Managing a Workload through Load Balancing Technique in Cloud Environment

By

Patel Srushtiben Rameshbhai

(Enrolment No: - 140400702017)

Internal Guide

Dr. Hiren B. Patel

(Professor, Computer Engineering Department, SPCE)

A Thesis Submitted to
Gujarat Technological University
in Partial Fulfillment of the Requirements for
the Master of Engineering Degree in Computer Engineering

May 2016

Department of Computer Engineering

Sankalchand Patel College of Engineering (040), Visnagar -384315

CERTIFICATE

This is to certify that research work embodied in this thesis entitled **"Managing a Workload through Load Balancing Technique in Cloud Environment "** was carried out by **Patel Srushtiben Rameshbhai** at **Sankalchand Patel College of Engineering, Visnagar (040)** for partial fulfillment of Master of Engineering degree in Computer Engineering to be awarded by Gujarat Technological University. This research work has been carried out under my Guidance and supervision and it is up to my satisfaction.

Date:

Place:

Signature and Name of Guide
Dr.Hiren B.Patel

Signature and Name of Principal
Dr. Dharmesh J. Shah

Seal of Institute

COMPLIANCE CERTIFICATE

This is to certify that research work embodied in this thesis entitled "Managing a Workload Through Load Balancing Technique in Cloud Environment " was carried out by **Patel Srushtiben Rameshbhai (140400702017)** at Sankalchand Patel College of Engineering (040) for partial fulfillment of Master of Engineering degree to be awarded by Gujarat Technological University. She has complied to the comments given by the Dissertation Phase – I as well as Mid Semester Thesis Reviewer to my satisfaction.

Date:

Place:

Signature and Name of Student
Patel Srushtiben Rameshbhai
(140400702017)

Signature and Name of Guide
Dr. Hiren B. Patel

PAPER PUBLICATION CERTIFICATE

This is to certify that research work embodied in this thesis entitled “Managing a Workload through Load Balancing Technique in Cloud Environment” carried out by **Patel Srushtiben Rameshbhai (140400702017)** at **Sankalchand Patel College of Engineering (040)** for partial fulfillment of Master of Engineering degree to be awarded by Gujarat Technological University, has published article entitled **Dynamic Load Balancing techniques for improving performance in Cloud Computing** for publication by the **International Journal of Computer Applications (IJCA) at NewYork, USA** on **25th March 2016.**

Date:
Place:

Candidate Name
Patel Srushtiben Rameshbhai
(140400702017)

Guide Name & Sign.
Dr. Hiren B. Patel

Signature Stamp and Name of Principal
Dr. D. J. Shah

(Seal of the Institute)

THESIS APPROVAL CERTIFICATE

This is to certify that research work embodied in this thesis entitled “Managing a Workload through Load Balancing Technique in Cloud Environment” was carried out by **Patel Srushtiben Rameshbhai (140400702017)** at Sankalchand Patel College of Engineering (040) is approved for the degree of Master of Engineering with specialization in Computer Engineering by Gujarat Technological University.

Date:

Place:

Examiner’s Sign and Name:

---------------------------------- ---------------------------------

( ) ( )

UNDERTAKING

We hereby certify that we are the sole authors of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other University or Institution.

We certify that, to the best of our knowledge, the current thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in our thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, we certify that we have obtained a written permission from the copyright owner(s) to include such material(s) in the current thesis and have included copies of such copyright clearances to our appendix.

We declare that this is a true copy of thesis, including any final revisions, as approved by thesis review committee.

We have checked write up of the present thesis using anti-plagiarism database and it is in allowable limit. Even though later on in case of any complaint pertaining of plagiarism, we are sole responsible for the same and we understand that as per UGC norms, University can even revoke Master of Engineering degree conferred to the student submitting this thesis.

Date:

Signature of Student: **Signature of Guide:**

Name of Student: Patel Srushtiben R. **Name of Guide: Dr. Hiren B. Patel**

Enrollment No: 140400702017 **Institute Code: 040**

ACKNOWLEDGEMENT

I am heartily thankful to my supervisor, Prof. Hiren B. Patel, whose encouragement, guidance and support from the initial to the final level enabled me to develop an attitude to carry out the research work for the selected area.

