybrid Cloud IaaS

Data confidentiality and trust formation are the major security concerns in the cloud. Therefore, there is a firm need to establish an Inherent trust on cloud service provider in order to assure the cloud behavior, data protection and make cloud technology globally acceptable and reliable for users. Data protection deals with protecting the individual or organization's private data which is shared over the cloud. It is possible only when the security and trustworthiness of both the service provider and user is ensured. Therefore there is a need to establish trust between both and hence we need to develop a trust management model.

Trust models are been used in all form of distributed environments ranging from MANETS (Mobile ad hoc network), Sensor network and Grid computing to validate the reliability of nodes over distributed network. In grid computing various trust models are been proposed to insure trust in term of security and reliability of the server or the node. Trust models are to resolve the problem of reliability in any heterogeneous environment, which contributed of nodes having different configuration spread over a network. There are many models being proposed in a cloud computing environment. Mostly trust models are proposed for cloud, SaaS (Software as a service) to overcome security issues. Some of the trust models proposed are being discussed in section II. The models proposed do not fit the cloud

IaaS (Infrastructure as a Service) environment because they do not take into consideration the performance of datacenter for trust and also most of the trust models proposed uses direct trust based on the static parameters but the trust vale may change as the performance of datacenter changes.

So in this approach, we have proposed a hybrid trust management model to overcome this problem, by taking into consideration datacenter characteristics which vary from datacenter to datacenter. In proposed trust model we have used direct trust and recommended trust value to evaluate the variable trust value over the period of time. These trust value are been used by the scheduling algorithm proposed to improve the scheduling of the resources for hybrid cloud.

3.3.1. Proposed Approach

The Trust Model we are proposing here will work for Private, Public and Hybrid Cloud. It will take into consideration both the Direct and Indirect Trust or recommended trust. For the Calculation of Trust Value we will take Memory (RAM), MIPS (Million Instruction per Cycle), Frequency (Frequency of data center) and Fault Rate. We will initially calculate only Direct Trust and as the time will evolve we will also consider the indirect trust. We will initiate the parameter MIPS and Fault with zero and Memory and MIPS will be according to the datacenter. With these we will calculate the Trust Value and initial load balancing will be done according to that. As the Time will evolve, we will calculate the Indirect or Recommended Trust also using Fault Rate which we initially considered zero and Response Time. Now, since the request from both public and private cloud will start arriving, we assume that some of them will not be accomplished due to technical faults and hence the parameter Fault Rate will have some value other than zero.

After Calculating Values for both we will rate the datacenters according to the trust value calculated, This Trust value will be combined of Direct and Indirect Trust. And hence allocate the private request to a server with high trust value since the request has lower chances of going down and allocate the public request to a server with low trust value since the request has higher chances of going down. We will keep this dynamic process going in order to ensure a Trust Based Load Balancing.

Steps for load balancing algorithm.

Pseudo Code

• Start by Calculating individual trust values of each data center.

• Consider only direct trust initially and as the time evolve indirect trust will also come in consideration.

• Take these factors into consideration for calculation of trust value for Direct Trust:

  - RAM (Memory)
  - MIPS
  - Frequency of data center
  - Fault Rate (initially 0)

• Now, according the size of request start dividing them into public or private.

• Send the private cloud requests to a data center with high trust value since they are less likely to fail.

• Send the public cloud requests to data center with low trust value as they are more likely to fail.

• Meanwhile calculate the Indirect Trust, which is based on:

  Fault Rate (number of task failed/total tasks)

  Response Time (Time taken to accomplish a standard request.)

• Update the trust value with the values from Indirect Trust and now allocate the load accordingly.

We will initially calculate only Direct Trust and as the time will evolve we will also consider the indirect trust. We will initiate the parameter MIPS and Fault with zero and Memory and MIPS will be according to the datacenter. With these we will calculate the Trust Value and initial load balancing will be done according to that. As the Time will evolve we will calculate the Indirect or Recommended Trust also using Fault Rate which we initially considered zero and Response Time. Now since the request from both public and private cloud will start arriving we assume that some of

them will not be accomplished due to technical faults and hence the parameter Fault Rate will have some value other than zero.

RAM: flash memory of the server.

MIPS: Millions of instructions per second of server.

Initial_Trust : Trust value of the server based on the executional power i.e. RAM and MIPS.

Fault_Rate : Number of faults in an server over a period of time 't'.

Updated_Trust : Updated Trust value based on initial trust and fault occurred over a period of time. α_1, α_2 & α_3: Constants

$$\text{Initial\_Trust} = \alpha_1 * \text{RAM} + \alpha_2 * \text{MIPS}; \quad (3.19)$$

Where

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \quad (3.20)$$

$$Updated\ Trust = InitialTrust + \alpha 3 * \left(\frac{1}{FaultRate}\right) \quad (3.21)$$

Algorithm : Load Balancing Algorithm

```
1. Load Balancing Algorithm(Req_list r)
   Input: Requests list r
2. Initialize servers
3. Calculate individual trust values of each data center
4. CheckDirect _TrustValue ( )
5. CheckIndirect _Value ( )
6. dividing them into public or private according the size of
   request.
7.  if (Data Center Value > Threshold) then
8.        Send the private cloud requests.
9.  else if (Data Center Value < Threshold)
10.       Send the public cloud requests.
11.  else
12. Keep Searching
Output: All request been scheduled.
```

Figure 3.16 Proposed trust based algorithm

Figure 3.16 shows the pseudo code for proposed algorithm with various steps to find the fittest server for each request.

3.3.2. Experiment and Results

Proposed trust model is simulated using Cloudsim API [24]. Cloudsim basically support cost estimation, and Random Load balancing of resource and has no support to study behaviour of datacenter and trust model. So to study the datacenter behaviour trust parameter is introduces as datacenter property, which depends on parameter defined in Cloudsim. Based on this attribute we have computed the result to show the real problem. For this we have considered two classes of budget range, with 3 datacenters and 300 user requests.

Table 3.3: Servers Parameters

Server Name	Fault rate	Hard disk (MB)	RAM (MB)	MIPS	Cores
Server1	0.143	1000000	20048	10000	4
Server2	0.125	1000000	20048	10000	6
Server3	0.5	1000000	20048	10000	4

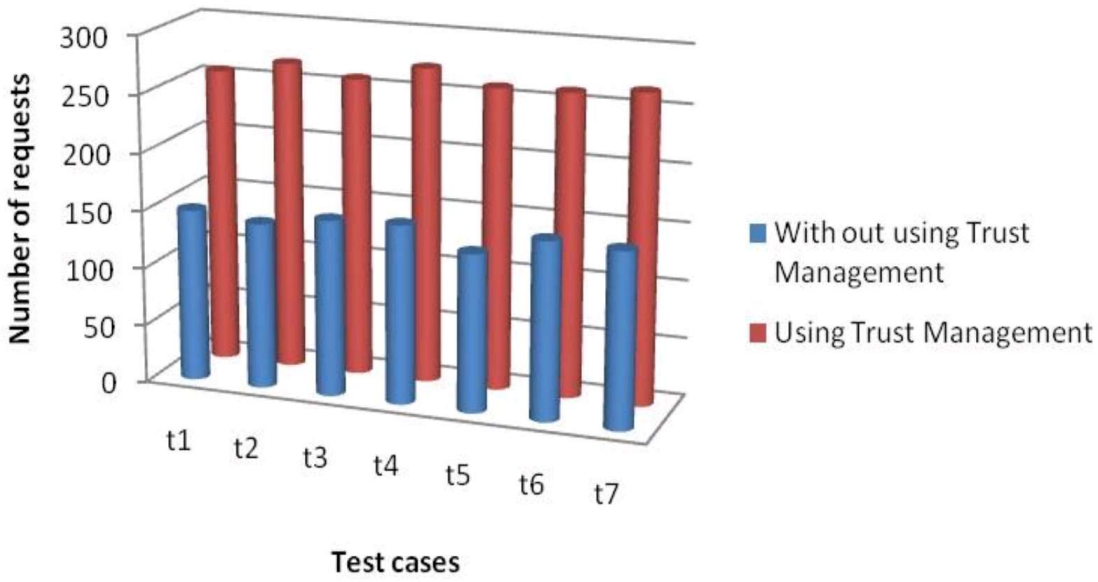


Figure 3.17 Total number of request completed in faulty environment

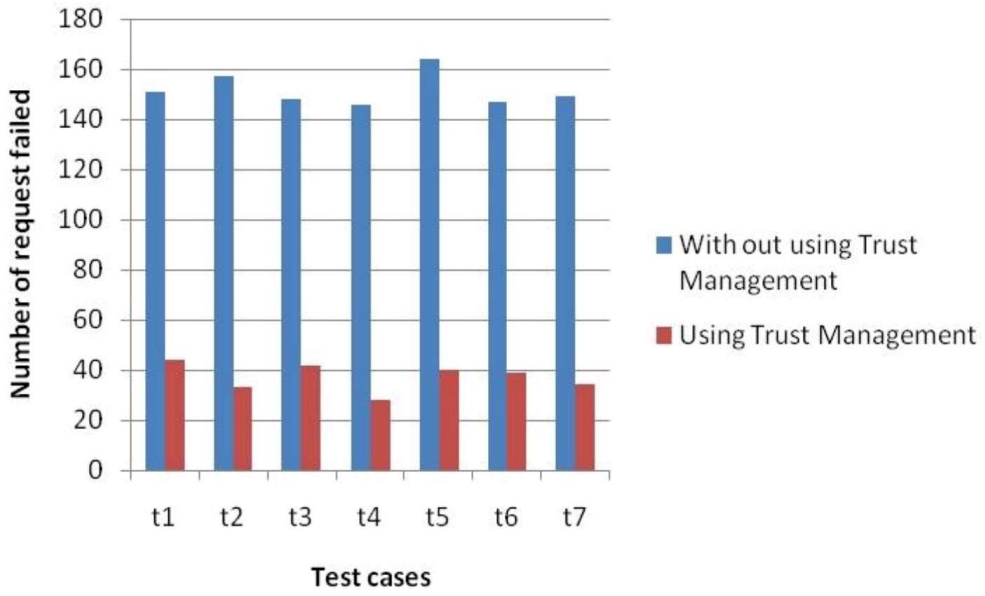
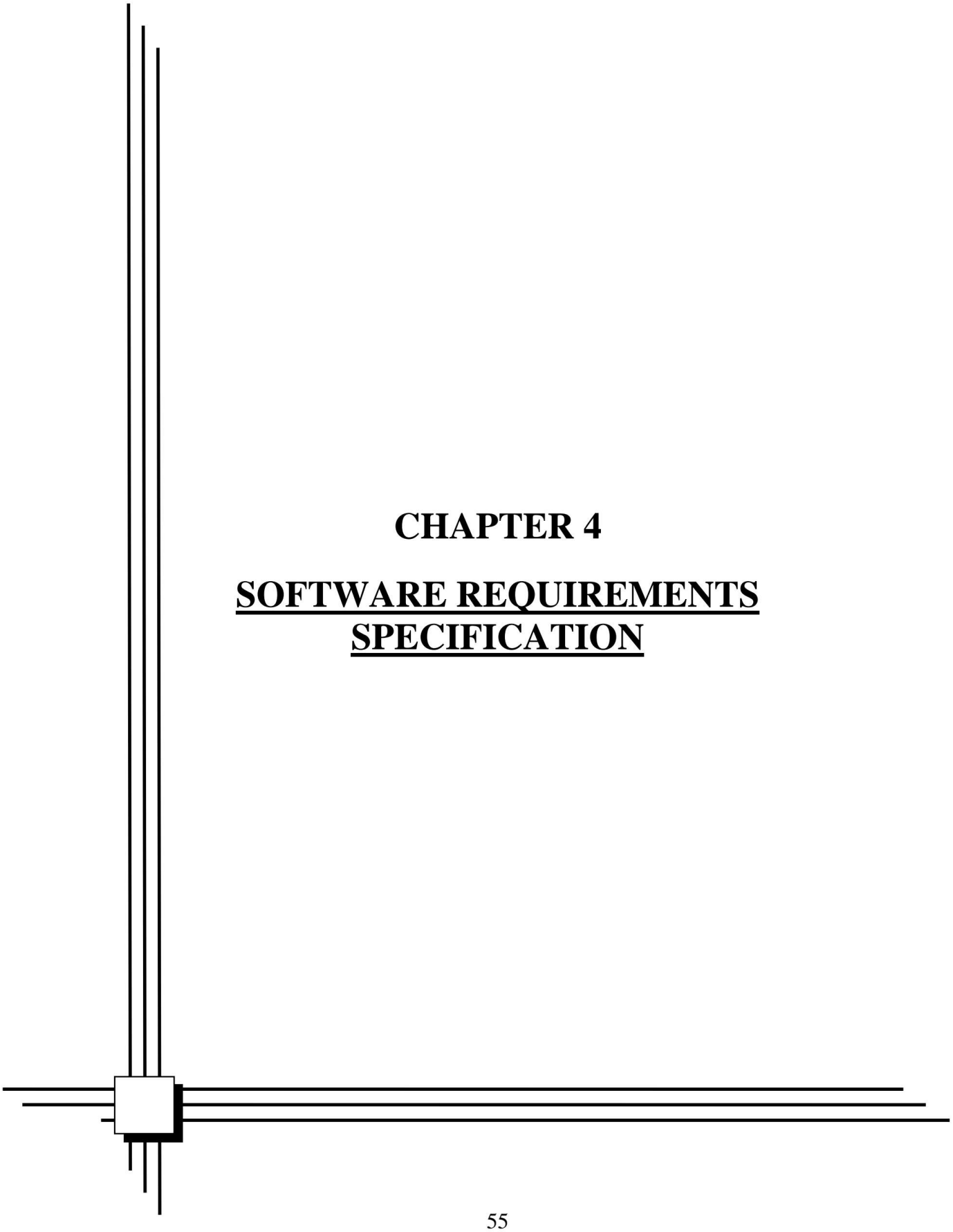


Figure 3.18 Total number of request failed.

Figure 3.17 and 3.18 represents the improvement in request failure and completed count using proposed algorithm as compared to existing algorithm without trust management.

CHAPTER 4

SOFTWARE REQUIREMENTS SPECIFICATION



4.1 Introduction

Distributed and cloud computing environments are presently slanting and more prevalent for computation by various organizations like Google, Amazon, Microsoft etc. as the cloud size increase there is huge expansion in power consumption over the server farms. Also with increase in request load over a datacenter increases the request failure probability. To overcome these issues request should be scheduled in more efficient manner, improving resource utilization, request failure count and reliability of system. Recent studies show that the failure probability of a server increases linearly with the increase of independent resources (processors), and result into request failure at datacenters. So to resolve this issue various approaches have been proposed to improve the performance of cloud environment.

In this section, we have proposed a set of learning based algorithms for task allocation to minimize the request failure and to improve QoS (Quality of Service) over a data center. Proposed approaches aim to provide a global best schedule with least scheduling time complexity. Proposed algorithms has proven to have better performance in term of load and request failure rate as compared to previously proposed task allocation algorithm for cloud IaaS.

.

4.2 Architecture

Proposed algorithm provides a benefit over existing static scheduling algorithm, that it can search for best global solution rather than assuming the local best solution as the best solution. Moreover, the proposed algorithm takes into consideration the faulty behavior of cloud, which helps in find a solution with similar high utilization and least failure probability. To overcome these issues a fault aware learning based resource allocation algorithm is been proposed using Genetic algorithm. Genetic algorithm helps us to find a solution, which cannot be, achieved my any static or dynamic algorithm. More over fault tolerant genetic algorithm help to find a fittest solution in term of least make span (Time taken to complete a request) and least request failure probability. Proposed algorithm uses Poisson probability distribution for random request failure at virtual machine i.e. at host and datacenter

level. On the other hand, request failure over a datacenter may occur randomly due to storage, network failure or VM crashes. Based on fault over a datacenter and computing capability of a system, we have proposed a task allocation policy to minimize the total make span over the system and reduce request failure probability. According to algorithm collect the information of data center resources and capability, and the count of failure occurred over a period of time on a datacenter.

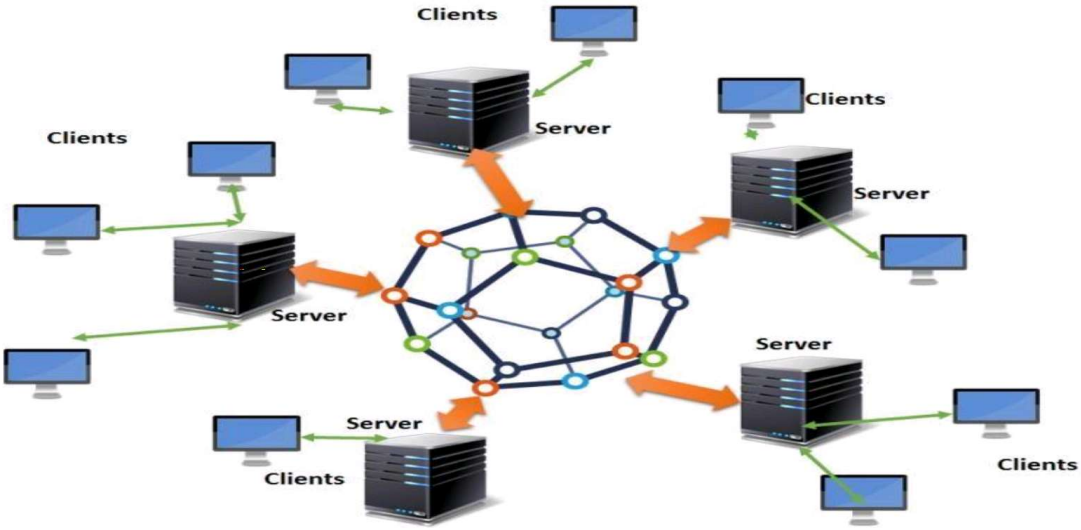


Figure: 4.2 Architecture of Distributed and cloud computing environments.

4.3 Functional Requirements

Approach 1: Fault and QoS based Genetic Algorithm for Task Allocation in Cloud Infrastructure

Proposed FGA (Fault aware Genetic Algorithm) is divided into four phases which are as follows:

a) **Initialization**

In this phase, the initial population for the genetic algorithm is generated. The population consists of a set of individuals, where each individual represents a potential solution to the problem. The individuals are encoded as chromosomes or

strings of genes, which can be binary or real-valued, depending on the specific problem.

b) **Evaluation and selection**

During this phase, each individual in the population is evaluated based on a fitness function. The fitness function assesses the quality or performance of each individual in solving the problem. The evaluation process takes into account both the fault tolerance and load balancing aspects of the storage mechanism. The individuals with higher fitness scores have a higher chance of being selected for the next phase.

c) **Crossover**

This phase involves applying genetic operators to the selected individuals to create new offspring. The genetic operators typically include crossover and mutation. Crossover involves combining genetic material from two parent individuals to create one or more offspring. Mutation introduces random changes in the genes of an individual to maintain diversity in the population. These operators allow for the exploration and exploitation of the solution space.

d) **Mutation**

After generating new offspring, the population is updated by replacing some individuals with the offspring. The replacement strategy can be based on elitism, where the best individuals from the current population and the offspring are selected to form the next generation. The process of evaluation, selection, genetic operators, and replacement continues iteratively until a termination condition is met. The termination condition can be a maximum number of generations, reaching a satisfactory solution, or a convergence criteria.

4.4 Non-Functional Requirements

Non-functional requirements (NFRs) are the criteria or attributes that specify the quality characteristics or constraints of a system rather than its specific functionalities.

These requirements define how the system should behave, perform, and interact with its environment. Here are some common categories of non-functional requirements:

1. Performance: Specifies the system's response time, throughput, scalability, and resource utilization under different conditions. For example, the system should be able to handle a certain number of concurrent users or process a specific number of transactions per second.
2. Reliability: Describes the system's ability to perform its intended functions consistently and accurately over a specified period. It includes measures such as availability, fault tolerance, error handling, and mean time between failures (MTBF).
3. Security: Addresses the protection of the system and its data from unauthorized access, data breaches, and other security threats. It encompasses aspects like authentication, authorization, encryption, data integrity, and auditability.
4. Usability: Focuses on the ease of use and user experience of the system. It includes factors such as user interface design, intuitiveness, accessibility, and documentation.
5. Scalability: Determines how well the system can handle increased workload or growing user base by adding additional resources. It covers aspects like load balancing, horizontal and vertical scaling, and system performance under varying workloads.
6. Maintainability: Refers to the ease with which the system can be modified, enhanced, or repaired over its lifecycle. It encompasses aspects such as code readability, modularity, documentation, and the use of standard coding practices.
7. Compatibility: Specifies the system's ability to operate and interact with other systems, platforms, or software components. It includes aspects like interoperability, integration capabilities, and adherence to industry standards.
8. Portability: Describes the system's ability to be deployed and run on different hardware, operating systems, or environments. It includes considerations like platform independence, adaptability, and ease of installation.
9. Compliance: Addresses adherence to legal, regulatory, or industry-specific requirements. It includes aspects such as data privacy, data protection, compliance with relevant standards or regulations, and auditability.

10. Performance Efficiency: Focuses on optimizing resource utilization, minimizing energy consumption, and maximizing system efficiency. It includes aspects like algorithmic efficiency, optimization techniques, and system responsiveness.

These are just a few examples of non-functional requirements. The specific non-functional requirements for a cloud secure storage mechanism based on fault and load balancing would depend on the context, goals, and stakeholders' needs for the system.

4.5 External Interface Requirements

External interface requirements define how a system interacts with external entities, including users, other systems, and external hardware or software components. These requirements specify the inputs, outputs, and communication protocols involved in these interactions. Here are some common types of external interface requirements:

1. User Interfaces:
    - Graphical User Interface (GUI): Describes the visual elements, layout, and functionality of the system's user interface. It includes elements such as menus, buttons, forms, and navigation.
    - Command-Line Interface (CLI): Specifies the commands, syntax, and options that users can use to interact with the system through a command-line interface.
    - Application Programming Interface (API): Defines the interface and protocols for integrating the system with other software applications. It includes methods, parameters, and data formats for communication.
2. Hardware Interfaces:
    - Input/Output (I/O) Devices: Specifies the hardware devices, such as keyboards, mice, printers, or scanners, that the system interfaces with to receive input or provide output.
    - Sensors or Actuators: Defines the interface with physical sensors or actuators, such as temperature sensors, motion detectors, or robotic arms.
3. Software Interfaces:

- Integration with External Systems: Describes the protocols, data formats, and communication mechanisms required to interface with external systems or services, such as databases, web services, or messaging systems.
- Libraries or Frameworks: Specifies the dependencies or integration requirements with external libraries, frameworks, or third-party software components.

4. Communication Interfaces:
    - Network Protocols: Defines the communication protocols and standards, such as HTTP, TCP/IP, REST, or SOAP, to be used for communication between the system and other systems over the network.
    - Data Formats: Specifies the data formats, such as JSON, XML, or CSV, to be used for exchanging data with external entities.
5. Security Interfaces:
    - Authentication and Authorization: Describes the mechanisms and protocols for user authentication, access control, and secure communication.
    - Encryption and Cryptographic Protocols: Specifies the encryption algorithms and cryptographic protocols to ensure secure data transmission and storage.
6. External Data Interfaces:
    - Data Import/Export: Specifies the data formats, protocols, and mechanisms for importing or exporting data from the system to external sources or vice versa.
    - Data Integration: Defines the requirements for integrating with external data sources or databases.

These are some examples of external interface requirements. The specific requirements will vary depending on the nature of the system, its intended users, and the integration needs with external entities. It is important to define clear and well-documented interfaces to ensure effective communication and interoperability with external components and systems.

4.6 Software Description

The software description provides an overview of the functionality, purpose, and components of a software system. It highlights the key features and capabilities of the software. Here's an example of a software description for a cloud secure storage mechanism based on fault and load balancing:

Title: Cloud Secure Storage Mechanism

Description: The Cloud Secure Storage Mechanism is a software system designed to provide a secure and reliable storage solution in a cloud computing environment. It employs fault and load balancing techniques to ensure data availability, fault tolerance, and optimal resource utilization. The system aims to protect sensitive data stored in the cloud while efficiently distributing the workload across multiple storage resources.

Key Features:

1. Fault-Tolerant Data Storage: The system utilizes fault tolerance mechanisms, such as data replication or erasure coding, to ensure the durability and availability of data even in the presence of hardware failures or network disruptions.
2. Load Balancing: Load balancing algorithms distribute the data and workload evenly across multiple storage nodes or servers. This optimization technique ensures efficient resource utilization, minimizes response time, and avoids overloading specific nodes.
3. Data Security: The system implements robust security measures to protect data stored in the cloud. This includes data encryption, access control mechanisms, secure communication protocols, and authentication mechanisms to prevent unauthorized access or data breaches.
4. Performance Optimization: The software focuses on performance optimization by continuously monitoring resource utilization, network conditions, and storage node capacities. It dynamically adjusts the load balancing strategy to optimize performance, responsiveness, and scalability.

5. Scalability: The system is designed to scale with the growing storage demands. Additional storage nodes can be added to the infrastructure seamlessly, and load balancing algorithms adapt to accommodate the expanded resources.
6. Fault Detection and Recovery: The system includes mechanisms to detect faults or failures in storage nodes. It initiates automatic fault recovery processes, such as data reconstruction or node replacement, to maintain data integrity and availability.
7. Monitoring and Reporting: The software provides comprehensive monitoring and reporting capabilities. It offers insights into system performance, resource utilization, fault events, and security audits to assist administrators in managing and maintaining the storage environment effectively.
8. Integration and Compatibility: The software is designed for seamless integration with existing cloud computing infrastructures, including various storage platforms, databases, and communication protocols. It ensures compatibility with industry standards and facilitates interoperability with external systems.

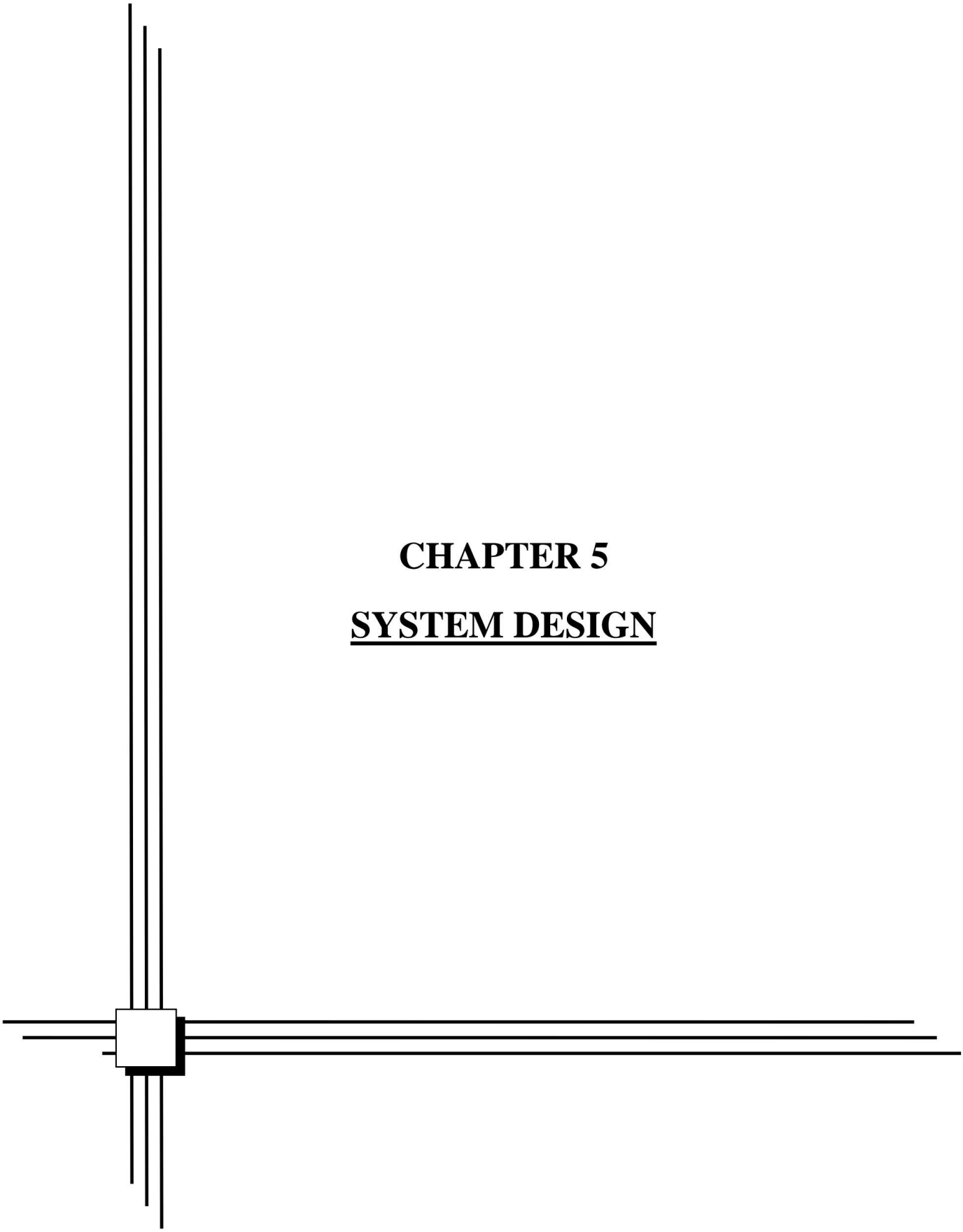
The Cloud Secure Storage Mechanism offers a robust and efficient solution for storing and managing data in the cloud, addressing the challenges of fault tolerance, load balancing, and data security. Its flexible architecture allows for scalability and adaptability, making it suitable for diverse cloud environments and storage requirements.

Operating System: We have chosen Windows operating system for its best Support and user-friendliness.

Cloud Computing Tools : eclipse 10, jdk-7u21-windows-i586, eclipse-SDK-3.7.1-win32 , etc.

CHAPTER 5

SYSTEM DESIGN



5.1 Design Objective

Design objectives in software engineering represent the specific goals and intentions that guide the design process of a software system. These objectives define the desired outcomes and qualities that the system should exhibit. Here's an example of a design objective for a cloud secure storage mechanism based on fault and load balancing:

Design Objective: Optimize Data Availability and Fault Tolerance

Description: The primary design objective of the cloud secure storage mechanism is to optimize data availability and fault tolerance in the cloud environment. The system should ensure that data stored in the cloud remains highly accessible, even in the presence of faults or failures. The objective encompasses the following key aspects:

1. Fault Tolerance: The system should be designed to handle and recover from various types of faults, such as hardware failures, network disruptions, or software errors. It should employ fault tolerance techniques, such as data replication or erasure coding, to ensure that data remains available and retrievable even when individual storage nodes or components experience failures.
2. Data Redundancy: The design should incorporate data redundancy strategies to store multiple copies of data across different storage nodes. This redundancy enhances fault tolerance by allowing the system to retrieve data from alternative sources if a particular node becomes unavailable or corrupted.
3. Load Balancing: The system should employ load balancing algorithms to evenly distribute the workload and data across multiple storage nodes. This approach optimizes resource utilization, minimizes response time, and avoids overloading specific nodes, thus enhancing both performance and fault tolerance.
4. Automated Fault Detection and Recovery: The design should include mechanisms for automated fault detection and recovery. It should continuously monitor the health and performance of storage nodes, detect any anomalies or faults, and initiate appropriate recovery processes. These processes may involve data reconstruction, node replacement, or redistribution of data to ensure uninterrupted access to data.

5. Performance Monitoring: The design should incorporate mechanisms for monitoring the performance and resource utilization of storage nodes. This monitoring enables the system to detect performance bottlenecks, identify overloaded nodes, and dynamically adjust the load balancing strategy to optimize system performance and fault tolerance.
6. Scalability: The design should support the scalability requirements of the cloud environment. It should be capable of seamlessly accommodating additional storage nodes as the data volume increases or the user base expands. The design should ensure that fault tolerance and load balancing mechanisms can adapt and scale effectively to maintain high availability and performance.

By emphasizing these design objectives, the cloud secure storage mechanism can achieve high levels of data availability, fault tolerance, and overall system reliability. The design should prioritize fault handling, redundancy, load balancing, and scalability to meet the demands of a dynamic and secure cloud storage environment.

5.2 Design Principle

Design principles in software engineering provide guidelines and best practices that help in creating well-designed and maintainable software systems. These principles serve as fundamental concepts to guide the decision-making process during the design phase. Here's an example of a design principle for a cloud secure storage mechanism based on fault and load balancing:

Design Principle: Separation of Concerns

Description: The design principle of separation of concerns advocates for dividing a system into distinct components or modules, each responsible for a specific aspect or concern. In the context of a cloud secure storage mechanism, the principle suggests separating different functionalities and responsibilities to achieve a more modular and maintainable design. This principle can be applied in the following ways:

1. Modularity: The system should be divided into modular components that encapsulate specific functionalities, such as data replication, fault detection, load balancing, or security measures. Each module should focus on its own

concern and have well-defined interfaces for interaction with other modules. This promotes code organization, reusability, and ease of maintenance.

2. Single Responsibility: Each module or component should have a single, well-defined responsibility. For example, a module could be dedicated to handling fault detection and recovery, while another module handles load balancing algorithms. By adhering to the single responsibility principle, modules become more focused, easier to understand, and less prone to introducing unintended side effects.
3. Loose Coupling: The design should strive for loose coupling between modules to minimize dependencies and increase flexibility. Modules should communicate through well-defined interfaces using standard protocols or messaging mechanisms. Loose coupling allows for easier modification or replacement of individual modules without affecting the overall system, enhancing maintainability and extensibility.
4. High Cohesion: Modules should exhibit high cohesion, meaning that the internal elements or functions within a module should be closely related and work together to achieve a specific goal. A module responsible for data replication, for instance, should contain all the necessary methods and logic related to that functionality, promoting clarity, readability, and ease of maintenance.
5. Abstraction and Encapsulation: The design should utilize abstraction and encapsulation to hide complex implementation details and expose only necessary interfaces or APIs to other modules. This reduces complexity and promotes information hiding, enabling modules to interact at a higher-level abstraction, which simplifies system understanding and promotes modular testing.

By following the principle of separation of concerns, the design of the cloud secure storage mechanism can achieve improved modularity, maintainability, and extensibility. It allows for more focused and cohesive modules that can be developed, tested, and maintained independently, leading to a more robust and adaptable software system.

5.3 Architecture Diagram

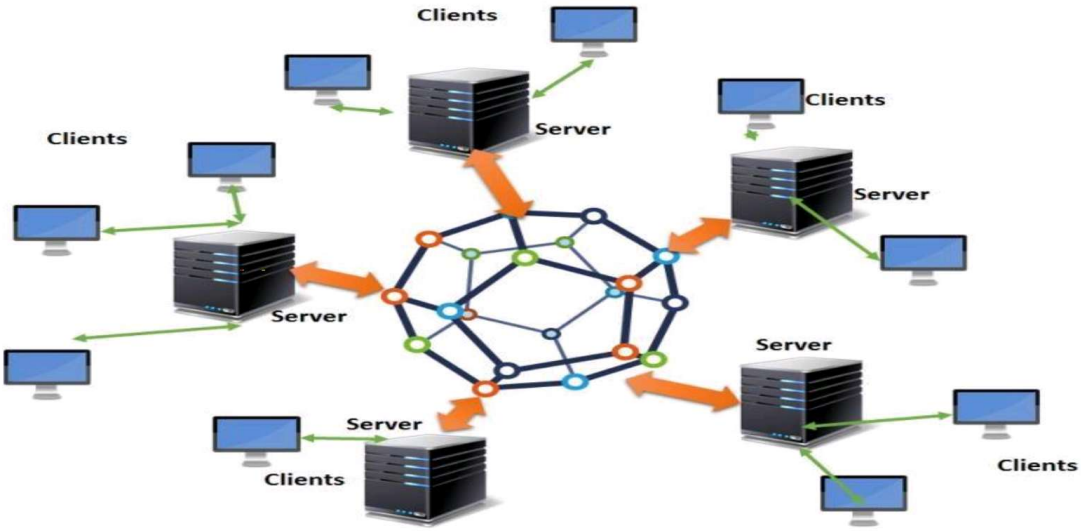


Figure: 5.3 Architecture Diagram of Distributed and cloud computing environments.

Transfer Learning Architecture

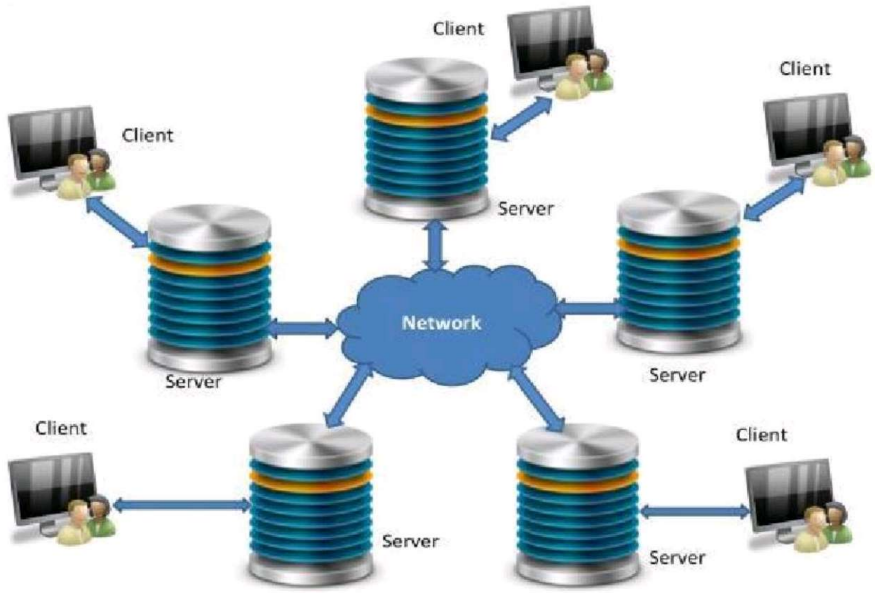


Figure 5.2 Transfer Learning Architecture

5.4 UML Description

UML (Unified Modeling Language) is a standardized visual modeling language used in software engineering to describe, specify, and document the structure and behavior of software systems. UML provides a set of notations and diagrams to represent different aspects of a system's design. Here's an example of a UML description for a cloud secure storage mechanism based on fault and load balancing:

Use Case Diagram:

- Use case diagrams illustrate the system's functionalities from a user's perspective.
- It identifies the different actors (users, systems, or external entities) and their interactions with the system.
- Use case diagrams depict the high-level system requirements and the main tasks or actions performed by the users.

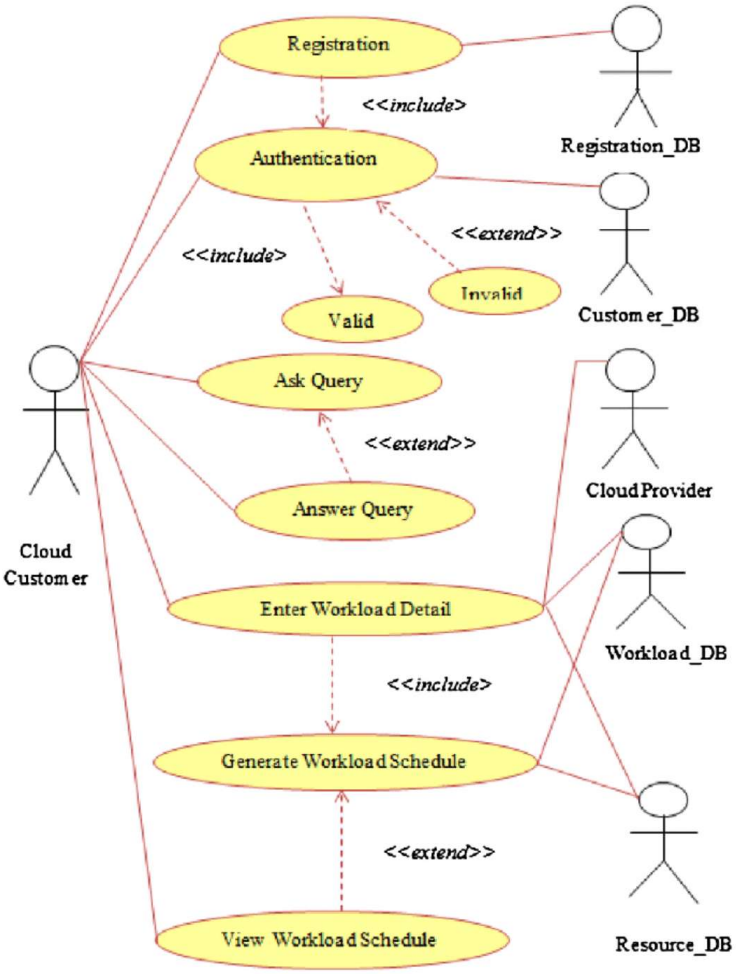


Figure 4.1 Use Case Diagram for a cloud secure storage mechanism based on fault and load balancing

Class Diagram:

- Class diagrams represent the static structure of the system.
- They depict the classes, their attributes, methods, and the relationships between the classes.
- Class diagrams show the organization of the system's classes and how they interact with each other.

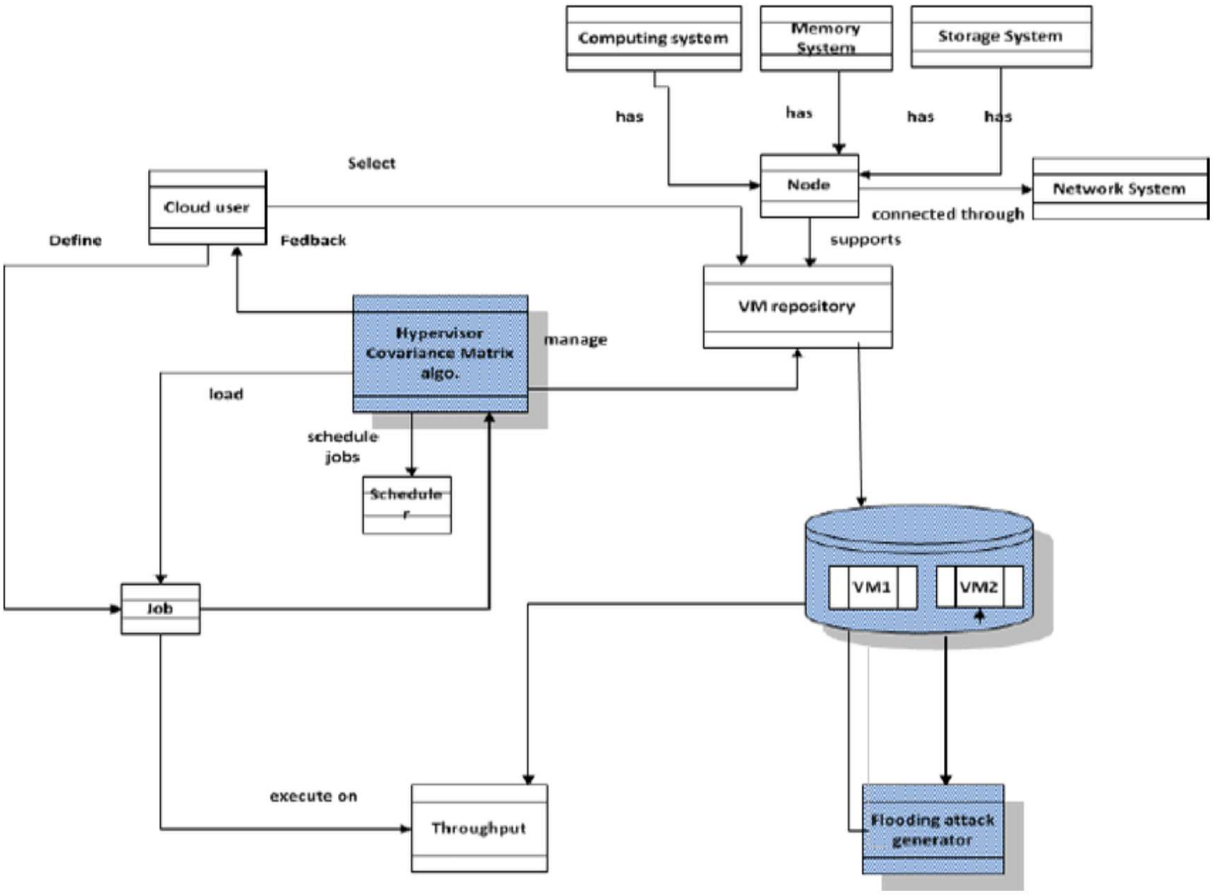


Figure 4.2 Class Diagram for a cloud secure storage mechanism based on fault and load balancing

Sequence Diagram:

- Sequence diagrams depict the dynamic behaviour of the system, particularly the interaction between objects or components over time.
- They illustrate the sequence of messages exchanged between objects and the order of their execution.

- Sequence diagrams help visualize the flow of control and data during a specific scenario or use case.

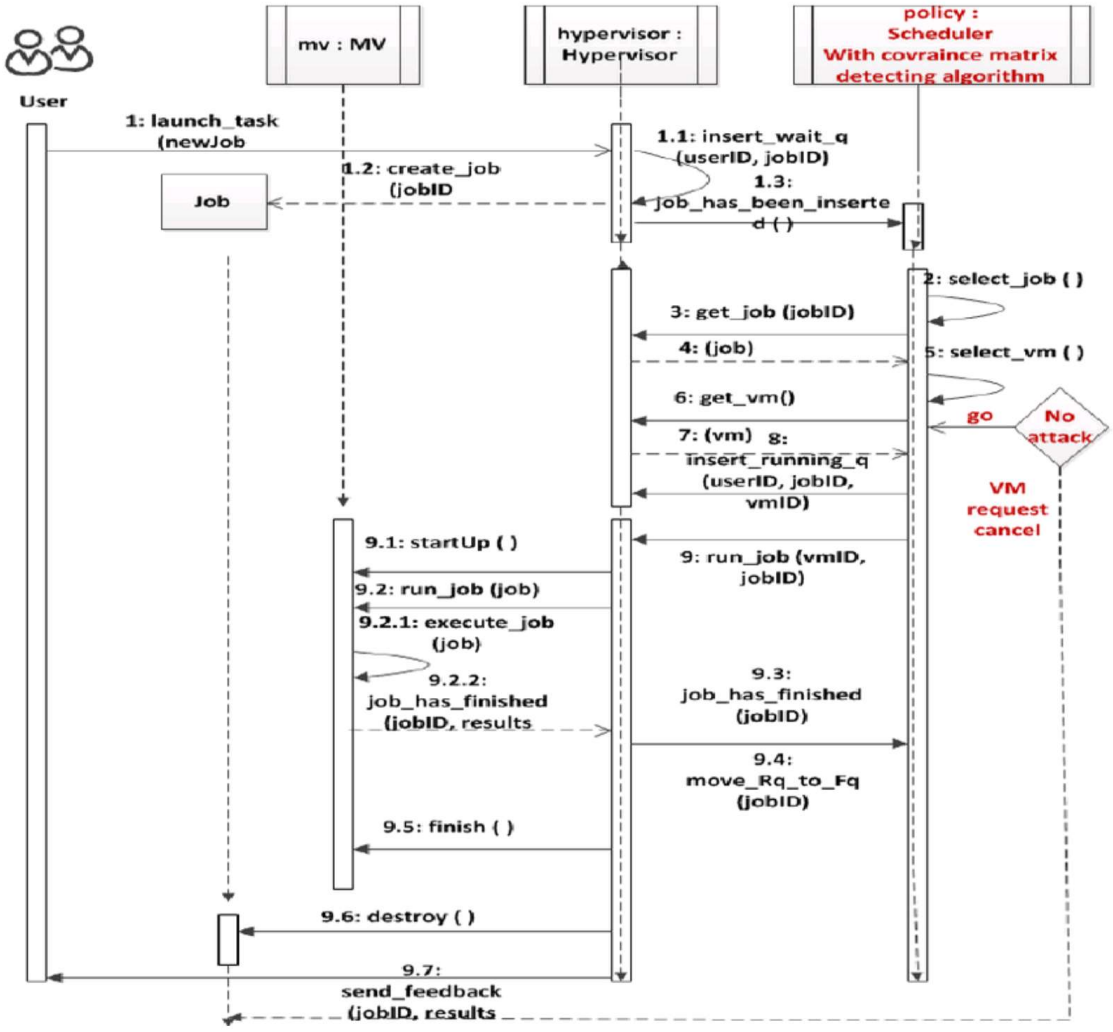


Figure 4.3 Sequence Diagram for a cloud secure storage mechanism based on fault and load balancing

Activity Diagram:

- Activity diagrams capture the workflow or business processes within the system.
- They represent the activities, actions, and decision points, and the flow of control between them.

- Activity diagrams provide a visual representation of the system's logic and the order of operations.

These UML diagrams can be used collectively to provide a comprehensive description of the waste or garbage classification system's structure, behaviour, and interactions. They help stakeholders understand the system's design, functionality, and how different components work together. Additionally, UML diagrams aid in communication and collaboration among developers, designers, and other project stakeholders, facilitating a shared understanding of the system's architecture and requirements.