Lastly, I offer my regards to all those who supported me in any respect directly or indirectly during the completion of this work.

Patel Srushtiben Rameshbhai

(140400702017)

TABLE OF CONTENTS

List of Figures

List of Tables

Managing a Workload through Load Balancing Technique in Cloud Environment

Submitted By

Srushti Patel

(140400702017)

Supervised By

Dr. Hiren B. Patel

Professor in Computer Engineering Department, SPCE, Visnagar

ABSTRACT

Being an emerging area in IT sector, Cloud computing enables a wide range of users to access distributed, scalable, virtualized hardware and/or software, applications and platforms which are provided over the Internet. Cloud Computing is a shared pool of Configurable computing resources which requires the proper distribution of dynamic workload among multiple computers to ensure that no single node is underloaded or overloaded. Load balancing aims to reduce response time of jobs, to reduce the migration time for VMs, to increase resource optimization, to maximize overall performance and avoid overload of any single node. In our work, we study various techniques related to load balancing in Cloud Environment and further, we proposed modified agent based dynamic load balancing algorithm by adding the standard deviation method to decide whether the host is overloaded or not. To provide a better load balancing in terms of better performance, proposed algorithm has been implemented and evaluated using CloudSim simulator. The simulator result shows reduction in CPU time, increase in resource utilization and also overall performance is improved as compared to the existing load balancing algorithm.

Chapter – 1

Introduction

Topics covered

1.1 Introduction to Cloud computing

1.1.1 Definition and characteristics

1.1.2 Advantages

1.1.3 Issues

1.1.4 Service Models

1.1.5 Deployment Models

1.1.6 Key parameter

1.2 Problem Statement

1.3 Motivation

1.4 Objective

1.1 Introduction to Cloud computing

1.1.1 Definition and characteristics [1]

National Institute of standard and Technology categorizes the computing resources into networks, servers, storage, applications and services. These resources are provisioned to demanding users and cost to the users is calculated on the bases of resources usage. This model is shown in Fig. 1.1.

This Cloud model support accessibility and collected of five essential characteristics, three service models, and four deployment models.