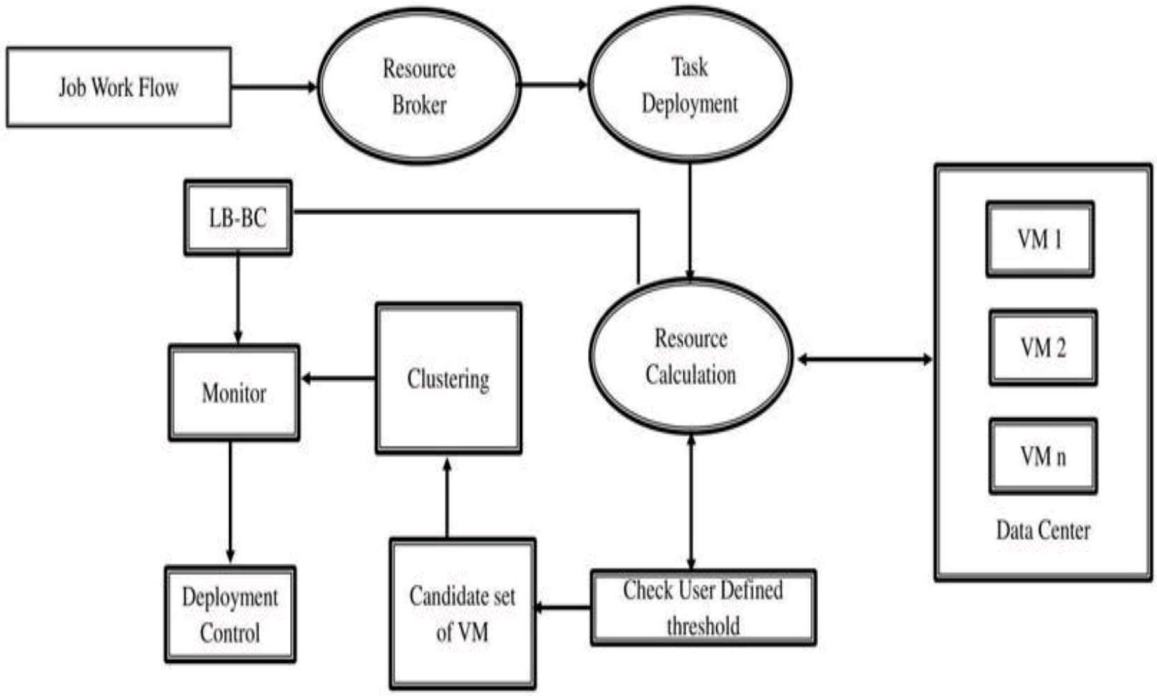


Figure 5.4 Activity Diagram for a cloud secure storage mechanism based on fault and load balancing.

5.5 UML Diagrams

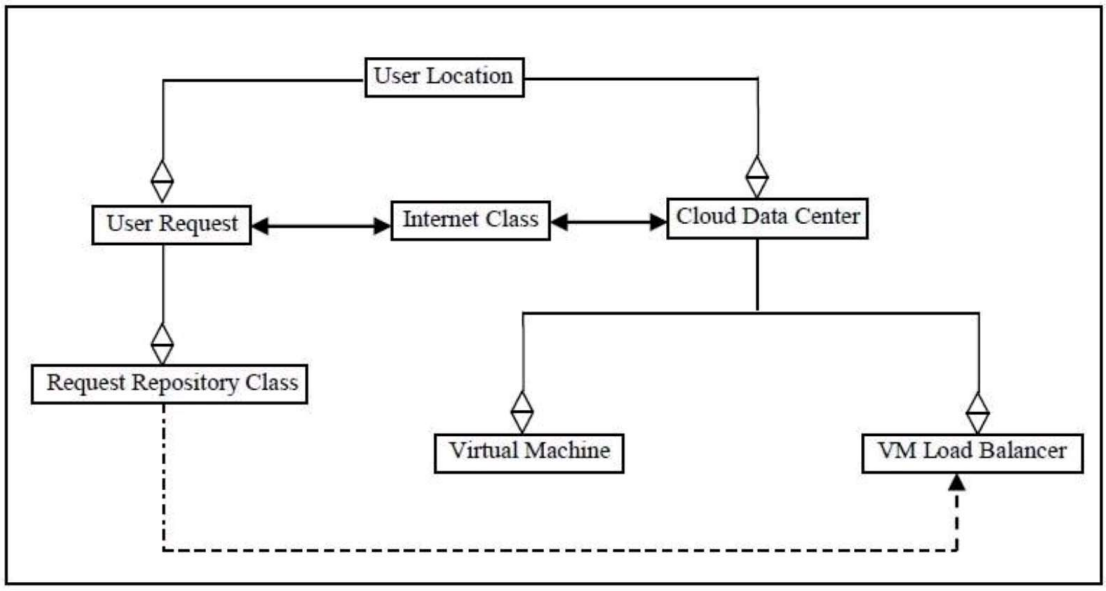


Figure : 5.5 UML Diagrams for a cloud secure storage mechanism based on fault and load balancing.

CHAPTER 6

IMPLEMENTATION

6.1 Modules

In the context of a cloud secure storage mechanism based on fault and load balancing, here are some example modules or components that can be part of the system:

1. Data Replication Module:
    - Responsible for replicating data across multiple storage nodes to ensure fault tolerance and data availability.
    - Manages the creation, update, and deletion of data replicas.
    - Monitors the health and status of storage nodes to initiate replication or recovery processes when necessary.
    - Implements replication strategies, such as data partitioning, data duplication, or erasure coding.
2. Load Balancer Module:
    - Distributes the workload and data evenly across storage nodes to optimize resource utilization and performance.
    - Monitors the load and capacity of storage nodes to determine the optimal node for data storage or retrieval.
    - Implements load balancing algorithms, such as round-robin, least-connection, or weighted distribution.
    - Adjusts the load balancing strategy dynamically based on current system conditions and node capacities.
3. Fault Detection and Recovery Module:
    - Detects faults or failures in storage nodes, such as hardware failures, network disruptions, or data corruption.
    - Monitors the availability and health of storage nodes through periodic checks or event-driven mechanisms.
    - Initiates fault recovery processes, such as data reconstruction, node replacement, or redistribution of data.
    - Notifies the system administrators or other relevant modules about detected faults or recovery actions.
4. Data Security Module:

- Implements data security measures to protect stored data from unauthorized access or data breaches.
- Handles data encryption, decryption, and key management to ensure data confidentiality.
- Enforces access control mechanisms, such as user authentication, authorization, and role-based access control.
- Integrates with secure communication protocols and cryptographic algorithms to ensure secure data transmission and storage.

5. User Interface Module:
    - Provides a user interface for system administrators or end-users to interact with the cloud secure storage mechanism.
    - Allows users to perform actions such as data upload, download, replication settings configuration, or monitoring system status.
    - Displays system alerts, reports, and visualizations related to data storage, load balancing, fault events, or security.
6. Communication Module:
    - Handles communication between different components or modules within the cloud secure storage system.
    - Implements communication protocols, such as HTTP, TCP/IP, or messaging protocols, for data transfer between storage nodes or with external systems.
    - Ensures reliable and secure communication channels between system components and external entities.

These modules represent distinct functional components that collaborate to provide fault-tolerant and load-balanced secure storage in the cloud environment. The specific modules and their functionalities may vary based on the design and requirements of the cloud secure storage mechanism.

6.2 Module Description

Here are detailed descriptions of the modules mentioned earlier in a cloud secure storage mechanism based on fault and load balancing:

1. Data Replication Module:

- Description: The Data Replication Module is responsible for replicating data across multiple storage nodes in order to ensure fault tolerance and data availability in the cloud storage system.
- Functionality:
    - Manages the creation, update, and deletion of data replicas across storage nodes.
    - Implements replication strategies such as data partitioning, data duplication, or erasure coding.
    - Monitors the health and status of storage nodes to determine the need for replication or recovery processes.
    - Initiates data replication based on predefined policies or in response to changes in the storage environment.

2. Load Balancer Module:
    - Description: The Load Balancer Module evenly distributes the workload and data across storage nodes to optimize resource utilization and enhance system performance in the cloud storage environment.
    - Functionality:
        - Monitors the load and capacity of storage nodes to determine the optimal node for storing or retrieving data.
        - Implements load balancing algorithms like round-robin, least-connection, or weighted distribution to assign tasks to storage nodes.
        - Dynamically adjusts the load balancing strategy based on the current system conditions and node capacities.
        - Collaborates with the Data Replication Module to consider the distribution of replicated data while load balancing.
3. Fault Detection and Recovery Module:
    - Description: The Fault Detection and Recovery Module detects faults or failures in storage nodes and initiates appropriate recovery processes to maintain data integrity and availability.
    - Functionality:
        - Monitors the availability and health of storage nodes through regular checks or event-driven mechanisms.

- Detects faults or failures, such as hardware failures, network disruptions, or data corruption.
- Triggers fault recovery processes, including data reconstruction, node replacement, or redistribution of data.
- Notifies system administrators or other relevant modules about detected faults and recovery actions.

4. Data Security Module:
    - Description: The Data Security Module ensures the confidentiality, integrity, and access control of data stored in the cloud storage system.
    - Functionality:
        - Implements data encryption, decryption, and key management to protect data confidentiality.
        - Enforces access control mechanisms, such as user authentication, authorization, and role-based access control.
        - Integrates with secure communication protocols and cryptographic algorithms for secure data transmission and storage.
        - Monitors and audits data access activities for compliance and security purposes.
5. User Interface Module:
    - Description: The User Interface Module provides a user-friendly interface for system administrators or end-users to interact with the cloud secure storage mechanism.
    - Functionality:
        - Enables users to perform actions such as data upload, download, replication settings configuration, or monitoring system status.
        - Presents system alerts, reports, and visualizations related to data storage, load balancing, fault events, or security.
        - Provides a graphical or command-line interface for easy system configuration and monitoring.
        - Supports user authentication and authorization for different levels of access and functionality.
6. Communication Module:

- Description: The Communication Module handles communication between different components or modules within the cloud secure storage system, as well as with external entities.
- Functionality:
    - Implements communication protocols like HTTP, TCP/IP, or messaging protocols for data transfer between storage nodes or with external systems.
    - Ensures reliable and secure communication channels between system components and external entities.
    - Handles data serialization, deserialization, and message formatting for efficient and standardized communication.
    - Manages network connectivity, error handling, and protocol-specific optimizations for efficient data transmission.

These modules work together to provide a secure, fault-tolerant, and load-balanced storage solution in the cloud environment. Each module has

6.3 Sample code

```
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;

import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.DatacenterCharacteristics;
import org.cloudbus.cloudsim.Host;
import org.cloudbus.cloudsim.Log;
import org.cloudbus.cloudsim.Pe;
import org.cloudbus.cloudsim.Storage;
import org.cloudbus.cloudsim.UtilizationModel;
import org.cloudbus.cloudsim.UtilizationModelFull;
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerTimeShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.core.SimEntity;
import org.cloudbus.cloudsim.core.SimEvent;
```

```
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;

public class Cloudsim {

	/** The cloudlet list. */
	public static List<Cloudlet> cloudletList;

	/** The vmList. */
	public static List<Vm> vmList;

	public static List<Vm> createVM(int userId, int vms, int idShift) {
		//Creates a container to store VMs. This list is passed to the broker
later
		LinkedList<Vm> list = new LinkedList<Vm>();

		//VM Parameters
		long size = 10000; //image size (MB)
		int ram = 512; //vm memory (MB)
		int mips = 250;
		long bw = 1000;
		int pesNumber = 1; //number of cpus
		String vmm = "Xen"; //VMM name

		//create VMs
		Vm[] vm = new Vm[vms];

		for(int i=0;i<vms;i++){
			vm[i] = new Vm(idShift + i, userId, mips, pesNumber, ram,
bw, size, vmm, new CloudletSchedulerTimeShared());
			list.add(vm[i]);
		}

		return list;
	}


	public static List<Cloudlet> createCloudlet(int userId, int cloudlets, int
idShift){
		// Creates a container to store Cloudlets
		LinkedList<Cloudlet> list = new LinkedList<Cloudlet>();

		//cloudlet parameters
		long length = 40000;
		long fileSize = 300;
		long outputSize = 300;
		int pesNumber = 1;
		UtilizationModel utilizationModel = new UtilizationModelFull();
```

```
                Cloudlet[] cloudlet = new Cloudlet[cloudlets];

                for(int i=0;i<cloudlets;i++){
                        cloudlet[i] = new Cloudlet(idShift + i, length, pesNumber,
fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel);
                        // setting the owner of these Cloudlets
                        cloudlet[i].setUserId(userId);
                        list.add(cloudlet[i]);
                }

                return list;
        }


        /////////////////////////// STATIC METHODS ///////////////////////

        /**
         * Creates main() to run this example
         */
        public static void main(String[] args) {
                Log.printLine("Starting CloudSimExample8...");

                try {
                        // First step: Initialize the CloudSim package. It should be
called
                        // before creating any entities.
                        int num_user = 2;   // number of grid users
                        Calendar calendar = Calendar.getInstance();
                        boolean trace_flag = false;  // mean trace events

                        // Initialize the CloudSim library
                        CloudSim.init(num_user, calendar, trace_flag);

                        GlobalBroker globalBroker = new
GlobalBroker("GlobalBroker");

                        // Second step: Create Datacenters
                        //Datacenters are the resource providers in CloudSim. We need
at list one of them to run a CloudSim simulation
                        Datacenter datacenter0 = createDatacenter("Datacenter_0");
                        Datacenter datacenter1 = createDatacenter("Datacenter_1");

                        //Third step: Create Broker
                        DatacenterBroker broker = createBroker("Broker_0");
                        int brokerId = broker.getId();

                        //Fourth step: Create VMs and Cloudlets and send them to
broker
                        vmList = createVM(brokerId, 5, 0); //creating 5 vms
```

```
                        cloudletList = createCloudlet(brokerId, 10, 0); // creating 10
cloudlets

                        broker.submitVmList(vmList);
                        broker.submitCloudletList(cloudletList);

                        // Fifth step: Starts the simulation
                        CloudSim.startSimulation();

                        // Final step: Print results when simulation is over
                        List<Cloudlet> newList = broker.getCloudletReceivedList();

        newList.addAll(globalBroker.getBroker().getCloudletReceivedList());

                        CloudSim.stopSimulation();

                        printCloudletList(newList);

                        //Print the debt of each user to each datacenter
                        datacenter0.printDebts();
                        datacenter1.printDebts();

                        Log.printLine("CloudSim Example8 finished!");
                }
                catch (Exception e)
                {
                        e.printStackTrace();
                        Log.printLine("The simulation has been terminated due to an
unexpected error");
                }
        }

        public static Datacenter createDatacenter(String name){

                // Here are the steps needed to create a PowerDatacenter:
                // 1. We need to create a list to store one or more
                //    Machines
                List<Host> hostList = new ArrayList<Host>();

                // 2. A Machine contains one or more PEs or CPUs/Cores. Therefore,
should
                //    create a list to store these PEs before creating
                //    a Machine.
                List<Pe> peList1 = new ArrayList<Pe>();

                int mips = 1000;

                // 3. Create PEs and add these into the list.
                //for a quad-core machine, a list of 4 PEs is required:
```

```
                peList1.add(new Pe(0, new PeProvisionerSimple(mips))); // need to
store Pe id and MIPS Rating
                peList1.add(new Pe(1, new PeProvisionerSimple(mips)));
                peList1.add(new Pe(2, new PeProvisionerSimple(mips)));
                peList1.add(new Pe(3, new PeProvisionerSimple(mips)));

                //Another list, for a dual-core machine
                List<Pe> peList2 = new ArrayList<Pe>();

                peList2.add(new Pe(0, new PeProvisionerSimple(mips)));
                peList2.add(new Pe(1, new PeProvisionerSimple(mips)));

                //4. Create Hosts with its id and list of PEs and add them to the list of
machines
                int hostId=0;
                int ram = 16384; //host memory (MB)
                long storage = 1000000; //host storage
                int bw = 10000;

                hostList.add(
                        new Host(
                                hostId,
                                new RamProvisionerSimple(ram),
                                new BwProvisionerSimple(bw),
                                storage,
                                peList1,
                                new VmSchedulerTimeShared(peList1)
                        )
                ); // This is our first machine

                hostId++;

                hostList.add(
                        new Host(
                                hostId,
                                new RamProvisionerSimple(ram),
                                new BwProvisionerSimple(bw),
                                storage,
                                peList2,
                                new VmSchedulerTimeShared(peList2)
                        )
                ); // Second machine

                // 5. Create a DatacenterCharacteristics object that stores the
                //    properties of a data center: architecture, OS, list of
                //    Machines, allocation policy: time- or space-shared, time zone
                //    and its price (G$/Pe time unit).
                String arch = "x86";      // system architecture
                String os = "Linux";          // operating system
                String vmm = "Xen";
```

```
                double time_zone = 10.0;         // time zone this resource located
                double cost = 3.0;              // the cost of using processing in this
resource
                double costPerMem = 0.05;               // the cost of using memory in
this resource
                double costPerStorage = 0.1;  // the cost of using storage in this
resource
                double costPerBw = 0.1;                         // the cost of using bw in
this resource
                LinkedList<Storage> storageList = new LinkedList<Storage>();
        //we are not adding SAN devices by now

                DatacenterCharacteristics characteristics = new
DatacenterCharacteristics(
           arch, os, vmm, hostList, time_zone, cost, costPerMem, costPerStorage,
costPerBw);


                // 6. Finally, we need to create a PowerDatacenter object.
                Datacenter datacenter = null;
                try {
                        datacenter = new Datacenter(name, characteristics, new
VmAllocationPolicySimple(hostList), storageList, 0);
                } catch (Exception e) {
                        e.printStackTrace();
                }

                return datacenter;
        }

        //We strongly encourage users to develop their own broker policies, to submit
vms and cloudlets according
        //to the specific rules of the simulated scenario
        public static DatacenterBroker createBroker(String name){

                DatacenterBroker broker = null;
                try {
                        broker = new DatacenterBroker(name);
                } catch (Exception e) {
                        e.printStackTrace();
                        return null;
                }
                return broker;
        }

        /**
         * Prints the Cloudlet objects
         * @param list  list of Cloudlets
         */
        public static void printCloudletList(List<Cloudlet> list) {
```

```
                int size = list.size();
                Cloudlet cloudlet;

                String indent = "    ";
                Log.printLine();
                Log.printLine("========== OUTPUT ==========");
                Log.printLine("Cloudlet ID" + indent + "STATUS" + indent +
                                "Data center ID" + indent + "VM ID" + indent + indent
+ "Time" + indent + "Start Time" + indent + "Finish Time");

                DecimalFormat dft = new DecimalFormat("###.##");
                for (int i = 0; i < size; i++) {
                        cloudlet = list.get(i);
                        Log.print(indent + cloudlet.getCloudletId() + indent + indent);

                        if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS){
                                Log.print("SUCCESS");

                                Log.printLine( indent + indent +
cloudlet.getResourceId() + indent + indent + indent + cloudlet.getVmId() +
                                                indent + indent + indent +
dft.format(cloudlet.getActualCPUTime()) +
                                                indent + indent +
dft.format(cloudlet.getExecStartTime())+ indent + indent + indent +
dft.format(cloudlet.getFinishTime()));
                        }
                }

        }

        public static class GlobalBroker extends SimEntity {

                public static final int CREATE_BROKER = 0;
                public List<Vm> vmList;
                public List<Cloudlet> cloudletList;
                public DatacenterBroker broker;

                public GlobalBroker(String name) {
                        super(name);
                }

                @Override
                public void processEvent(SimEvent ev) {
                        switch (ev.getTag()) {
                        case CREATE_BROKER:
                                setBroker(createBroker(super.getName()+"_"));

                                //Create VMs and Cloudlets and send them to broker
                                setVmList(createVM(getBroker().getId(), 5, 100));
//creating 5 vms
```

```
                        setCloudletList(createCloudlet(getBroker().getId(), 10,
100)); // creating 10 cloudlets

                        broker.submitVmList(getVmList());
                        broker.submitCloudletList(getCloudletList());

                        CloudSim.resumeSimulation();

                        break;

                default:
                        Log.printLine(getName() + ": unknown event type");
                        break;
                }
        }

        @Override
        public void startEntity() {
                Log.printLine(super.getName()+" is starting...");
                schedule(getId(), 200, CREATE_BROKER);
        }

        @Override
        public void shutdownEntity() {
        }

        public List<Vm> getVmList() {
                return vmList;
        }

        protected void setVmList(List<Vm> vmList) {
                this.vmList = vmList;
        }

        public List<Cloudlet> getCloudletList() {
                return cloudletList;
        }

        protected void setCloudletList(List<Cloudlet> cloudletList) {
                this.cloudletList = cloudletList;
        }

        public DatacenterBroker getBroker() {
                return broker;
        }

        protected void setBroker(DatacenterBroker broker) {
                this.broker = broker;
        }
```

```
        }
    }
```

CHAPTER 7

RESULTS

7.1 Screen shots

```
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will be changed number of VMs:5
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:6032.0977
LCA Threshold for this season:2121.3716
Season 8
League 0 will be changed number of VMs:0
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:6
League 3 will be changed number of VMs:9
League 4 will be changed number of VMs:2
League 5 will be changed number of VMs:7
League 6 will be changed number of VMs:5
League 7 will be changed number of VMs:2
League 8 will be changed number of VMs:8
League 9 will be changed number of VMs:4
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:1
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:2
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:9
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:0
Mean fitness for this season:13653.094
LCA Threshold for this season:4801.528
Season 9
League 0 will NOT be changed
League 1 will be changed number of VMs:0
League 2 will be changed number of VMs:3
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:5
League 7 will be changed number of VMs:3
```

```
League 8 will NOT be changed
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:1
League 11 will be changed number of VMs:8
League 12 will be changed number of VMs:8
League 13 will NOT be changed
League 14 will be changed number of VMs:1
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:0
League 19 will be changed number of VMs:5
Mean fitness for this season:5998.4424
LCA Threshold for this season:2109.5356
Season 10
League 0 will be changed number of VMs:9
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:4
League 3 will be changed number of VMs:6
League 4 will be changed number of VMs:4
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:6
League 8 will be changed number of VMs:3
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:6
League 12 will be changed number of VMs:3
League 13 will be changed number of VMs:1
League 14 will be changed number of VMs:7
League 15 will be changed number of VMs:5
League 16 will be changed number of VMs:9
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:5
Mean fitness for this season:143650.95
LCA Threshold for this season:50519.25
Best solution found at index:9
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #5 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
Task 58, executing on Vm:2
Task 72, executing on Vm:0
Task 77, executing on Vm:1
Task 79, executing on Vm:3
Task 82, executing on Vm:2
Task 90, executing on Vm:2
Task 94, executing on Vm:2
Task 99, executing on Vm:3
Simulation completed.
All tasks completed
Season 1
```

```
League 0 will be changed number of VMs:4
League 1 will be changed number of VMs:5
League 2 will be changed number of VMs:8
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:6
League 5 will be changed number of VMs:2
League 6 will be changed number of VMs:4
League 7 will be changed number of VMs:0
League 8 will be changed number of VMs:0
League 9 will be changed number of VMs:2
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:9
League 13 will be changed number of VMs:7
League 14 will be changed number of VMs:0
League 15 will be changed number of VMs:1
League 16 will be changed number of VMs:4
League 17 will be changed number of VMs:4
League 18 will be changed number of VMs:1
League 19 will be changed number of VMs:4
Mean fitness for this season:140194.08
LCA Threshold for this season:96810.31
Season 2
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will be changed number of VMs:1
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:9003.954
LCA Threshold for this season:6217.6353
Season 3
League 0 will be changed number of VMs:4
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will be changed number of VMs:1
League 5 will be changed number of VMs:0
League 6 will NOT be changed
League 7 will be changed number of VMs:4
League 8 will NOT be changed
League 9 will be changed number of VMs:4
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
```

```
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will be changed number of VMs:0
League 16 will be changed number of VMs:3
League 17 will be changed number of VMs:2
League 18 will NOT be changed
League 19 will be changed number of VMs:3
Mean fitness for this season:6346.7505
LCA Threshold for this season:4382.717
Season 4
League 0 will be changed number of VMs:9
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:3
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:1
League 8 will NOT be changed
League 9 will be changed number of VMs:9
League 10 will NOT be changed
League 11 will be changed number of VMs:6
League 12 will NOT be changed
League 13 will be changed number of VMs:1
League 14 will NOT be changed
League 15 will be changed number of VMs:8
League 16 will be changed number of VMs:5
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will be changed number of VMs:2
Mean fitness for this season:4124.3955
LCA Threshold for this season:2848.0806
Season 5
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:6
League 2 will be changed number of VMs:2
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:6
League 5 will be changed number of VMs:5
League 6 will be changed number of VMs:2
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:3
League 9 will be changed number of VMs:8
League 10 will be changed number of VMs:2
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:3
League 14 will be changed number of VMs:2
League 15 will be changed number of VMs:4
League 16 will be changed number of VMs:6
League 17 will be changed number of VMs:6
League 18 will be changed number of VMs:7
League 19 will be changed number of VMs:5
Mean fitness for this season:5665.3325
LCA Threshold for this season:3912.1667
Season 6
League 0 will be changed number of VMs:7
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:4
```

```
League 3 will be changed number of VMs:9
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:7
League 6 will be changed number of VMs:6
League 7 will NOT be changed
League 8 will be changed number of VMs:2
League 9 will NOT be changed
League 10 will be changed number of VMs:9
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:9
League 15 will be changed number of VMs:0
League 16 will be changed number of VMs:4
League 17 will NOT be changed
League 18 will be changed number of VMs:6
League 19 will NOT be changed
Mean fitness for this season:4554.3135
LCA Threshold for this season:3144.9583
Season 7
League 0 will be changed number of VMs:7
League 1 will be changed number of VMs:3
League 2 will be changed number of VMs:3
League 3 will be changed number of VMs:1
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:6
League 6 will be changed number of VMs:4
League 7 will be changed number of VMs:8
League 8 will be changed number of VMs:2
League 9 will be changed number of VMs:4
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:7
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:8
League 14 will be changed number of VMs:8
League 15 will be changed number of VMs:6
League 16 will be changed number of VMs:8
League 17 will be changed number of VMs:8
League 18 will be changed number of VMs:6
League 19 will be changed number of VMs:2
Mean fitness for this season:7875.7295
LCA Threshold for this season:5438.5454
Season 8
League 0 will NOT be changed
League 1 will be changed number of VMs:1
League 2 will be changed number of VMs:9
League 3 will NOT be changed
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:7
League 6 will be changed number of VMs:9
League 7 will NOT be changed
League 8 will be changed number of VMs:8
League 9 will be changed number of VMs:5
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
```

```
League 16 will NOT be changed
League 17 will be changed number of VMs:1
League 18 will be changed number of VMs:6
League 19 will be changed number of VMs:9
Mean fitness for this season:4679.076
LCA Threshold for this season:3231.1125
Season 9
League 0 will be changed number of VMs:2
League 1 will be changed number of VMs:5
League 2 will be changed number of VMs:2
League 3 will be changed number of VMs:8
League 4 will be changed number of VMs:4
League 5 will be changed number of VMs:7
League 6 will NOT be changed
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:0
League 9 will be changed number of VMs:5
League 10 will be changed number of VMs:8
League 11 will NOT be changed
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:8
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:3
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will be changed number of VMs:8
Mean fitness for this season:138127.45
LCA Threshold for this season:95383.22
Season 10
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will be changed number of VMs:6
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:6572.9634
LCA Threshold for this season:4538.927
Best solution found at index:1
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
```

```
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #7 in Datacenter_0
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #8 in Datacenter_0
0.0: Broker: Trying to Create VM #6 in Datacenter_0
Task 4, executing on Vm:4 for 149 cycles
Task 9, executing on Vm:4
Task 10, executing on Vm:4
Task 11, executing on Vm:4
Task 12, executing on Vm:4
Task 13, executing on Vm:4
Task 14, executing on Vm:4
Task 15, executing on Vm:4
Task 16, executing on Vm:4
Task 17, executing on Vm:4
Task 18, executing on Vm:4
Task 19, executing on Vm:4
Task 20, executing on Vm:4
Task 21, executing on Vm:4
Task 22, executing on Vm:4
Task 23, executing on Vm:4
Task 24, executing on Vm:4 for 1 cycles
Task 29, executing on Vm:4
Task 30, executing on Vm:4
Task 31, executing on Vm:4
Task 32, executing on Vm:4
Task 33, executing on Vm:4
Task 34, executing on Vm:4
Task 35, executing on Vm:4
Task 36, executing on Vm:4
Task 37, executing on Vm:4
Task 38, executing on Vm:4
Task 39, executing on Vm:4
Task 40, executing on Vm:4
Task 41, executing on Vm:4
Task 42, executing on Vm:4 for 11 cycles
Task 47, executing on Vm:4
Task 48, executing on Vm:4
Task 49, executing on Vm:4
Task 50, executing on Vm:4
Task 51, executing on Vm:4
Task 52, executing on Vm:4
Task 53, executing on Vm:4
Task 54, executing on Vm:4
Task 55, executing on Vm:4
Task 56, executing on Vm:4
Task 57, executing on Vm:4
Task 58, executing on Vm:4
Task 59, executing on Vm:4 for 43 cycles
Task 64, executing on Vm:4
Task 65, executing on Vm:4 for 146 cycles
Task 70, executing on Vm:4
Task 71, executing on Vm:4
Task 72, executing on Vm:4
Task 73, executing on Vm:4 for 120 cycles
Task 77, executing on Vm:3
Task 78, executing on Vm:4 for 149 cycles
Task 79, executing on Vm:0
Task 82, executing on Vm:3
```

```
Task 83, executing on Vm:4
Task 84, executing on Vm:4 for 134 cycles
Task 89, executing on Vm:4 for 149 cycles
Task 90, executing on Vm:0
Task 94, executing on Vm:4
Task 95, executing on Vm:4 for 149 cycles
Task 99, executing on Vm:3
Task 100, executing on Vm:4 for 149 cycles
Simulation completed.
All tasks completed
Season 1
League 0 will be changed number of VMs:0
League 1 will be changed number of VMs:8
League 2 will be changed number of VMs:9
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:0
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:1
League 8 will be changed number of VMs:7
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:2
League 12 will be changed number of VMs:5
League 13 will be changed number of VMs:1
League 14 will be changed number of VMs:9
League 15 will be changed number of VMs:1
League 16 will be changed number of VMs:7
League 17 will be changed number of VMs:6
League 18 will be changed number of VMs:1
League 19 will be changed number of VMs:4
Mean fitness for this season:17532.29
LCA Threshold for this season:7866.2417
Season 2
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:5
League 2 will NOT be changed
League 3 will be changed number of VMs:4
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:0
League 6 will be changed number of VMs:7
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:7
League 12 will be changed number of VMs:3
League 13 will NOT be changed
League 14 will be changed number of VMs:2
League 15 will be changed number of VMs:1
League 16 will NOT be changed
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:6
League 19 will be changed number of VMs:7
Mean fitness for this season:22269.035
LCA Threshold for this season:9991.485
Season 3
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:4
```

```
League 2 will NOT be changed
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:0
League 6 will be changed number of VMs:2
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will be changed number of VMs:0
League 11 will NOT be changed
League 12 will be changed number of VMs:9
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will be changed number of VMs:1
League 16 will NOT be changed
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:7
League 19 will be changed number of VMs:2
Mean fitness for this season:10125.703
LCA Threshold for this season:4543.116
Season 4
League 0 will be changed number of VMs:0
League 1 will be changed number of VMs:4
League 2 will be changed number of VMs:7
League 3 will be changed number of VMs:1
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:5
League 7 will be changed number of VMs:4
League 8 will be changed number of VMs:7
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:7
League 11 will be changed number of VMs:1
League 12 will be changed number of VMs:7
League 13 will be changed number of VMs:9
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:8
League 17 will be changed number of VMs:1
League 18 will be changed number of VMs:8
League 19 will be changed number of VMs:5
Mean fitness for this season:13103.193
LCA Threshold for this season:5879.032
Season 5
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:9
League 2 will be changed number of VMs:9
League 3 will be changed number of VMs:9
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:9
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:5
League 8 will be changed number of VMs:3
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:5
League 12 will be changed number of VMs:0
League 13 will be changed number of VMs:3
League 14 will be changed number of VMs:8
```

```
League 15 will be changed number of VMs:5
League 16 will be changed number of VMs:9
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:4
Mean fitness for this season:143030.61
LCA Threshold for this season:64173.785
Season 6
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will be changed number of VMs:9
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:11592.811
LCA Threshold for this season:5201.366
Season 7
League 0 will be changed number of VMs:0
League 1 will be changed number of VMs:3
League 2 will be changed number of VMs:8
League 3 will be changed number of VMs:8
League 4 will be changed number of VMs:3
League 5 will be changed number of VMs:9
League 6 will be changed number of VMs:8
League 7 will be changed number of VMs:0
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:3
League 10 will be changed number of VMs:8
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:3
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:7
League 15 will be changed number of VMs:3
League 16 will be changed number of VMs:8
League 17 will be changed number of VMs:9
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:6
Mean fitness for this season:150234.02
LCA Threshold for this season:67405.75
Season 8
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
```

```
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will be changed number of VMs:8
League 8 will NOT be changed
League 9 will be changed number of VMs:2
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:13897.711
LCA Threshold for this season:6235.51
Season 9
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:8
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:0
League 5 will be changed number of VMs:6
League 6 will be changed number of VMs:8
League 7 will be changed number of VMs:7
League 8 will be changed number of VMs:9
League 9 will be changed number of VMs:6
League 10 will be changed number of VMs:1
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:1
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:5
League 15 will be changed number of VMs:6
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:7
Mean fitness for this season:19268.613
LCA Threshold for this season:8645.281
Season 10
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:3
League 2 will NOT be changed
League 3 will be changed number of VMs:7
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will be changed number of VMs:4
League 7 will be changed number of VMs:0
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:6
League 15 will be changed number of VMs:4
League 16 will NOT be changed
League 17 will be changed number of VMs:5
```

```
League 18 will be changed number of VMs:3
League 19 will NOT be changed
Mean fitness for this season:18405.582
LCA Threshold for this season:8258.063
Best solution found at index:12
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #7 in Datacenter_0
0.0: Broker: Trying to Create VM #8 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #6 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
Task 5, executing on Vm:5
Task 6, executing on Vm:5
Task 7, executing on Vm:5
Task 8, executing on Vm:5 for 10 cycles
Task 14, executing on Vm:5
Task 15, executing on Vm:5 for 10 cycles
Task 21, executing on Vm:5
Task 22, executing on Vm:5
Task 23, executing on Vm:5
Task 24, executing on Vm:5 for 3 cycles
Task 30, executing on Vm:5
Task 31, executing on Vm:5 for 10 cycles
Task 37, executing on Vm:5
Task 38, executing on Vm:5 for 21 cycles
Task 44, executing on Vm:5
Task 45, executing on Vm:5 for 9 cycles
Task 51, executing on Vm:5
Task 52, executing on Vm:5
Task 53, executing on Vm:5 for 5 cycles
Task 58, executing on Vm:4
Task 59, executing on Vm:5 for 24 cycles
Task 65, executing on Vm:5 for 24 cycles
Task 71, executing on Vm:5
Task 72, executing on Vm:5
Task 73, executing on Vm:5 for 9 cycles
Task 77, executing on Vm:3
Task 79, executing on Vm:5
Task 80, executing on Vm:5 for 24 cycles
Task 82, executing on Vm:1
Task 86, executing on Vm:5
Task 87, executing on Vm:5 for 8 cycles
Task 90, executing on Vm:2
Task 93, executing on Vm:5
Task 94, executing on Vm:5
Task 95, executing on Vm:5 for 21 cycles
Task 99, executing on Vm:3
Simulation completed.
All tasks completed
Season 1
League 0 will be changed number of VMs:3
League 1 will be changed number of VMs:4
League 2 will be changed number of VMs:5
```

```
League 3 will be changed number of VMs:8
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:9
League 6 will be changed number of VMs:8
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:5
League 9 will be changed number of VMs:7
League 10 will be changed number of VMs:4
League 11 will be changed number of VMs:6
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:8
League 14 will be changed number of VMs:8
League 15 will be changed number of VMs:4
League 16 will be changed number of VMs:8
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:6
League 19 will be changed number of VMs:0
Mean fitness for this season:6430.5986
LCA Threshold for this season:3460.2568
Season 2
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:6
League 2 will be changed number of VMs:1
League 3 will NOT be changed
League 4 will be changed number of VMs:5
League 5 will NOT be changed
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:3
League 8 will be changed number of VMs:4
League 9 will NOT be changed
League 10 will be changed number of VMs:4
League 11 will be changed number of VMs:1
League 12 will be changed number of VMs:1
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will be changed number of VMs:0
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:0
League 19 will be changed number of VMs:5
Mean fitness for this season:10530.6
LCA Threshold for this season:5666.4365
Season 3
League 0 will be changed number of VMs:3
League 1 will be changed number of VMs:3
League 2 will be changed number of VMs:0
League 3 will NOT be changed
League 4 will be changed number of VMs:1
League 5 will NOT be changed
League 6 will be changed number of VMs:0
League 7 will be changed number of VMs:4
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will be changed number of VMs:4
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
```

```
League 16 will NOT be changed
League 17 will be changed number of VMs:7
League 18 will NOT be changed
League 19 will be changed number of VMs:5
Mean fitness for this season:4816.1523
LCA Threshold for this season:2591.5354
Season 4
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:7
League 2 will be changed number of VMs:8
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:3
League 5 will be changed number of VMs:4
League 6 will be changed number of VMs:8
League 7 will be changed number of VMs:4
League 8 will be changed number of VMs:2
League 9 will be changed number of VMs:3
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:4
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:7
League 15 will be changed number of VMs:3
League 16 will be changed number of VMs:6
League 17 will be changed number of VMs:7
League 18 will be changed number of VMs:8
League 19 will be changed number of VMs:8
Mean fitness for this season:6833.5273
LCA Threshold for this season:3677.0698
Season 5
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:1
League 2 will be changed number of VMs:2
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:2
League 5 will be changed number of VMs:5
League 6 will be changed number of VMs:6
League 7 will be changed number of VMs:4
League 8 will be changed number of VMs:0
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:6
League 11 will be changed number of VMs:6
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:5
League 14 will NOT be changed
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:6
League 18 will be changed number of VMs:2
League 19 will NOT be changed
Mean fitness for this season:6395.1436
LCA Threshold for this season:3441.1787
Season 6
League 0 will be changed number of VMs:3
League 1 will be changed number of VMs:4
League 2 will be changed number of VMs:1
League 3 will be changed number of VMs:2
League 4 will be changed number of VMs:8
League 5 will be changed number of VMs:9
```

```
League 6 will be changed number of VMs:5
League 7 will be changed number of VMs:1
League 8 will be changed number of VMs:7
League 9 will be changed number of VMs:5
League 10 will be changed number of VMs:7
League 11 will NOT be changed
League 12 will be changed number of VMs:5
League 13 will be changed number of VMs:6
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will be changed number of VMs:5
League 17 will be changed number of VMs:9
League 18 will be changed number of VMs:3
League 19 will NOT be changed
Mean fitness for this season:5192.62
LCA Threshold for this season:2794.1099
Season 7
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:5
League 2 will be changed number of VMs:7
League 3 will be changed number of VMs:1
League 4 will be changed number of VMs:2
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:9
League 7 will be changed number of VMs:8
League 8 will be changed number of VMs:3
League 9 will be changed number of VMs:5
League 10 will be changed number of VMs:7
League 11 will be changed number of VMs:5
League 12 will be changed number of VMs:0
League 13 will be changed number of VMs:1
League 14 will be changed number of VMs:8
League 15 will be changed number of VMs:8
League 16 will be changed number of VMs:8
League 17 will be changed number of VMs:1
League 18 will be changed number of VMs:9
League 19 will be changed number of VMs:0
Mean fitness for this season:11161.975
LCA Threshold for this season:6006.175
Season 8
League 0 will NOT be changed
League 1 will be changed number of VMs:4
League 2 will NOT be changed
League 3 will be changed number of VMs:1
League 4 will be changed number of VMs:3
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:6
League 10 will NOT be changed
League 11 will be changed number of VMs:7
League 12 will be changed number of VMs:3
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
```

```
League 19 will NOT be changed
Mean fitness for this season:4823.842
LCA Threshold for this season:2595.673
Season 9
League 0 will be changed number of VMs:2
League 1 will be changed number of VMs:8
League 2 will be changed number of VMs:0
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:7
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:3
League 8 will be changed number of VMs:2
League 9 will be changed number of VMs:6
League 10 will be changed number of VMs:4
League 11 will be changed number of VMs:2
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:2
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:1
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:8
League 18 will be changed number of VMs:9
League 19 will be changed number of VMs:7
Mean fitness for this season:5819.3916
LCA Threshold for this season:3131.3708
Season 10
League 0 will be changed number of VMs:9
League 1 will be changed number of VMs:0
League 2 will NOT be changed
League 3 will be changed number of VMs:8
League 4 will be changed number of VMs:7
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:2
League 7 will be changed number of VMs:7
League 8 will be changed number of VMs:2
League 9 will be changed number of VMs:9
League 10 will be changed number of VMs:8
League 11 will NOT be changed
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:7
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:9
League 16 will NOT be changed
League 17 will be changed number of VMs:2
League 18 will NOT be changed
League 19 will be changed number of VMs:8
Mean fitness for this season:4041.6323
LCA Threshold for this season:2174.772
Best solution found at index:6
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #6 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
```

```
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
Task 0, executing on Vm:0 for 219 cycles
Task 5, executing on Vm:0
Task 6, executing on Vm:0
Task 7, executing on Vm:0
Task 8, executing on Vm:0
Task 9, executing on Vm:0
Task 10, executing on Vm:0
Task 11, executing on Vm:0
Task 12, executing on Vm:0
Task 13, executing on Vm:0
Task 14, executing on Vm:0
Task 15, executing on Vm:0
Task 16, executing on Vm:0
Task 17, executing on Vm:0
Task 18, executing on Vm:0
Task 19, executing on Vm:0
Task 20, executing on Vm:0
Task 21, executing on Vm:0
Task 22, executing on Vm:0
Task 23, executing on Vm:0
Task 24, executing on Vm:0
Task 25, executing on Vm:0
Task 26, executing on Vm:0 for 12 cycles
Task 31, executing on Vm:0
Task 32, executing on Vm:0
Task 33, executing on Vm:0
Task 34, executing on Vm:0
Task 35, executing on Vm:0
Task 36, executing on Vm:0
Task 37, executing on Vm:0
Task 38, executing on Vm:0
Task 39, executing on Vm:0
Task 40, executing on Vm:0
Task 41, executing on Vm:0
Task 42, executing on Vm:0
Task 43, executing on Vm:0
Task 44, executing on Vm:0
Task 45, executing on Vm:0
Task 46, executing on Vm:0
Task 47, executing on Vm:0
Task 48, executing on Vm:0
Task 49, executing on Vm:0
Task 50, executing on Vm:0
Task 51, executing on Vm:0
Task 52, executing on Vm:0
Task 53, executing on Vm:0 for 3 cycles
Task 58, executing on Vm:0
Task 59, executing on Vm:0 for 219 cycles
Task 64, executing on Vm:0
Task 65, executing on Vm:0 for 216 cycles
Task 70, executing on Vm:0
Task 71, executing on Vm:0
Task 72, executing on Vm:0
Task 73, executing on Vm:0 for 190 cycles
Task 77, executing on Vm:4
Task 78, executing on Vm:0 for 219 cycles
Task 79, executing on Vm:1
```

```
Task 82, executing on Vm:4
Task 83, executing on Vm:0
Task 84, executing on Vm:0 for 204 cycles
Task 89, executing on Vm:0
Task 90, executing on Vm:0
Task 91, executing on Vm:0 for 3 cycles
Task 94, executing on Vm:3
Task 96, executing on Vm:0 for 219 cycles
Task 99, executing on Vm:3
Simulation completed.
All tasks completed
Season 1
League 0 will be changed number of VMs:4
League 1 will be changed number of VMs:0
League 2 will be changed number of VMs:0
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:8
League 5 will be changed number of VMs:5
League 6 will be changed number of VMs:8
League 7 will be changed number of VMs:1
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:4
League 11 will be changed number of VMs:2
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:3
League 14 will be changed number of VMs:1
League 15 will be changed number of VMs:8
League 16 will be changed number of VMs:9
League 17 will be changed number of VMs:8
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:0
Mean fitness for this season:9868.768
LCA Threshold for this season:2530.0784
Season 2
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:6
League 2 will be changed number of VMs:0
League 3 will be changed number of VMs:4
League 4 will be changed number of VMs:7
League 5 will be changed number of VMs:3
League 6 will be changed number of VMs:2
League 7 will be changed number of VMs:5
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:2
League 11 will be changed number of VMs:7
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:8
League 14 will be changed number of VMs:6
League 15 will be changed number of VMs:8
League 16 will be changed number of VMs:8
League 17 will be changed number of VMs:7
League 18 will be changed number of VMs:8
League 19 will be changed number of VMs:6
Mean fitness for this season:5470.8735
LCA Threshold for this season:1402.5803
Season 3
League 0 will be changed number of VMs:5
```

```
League 1 will be changed number of VMs:5
League 2 will be changed number of VMs:0
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:8
League 5 will be changed number of VMs:5
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:4
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:5
League 11 will be changed number of VMs:1
League 12 will be changed number of VMs:4
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:3
League 15 will be changed number of VMs:8
League 16 will be changed number of VMs:4
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:9
League 19 will be changed number of VMs:2
Mean fitness for this season:5887.7686
LCA Threshold for this season:1509.4606
Season 4
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:1
League 2 will be changed number of VMs:5
League 3 will be changed number of VMs:6
League 4 will be changed number of VMs:0
League 5 will be changed number of VMs:9
League 6 will be changed number of VMs:8
League 7 will be changed number of VMs:3
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:6
League 11 will be changed number of VMs:8
League 12 will be changed number of VMs:0
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:9
League 15 will be changed number of VMs:5
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:0
League 19 will be changed number of VMs:4
Mean fitness for this season:5878.199
LCA Threshold for this season:1507.0073
Season 5
League 0 will be changed number of VMs:9
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:8
League 3 will be changed number of VMs:9
League 4 will be changed number of VMs:4
League 5 will be changed number of VMs:7
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:5
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:5
League 10 will be changed number of VMs:8
League 11 will be changed number of VMs:2
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:8
```

```
League 14 will be changed number of VMs:9
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:1
League 17 will be changed number of VMs:8
League 18 will be changed number of VMs:8
League 19 will be changed number of VMs:7
Mean fitness for this season:6642.4727
LCA Threshold for this season:1702.9458
Season 6
League 0 will be changed number of VMs:7
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:9
League 3 will be changed number of VMs:6
League 4 will be changed number of VMs:1
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:0
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:2
League 9 will be changed number of VMs:8
League 10 will be changed number of VMs:5
League 11 will be changed number of VMs:6
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:3
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:1
League 17 will be changed number of VMs:5
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:7
Mean fitness for this season:6184.3135
LCA Threshold for this season:1585.4865
Season 7
League 0 will be changed number of VMs:8
League 1 will be changed number of VMs:3
League 2 will be changed number of VMs:4
League 3 will be changed number of VMs:8
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:6
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:6
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:2
League 11 will be changed number of VMs:8
League 12 will be changed number of VMs:0
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:5
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:9
Mean fitness for this season:9637.272
LCA Threshold for this season:2470.7295
Season 8
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:4
League 2 will be changed number of VMs:5
League 3 will be changed number of VMs:4
```

```
League 4 will be changed number of VMs:7
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:1
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:6
League 10 will be changed number of VMs:7
League 11 will be changed number of VMs:0
League 12 will be changed number of VMs:7
League 13 will be changed number of VMs:3
League 14 will be changed number of VMs:6
League 15 will be changed number of VMs:1
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:8
League 19 will be changed number of VMs:9
Mean fitness for this season:8370.452
LCA Threshold for this season:2145.952
Season 9
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:7
League 2 will be changed number of VMs:2
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:8
League 5 will be changed number of VMs:6
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:8
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:0
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:9
League 17 will be changed number of VMs:9
League 18 will be changed number of VMs:0
League 19 will be changed number of VMs:7
Mean fitness for this season:4700.8057
LCA Threshold for this season:1205.1562
Season 10
League 0 will be changed number of VMs:9
League 1 will be changed number of VMs:4
League 2 will be changed number of VMs:1
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:2
League 5 will be changed number of VMs:1
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:3
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:6
League 10 will be changed number of VMs:9
League 11 will be changed number of VMs:7
League 12 will be changed number of VMs:0
League 13 will be changed number of VMs:6
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:3
League 16 will be changed number of VMs:2
```

```
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:8
Mean fitness for this season:9668.362
LCA Threshold for this season:2478.7
Best solution found at index:17
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #6 in Datacenter_0
0.0: Broker: Trying to Create VM #7 in Datacenter_0
0.0: Broker: Trying to Create VM #0 in Datacenter_0
Task 1, executing on Vm:1 for 38 cycles
Task 4, executing on Vm:1 for 38 cycles
Task 7, executing on Vm:1
Task 8, executing on Vm:1
Task 9, executing on Vm:1
Task 10, executing on Vm:1 for 11 cycles
Task 13, executing on Vm:1
Task 14, executing on Vm:1
Task 15, executing on Vm:1
Task 16, executing on Vm:1 for 7 cycles
Task 19, executing on Vm:1
Task 20, executing on Vm:1
Task 21, executing on Vm:1
Task 22, executing on Vm:1
Task 23, executing on Vm:1
Task 24, executing on Vm:1 for 9 cycles
Task 27, executing on Vm:1
Task 28, executing on Vm:1
Task 29, executing on Vm:1
Task 30, executing on Vm:1 for 6 cycles
Task 33, executing on Vm:1
Task 34, executing on Vm:1
Task 35, executing on Vm:1
Task 36, executing on Vm:1
Task 37, executing on Vm:1
Task 38, executing on Vm:1 for 1 cycles
Task 41, executing on Vm:1
Task 42, executing on Vm:1
Task 43, executing on Vm:1
Task 44, executing on Vm:1 for 5 cycles
Task 47, executing on Vm:1
Task 48, executing on Vm:1
Task 49, executing on Vm:1
Task 50, executing on Vm:1
Task 51, executing on Vm:1
Task 52, executing on Vm:1 for 12 cycles
Task 55, executing on Vm:1
Task 56, executing on Vm:1
Task 57, executing on Vm:1
Task 58, executing on Vm:1
Task 59, executing on Vm:1 for 10 cycles
Task 62, executing on Vm:1
Task 63, executing on Vm:1
Task 64, executing on Vm:1
```

```
Task 65, executing on Vm:1 for 17 cycles
Task 68, executing on Vm:1
Task 69, executing on Vm:1
Task 70, executing on Vm:1 for 2 cycles
Task 72, executing on Vm:0
Task 73, executing on Vm:1 for 38 cycles
Task 76, executing on Vm:1
Task 77, executing on Vm:1
Task 78, executing on Vm:1 for 35 cycles
Task 79, executing on Vm:2
Task 81, executing on Vm:1
Task 82, executing on Vm:1
Task 83, executing on Vm:1
Task 84, executing on Vm:1 for 19 cycles
Task 87, executing on Vm:1
Task 88, executing on Vm:1
Task 89, executing on Vm:1 for 18 cycles
Task 90, executing on Vm:2
Task 92, executing on Vm:1 for 38 cycles
Task 94, executing on Vm:0
Task 95, executing on Vm:1 for 38 cycles
Task 98, executing on Vm:1
Task 99, executing on Vm:1
Task 100, executing on Vm:1 for 31 cycles
Simulation completed.
All tasks completed
Season 1
League 0 will be changed number of VMs:1
League 1 will be changed number of VMs:7
League 2 will be changed number of VMs:6
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:6
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:5
League 7 will be changed number of VMs:4
League 8 will be changed number of VMs:7
League 9 will be changed number of VMs:3
League 10 will be changed number of VMs:8
League 11 will be changed number of VMs:5
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:9
League 14 will be changed number of VMs:6
League 15 will be changed number of VMs:9
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:5
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:3
Mean fitness for this season:16820.17
LCA Threshold for this season:2054.3572
Season 2
League 0 will be changed number of VMs:5
League 1 will be changed number of VMs:3
League 2 will be changed number of VMs:1
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:4
League 6 will be changed number of VMs:9
League 7 will be changed number of VMs:1
League 8 will be changed number of VMs:1
```

```
League 9 will be changed number of VMs:8
League 10 will be changed number of VMs:5
League 11 will be changed number of VMs:1
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:8
League 14 will be changed number of VMs:7
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:6
Mean fitness for this season:9536.009
LCA Threshold for this season:1164.6951
Season 3
League 0 will be changed number of VMs:3
League 1 will be changed number of VMs:7
League 2 will be changed number of VMs:3
League 3 will be changed number of VMs:2
League 4 will be changed number of VMs:2
League 5 will be changed number of VMs:9
League 6 will be changed number of VMs:6
League 7 will be changed number of VMs:6
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:9
League 10 will be changed number of VMs:5
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:1
League 13 will be changed number of VMs:9
League 14 will be changed number of VMs:3
League 15 will be changed number of VMs:6
League 16 will be changed number of VMs:5
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:0
League 19 will be changed number of VMs:4
Mean fitness for this season:10632.193
LCA Threshold for this season:1298.5792
Season 4
League 0 will be changed number of VMs:7
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:3
League 3 will be changed number of VMs:0
League 4 will be changed number of VMs:2
League 5 will be changed number of VMs:5
League 6 will be changed number of VMs:2
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:6
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:2
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:2
League 14 will be changed number of VMs:8
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:0
Mean fitness for this season:17442.605
LCA Threshold for this season:2130.3794
```

```
Season 5
League 0 will be changed number of VMs:3
League 1 will be changed number of VMs:9
League 2 will be changed number of VMs:7
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:7
League 5 will be changed number of VMs:4
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:3
League 8 will be changed number of VMs:7
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:6
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:1
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:9
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:4
Mean fitness for this season:22152.863
LCA Threshold for this season:2705.6738
Season 6
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:0
League 2 will be changed number of VMs:6
League 3 will be changed number of VMs:7
League 4 will be changed number of VMs:0
League 5 will be changed number of VMs:5
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:6
League 8 will be changed number of VMs:0
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:2
League 12 will be changed number of VMs:7
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:2
League 15 will be changed number of VMs:9
League 16 will be changed number of VMs:7
League 17 will be changed number of VMs:8
League 18 will be changed number of VMs:8
League 19 will be changed number of VMs:7
Mean fitness for this season:9132.197
LCA Threshold for this season:1115.3749
Season 7
League 0 will be changed number of VMs:9
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:8
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:8
League 5 will be changed number of VMs:4
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:3
League 8 will be changed number of VMs:0
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:9
League 11 will be changed number of VMs:0
```

```
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:6
League 14 will be changed number of VMs:9
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:6
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:0
League 19 will be changed number of VMs:5
Mean fitness for this season:23586.69
LCA Threshold for this season:2880.7964
Season 8
League 0 will be changed number of VMs:0
League 1 will be changed number of VMs:5
League 2 will be changed number of VMs:7
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:4
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:0
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:1
League 11 will be changed number of VMs:4
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:6
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:6
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:5
League 19 will be changed number of VMs:2
Mean fitness for this season:9860.534
LCA Threshold for this season:1204.3314
Season 9
League 0 will be changed number of VMs:7
League 1 will be changed number of VMs:6
League 2 will be changed number of VMs:1
League 3 will be changed number of VMs:4
League 4 will be changed number of VMs:4
League 5 will be changed number of VMs:6
League 6 will be changed number of VMs:7
League 7 will be changed number of VMs:5
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:5
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:8
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:1
League 14 will be changed number of VMs:1
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:1
League 17 will be changed number of VMs:9
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:5
Mean fitness for this season:9065.4375
LCA Threshold for this season:1107.2211
Season 10
League 0 will be changed number of VMs:4
League 1 will be changed number of VMs:8
```

```
League 2 will be changed number of VMs:1
League 3 will be changed number of VMs:1
League 4 will be changed number of VMs:3
League 5 will be changed number of VMs:2
League 6 will be changed number of VMs:3
League 7 will be changed number of VMs:2
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:3
League 10 will be changed number of VMs:0
League 11 will be changed number of VMs:3
League 12 will be changed number of VMs:4
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:0
League 15 will be changed number of VMs:3
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:2
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:4
Mean fitness for this season:12155.194
LCA Threshold for this season:1484.5933
Best solution found at index:6
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #7 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
Task 0, executing on Vm:0 for 15 cycles
Task 1, executing on Vm:1 for 50 cycles
Task 3, executing on Vm:0 for 15 cycles
Task 4, executing on Vm:1 for 50 cycles
Task 6, executing on Vm:0
Task 7, executing on Vm:0
Task 8, executing on Vm:0 for 4 cycles
Task 9, executing on Vm:1
Task 10, executing on Vm:1
Task 11, executing on Vm:1
Task 12, executing on Vm:1 for 9 cycles
Task 14, executing on Vm:0
Task 15, executing on Vm:0 for 1 cycles
Task 16, executing on Vm:1
Task 17, executing on Vm:1
Task 18, executing on Vm:1
Task 19, executing on Vm:1
Task 20, executing on Vm:1
Task 21, executing on Vm:1
Task 22, executing on Vm:1 for 7 cycles
Task 24, executing on Vm:0
Task 25, executing on Vm:0 for 1 cycles
Task 26, executing on Vm:1
Task 27, executing on Vm:1
Task 28, executing on Vm:1
Task 29, executing on Vm:1
Task 30, executing on Vm:1 for 4 cycles
Task 32, executing on Vm:0
Task 33, executing on Vm:0
```

```
Task 34, executing on Vm:0
Task 35, executing on Vm:0
Task 36, executing on Vm:1
Task 37, executing on Vm:1
Task 38, executing on Vm:1
Task 39, executing on Vm:1 for 4 cycles
Task 41, executing on Vm:0
Task 42, executing on Vm:0 for 13 cycles
Task 43, executing on Vm:1
Task 44, executing on Vm:1
Task 45, executing on Vm:1
Task 46, executing on Vm:1 for 2 cycles
Task 48, executing on Vm:0
Task 49, executing on Vm:0
Task 50, executing on Vm:0 for 11 cycles
Task 51, executing on Vm:1
Task 52, executing on Vm:1
Task 53, executing on Vm:1
Task 54, executing on Vm:1 for 10 cycles
Task 56, executing on Vm:0
Task 57, executing on Vm:0 for 5 cycles
Task 58, executing on Vm:1
Task 59, executing on Vm:1 for 50 cycles
Task 61, executing on Vm:0 for 15 cycles
Task 62, executing on Vm:1
Task 63, executing on Vm:1
Task 64, executing on Vm:1
Task 65, executing on Vm:1 for 29 cycles
Task 67, executing on Vm:0
Task 68, executing on Vm:0 for 9 cycles
Task 69, executing on Vm:1
Task 70, executing on Vm:1
Task 71, executing on Vm:1
Task 72, executing on Vm:1
Task 73, executing on Vm:1 for 7 cycles
Task 75, executing on Vm:0
Task 76, executing on Vm:1
Task 77, executing on Vm:1
Task 78, executing on Vm:1 for 47 cycles
Task 79, executing on Vm:2
Task 80, executing on Vm:0 for 15 cycles
Task 81, executing on Vm:1
Task 82, executing on Vm:1
Task 83, executing on Vm:1
Task 84, executing on Vm:1 for 31 cycles
Task 86, executing on Vm:0 for 15 cycles
Task 87, executing on Vm:1
Task 88, executing on Vm:1
Task 89, executing on Vm:1 for 30 cycles
Task 90, executing on Vm:2
Task 91, executing on Vm:0 for 15 cycles
Task 92, executing on Vm:1 for 50 cycles
Task 94, executing on Vm:0
Task 95, executing on Vm:0 for 15 cycles
Task 96, executing on Vm:1 for 50 cycles
Task 98, executing on Vm:0
Task 99, executing on Vm:0
Task 100, executing on Vm:0 for 8 cycles
Simulation completed.
```

```
All tasks completed
Season 1
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:3
League 2 will be changed number of VMs:8
League 3 will be changed number of VMs:9
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:6
League 6 will be changed number of VMs:5
League 7 will be changed number of VMs:3
League 8 will be changed number of VMs:5
League 9 will be changed number of VMs:7
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:2
League 12 will be changed number of VMs:9
League 13 will be changed number of VMs:1
League 14 will be changed number of VMs:6
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:7
League 17 will be changed number of VMs:2
League 18 will be changed number of VMs:9
League 19 will be changed number of VMs:8
Mean fitness for this season:142977.77
LCA Threshold for this season:114257.516
Season 2
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will be changed number of VMs:6
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:11499.325
LCA Threshold for this season:9189.432
Season 3
League 0 will be changed number of VMs:8
League 1 will be changed number of VMs:0
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will be changed number of VMs:9
League 6 will be changed number of VMs:9
League 7 will NOT be changed
League 8 will be changed number of VMs:9
League 9 will be changed number of VMs:0
League 10 will be changed number of VMs:1
```

```
League 11 will be changed number of VMs:2
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will be changed number of VMs:2
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:139681.75
LCA Threshold for this season:111623.58
Season 4
League 0 will NOT be changed
League 1 will be changed number of VMs:9
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:8143.6616
LCA Threshold for this season:6507.827
Season 5
League 0 will be changed number of VMs:5
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will be changed number of VMs:2
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will be changed number of VMs:5
League 9 will NOT be changed
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:3
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:8
League 14 will be changed number of VMs:0
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will be changed number of VMs:3
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:10678.571
LCA Threshold for this season:8533.544
Season 6
League 0 will be changed number of VMs:7
```

```
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will be changed number of VMs:9
League 9 will NOT be changed
League 10 will be changed number of VMs:9
League 11 will be changed number of VMs:7
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will be changed number of VMs:2
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will be changed number of VMs:8
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:5063.099
LCA Threshold for this season:4046.0637
Season 7
League 0 will be changed number of VMs:2
League 1 will be changed number of VMs:7
League 2 will be changed number of VMs:4
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:7
League 5 will be changed number of VMs:8
League 6 will be changed number of VMs:4
League 7 will be changed number of VMs:4
League 8 will be changed number of VMs:4
League 9 will be changed number of VMs:9
League 10 will be changed number of VMs:6
League 11 will be changed number of VMs:4
League 12 will be changed number of VMs:4
League 13 will be changed number of VMs:7
League 14 will be changed number of VMs:5
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:1
League 17 will be changed number of VMs:8
League 18 will be changed number of VMs:7
League 19 will be changed number of VMs:1
Mean fitness for this season:144319.36
LCA Threshold for this season:115329.625
Season 8
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will be changed number of VMs:5
League 12 will NOT be changed
League 13 will NOT be changed
```

```
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:12803.758
LCA Threshold for this season:10231.84
Season 9
League 0 will be changed number of VMs:2
League 1 will NOT be changed
League 2 will be changed number of VMs:1
League 3 will be changed number of VMs:0
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will be changed number of VMs:7
League 8 will be changed number of VMs:3
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will be changed number of VMs:5
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:403471.8
LCA Threshold for this season:322425.56
Season 10
League 0 will be changed number of VMs:0
League 1 will NOT be changed
League 2 will be changed number of VMs:8
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will be changed number of VMs:5
League 9 will NOT be changed
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:9136.373
LCA Threshold for this season:7301.1304
Best solution found at index:3
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
```

```
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #9 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #8 in Datacenter_0
Task 2, executing on Vm:2 for 51 cycles
Task 6, executing on Vm:2
Task 7, executing on Vm:2
Task 8, executing on Vm:2
Task 9, executing on Vm:2
Task 10, executing on Vm:2
Task 11, executing on Vm:2 for 2 cycles
Task 15, executing on Vm:2
Task 16, executing on Vm:2
Task 17, executing on Vm:2
Task 18, executing on Vm:2
Task 19, executing on Vm:2 for 3 cycles
Task 23, executing on Vm:2
Task 24, executing on Vm:2
Task 25, executing on Vm:2
Task 26, executing on Vm:2
Task 27, executing on Vm:2
Task 28, executing on Vm:2 for 4 cycles
Task 32, executing on Vm:2
Task 33, executing on Vm:2
Task 34, executing on Vm:2
Task 35, executing on Vm:2
Task 36, executing on Vm:2
Task 37, executing on Vm:2
Task 38, executing on Vm:2 for 12 cycles
Task 42, executing on Vm:2
Task 43, executing on Vm:2
Task 44, executing on Vm:2
Task 45, executing on Vm:2 for 5 cycles
Task 49, executing on Vm:2
Task 50, executing on Vm:2
Task 51, executing on Vm:2
Task 52, executing on Vm:2
Task 53, executing on Vm:2 for 14 cycles
Task 57, executing on Vm:2
Task 58, executing on Vm:2
Task 59, executing on Vm:2 for 36 cycles
Task 63, executing on Vm:2
Task 64, executing on Vm:2
Task 65, executing on Vm:2 for 39 cycles
Task 69, executing on Vm:2
Task 70, executing on Vm:2
Task 71, executing on Vm:2
Task 72, executing on Vm:2
Task 73, executing on Vm:2 for 8 cycles
Task 77, executing on Vm:2
Task 78, executing on Vm:2 for 51 cycles
Task 79, executing on Vm:3
Task 82, executing on Vm:2
Task 83, executing on Vm:2
Task 84, executing on Vm:2 for 36 cycles
Task 88, executing on Vm:2
Task 89, executing on Vm:2 for 46 cycles
```

```
Task 90, executing on Vm:3
Task 93, executing on Vm:2
Task 94, executing on Vm:2
Task 95, executing on Vm:2 for 48 cycles
Task 99, executing on Vm:2
Task 100, executing on Vm:2 for 51 cycles
Simulation completed.
All tasks completed
Season 1
League 0 will be changed number of VMs:9
League 1 will be changed number of VMs:9
League 2 will be changed number of VMs:3
League 3 will be changed number of VMs:2
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:5
League 6 will be changed number of VMs:9
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:3
League 10 will be changed number of VMs:4
League 11 will be changed number of VMs:5
League 12 will be changed number of VMs:7
League 13 will be changed number of VMs:6
League 14 will be changed number of VMs:7
League 15 will be changed number of VMs:2
League 16 will be changed number of VMs:2
League 17 will be changed number of VMs:3
League 18 will be changed number of VMs:0
League 19 will be changed number of VMs:8
Mean fitness for this season:137667.48
LCA Threshold for this season:70826.06
Season 2
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will NOT be changed
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will NOT be changed
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will be changed number of VMs:9
League 10 will NOT be changed
League 11 will NOT be changed
League 12 will NOT be changed
League 13 will NOT be changed
League 14 will NOT be changed
League 15 will NOT be changed
League 16 will NOT be changed
League 17 will NOT be changed
League 18 will NOT be changed
League 19 will NOT be changed
Mean fitness for this season:6038.7104
LCA Threshold for this season:3106.747
Season 3
League 0 will NOT be changed
League 1 will NOT be changed
League 2 will be changed number of VMs:2
League 3 will be changed number of VMs:4
```

```
League 4 will be changed number of VMs:1
League 5 will be changed number of VMs:7
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will be changed number of VMs:3
League 9 will NOT be changed
League 10 will be changed number of VMs:1
League 11 will be changed number of VMs:4
League 12 will be changed number of VMs:1
League 13 will be changed number of VMs:9
League 14 will be changed number of VMs:9
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:3
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:1
League 19 will NOT be changed
Mean fitness for this season:4490.2666
LCA Threshold for this season:2310.1162
Season 4
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:9
League 2 will be changed number of VMs:9
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:0
League 5 will be changed number of VMs:9
League 6 will be changed number of VMs:5
League 7 will be changed number of VMs:4
League 8 will be changed number of VMs:7
League 9 will be changed number of VMs:3
League 10 will be changed number of VMs:9
League 11 will be changed number of VMs:3
League 12 will be changed number of VMs:8
League 13 will be changed number of VMs:0
League 14 will be changed number of VMs:2
League 15 will be changed number of VMs:1
League 16 will be changed number of VMs:4
League 17 will be changed number of VMs:8
League 18 will be changed number of VMs:9
League 19 will be changed number of VMs:0
Mean fitness for this season:5292.026
LCA Threshold for this season:2722.5989
Season 5
League 0 will be changed number of VMs:5
League 1 will be changed number of VMs:5
League 2 will be changed number of VMs:4
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:6
League 5 will be changed number of VMs:2
League 6 will be changed number of VMs:4
League 7 will be changed number of VMs:9
League 8 will be changed number of VMs:8
League 9 will be changed number of VMs:4
League 10 will be changed number of VMs:6
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:9
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:3
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:2
```