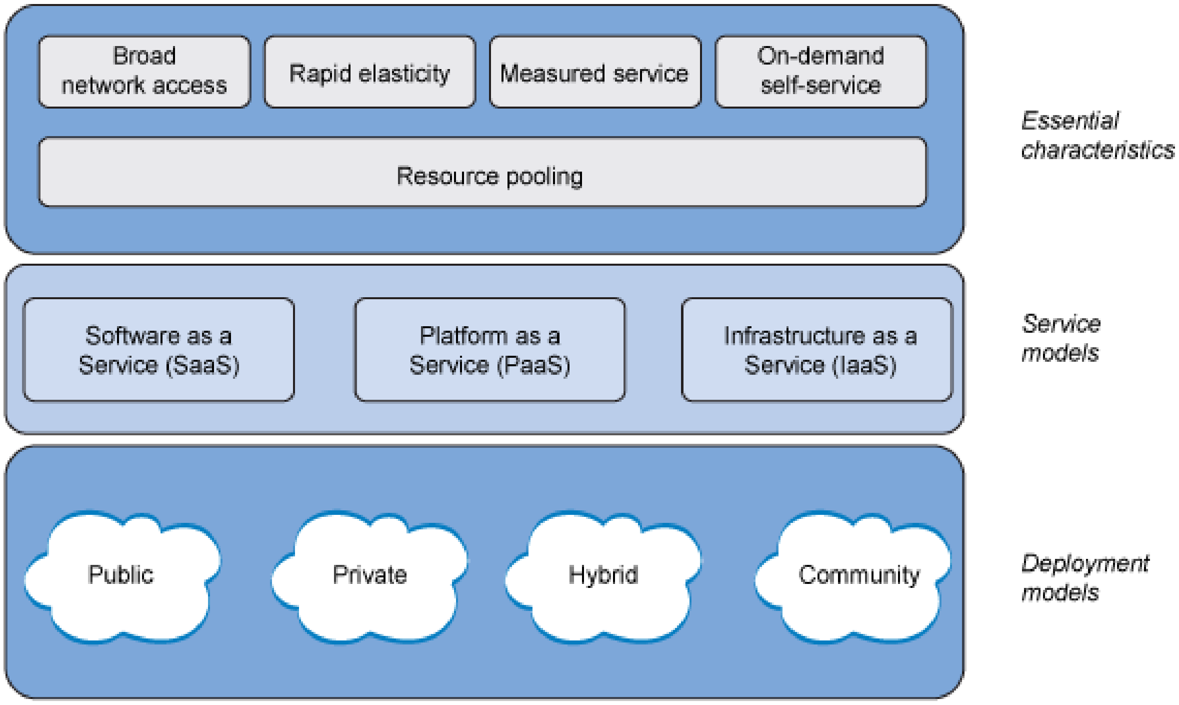


Fig. 1.1. Cloud Model

http://www.ibm.com/developerworks/Cloud/library/cl-extenddevtoCloudbigdata/nist-Cloud

Characteristics:

Five important characteristics of Cloud computing is following:

1. On-demand self-service:

A client can provision computing resources (for ex. Server time and network storage) as per their need without any interaction with service provider.

2. **Broad network access**:

All the types of client is provided platform independent access because Cloud is available over entire network which is accessed by standard mechanisms which includes a mixture of heterogeneous OS, and thick & thin platforms such as mobile phones, laptops etc.

3. **Resource pooling**:

By the use of multi-tenant model multiple clients are served by pooled computing resources of provider with different virtual and physical resources are dynamically allocated and reallocated as per client requirement. Customer has no idea of where the provided resources are located. Storage, processing, memory, and network bandwidth are the example of resources.

4. **Rapid elasticity:**

Resources are allocated and reallocated automatically. Resources in Cloud computing are limitless. They can be purchased at any time and in any quantity.

5. **Measured service:**

Based on metered system the use of Cloud system resources is measured, audited and reported to the client.

1.1.2 Advantages [1]

1. **Flexibility:**

Cloud computing provide high rate of flexibility to client. Client can access their application and data from anywhere on any system in the world.

2. **Low Cost:**

Cloud is offered to the clients at very low rate unlike traditional desktop software which result in decrease in operating cost of IT companies.

3. **Better storage capacity:**

Client can save more data in Cloud computing environment than personal computer.

Because of that client need not have to upgrade their computer's memory which reduces the cost for companies and user.

4. **Flexible Compatibility:**

Services of Cloud computing can be used using different electronic devices having internet access (ex. Laptop, desktop, mobile phone)

5. **Automatic Updating:**

Cloud computing services enables users to download updates for software, once the server is updated client do not have to do anything which saves company's time and effort.

6. **Easier organization of Data and Information:**

Because of centralized storage of data it is easy to maintain them.

7. **Backup and Recovery:**

Recovery of information is done very efficiently so that taking a backup and restoring data is very easier in Cloud.