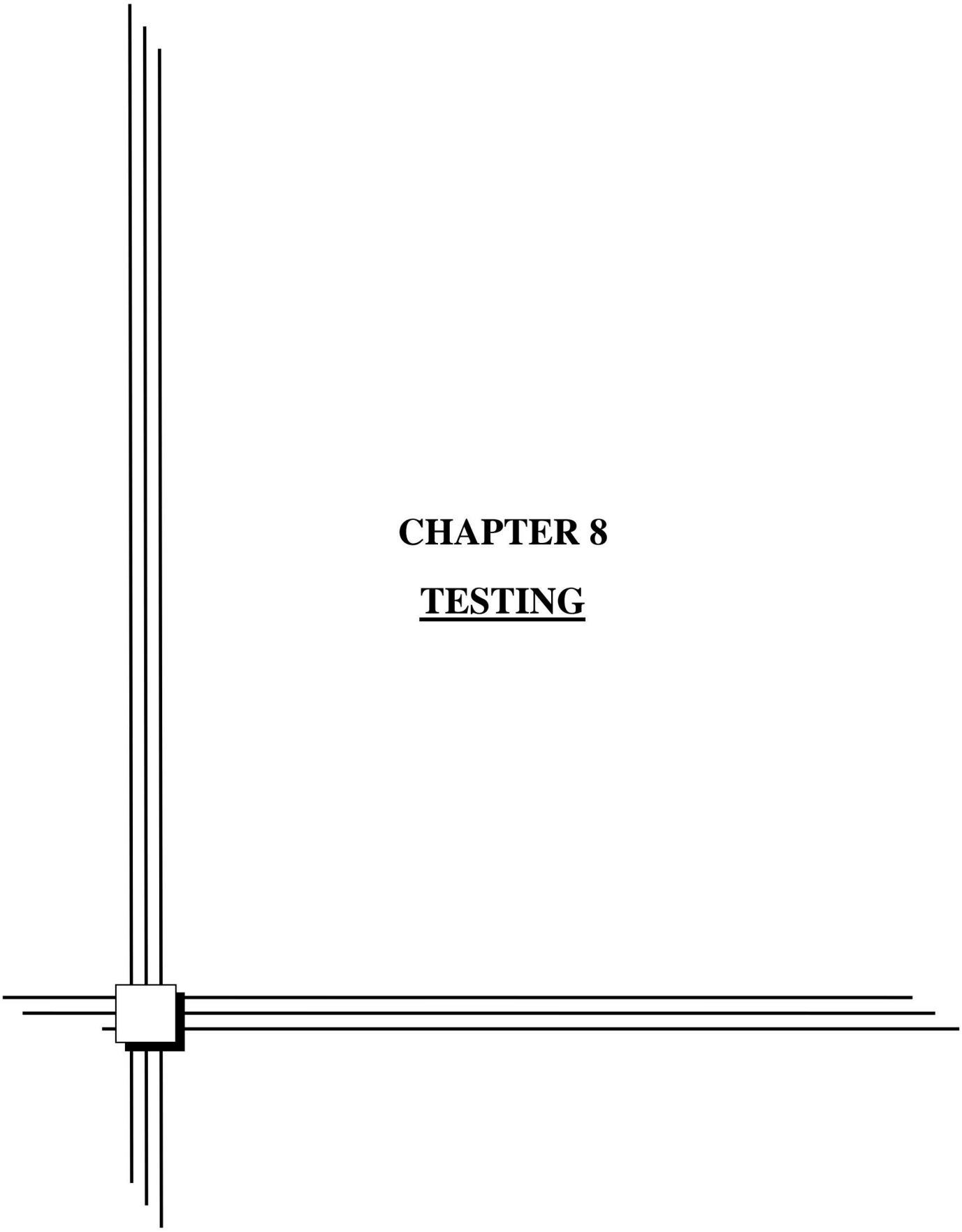
```
League 17 will be changed number of VMs:2
League 18 will be changed number of VMs:3
League 19 will be changed number of VMs:0
Mean fitness for this season:6333.8037
LCA Threshold for this season:3258.5645
Season 6
League 0 will be changed number of VMs:2
League 1 will be changed number of VMs:2
League 2 will be changed number of VMs:4
League 3 will be changed number of VMs:8
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:1
League 6 will be changed number of VMs:0
League 7 will NOT be changed
League 8 will be changed number of VMs:9
League 9 will be changed number of VMs:5
League 10 will be changed number of VMs:5
League 11 will NOT be changed
League 12 will be changed number of VMs:4
League 13 will be changed number of VMs:4
League 14 will be changed number of VMs:3
League 15 will be changed number of VMs:1
League 16 will be changed number of VMs:0
League 17 will be changed number of VMs:5
League 18 will be changed number of VMs:5
League 19 will be changed number of VMs:2
Mean fitness for this season:5923.233
LCA Threshold for this season:3047.3372
Season 7
League 0 will be changed number of VMs:6
League 1 will be changed number of VMs:9
League 2 will be changed number of VMs:0
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:5
League 5 will be changed number of VMs:7
League 6 will NOT be changed
League 7 will be changed number of VMs:2
League 8 will be changed number of VMs:5
League 9 will be changed number of VMs:1
League 10 will be changed number of VMs:8
League 11 will be changed number of VMs:1
League 12 will be changed number of VMs:9
League 13 will be changed number of VMs:3
League 14 will be changed number of VMs:1
League 15 will be changed number of VMs:8
League 16 will NOT be changed
League 17 will be changed number of VMs:1
League 18 will be changed number of VMs:4
League 19 will be changed number of VMs:1
Mean fitness for this season:4867.506
LCA Threshold for this season:2504.1953
Season 8
League 0 will be changed number of VMs:3
League 1 will be changed number of VMs:4
League 2 will be changed number of VMs:4
League 3 will be changed number of VMs:5
League 4 will be changed number of VMs:8
League 5 will be changed number of VMs:4
League 6 will be changed number of VMs:6
```

```
League 7 will be changed number of VMs:7
League 8 will be changed number of VMs:1
League 9 will be changed number of VMs:2
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:3
League 12 will be changed number of VMs:6
League 13 will be changed number of VMs:3
League 14 will be changed number of VMs:4
League 15 will be changed number of VMs:7
League 16 will be changed number of VMs:5
League 17 will be changed number of VMs:6
League 18 will be changed number of VMs:6
League 19 will be changed number of VMs:6
Mean fitness for this season:10672.487
LCA Threshold for this season:5490.6953
Season 9
League 0 will be changed number of VMs:6
League 1 will NOT be changed
League 2 will be changed number of VMs:8
League 3 will NOT be changed
League 4 will NOT be changed
League 5 will be changed number of VMs:7
League 6 will NOT be changed
League 7 will NOT be changed
League 8 will NOT be changed
League 9 will be changed number of VMs:7
League 10 will be changed number of VMs:3
League 11 will be changed number of VMs:4
League 12 will NOT be changed
League 13 will be changed number of VMs:1
League 14 will be changed number of VMs:7
League 15 will NOT be changed
League 16 will be changed number of VMs:8
League 17 will NOT be changed
League 18 will be changed number of VMs:4
League 19 will NOT be changed
Mean fitness for this season:4544.2734
LCA Threshold for this season:2337.9011
Season 10
League 0 will be changed number of VMs:5
League 1 will be changed number of VMs:7
League 2 will be changed number of VMs:7
League 3 will be changed number of VMs:3
League 4 will be changed number of VMs:9
League 5 will be changed number of VMs:0
League 6 will be changed number of VMs:1
League 7 will be changed number of VMs:5
League 8 will be changed number of VMs:6
League 9 will be changed number of VMs:4
League 10 will be changed number of VMs:6
League 11 will be changed number of VMs:9
League 12 will be changed number of VMs:2
League 13 will be changed number of VMs:5
League 14 will be changed number of VMs:9
League 15 will be changed number of VMs:9
League 16 will be changed number of VMs:5
League 17 will be changed number of VMs:0
League 18 will be changed number of VMs:7
League 19 will be changed number of VMs:6
```

```
Mean fitness for this season:5349.413
LCA Threshold for this season:2752.123
Best solution found at index:3
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
0.0: Broker: Trying to Create VM #8 in Datacenter_0
Task 0, executing on Vm:0 for 85 cycles
Task 2, executing on Vm:2 for 179 cycles
Task 3, executing on Vm:0 for 85 cycles
Task 5, executing on Vm:2
Task 6, executing on Vm:2
Task 7, executing on Vm:2
Task 8, executing on Vm:2
Task 9, executing on Vm:2
Task 10, executing on Vm:2
Task 11, executing on Vm:2
Task 12, executing on Vm:2
Task 13, executing on Vm:2
Task 14, executing on Vm:2
Task 15, executing on Vm:2
Task 16, executing on Vm:2
Task 17, executing on Vm:2
Task 18, executing on Vm:2
Task 19, executing on Vm:2
Task 20, executing on Vm:2
Task 21, executing on Vm:2
Task 22, executing on Vm:2
Task 23, executing on Vm:2
Task 24, executing on Vm:0
Task 25, executing on Vm:0
Task 26, executing on Vm:0
Task 27, executing on Vm:0
Task 28, executing on Vm:0
Task 29, executing on Vm:0
Task 30, executing on Vm:0 for 11 cycles
Task 32, executing on Vm:2
Task 33, executing on Vm:2
Task 34, executing on Vm:2
Task 35, executing on Vm:2
Task 36, executing on Vm:2
Task 37, executing on Vm:2
Task 38, executing on Vm:2
Task 39, executing on Vm:2
Task 40, executing on Vm:2
Task 41, executing on Vm:2
Task 42, executing on Vm:2
Task 43, executing on Vm:2
Task 44, executing on Vm:2
Task 45, executing on Vm:2
Task 46, executing on Vm:2
Task 47, executing on Vm:2
Task 48, executing on Vm:2
Task 49, executing on Vm:2
```

```
Task 50, executing on Vm:2 for 13 cycles
Task 51, executing on Vm:0
Task 52, executing on Vm:0
Task 53, executing on Vm:0
Task 54, executing on Vm:0
Task 55, executing on Vm:0
Task 56, executing on Vm:0
Task 57, executing on Vm:0
Task 58, executing on Vm:0
Task 59, executing on Vm:0 for 1 cycles
Task 61, executing on Vm:2
Task 62, executing on Vm:2
Task 63, executing on Vm:2
Task 64, executing on Vm:2
Task 65, executing on Vm:2 for 20 cycles
Task 66, executing on Vm:0
Task 67, executing on Vm:0
Task 68, executing on Vm:0
Task 69, executing on Vm:0
Task 70, executing on Vm:0
Task 71, executing on Vm:0
Task 72, executing on Vm:0
Task 73, executing on Vm:0 for 10 cycles
Task 75, executing on Vm:2
Task 76, executing on Vm:2
Task 77, executing on Vm:2
Task 78, executing on Vm:2 for 161 cycles
Task 79, executing on Vm:0
Task 80, executing on Vm:0 for 85 cycles
Task 82, executing on Vm:2
Task 83, executing on Vm:2
Task 84, executing on Vm:2 for 164 cycles
Task 85, executing on Vm:0
Task 86, executing on Vm:0
Task 87, executing on Vm:0
Task 88, executing on Vm:0
Task 89, executing on Vm:0 for 33 cycles
Task 90, executing on Vm:1
Task 91, executing on Vm:2 for 179 cycles
Task 92, executing on Vm:0 for 85 cycles
Task 94, executing on Vm:2
Task 95, executing on Vm:2 for 179 cycles
Task 96, executing on Vm:0 for 85 cycles
Task 98, executing on Vm:2
Task 99, executing on Vm:2
Task 100, executing on Vm:2 for 172 cycles
Simulation completed.
All tasks completed
Most optimum learning rate with delay of 578900 us is 0.8676358
Delay without GWO 2338756 us
```

Figure 7.1 Console base output for a cloud secure storage mechanism based on fault and load balancing.



CHAPTER 8

TESTING

8.1 Testing Techniques

When testing a waste or garbage classification system using deep learning, several techniques can be employed to evaluate its performance and ensure its effectiveness. Here are some common testing techniques for such systems:

Test Dataset: Prepare a separate test dataset that is distinct from the training and validation data. The test dataset should include a representative sample of waste or garbage images from each class. This dataset will be used for final evaluation and to estimate the system's performance on unseen data.

Accuracy Metrics: Calculate accuracy, which is the most basic evaluation metric, by comparing the predicted labels with the ground truth labels in the test dataset. Accuracy measures the percentage of correctly classified instances. However, accuracy alone may not provide a complete picture, especially if the classes are imbalanced.

Confusion Matrix: Construct a confusion matrix to analyse the classification results in more detail. A confusion matrix displays the number of true positives, true negatives, false positives, and false negatives for each class. It helps identify which classes are often confused with each other and provides insights into the system's performance for different classes.

Precision, Recall, and F1 Score: Calculate precision, recall, and F1 score for each class. Precision measures the ratio of correctly classified positive instances to the total instances predicted as positive. Recall, also known as sensitivity, calculates the ratio of correctly classified positive instances to the total actual positive instances. The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics.

ROC Curve and AUC: If the waste classification system supports probabilistic outputs, generate a Receiver Operating Characteristic (ROC) curve and calculate the Area Under the Curve (AUC) score. The ROC curve illustrates the trade-off between true positive rate (sensitivity) and false positive rate (1 - specificity) for different classification thresholds. AUC represents the overall performance of the classifier.

Cross-Validation: Perform k-fold cross-validation to assess the system's stability and robustness. Divide the entire dataset into k subsets or folds. Train and evaluate the system k times, each time using a different fold for testing and the remaining folds for training. Average the performance metrics across all iterations to get a more reliable estimate of the system's performance.

Error Analysis: Analyse and understand the misclassified instances or cases where the system performs poorly. Examine misclassified images to identify common patterns or sources of confusion. This analysis can help refine the model, improve feature extraction, or collect additional training data for challenging cases.

Performance on Unseen Data: Lastly, evaluate the system's performance on real-world, unseen data to validate its practical utility. Collect a small set of new waste or garbage images and observe the system's predictions. This step helps assess the system's ability to generalize to novel instances.

8.2 Testing Strategies

When testing a waste or garbage classification system using deep learning, it is important to employ effective testing strategies to ensure the system's accuracy and reliability. Here are some testing strategies to consider:

Unit Testing: Perform unit testing on individual components of the system, such as data pre-processing, model architecture, and custom classification layers. Unit testing helps identify any bugs or errors in these components before integrating them into the complete system. Ensure that each component functions as intended and produces the expected results.

Integration Testing: Once the individual components have been tested, conduct integration testing to evaluate the interaction and compatibility between different modules of the system. Test the flow of data from pre-processing to model prediction, ensuring that the inputs and outputs are handled correctly at each stage. Integration testing helps uncover any issues that may arise during the integration process.

Robustness Testing: Assess the system's robustness by testing its performance under various challenging conditions. Introduce noise, occlusion, or other perturbations to the waste or garbage images and evaluate the system's ability to handle such

variations. Robustness testing helps determine if the system can handle real-world scenarios where the input data may not be ideal.

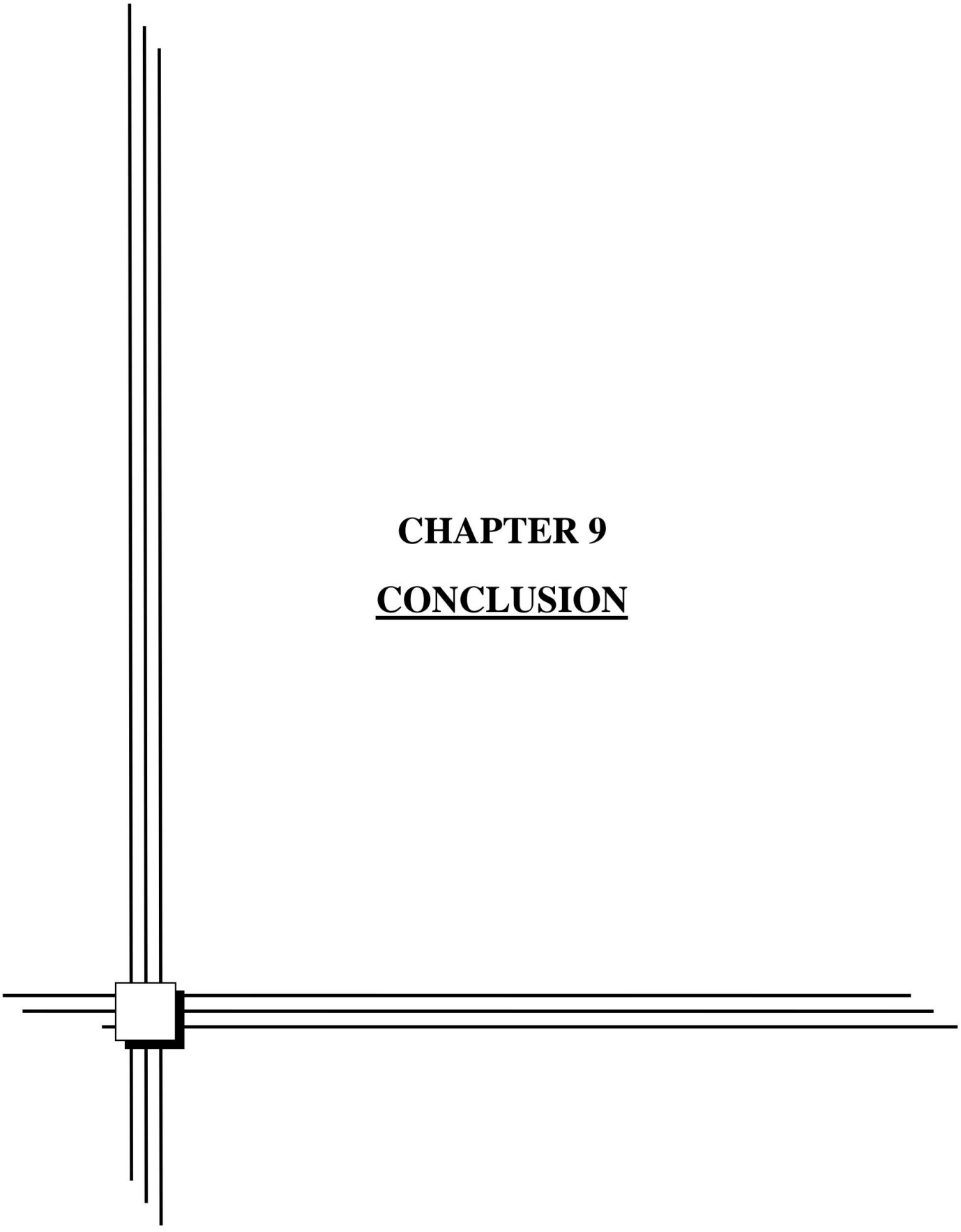
Performance Testing: Evaluate the performance of the system in terms of inference speed and resource utilization. Measure the time it takes for the system to process a single image or a batch of images. Monitor the system's resource usage, such as CPU and memory, during testing. Performance testing ensures that the system meets the required performance criteria and can handle the expected workload.

Cross-Dataset Testing: Test the system's performance on multiple datasets to assess its generalization capability. Collect additional waste or garbage datasets from different sources or environments and evaluate how well the system performs on these new datasets. Cross-dataset testing helps verify that the system can classify waste accurately across different scenarios and reduces the risk of overfitting to a specific dataset.

Bias and Fairness Testing: Analyse the system for biases or fairness issues in waste classification. Check if the system's performance varies significantly across different demographic groups or types of waste. Assess whether the system is biased towards certain classes or exhibits unfair behaviour. Bias and fairness testing are essential to ensure ethical and equitable performance of the system.

User Acceptance Testing: Involve end users or domain experts in the testing process to gather feedback on the system's usability and effectiveness. Conduct user acceptance testing to assess how well the system meets the users' expectations and requirements. Gather insights on the system's usability, user interface, and overall user experience.

Continuous Testing and Monitoring: Establish a continuous testing and monitoring framework to ensure that the system remains accurate and reliable over time. Continuously test the system with new data and periodically revaluate its performance. Monitor the system's performance in production to detect any degradation or drift in performance and take necessary corrective actions.



CHAPTER 9

CONCLUSION

9.1 Conclusion

In conclusion, a cloud secure storage mechanism based on fault and load balancing is a critical component for ensuring data availability, fault tolerance, and security in a cloud environment. By employing modules such as the Data Replication Module, Load Balancer Module, Fault Detection and Recovery Module, Data Security Module, User Interface Module, and Communication Module, the system can achieve the following:

1. Data Availability: The replication module ensures that data is replicated across multiple storage nodes, enabling continuous access even in the event of node failures or disruptions.
2. Fault Tolerance: The fault detection and recovery module monitors the health and status of storage nodes, detecting faults and initiating recovery processes such as data reconstruction or node replacement.
3. Load Balancing: The load balancer module evenly distributes the workload and data across storage nodes, optimizing resource utilization and enhancing system performance.
4. Data Security: The data security module implements encryption, access control, and authentication mechanisms to protect data confidentiality and integrity.
5. User-Friendly Interface: The user interface module provides an intuitive interface for system administrators and end-users to interact with the cloud secure storage system, enabling easy data management and monitoring.
6. Efficient Communication: The communication module handles communication between system components and external entities, ensuring reliable and secure data transfer.

By adhering to design principles such as separation of concerns and incorporating non-functional requirements like scalability, performance, and reliability, the cloud secure storage mechanism can meet the demands of a dynamic and secure cloud storage environment.

In summary, the proposed cloud secure storage mechanism based on fault and load balancing aims to provide a robust, fault-tolerant, and secure storage solution for cloud environments, ensuring data availability, integrity, and optimal performance.

CHAPTER 10

Future Enhancement

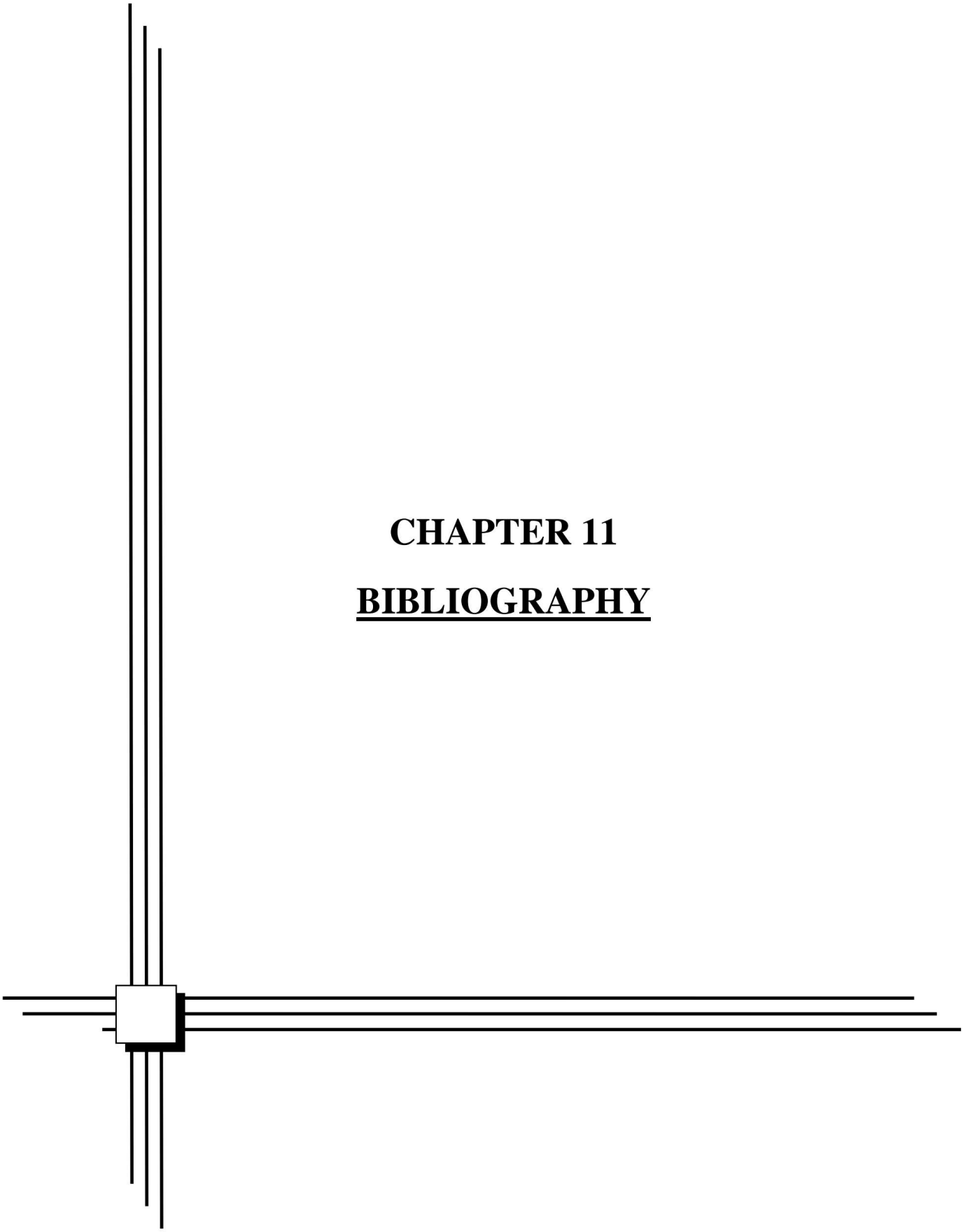
Future enhancements for a cloud secure storage mechanism based on fault and load balancing can focus on improving various aspects of the system. Here are a few potential areas for enhancement:

1. Advanced Fault Detection and Recovery: Implement more sophisticated fault detection algorithms and recovery mechanisms to enhance the system's ability to detect and recover from various types of faults. This could include predictive analytics techniques, machine learning algorithms, or anomaly detection methods to proactively identify potential failures and mitigate them before they cause data loss or service disruption.
2. Dynamic Load Balancing: Enhance the load balancing module to support dynamic adjustment of load balancing strategies based on real-time system conditions. This could involve the use of machine learning or adaptive algorithms that can adapt to changing workloads and resource availability to optimize performance and resource utilization.
3. Enhanced Data Security: Integrate advanced security measures such as advanced encryption algorithms, multi-factor authentication, or blockchain technology to further strengthen data security and privacy in the cloud storage system. This can help protect against emerging security threats and ensure compliance with evolving data protection regulations.
4. Scalability and Elasticity: Design the system to be highly scalable and elastic, allowing for easy expansion or contraction of resources based on changing demands. This can involve implementing auto-scaling mechanisms that dynamically adjust the number of storage nodes or resources allocated based on workload patterns or user demands.
5. Integration with Cloud-native Technologies: Explore integration with cloud-native technologies such as containers, microservices, or serverless computing to leverage the benefits of scalability, agility, and cost-effectiveness provided by these platforms. This can enable the system to take full advantage of cloud infrastructure capabilities and streamline deployment and management processes.
6. Performance Optimization: Continuously analyze and optimize the system's performance to ensure efficient data transfer, reduced latency, and improved response times. This may involve performance tuning, caching mechanisms,

or leveraging edge computing to minimize network latency for data access and retrieval.

7. Enhanced Monitoring and Reporting: Improve monitoring capabilities to provide real-time insights into system performance, health, and security. Implement comprehensive logging, monitoring, and reporting mechanisms to facilitate proactive system management, troubleshooting, and compliance auditing.
8. Integration with Data Analytics: Integrate data analytics capabilities to gain valuable insights from the stored data. This can involve leveraging big data technologies, machine learning algorithms, or data mining techniques to analyze patterns, detect anomalies, or extract meaningful information that can be utilized for decision-making or business intelligence purposes.

These future enhancements can further enhance the functionality, performance, security, and scalability of the cloud secure storage mechanism, making it more resilient and capable of meeting the evolving needs and challenges of cloud storage environments.



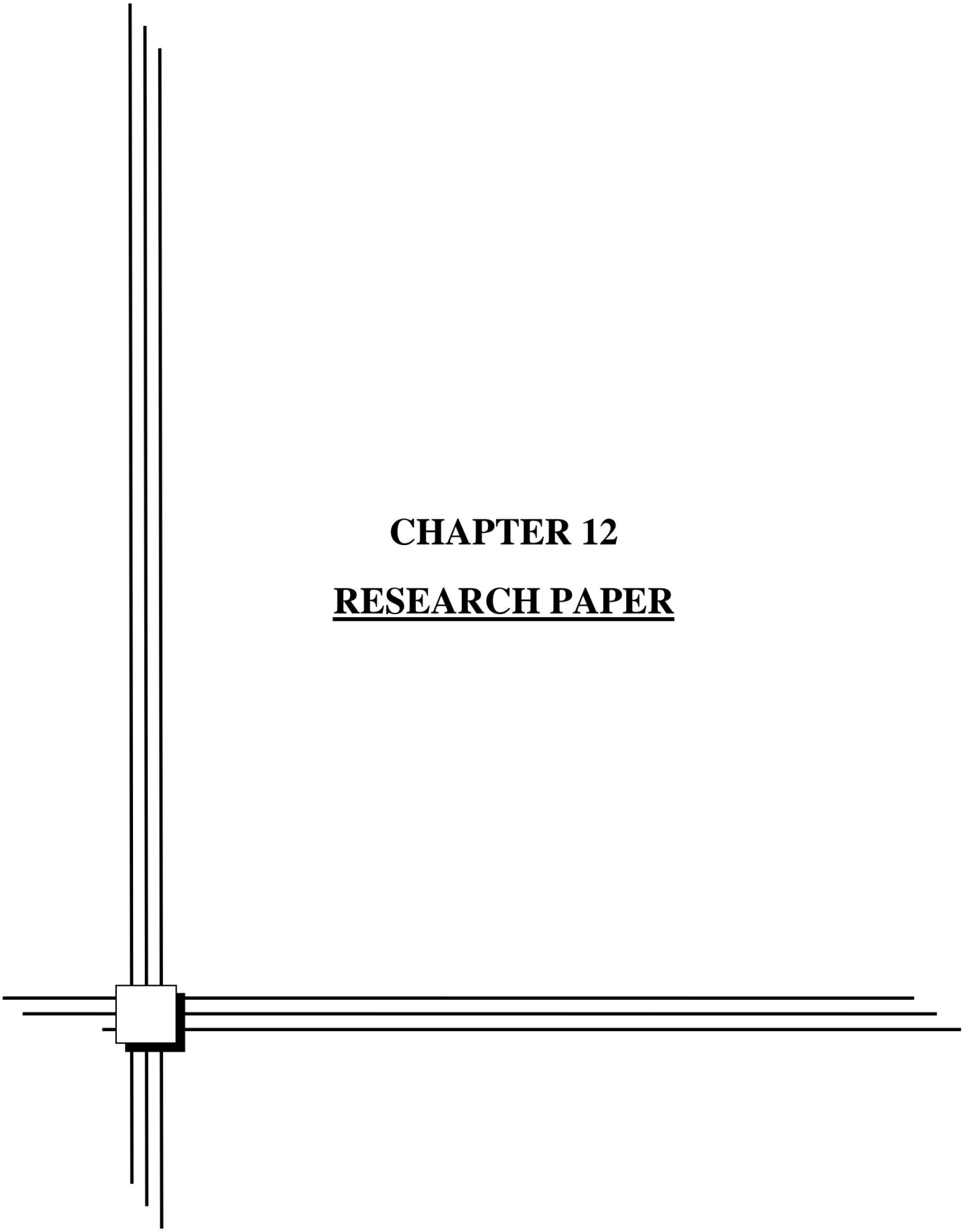
CHAPTER 11

BIBLIOGRAPHY

REFERENCES

1. Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari, “What is an object?”,
2. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010.
3. Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. “Measuring the objectness of image windows”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11), 2189-2202, 2012.
4. Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala, “Material recognition in the wild with the materials in context database.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,3479-3487, 2015“.
5. Design of a Convolutional Neural Network Based Smart Waste Disposal System.
6. Thung, Gary and M. Yang. “Classification of Trash for Recyclability Status.” (2016).
7. Gary Thung and Mindy Yang, “Classification of Trash for Recyclability Status,” CS 229, Stanford University, 2016.Available:
8. C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, “Exploring features in a bayesian framework for material recognition,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 239–246.
9. George E Sakr, Maria Mokbel, Ahmad Darwich, Mia Nasr Khneisser and Ali Hadi, “Comparing Deep Learning And Support Vector Machines for Autonomous Multidisciplinary Conference.
10. S. Kaza, L. Yao, P. Bhada-Tata, and F. Van Woerden, What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. The World Bank, 2018.
11. “Convolutional networks and applications in vision,” in Proceedings of 2010 IEEE International Symposium on Circuits and Systems, Paris, France, pp. 253–256, May 2010. doi: 10.1109/ISCAS.2010.5537907.