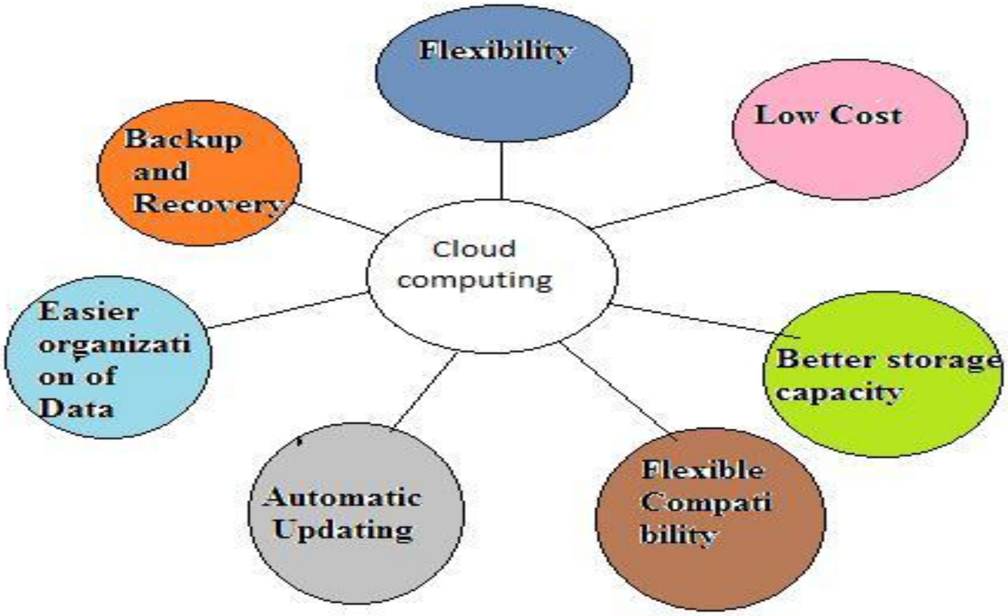


Fig. 1.2 Cloud computing advantage

Source:http//www.defenginc.com/sites/all/images/Cloudadvantage.jpg

1.1.3 Issues [1]

1. Privacy and security in the Cloud:

Privacy and security are the sensitive issue of Cloud computing. Because of remote Cloud based infrastructure company may give away private and secret information. Companies have to rely on Cloud provider that they will protect their information from unauthorized users.

2. Dependency and vendor lock-in:

If user wants change the provider then that can be very difficult for him to transfer the vast data from one provider to another provider.

3. Requires a Constant internet connection:

Cloud services makes the business relies on internet connection. Without internet it is impossible to get Cloud services.

4. Increased Vulnerability:

If there is compromise on the machine where the data is stored all the personnel information is leaked. Personnel information may get leaked which may be targeted by hackers and malicious user.

5. Availability:

When user's applications are moved to the Cloud then user expected from service level provider to availability of services and its overall performance.

1.1.4 Service Models [1]

There are three service model of Cloud computing:

1. Infrastructure as a Service:

Processing, storage, networks, and other fundamental computing resources capabilities are provided to the customer where customer can set up and run random software which can include applications and operating system. Customer has control over the operating systems, storage, deployed applications without managing and controlling underlying Cloud physical infrastructure.

2. Platform as a Service:

Using programming languages and tools supported by the Cloud service provider customer can create and deploy applications on Cloud environment. The customer need not have to worry about servers, operating systems, or storage but has control over the deployed applications.

3. Software as a Service:

Customer is provided the facility to use client application running on the Cloud Environment**.** Customer can access their application through the Customer interface. Customer need not have to worry about underlying Cloud infrastructure like network, servers, operating systems, storage.

1.1.5 Deployment Models

There are four types of Deployment models are following:

1. Public Cloud:

Public Cloud as a Cloud made available in pay-as-you-manner to the general public

2. Private Cloud:

Private Cloud as a internal Data center of Business or other Organization and not made available to the general public.

3. Hybrid Cloud:

It is used when private Cloud is Supplemented with computing Capacity from public Cloud.

4. Community Cloud:

A Community Cloud is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy etc.)

1.1.6 Key parameters of Cloud computing [1]

1. Fault tolerance:

When fault enters in to the system we have to apply fault tolerance methods. Fault tolerance used to forecast failure and then take appropriate action which includes failure Recovery, lower cost, improved performance metrics. Load Balancing plays very important role to facilitate this parameters.

2. Load balancing:

Load balancing is used to optimize resource usage, decrease response time, increase throughput and avoid overload at single resource. Which is done by diving work of a single computer between more than one computer.

3. Power Saving:

Ideal power wasted When server run at low utilization which is major cause of more energy consumption. In Cloud environment applications of multiple datacenter are stored on set of servers this is allows for better utilization amortizing the idle power costs more efficiently.

4. Hardware Maintenance:

Because application need not be installed on each computer maintenance of Cloud computing is easier. Application can be accessed from different location.