12. N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, vol. 1, pp. 886–893, 2005. doi: 10.1109/ CVPR.2005. 177.
13. K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image.
14. Recognition,” in arXiv:1409.1556 [cs], San Diego, CA, USA, pp. 1–14, May 2015, [Online]. Available: http://arxiv.org/abs/1409.1556 [Accessed: Sep. 27, 2019].
15. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in 2016 IEEE
16. Conference on Computer Vision and Pattern
17. Recognition (CVPR), Las Vegas, NV, USA, pp. 770–778, Jun. 2016. doi: 10.1109/CVPR.2016.90.
18. C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 1– 27, Apr. 2011. doi: 10.1145/1961189.1961199.
19. M. Yang and G. Thung, “Classification of Trash for Recyclability Status,” Stanford University, CS229, 2016.
20. C. Bircanoglu, M. Atay, F. Beser, O. Genc, and M. A. Kizrak, “RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks,” in 2018 Innovations in Intelligent Systems and Applications (INISTA), Thessaloniki, pp.1–7, Jul. 2018. doi: 10.1109/INISTA.2018.8466276.
21. H. Khan, “Transfer learning using mobilenet,” 2019. [Accessed Sep. 23, 2019].



CHAPTER 12

RESEARCH PAPER

Review On Challenges and Issues in Data Mining

Omkar K. Khadke[1], Lowlesh N. Yadav[2], Neehal B. Jiwane[3]

B-Tech Final Year Student, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India[1]

Assistant Professor, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India[2]

Assistant Professor, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India[3]

Abstract: Data Mining data and data from high databases has been recognized by several researchers as a key analysis topic in machine learning and info system and in several industrial corporations as a crucial space with a chance of major revenues .we gift a two-dimensional read of information mining. the foremost dimensions area unit data, knowledge, technologies and application. Researchers in some totally different fields have shown their nice interest in data mining. during this section, we tend to in brief define methodology, user interaction. data processing analysis has powerfully impact society and can still do thus within the future.

Keywords: Data Mining, Machine Learning, Data Mining Issues.

1. INTRODUCTION

Data mining is that the method of motion numerous queries and extracting helpful info, patterns, and trends typically antecedently unknown from massive quantities of knowledge probably keep in databases. basically, for several organizations, the goals of knowledge mining embrace up selling capabilities, detection abnormal patterns, and predicting the long run supported past experiences and current trends. there's clearly a requirement for this technology. There square measure massive amounts of current and historical information being keep. Therefore, as databases become larger, it becomes progressively tough to support higher cognitive process. additionally, the information might be from multiple sources and multiple domains. there's clear have to be compelled to associatealyze the information to support designing and alternative functions of an enterprise. numerous terms are wont to talk to data processing. These embrace knowledge/data/information discovery and knowledge/data/information extraction. so data processing is that the natural evolution of question and news tools. Everyone, World Health Organization creates queries and reports, advantages from having data processing capabilities

2. CHARACTERISTIC OF DATA MINING

Characteristics of information Mining:
Data mining service is a simple sort of military operation methodology whereby that all the relevant data goes through some style of identification method. And eventually at the tip of this method, one will confirm all the characteristics of the information mining method.

2.1. Enhanced quantities of data: In earlier days, the information mining system will be determined with the assistance of their shoppers and customers, however in today's date, one will acquire any range of data while not the assistance of these shoppers. Moreover, once this sort of revolution within the mining system, it conjointly extra a new downside which is massive quantities of labor. With the assistance of this data technology, one will acquire an oversized range of data with none additional burden or bother.

2.2. Provides incomplete data: Most of the folks give incomplete data concerning themselves in a number of the survey conducted with the assistance of information mining systems. Therefore, folks ignore the worth of their data which is why they supply incomplete data concerning themselves in those surveys conducted for the good thing about the mining systems. Moreover, these mining systems modified the attitude of individuals and since of that, folks worry the exchange of their personal data.

2.3. difficult information structure: Data mining could be a kind whereby that all the knowledge is gathered and incorporated with the assistance of data assortment techniques. These data collection techniques area unit additional of manual and rest area unit technological. Therefore, most of the understanding and determination of those mining will be a touch difficult than different structures of data technology.

3. DATA MINING APPLICATIONS

Data mining is generally employed by several of the large gaints within the info technology sector and conjointly some little industries by creating use of their own techniques. a number of the favored domains ar, Market Analysis and Management Corporate Analysis & Risk Management Fraud Detection

3.1. market research and Management: The following mentioned ar the varied fields of the market wherever the information mining method is effectively used,

- ★ Customer identification
- ★ Finding client necessities
- ★ Cross-market analysis
- ★ Target selling
- ★ Determining client buying pattern
- ★ Provides outline info

3.2. company Analysis and Risk Management: The following mentioned ar the varied fields of the company sector wherever the information mining method is effectively used,

- ★ Finance designing
- ★ Asset analysis
- ★ Resource designing
- ★ Competition

3.3. Fraud Detection: Frauds and malware is one in every of the foremost dangerous threats on the net. it's virtually a sort of crime that's increasing day once day. The fraud detection method is chiefly used through MasterCard services and telecommunication. With the assistance of the services most of the necessary info like length of the decision, location, the time and day etc is non inheritable that helps in success.

4. DATA MINING CHALLENGES

These days data processing and knowledge revealing area unit developing crucial innovations for researchers and businesses in varied areas. data processing was forming into a setup and confided up to the mark, heretofore forthcoming data processing challenges should be tackled.

4.1. Security and Social Challenges

Dynamic techniques area unit done through knowledge assortment sharing, which needs spectacular security. non-public data concerning individuals and touchy data is gathered for the client's profiles, shopper normal of conduct understanding—illicit admittance to data and therefore the secret plan of knowledge turning into a big issue.

4.2. Clanging and Incomplete knowledge

Data Mining could be a thanks to get data from Brobdingnagian volumes of information. This gift reality of knowledge is clanging, incomplete, and heterogeneous. knowledge in Brobdingnagian amounts often are unreliable or inaccurate. These problems may be owing to human mistakes, blunders, or errors within the instruments that live the info.

4.3. Distributed knowledge

True knowledge is generally place away at varied stages in distributed process conditions. it's going to get on the web, individual systems, or maybe databases. it's basically arduous to hold all {the knowledge|the info|the information} to a unified data archive chiefly owing to technical and structure reasons.

4.4. Complicated knowledge

True knowledge is heterogeneous, and it's going to be media knowledge, as well as tongue text, statistic, spatial knowledge, temporal knowledge, complicated knowledge, audio or video, images, etc. it's actually arduous to alter these varied forms of knowledge and think about the required data. a lot of typically than not, new apparatuses and systems would want to be created to separate vital data.

4.5. Performance

The presentation of the info mining framework primarily depends upon the productivity of techniques and algorithms used. On the off likelihood that the techniques and algorithms planned aren't sufficient , at that time, it'll influence the presentation of the info mining live unfavorably.

4.6. Quantifiability and potency of the Algorithms

The Data Mining rule ought to be climbable and economical to disengage data from tremendous measures {of knowledge|of knowledge|of information} within the data set.

4.7. Improvement of Mining Algorithms

Factors, for instance, the issue of information mining approaches, the large size of the information, and therefore the entire knowledge flow, inspire the distribution and creation of parallel data processing algorithms.

4.8. Incorporation of information

In the event that information will be consolidated, a lot of correct and reliable data processing arrangements will be found. prognosticative tasks will create a lot of correct predictions, whereas descriptive tasks will come back up with a lot of helpful findings. Be that because it could, gathering and as well as foundation information is unpredictable.

4.9. Knowledge visualization

Data visualization could be a very important cycle in data processing since it's the foremost interaction that shows the output in an exceedingly respectable thanks to the shopper. the knowledge freed have to be compelled to pass away the importance of what it plans to pass away. However, ordinarily, it's actually arduous to handle the knowledge exactly and squarely to the top user. The output data and computer file being terribly effective, successful, and sophisticated knowledge perception strategies ought to be applied to create it fruitful.

4.10. Knowledge Privacy and Security

Data mining generally prompts important governance, privacy, and knowledge security problems. as an example, once a distributor investigates the acquisition details, it uncovers data concerning buying propensities and decisions of shoppers while not their authorization.

4.11. Interface

The information is set utilizing data processing devices is effective simply within the event that it's fascinating or a lot of all affordable by the shopper. From nice illustration translation of information, mining results will be expedited, and betters comprehend their conditions. several explorations area unit in deep trouble monumental knowledge sets that manipulate and show strip-mined information to urge an excellent perception.

4.12. Mining keen about Level of Abstraction

Data Mining measures ought to be community-oriented in light-weight of the very fact that it permits shoppers to specialize in example optimizing, presenting, and pattern finding for data processing keen about conveyance results back.

4.13. Integration of information

Previous data can be accustomed communicate examples to precise discovered patterns and direct the exploration method.

4.14. Mining Methodology Challenges

These difficulties area unit known with data processing strategies and their limits. Mining strategies that cause the difficulty area unit the management and handling of noise in knowledge, the spatiality of the domain, the range of information accessible, the flexibility of the mining methodology, and so on.

5. DATA MINING ISSUES

5.1. Mining Strategies & User Interaction Problems

The issues may be with finding out the best-fit methodologies of knowledge mining, that area unit association, classification, clump analysis, prediction, sequent patterns or pattern pursuit, call trees, outlier analysis or anomaly detection, and neural network.The professionals with active expertise during this domain usually struggle with these problems whereas victimization the same data processing strategies.

5.1.1.Hard to Derive information for numerous Domains

The beneficiary may be associated with totally different industries and domains. thus do dissent their demand and information discovery, that need niche-based knowledge extraction, transformation, & loading of numerous forms like visuals, text, or numbers. This method covers a broad vary of information discovery processes, that may be a challenge.

5.1.2.Lack of Interactive Models

Interactive modelling makes pattern searches easier. For this purpose, datasets area unit extracted, refined, converted, and cleaned to make sure that they manufacture the intelligence that's needed through the mining request.

5.1.3.Lacking skilled information

For every level, from net knowledge extraction to process, & modelling, you would like knowledgeable matter specialists or knowledge miners. Their active expertise guides them through the complete discovery method, that is to work out patterns. Their qualification and operating expertise guarantee golf stroke patterns during a crisp and comprehensive format. Finding such professionals with answer ability is so Associate in Nursing uphill battle.

5.1.4.Ad hoc data processing isn't straightforward

Ad-hoc mining answers to a selected business demand or question, that insights will create it easier to abstract. The Structured search language is that the search language that supports the accidental information discovery method. This language guides users through mining tasks by mining or knowledge analysis corporations, that work on the optimized & versatile structure of records. This task is once more not very easy.

5.1.5.Comprehensiveness may be a Challenge

The visual presentation ought to be impactful to let the analyst simply see and extract insights in no time. Here again, the team ought to be competent to run totally different tools like Tableau, Sisence, Excel, etc. for a good visual presentation victimization charts, graphs, etc. Simply put, the presentation of models ought to be perceivable, that isn't sort of a walkover.

5.1.6.Noisy knowledge is tough to Handle

Noisy knowledge talk over with redundancies or useless knowledge within the info or massive knowledge. addressing the noises like duplicity, incomplete info, and errors may be an enormous challenge. But, going ahead with them will disturb the effectiveness of consequence. Result? The pattern are poor and useless.

5.1.7.Measuring Patterns

The patterns filtered, tested, & evaluated ought to be attention-grabbing as a result of they represent either intelligence or possible solutions or lack novelty.

5.2. Performance problems

It is necessary that the modelling ought to be versatile and qualified through quality tests. These models area unit seemingly to be AI via machine learning. Here, the performance problems should be acknowledged to sail across them with previous solutions.

5.2.1.Inefficiency and issue in measurability

Certainly, a info is needed, principally colossal-sized niche-based records. net knowledge extraction & capturing create it approach easier. although it's paper-bound, the OCR conversion and cleansing practices facilitate in making ready a info. But, there area unit challenges like king protea lure, captcha, and privacy settings which will hamper the provision of significant details. Sometimes, lacking tools may prove an enormous barrier to shaping the potency and measurability of knowledge mining.

5.2.2.Inability to figure with Parallel, Distributed, and progressive Algorithms

The metrics like prodigious size of databases, wide distribution, and complexness of knowledge mining strategies push to derive parallel and distributed data processing algorithms. These algorithms split records into elements, that area unit any processed within the same manner. Finally, the results of all elements area unit compiled along. this is often however the progressive algorithms still update databases while not mining the info once more from scratch.

5.3. Numerous knowledge varieties problems

Data has several faces. you'll notice it in its visual, audio, text, and numeric forms. process these differing types and so, mining could also be troublesome.

5.3.1.Dealing with relative and complicated sorts of knowledge

Your supply knowledge might have PDF, transmission objects, abstraction knowledge, temporal or alternative sorts of datasets. this is often so a bone-breaking expertise to form such a customary tool which will ideally method every kind of knowledge within the same approach.

So, you would like to customize or access a specifically designed tool to mine from a specific kind of datasets, that is a rich deal. And also, you must have the power to manage that tool.

5.3.2.Modelling from Heterogeneous Databases

As there area unit variety of sources to access knowledge like local area network or WAN, you'll not expect to own records in Associate in Nursing ideally similar type and format. it's just because of their storage during a structured, semistructured, or unstructured type. Therefore, mining information from them isn't straightforward.

5.4. knowledge Security & Privacy

Personally, place able knowledge is sensitive and other people don't wish to share it with anyone. Here, a threat to its privacy & security may be a reason to significantly suppose.

5.4.1.Security

Mostly, knowledge area unit shared over the web, the cloud, and servers to make sure their access 24X7 remotely. This access may be dangerous if it's done through a public network, that isn't secure. Vulnerability poses an enormous risk. So, the inter changeableness of any record ought to be outlined through encoding.

5.4.2.Privacy considerations

Dynamic techniques area unit adopted to gather info from numerous resources, particularly from knowledge subjects. This assortment isn't unhazardous, as they carry in person place-able info. Hackers tend to interrupt in and exclude these credentials. Here, privacy controls, authorization, and knowledge compliance like GDPR seem during a major safeguarding role.

6. ADVANTAGES OF DATA MINING TECHNIQUES

There square measure many styles of advantages and blessings of information mining systems. one among the essential matters of those mining creates an entire structure of research of mining techniques.

6. 1.It's useful to predict future trends: Most of the operating nature of the info mining systems carries on all the informational factors of the weather and their structure.One of the common advantages that may be derived with these data processing systems is that they will be useful whereas predicting future trends. which is sort of potential with the assistance of technology and behavioral changes adopted by the individuals.

6.2. It signifies client habits: For example, whereas operating within the selling trade one will perceive all the matters of client behaviour and their habits. which is feasible with the assistance of information mining systems. As these data processing systems handle all the knowledge deed techniques. it's useful keep track of client habits and their behavior.

6.3. Helps in call making: There square measure some folks that create use of those data processing techniques to assist them with some reasonably deciding. Nowadays, all the knowledge concerning something may be determined simply with the assistance of technology and equally, with the assistance of such technology one will create an explicit call concerning one thing unknown and surprising.

6.4. Increase company revenue: As it has been explained earlier that methoding} could be a process whereby that it involves some form of technology to accumulate some info concerning something potential. And this kind of technology makes things easier for his or her profit earning magnitude relation. As individuals will collect info concerning the marketed merchandise on-line, that eventually reduces the price of the merchandise and their services.

6.5. It depends upon market-based analysis: Data mining method could be a system whereby that all {the info|the knowledge|the data} has been gathered on the idea of market information. Nowadays, technology plays a vital role in everything which casualty may be seen in these data processing systems. Therefore, all the knowledge collected through these data processing is largely from selling analysis.

6.6. Fast fraud detection: Most components of {the data|the info|the info} mining method is largely from information gathered with the assistance of promoting analysis. With the assistance of such selling analysis, one also can establish those dishonest acts and merchandise obtainable within the market. Moreover, with the assistance of it one will perceive the importance of correct info.

7. DISADVANTAGES OF DATA MINING TECHNIQUES

Data mining technology are some things that helps one person in their {decision making|deciding|higher cognitive method} which deciding could be a process whereby that all the factors of mining is concerned exactly. And whereas the involvement of those mining systems, one will stumble upon many disadvantages of information mining and that they square measure as follows.

7.1. It violates user privacy: It is a far-famed incontrovertible fact that data processing collects info concerning individuals mistreatment some market-based techniques and data technology. And these methoding} process involves many numbers of things. But whereas involving those factors, data processing system violates the privacy of its user which is why it lacks within the matters of safety and security of its users. Eventually, it creates miscommunication between individuals.

7.2. Further extraneous information: The main functions of {the data|the info|the info} mining systems produce a relevant house for useful information. But the most drawback with these info collections is that there's a prospect that the gathering of data processes may be a touch overwhelming for all. Therefore, it's noticeably essential to take care of a minimum level of limit for all the info mining techniques.

7.3. Misuse of information: As it has been explained earlier that within the data processing system the chance of safety and security live square measure extremely minimal . which is why some will misuse this info to hurt others in their own manner. Therefore, the info mining system has to amendment its course of operating in order that it will scale back the magnitude relation of misuse of data through the mining method.

7.4. Accuracy of data: Most of the time whereas collection info concerning sure components one wont to ask for facilitate from their shoppers, however today everything has modified. And currently the method of data assortment created things simple with the mining technology and their ways. One of the foremost potential limitations of this data processing system is that it will offer accuracy of information with its own limits.

8.CONCLUSION

Data mining seeks to extract hidden information from large amount of information. methoding} is that the process of extracting and valuable attention-grabbing patterns from raw collection of information. data processing may be accustomed uncover patterns within the information however it's typically dispensed solely on the samples of information. This mining method are ineffective if the samples don't seem to be a decent illustration of the larger body of the info.And beside this, today‘s competition is one among the most necessary challenges facing by all organizations and industries in data processing problems. that's onerous to seek out in a particular organization or business that has no rival to him.This paper describes numerous tasks; goals and limitations of information mining. Additionally-this paper additionally discussed regarding the varied valuable problems; future challenges and problems in field of information mining that I important to try to to more more practical analysis during this emerging field.

REFERENCES

[1] M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.

[2] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[3] Chieh-Yuan, T. and T. Min-Hong. A dynamic Web service based data mining process system. in Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on. 2005.

[4] Ding, P. A formal framework for Data Mining process model. In Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on. 2009.

[5] Cooley, B. Mobasher and J. Srivsatava. Web Mining: Information and Pattern Discoveryon the Word Wide Web. In Proceedings of the 9th IEEE International Conference on Tools with AI (ICTAI,97), Nov. 1997.

[6] Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93.

[7] J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98.

[8] Huhtala, J. Kärkkäinen, P. Porkka, H. Toivonen. Efficient Discovery of Functional and Approximate Dependencies Using Partitions. ICDE'98.

[9] W. Shavlik and T. G. Dietterich. Readings in Machine Learning.Morgan Kaufmann, 1990.

[10] P.Tan, M. Steinbach, and V. Kumar. Introduction to Data Mining.Addison Wesley, 2005.

[11] S. M. Weiss and C. A. Kulikowski. Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.Morgan Kaufman, 1991.

[12] Xinjian Guo, Yilong Yin1, Cailing Dong, Gongping Yang, Guangtong Zhou,"On the Class Imbalance Problem" Fourth International Conference on Natural Computation, 2008.

[13]

[14] David P. Williams, Member, Vincent Myers, and Miranda Schatten Silvious, "Mine Classification With Imbalanced Data", IEEE Geosciences And Remote Sensing Letters, Vol. 6, No.3, July 2009.

[15] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, Amri Napolitano "A Comparative Study of Data Sampling and Cost Sensitive Learning" , IEEE International Conference on Data Mining Workshops.15-19Dec.2008.

[16]

[17] Clifton, Christopher "Encyclopedia Britannica: Definition of Data Mining", Retrieved 2010-12-09.

[18] Ian H. Witten; Eibe Frank; Mark A. Hall, ―Data Mining: Practical Machine Learning Tools and Techniques (3rd Ed.)‖, Elsevier, 30 January 2011.

[19] Kantardzic, Mehmed ―Data Mining: Concepts, Models, Methods, and Algorithms‖, John Wiley & Sons, 2003.

[20] Günnemann, S.; Kremer, H.; Seidl, T. "An extension of the PMML standard to subspace clustering models". Proceedings of the 2011 workshop on Predictive markup language modeling - PMML '11.

[21] Ralf Mikut; Markus Reischl, "Data Mining Tools". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Retrieved October 21, 2011.

[22] Domenico Talia; Paolo Trunfio"How distributed data mining tasks can thrive as knowledge services". Communications of the ACM , 2010 .

[23] Karl Rexer, Heather Allen, & Paul Gearan ―Data Miner Survey – Overcoming Data Mining's Top Challenges‖, 2011.

[24] Hans-Peter Kriegel , Karsten M. Borgwardt et.al ―Future trends in data mining‖, sprinzer, 2007.

[25] QIANG YANG,XINDONG WU, ―10 Challenging Problems in Data Mining Research‖.

[26] Michael Goebel,Le Gruenwald ―A Survey of Data Mining and Knowledge Discovery Software Tools‖, ACM, 1999.

[27] DianshengGuo, Jeremy Mennis, ―Spatial data mining and geographic knowledge discovery—An introduction‖, 2009 Elsevier

[28] Pavel Berkhin Accrue Software, Inc.‖Survey of Clustering Data Mining Techniques‖.

TCET

Celebrating its 21st years of Imparting Quality Education

Estd. 2001

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

आज़ादी का अमृत महोत्सव

G20 भारत 2023 INDIA

14 CONFERENCE

MCW 2023 Invention & Innovation

MULTICON-W 2023

IC-ICN 2023

- A platform for Multiple Conferences and Workshops

14th International & National Conferences and Workshops

Certificate

APPRECIATION

This is to certify that Dr./Mr./Ms. **Omkar K. Khadke** has presented/participated/contributed a **Short** length paper with the title **Review On Challenges and Issues in Data Mining** in the **International Conference on Intelligent Computing & Networking (IC-ICN 2023)** organized during February, 24th & 25th, 2023 at **Thakur College of Engineering and Technology,** Kandivali (E), Mumbai.

Dr. B. K. Mishra
Principal & Program Chair

Jagdu Singh Charitable Trust's (Regd.)

THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY

Autonomous College Affiliated to University of Mumbai

Approved by All India Council for Technical Education(AICTE) and Government of Maharashtra

A - Block, Thakur Educational Campus, Shyamnarayan Thakur Marg, Thakur Village, Kandivali (East), Mumbai - 400 101

Tel.: 022-6730 8000 / 8106 / 8107 Telefax: 022-2846 1890 • Email: tcet@thakureducation.org

Website: www.tcetmumbai.in www.thakureducation.org

TCET

Celebrating its 21st years of Imparting Quality Education

Estd. 2001

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

आज़ादी का अमृत महोत्सव

G20 भारत 2023 INDIA

14 CONFERENCE

MCW 2023 Invention & Innovation

MULTICON-W 2023

IC-ICN 2023

- A platform for Multiple Conferences and Workshops

14th International & National Conferences and Workshops

Certificate

APPRECIATION

This is to certify that Dr./Mr./Ms. **Lowlesh Yadav** has presented/participated/contributed a **Short** length paper with the title **Review On Challenges and Issues in Data Mining** in the **International Conference on Intelligent Computing & Networking (IC-ICN 2023)** organized during February, 24th & 25th, 2023 at **Thakur College of Engineering and Technology,** Kandivali (E), Mumbai.

Dr. B. K. Mishra
Principal & Program Chair

Jagdu Singh Charitable Trust's (Regd.)

THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY

Autonomous College Affiliated to University of Mumbai

Approved by All India Council for Technical Education(AICTE) and Government of Maharashtra

A - Block, Thakur Educational Campus, Shyamnarayan Thakur Marg, Thakur Village, Kandivali (East), Mumbai - 400 101

Tel.: 022-6730 8000 / 8106 / 8107 Telefax: 022-2846 1890 • Email: tcet@thakureducation.org

Website: www.tcetmumbai.in www.thakureducation.org

TCET

Celebrating its 21st years of Imparting Quality Education

Estd. 2001

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

TSDW

आज़ादी का अमृत महोत्सव

G20 भारत 2023 INDIA

14 CONFERENCE

MCW 2023 Invention & Innovation

MULTICON-W 2023

IC-ICN 2023

- A platform for Multiple Conferences and Workshops

14th International & National Conferences and Workshops

Certificate

APPRECIATION

This is to certify that Dr./Mr./Ms. **Neehal B. Jiwane** has presented/participated/contributed a **Short** length paper with the title **Review On Challenges and Issues in Data Mining** in the **International Conference on Intelligent Computing & Networking (IC-ICN 2023)** organized during February, 24th & 25th, 2023 at **Thakur College of Engineering and Technology,** Kandivali (E), Mumbai.

Dr. B. K. Mishra
Principal & Program Chair

Jagdu Singh Charitable Trust's (Regd.)

THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY

Autonomous College Affiliated to University of Mumbai

Approved by All India Council for Technical Education(AICTE) and Government of Maharashtra

A - Block, Thakur Educational Campus, Shyamnarayan Thakur Marg, Thakur Village, Kandivali (East), Mumbai - 400 101

Tel.: 022-6730 8000 / 8106 / 8107 Telefax: 022-2846 1890 • Email: tcet@thakureducation.org

Website: www.tcetmumbai.in www.thakureducation.org

TCET

Celebrating its 21st years of Imparting Quality Education

Estd. 2001

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

आज़ादी का अमृत महोत्सव

G20 भारत 2023 INDIA

14 CONFERENCE

MCW 2023 Invention & Innovation

MULTICON-W 2023

IC-ICN 2023

- A platform for Multiple Conferences and Workshops

14th International & National Conferences and Workshops

Certificate

APPRECIATION

This is to certify that Dr./Mr./Ms. **Vijay Rakhade** has presented/participated/contributed a **Short** length paper with the title **Review On Challenges and Issues in Data Mining** in the **International Conference on Intelligent Computing & Networking (IC-ICN 2023)** organized during February, 24th & 25th, 2023 at **Thakur College of Engineering and Technology,** Kandivali (E), Mumbai.

Dr. B. K. Mishra
Principal & Program Chair

Jagdu Singh Charitable Trust's (Regd.)

THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY

Autonomous College Affiliated to University of Mumbai

Approved by All India Council for Technical Education(AICTE) and Government of Maharashtra

A - Block, Thakur Educational Campus, Shyamnarayan Thakur Marg, Thakur Village, Kandivali (East), Mumbai - 400 101

Tel.: 022-6730 8000 / 8106 / 8107 Telefax: 022-2846 1890 • Email: tcet@thakureducation.org

Website: www.tcetmumbai.in www.thakureducation.org

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 11, November 2022

DOI: 10.17148/IJARCCE.2022.111149

Review On Challenges and Issues in Data Mining

Omkar K. Khadke[1], Lowlesh N. Yadav[2], Neehal B. Jiwane[3]

B-Tech Final Year Student, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India[1]

Assistant Professor, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India[2]

Assistant Professor, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India[3]

Abstract: Data Mining data and data from high databases has been recognized by several researchers as a key analysis topic in machine learning and info system and in several industrial corporations as a crucial space with a chance of major revenues .we gift a two-dimensional read of information mining. the foremost dimensions area unit data, knowledge, technologies and application. Researchers in some totally different fields have shown their nice interest in data mining. during this section, we tend to in brief define methodology, user interaction. data processing analysis has powerfully impact society and can still do thus within the future.

Keywords: Data Mining, Machine Learning, Data Mining Issues.

1. INTRODUCTION

Data mining is that the method of motion numerous queries and extracting helpful info, patterns, and trends typically antecedently unknown from massive quantities of knowledge probably keep in databases. basically, for several organizations, the goals of knowledge mining embrace up selling capabilities, detection abnormal patterns, and predicting the long run supported past experiences and current trends. there's clearly a requirement for this technology. There square measure massive amounts of current and historical information being keep. Therefore, as databases become larger, it becomes progressively tough to support higher cognitive process. additionally, the information might be from multiple sources and multiple domains. there's clear have to be compelled to associatealyze the information to support designing and alternative functions of an enterprise. numerous terms are wont to talk to data processing. These embrace knowledge/data/information discovery and knowledge/data/information extraction. so data processing is that the natural evolution of question and news tools. Everyone, World Health Organization creates queries and reports, advantages from having data processing capabilities

2. CHARACTERISTIC OF DATA MINING

Characteristics of information Mining:
Data mining service is a simple sort of military operation methodology whereby that all the relevant data goes through some style of identification method. And eventually at the tip of this method, one will confirm all the characteristics of the information mining method.

2.1. Enhanced quantities of data: In earlier days, the information mining system will be determined with the assistance of their shoppers and customers, however in today's date, one will acquire any range of data while not the assistance of these shoppers. Moreover, once this sort of revolution within the mining system, it conjointly extra a new downside which is massive quantities of labor. With the assistance of this data technology, one will acquire an oversized range of data with none additional burden or bother.

2.2. Provides incomplete data: Most of the folks give incomplete data concerning themselves in a number of the survey conducted with the assistance of information mining systems. Therefore, folks ignore the worth of their data which is why they supply incomplete data concerning themselves in those surveys conducted for the good thing about the mining

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 11, November 2022
DOI: 10.17148/IJARCCE.2022.111149

systems. Moreover, these mining systems modified the attitude of individuals and since of that, folks worry the exchange of their personal data.

2.3. difficult information structure: Data mining could be a kind whereby that all the knowledge is gathered and incorporated with the assistance of data assortment techniques. These data collection techniques area unit additional of manual and rest area unit technological. Therefore, most of the understanding and determination of those mining will be a touch difficult than different structures of data technology.

3. DATA MINING APPLICATIONS

Data mining is generally employed by several of the large gaints within the info technology sector and conjointly some little industries by creating use of their own techniques. a number of the favored domains ar, Market Analysis and Management Corporate Analysis & Risk Management Fraud Detection

3.1. market research and Management: The following mentioned ar the varied fields of the market wherever the information mining method is effectively used,

- Customer identification
- Finding client necessities
- Cross-market analysis
- Target selling
- Determining client buying pattern
- Provides outline info

3.2. company Analysis and Risk Management: The following mentioned ar the varied fields of the company sector wherever the information mining method is effectively used,

- Finance designing
- Asset analysis
- Resource designing
- Competition

3.3. Fraud Detection: Frauds and malware is one in every of the foremost dangerous threats on the net. it's virtually a sort of crime that's increasing day once day. The fraud detection method is chiefly used through MasterCard services and telecommunication. With the assistance of the services most of the necessary info like length of the decision, location, the time and day etc is non inheritable that helps in success.

4. DATA MINING CHALLENGES

These days data processing and knowledge revealing area unit developing crucial innovations for researchers and businesses in varied areas. data processing was forming into a setup and confided up to the mark, heretofore forthcoming data processing challenges should be tackled.

4.1. Security and Social Challenges
Dynamic techniques area unit done through knowledge assortment sharing, which needs spectacular security. non-public data concerning individuals and touchy data is gathered for the client's profiles, shopper normal of conduct understanding—illicit admittance to data and therefore the secret plan of knowledge turning into a big issue.

4.2. Clanging and Incomplete knowledge
Data Mining could be a thanks to get data from Brobdingnagian volumes of information. This gift reality of knowledge is clanging, incomplete, and heterogeneous. knowledge in Brobdingnagian amounts often are unreliable or inaccurate. These problems may be owing to human mistakes, blunders, or errors within the instruments that live the info.

4.3. Distributed knowledge
True knowledge is generally place away at varied stages in distributed process conditions. it's going to get on the web, individual systems, or maybe databases. it's basically arduous to hold all {the knowledge|the info|the information} to a unified data archive chiefly owing to technical and structure reasons.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 11, November 2022
DOI: 10.17148/IJARCCE.2022.111149

4.4. Complicated knowledge

True knowledge is heterogeneous, and it's going to be media knowledge, as well as tongue text, statistic, spatial knowledge, temporal knowledge, complicated knowledge, audio or video, images, etc. it's actually arduous to alter these varied forms of knowledge and think about the required data. a lot of typically than not, new apparatuses and systems would want to be created to separate vital data.

4.5. Performance

The presentation of the info mining framework primarily depends upon the productivity of techniques and algorithms used. On the off likelihood that the techniques and algorithms planned aren't sufficient , at that time, it'll influence the presentation of the info mining live unfavorably.

4.6. Quantifiability and potency of the Algorithms

The Data Mining rule ought to be climbable and economical to disengage data from tremendous measures {of knowledge|of knowledge|of information} within the data set.

4.7. Improvement of Mining Algorithms

Factors, for instance, the issue of information mining approaches, the large size of the information, and therefore the entire knowledge flow, inspire the distribution and creation of parallel data processing algorithms.

4.8. Incorporation of information

In the event that information will be consolidated, a lot of correct and reliable data processing arrangements will be found. prognosticative tasks will create a lot of correct predictions, whereas descriptive tasks will come back up with a lot of helpful findings. Be that because it could, gathering and as well as foundation information is unpredictable.

4.9. Knowledge visualization

Data visualization could be a very important cycle in data processing since it's the foremost interaction that shows the output in an exceedingly respectable thanks to the shopper. the knowledge freed have to be compelled to pass away the importance of what it plans to pass away. However, ordinarily, it's actually arduous to handle the knowledge exactly and squarely to the top user. The output data and computer file being terribly effective, successful, and sophisticated knowledge perception strategies ought to be applied to create it fruitful.

4.10. Knowledge Privacy and Security

Data mining generally prompts important governance, privacy, and knowledge security problems. as an example, once a distributor investigates the acquisition details, it uncovers data concerning buying propensities and decisions of shoppers while not their authorization.

4.11. Interface

The information is set utilizing data processing devices is effective simply within the event that it's fascinating or a lot of all affordable by the shopper. From nice illustration translation of information, mining results will be expedited, and betters comprehend their conditions. several explorations area unit in deep trouble monumental knowledge sets that manipulate and show strip-mined information to urge an excellent perception.

4.12. Mining keen about Level of Abstraction

Data Mining measures ought to be community-oriented in light-weight of the very fact that it permits shoppers to specialize in example optimizing, presenting, and pattern finding for data processing keen about conveyance results back.

4.13. Integration of information

Previous data can be accustomed communicate examples to precise discovered patterns and direct the exploration method.

4.14. Mining Methodology Challenges

These difficulties area unit known with data processing strategies and their limits. Mining strategies that cause the difficulty area unit the management and handling of noise in knowledge, the spatiality of the domain, the range of information accessible, the flexibility of the mining methodology, and so on.

5. DATA MINING ISSUES

5.1. Mining Strategies & User Interaction Problems

The issues may be with finding out the best-fit methodologies of knowledge mining, that area unit association, classification, clump analysis, prediction, sequent patterns or pattern pursuit, call trees, outlier analysis or anomaly

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 11, November 2022
DOI: 10.17148/IJARCCE.2022.111149

detection, and neural network.The professionals with active expertise during this domain usually struggle with these problems whereas victimization the same data processing strategies.

5.1.1.Hard to Derive information for numerous Domains

The beneficiary may be associated with totally different industries and domains. thus do dissent their demand and information discovery, that need niche-based knowledge extraction, transformation, & loading of numerous forms like visuals, text, or numbers. This method covers a broad vary of information discovery processes, that may be a challenge.

5.1.2.Lack of Interactive Models

Interactive modelling makes pattern searches easier. For this purpose, datasets area unit extracted, refined, converted, and cleaned to make sure that they manufacture the intelligence that's needed through the mining request.

5.1.3.Lacking skilled information

For every level, from net knowledge extraction to process, & modelling, you would like knowledgeable matter specialists or knowledge miners. Their active expertise guides them through the complete discovery method, that is to work out patterns. Their qualification and operating expertise guarantee golf stroke patterns during a crisp and comprehensive format. Finding such professionals with answer ability is so Associate in Nursing uphill battle.

5.1.4.Ad hoc data processing isn't straightforward

Ad-hoc mining answers to a selected business demand or question, that insights will create it easier to abstract. The Structured search language is that the search language that supports the accidental information discovery method. This language guides users through mining tasks by mining or knowledge analysis corporations, that work on the optimized & versatile structure of records. This task is once more not very easy.

5.1.5.Comprehensiveness may be a Challenge

The visual presentation ought to be impactful to let the analyst simply see and extract insights in no time. Here again, the team ought to be competent to run totally different tools like Tableau, Sisence, Excel, etc. for a good visual presentation victimization charts, graphs, etc. Simply put, the presentation of models ought to be perceivable, that isn't sort of a walkover.

5.1.6.Noisy knowledge is tough to Handle

Noisy knowledge talk over with redundancies or useless knowledge within the info or massive knowledge. addressing the noises like duplicity, incomplete info, and errors may be an enormous challenge. But, going ahead with them will disturb the effectiveness of consequence. Result? The pattern are poor and useless.

5.1.7.Measuring Patterns

The patterns filtered, tested, & evaluated ought to be attention-grabbing as a result of they represent either intelligence or possible solutions or lack novelty.

5.2. Performance problems

It is necessary that the modelling ought to be versatile and qualified through quality tests. These models area unit seemingly to be AI via machine learning. Here, the performance problems should be acknowledged to sail across them with previous solutions.

5.2.1.Inefficiency and issue in measurability

Certainly, a info is needed, principally colossal-sized niche-based records. net knowledge extraction & capturing create it approach easier. although it's paper-bound, the OCR conversion and cleansing practices facilitate in making ready a info. But, there area unit challenges like king protea lure, captcha, and privacy settings which will hamper the provision of significant details. Sometimes, lacking tools may prove an enormous barrier to shaping the potency and measurability of knowledge mining.

5.2.2.Inability to figure with Parallel, Distributed, and progressive Algorithms

The metrics like prodigious size of databases, wide distribution, and complexness of knowledge mining strategies push to derive parallel and distributed data processing algorithms. These algorithms split records into elements, that area unit any processed within the same manner. Finally, the results of all elements area unit compiled along. this is often however the progressive algorithms still update databases while not mining the info once more from scratch.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 11, November 2022
DOI: 10.17148/IJARCCE.2022.111149

5.3. Numerous knowledge varieties problems

Data has several faces. you'll notice it in its visual, audio, text, and numeric forms. process these differing types and so, mining could also be troublesome.

5.3.1.Dealing with relative and complicated sorts of knowledge

Your supply knowledge might have PDF, transmission objects, abstraction knowledge, temporal or alternative sorts of datasets. this is often so a bone-breaking expertise to form such a customary tool which will ideally method every kind of knowledge within the same approach.

So, you would like to customize or access a specifically designed tool to mine from a specific kind of datasets, that is a rich deal. And also, you must have the power to manage that tool.

5.3.2.Modelling from Heterogeneous Databases

As there area unit variety of sources to access knowledge like local area network or WAN, you'll not expect to own records in Associate in Nursing ideally similar type and format. it's just because of their storage during a structured, semi-structured, or unstructured type. Therefore, mining information from them isn't straightforward.

5.4. knowledge Security & Privacy

Personally, place able knowledge is sensitive and other people don’t wish to share it with anyone. Here, a threat to its privacy & security may be a reason to significantly suppose.

5.4.1.Security

Mostly, knowledge area unit shared over the web, the cloud, and servers to make sure their access 24X7 remotely. This access may be dangerous if it's done through a public network, that isn't secure. Vulnerability poses an enormous risk. So, the inter changeableness of any record ought to be outlined through encoding.

5.4.2.Privacy considerations

Dynamic techniques area unit adopted to gather info from numerous resources, particularly from knowledge subjects. This assortment isn't unhazardous, as they carry in person place-able info. Hackers tend to interrupt in and exclude these credentials. Here, privacy controls, authorization, and knowledge compliance like GDPR seem during a major safeguarding role.

6. ADVANTAGES OF DATA MINING TECHNIQUES

There square measure many styles of advantages and blessings of information mining systems. one among the essential matters of those mining creates an entire structure of research of mining techniques.

6.1.It's useful to predict future trends: Most of the operating nature of the info mining systems carries on all the informational factors of the weather and their structure.One of the common advantages that may be derived with these data processing systems is that they will be useful whereas predicting future trends. which is sort of potential with the assistance of technology and behavioral changes adopted by the individuals.

6.2. It signifies client habits: For example, whereas operating within the selling trade one will perceive all the matters of client behaviour and their habits. which is feasible with the assistance of information mining systems. As these data processing systems handle all the knowledge deed techniques. it's useful keep track of client habits and their behavior.

6.3. Helps in call making: There square measure some folks that create use of those data processing techniques to assist them with some reasonably deciding. Nowadays, all the knowledge concerning something may be determined simply with the assistance of technology and equally, with the assistance of such technology one will create an explicit call concerning one thing unknown and surprising.

6.4. Increase company revenue: As it has been explained earlier that methoding} could be a process whereby that it involves some form of technology to accumulate some info concerning something potential. And this kind of technology makes things easier for his or her profit earning magnitude relation. As individuals will collect info concerning the marketed merchandise on-line, that eventually reduces the price of the merchandise and their services.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 11, November 2022
DOI: 10.17148/IJARCCE.2022.111149

6.5. It depends upon market-based analysis: Data mining method could be a system whereby that all {the info|the knowledge|the data} has been gathered on the idea of market information. Nowadays, technology plays a vital role in everything which casualty may be seen in these data processing systems. Therefore, all the knowledge collected through these data processing is largely from selling analysis.

6.6. Fast fraud detection: Most components of {the data|the info|the info} mining method is largely from information gathered with the assistance of promoting analysis. With the assistance of such selling analysis, one also can establish those dishonest acts and merchandise obtainable within the market. Moreover, with the assistance of it one will perceive the importance of correct info.

7. DISADVANTAGES OF DATA MINING TECHNIQUES

Data mining technology are some things that helps one person in their {decision making|deciding|higher cognitive method} which deciding could be a process whereby that all the factors of mining is concerned exactly. And whereas the involvement of those mining systems, one will stumble upon many disadvantages of information mining and that they square measure as follows.

7.1. It violates user privacy: It is a far-famed incontrovertible fact that data processing collects info concerning individuals mistreatment some market-based techniques and data technology. And these methoding} process involves many numbers of things. But whereas involving those factors, data processing system violates the privacy of its user which is why it lacks within the matters of safety and security of its users. Eventually, it creates miscommunication between individuals.

7.2. Further extraneous information: The main functions of {the data|the info|the info} mining systems produce a relevant house for useful information. But the most drawback with these info collections is that there's a prospect that the gathering of data processes may be a touch overwhelming for all. Therefore, it's noticeably essential to take care of a minimum level of limit for all the info mining techniques.

7.3. Misuse of information: As it has been explained earlier that within the data processing system the chance of safety and security live square measure extremely minimal . which is why some will misuse this info to hurt others in their own manner. Therefore, the info mining system has to amendment its course of operating in order that it will scale back the magnitude relation of misuse of data through the mining method.

7.4. Accuracy of data: Most of the time whereas collection info concerning sure components one wont to ask for facilitate from their shoppers, however today everything has modified. And currently the method of data assortment created things simple with the mining technology and their ways. One of the foremost potential limitations of this data processing system is that it will offer accuracy of information with its own limits.

8.CONCLUSION

Data mining seeks to extract hidden information from large amount of information. methoding} is that the process of extracting and valuable attention-grabbing patterns from raw collection of information. data processing may be accustomed uncover patterns within the information however it's typically dispensed solely on the samples of information. This mining method are ineffective if the samples don't seem to be a decent illustration of the larger body of the info.And beside this, today‘s competition is one among the most necessary challenges facing by all organizations
and industries in data processing problems. that's onerous to seek out in a particular organization or business that has no rival to him.This paper describes numerous tasks; goals and limitations of information mining. Additionally-this paper additionally discussed regarding the varied valuable problems; future challenges and problems in field of information mining that I important to try to to more more practical analysis during this emerging field.

REFERENCES

[1] M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.
[2] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.
[3] Chieh-Yuan, T. and T. Min-Hong. A dynamic Web service based data mining process system. in Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on. 2005.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 11, November 2022
DOI: 10.17148/IJARCCE.2022.111149

[4] Ding, P. A formal framework for Data Mining process model. In Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on. 2009.

[5] Cooley, B. Mobasher and J. Srivsatava. Web Mining: Information and Pattern Discoveryon the Word Wide Web. In Proceedings of the 9th IEEE International Conference on Tools with AI (ICTAI,97), Nov. 1997.

[6] Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93.

[7] J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98.

[8] Huhtala, J. Kärkkäinen, P. Porkka, H. Toivonen. Efficient Discovery of Functional and Approximate Dependencies Using Partitions. ICDE'98.

[9] W. Shavlik and T. G. Dietterich. Readings in Machine Learning.Morgan Kaufmann, 1990.

[10] P.Tan, M. Steinbach, and V. Kumar. Introduction to Data Mining.Addison Wesley, 2005.

[11] S. M. Weiss and C. A. Kulikowski. Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.Morgan Kaufman, 1991.

[12] Xinjian Guo, Yilong Yin1, Cailing Dong, Gongping Yang, Guangtong Zhou,"On the Class Imbalance Problem" Fourth International Conference on Natural Computation, 2008.
[13]
[14] David P. Williams, Member, Vincent Myers, and Miranda Schatten Silvious, "Mine Classification With Imbalanced Data", IEEE Geosciences And Remote Sensing Letters, Vol. 6, No.3, July 2009.

[15] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, Amri Napolitano "A Comparative Study of Data Sampling and Cost Sensitive Learning" , IEEE International Conference on Data Mining Workshops.15-19Dec.2008.
[16]
[17] Clifton, Christopher "Encyclopedia Britannica: Definition of Data Mining", Retrieved 2010-12-09.

[18] Ian H. Witten; Eibe Frank; Mark A. Hall, ―Data Mining: Practical Machine Learning Tools and Techniques (3rd Ed.)‖, Elsevier, 30 January 2011.

[19] Kantardzic, Mehmed ―Data Mining: Concepts, Models, Methods, and Algorithms‖, John Wiley & Sons, 2003.

[20] Günnemann, S.; Kremer, H.; Seidl, T. "An extension of the PMML standard to subspace clustering models". Proceedings of the 2011 workshop on Predictive markup language modeling - PMML '11.

[21] Ralf Mikut; Markus Reischl, "Data Mining Tools". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Retrieved October 21, 2011.

[22] Domenico Talia; Paolo Trunfio"How distributed data mining tasks can thrive as knowledge services". Communications of the ACM , 2010 .

[23] Karl Rexer, Heather Allen, & Paul Gearan ―Data Miner Survey – Overcoming Data Mining's Top Challenges‖, 2011.

[24] Hans-Peter Kriegel , Karsten M. Borgwardt et.al ―Future trends in data mining‖, sprinzer, 2007.

[25] QIANG YANG,XINDONG WU, ―10 Challenging Problems in Data Mining Research‖.
[26] Michael Goebel,Le Gruenwald ―A Survey of Data Mining and Knowledge Discovery Software Tools‖, ACM, 1999.
[27] DianshengGuo, Jeremy Mennis, ―Spatial data mining and geographic knowledge discovery—An introduction‖, 2009 Elsevier
[28] Pavel Berkhin Accrue Software, Inc.‖Survey of Clustering Data Mining Techniques‖.

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH INCOMPUTER AND COMMUNICATION ENGINEERING

A monthly peer-reviewed journal

Impact Factor 7.918

Indexed by Google Scholar, Mendeley, NAAS Accredited Science Journal, ISO 3297:2007 Certified
Thomson Reuters ID I-8645-2017

Google Scholar doi Crossref MENDELEY PlumX Metrics

CERTIFICATE OF PUBLICATION

OMKAR K. KHADKE

B-Tech Final Year Student, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India

Published a paper entitled

Review On Challenges and Issues in Data Mining

Volume 11, Issue 11, November 2022

DOI: 10.17148/IJARCCE.2022.111149

Certificate#:IJARCCE/2022/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

TejassPublisheers
ORGANIZATION

Editor-in-Chief
IJARCCE

SERTIFIKAAT 証明書 شهادة Sertifikat CERTYFIKAT 증명서 CERTIFICADO СЕРТИФИКАТ 证书 POTVRDA SERTİFİKA

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH INCOMPUTER AND COMMUNICATION ENGINEERING

A monthly peer-reviewed journal

Impact Factor 7.918

Indexed by Google Scholar, Mendeley, NAAS Accredited Science Journal, ISO 3297:2007 Certified
Thomson Reuters ID I-8645-2017

Google Scholar doi Crossref MENDELEY PlumX Metrics

CERTIFICATE OF PUBLICATION

LOWLESH N. YADAV

Assistant Professor, Computer Science And Engineering, Shri Sai College Of Engineering And Technology, Bhadrawati, Maharashtra, India

Published a paper entitled

Review On Challenges and Issues in Data Mining

Volume 11, Issue 11, November 2022

DOI: 10.17148/IJARCCE.2022.111149

Certificate#:IJARCCE/2022/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

TejassPublisheers
ORGANIZATION

Editor-in-Chief
IJARCCE

SERTIFIKAAT 証明書 Sertifikat CERTYFIKAT 증명서 CERTIFICADO СЕРТИФИКАТ 证书 POTVRDA SERTİFİKA

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH INCOMPUTER AND COMMUNICATION ENGINEERING

A monthly peer-reviewed journal

Impact Factor 7.918

Indexed by Google Scholar, Mendeley, NAAS Accredited Science Journal, ISO 3297:2007 Certified
Thomson Reuters ID I-8645-2017

Google Scholar doi Crossref MENDELEY PlumX Metrics

CERTIFICATE OF PUBLICATION

NEEHAL B. JIWANE

Assistant Professor, Computer Science And Engineering, Shri Sai College Of Engineering And Technology,Bhadrawati, Maharashtra, India

Published a paper entitled

Review On Challenges and Issues in Data Mining

Volume 11, Issue 11, November 2022

DOI: 10.17148/IJARCCE.2022.111149

Certificate#:IJARCCE/2022/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

TejassPublisheers
ORGANIZATION

Editor-in-Chief
IJARCCE

SERTIFIKAAT 証明書 Sertifikat CERTYFIKAT 증명서 CERTIFICADO СЕРТИФИКАТ 证书 POTVRDA SERTİFİKA