1.2 Problem Statement

Improper resource utilization in Cloud computing leads to many issues viz. imbalanced workload and performance degradation various load balancing techniques can be used to balance the workload which may result into improvement in performance. We aims to address the issues using modified agent based load balancing technique to minimize the CPU time, increase the resource utilization & decrease the Migration time etc

1.3 Motivation

During the load balancing process, few issues are yet to be fully addressed. Couple of them are: Some of the nodes are overutilized or some of the nodes are underutilized, Improper workload in Cloud environment results into overhead in resource utilization and in turn inefficient usage of energy, response time of jobs, communication cost of servers, maintain cost of VMs, throughput and overload of any single node. By addressing the concern of load balancing, we aim to address multiple facets of Cloud viz. (a) resource utilization (b) CPU time (c) Migration time.

1.4 Objectives

We intend to present a technique to clear up the problem (like Resource utilization, overall performance, CPU Time and overload of any single node) associated with Load Balancing in Cloud Environment.

Chapter – 2

Background Theory

Topic Covered

2.1 Introduction to Load balancing

2.2 Goals of Load balancing

2.3 Types of Load balancing algorithms

- 2.3.1 Static load balancing algorithms
- 2.3.2 Dynamic Load balancing algorithm
- 2.3.3 Comparison of static and dynamic load balancing algorithm

2.1 Introduction to Load Balancing

Load Balancing is the major issues in Cloud Environment. It is the mechanism for Distributing Dynamic workload among the all nodes in Cloud. This is also remove the condition in which some nodes are overloaded while other nodes are under loaded which is used to achieving improvement in Resource utilization and User satisfaction. To removing imbalanced load into the node we can reassign the total load across all nodes into the system and it is also used to improve response time of job. Hence this will increase the overall performance. It is also used to prevent the bottlenecks of system which is occurs due to load imbalance. Proper Load balancing can decrease the Resource Consumption and also make enterprises greener[2]. Implementation of Dynamic load balancing algorithm is most important things for estimation of load, performance of system, nature of work to be transferred, selecting of nodes and many other ones[14]. This load considered can be in terms of CPU load, amount of memory used, delay or Network load.

2.2 Goals of Load Balancing

- performance increases
- When the system is failed then provide the backup plan
- For maintaining stability of system[14]

2.3 Types of Load balancing algorithms

Load balancing techniques can be classified as mainly two categories: static algorithms and dynamic algorithms [6,12] that have been developed for Cloud computing which is shown in fig.2.1.

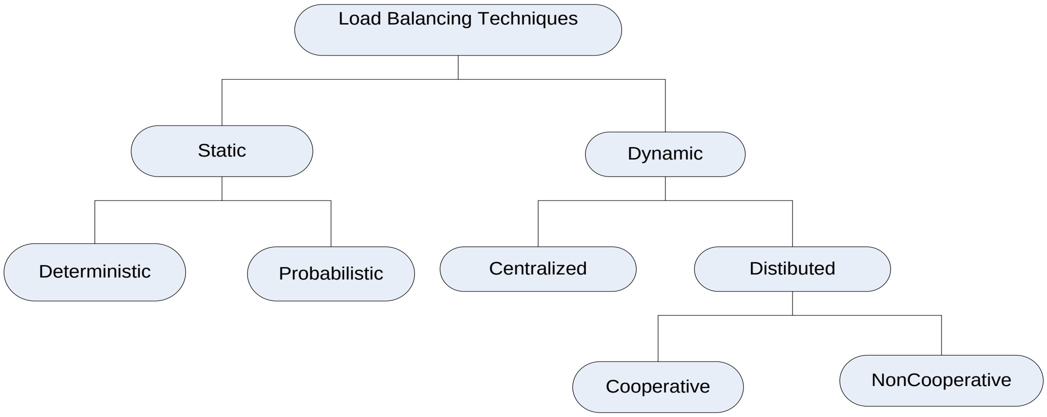


Fig. 2.1 Classification of Load balancing techniques.[13]

2.3.1 Static load balancing algorithms

These algorithms assign the tasks to the nodes to process new requests based on prior knowledge of the properties and capabilities of node. Static algorithms do not change the node’s processing power, memory and storage capacity at run time[2]. Long connectivity applications are cannot be handled by static load balancing algorithms.

2.3.2 Dynamic Load balancing algorithms

In Dynamic load balancing algorithms jobs are assign dynamically and reassign the jobs to another node based on workload of node. Dynamic algorithms are constant monitoring on the jobs and nodes therefore it will give more accuracy than the static load balancing algorithms[6,12].

Dynamic Load Balancing algorithms categorized into two parts: Distributed and non-distributed load balancing algorithms. In distributed, load balancing perform by all nodes into the system n jobs are also assigned to those nodes. This algorithm further divided into part: Cooperative and non-cooperative[14]. This algorithms are used for improving the response time of jobs, first is used for improve the overall response time and second one is used for improving the response time of a local jobs[14].

Non Distributed load balancing algorithms are grouped into two forms: Centralized and decentralized. In centralized load balancing algorithm only central node can balance the whole system load and also maintain the overall load of system. Whenever one central node is crashed then this centralized load balancing system become useless and also it causes a bottleneck. This algorithm is used for small networks.

Policies in dynamic load balancing:

There are four policies[14] of load balancing:

- **Transfer Policy:** Transfer policy is used for transferring jobs from local node to remote node.
- **Selection Policy:** It will selecting a processor for exchanging a load..
- **Location Policy**: It will specify a location of destination node for transferring a job.
- **Information Policy:** Using this policy it will collecting the information about nodes into the system.

Chapter – 3

Review of Literatures

Topics covered

1.2 Study of Research papers

1.3 Load balancing techniques

1.4 Comparison and discussion of load balancing techniques.

3.1 Review of Research papers

We have made survey on various recent research papers relating to Load balancing in Cloud environment. To present our survey in better form, we express the outcome in a tabular form for more readability and understanding. Table 3.1 shows the survey categorized by title, publication, year of publication and more importantly, research gap.

Table 3.1 Review of Research papers

NO.	Paper Title	Pub	Year	Tool	Problem Identified	Solution Proposed
1.	Cloud Task scheduling based on Load Balancing Ant Colony Optimization[3]	IEEE	2011	CloudSim	Difficult to manually assign tasks to computing resources in Clouds.	proposed a Load Balancing Ant Colony Optimization (LBACO) algorithm to find the optimal resource allocation for each task in the dynamic Cloud system.
2.	A green energy-efficient scheduling algorithm using the DVFS technique for Cloud Datacenter[4].	Science Direct	2013	CloudSim	To increasing the utilization and efficiency of hardware equipment, The job scheduler is needed by a Cloud datacenter to arrange resources for executing jobs.	dynamic voltage frequency scaling(DVFS) technique.

3.	Agent Based Dynamic Load Balancing in Cloud Computing[5].	IEEE	2013	Cloudsim	CPU time consumption is high in centralized server based load balancing scheme	Proposed a Agent based Dynamic Load Balancing scheme(ABDLB) which is used to decreasing the CPU time consumption
4.	Towards energy-efficient scheduling for real-time tasks under uncertain Cloud computing environment[6]	Science Direct	2014	CloudSim	Scheduling the Task in uncertain environment(user demand is not perfectly predictable)	Partial Re-Scheduling (PRS) scheduling algorithm(proactive, reactive).
5.	DeMS: A hybrid scheme of task scheduling and load balancing in computing clusters[7].	Elsevier	2015	Cloudsim	High Response time & Job Correlation.	Proposed a DeMS consists of three algorithms, including On-Demand scheduling, Querying and Migrating Task (QMT) and Staged Task Migration (STM).

After reviewed these papers we seen the issues related with load balancing like some node are overloaded and some are underloaded. To solving these problem various techniques are proposed by the authors to addressing a multiple facets' of Cloud viz. like resource utilization, removing a condition of improper workload and there for overall performance will be increases. All the techniques are implemented in Cloudsim environment.

3.2 Different Load balancing Techniques:

Here fig.3.1 shows that the all Load balancing Techniques which is used for balancing the overall workload of all VMs into the system. Job manager having a several VMs, using this list of VM it assign the desire job to the appropriate VM. If not any VM is free at that time the job manager wait for the client request and place that job into queue for the fast processing[8]

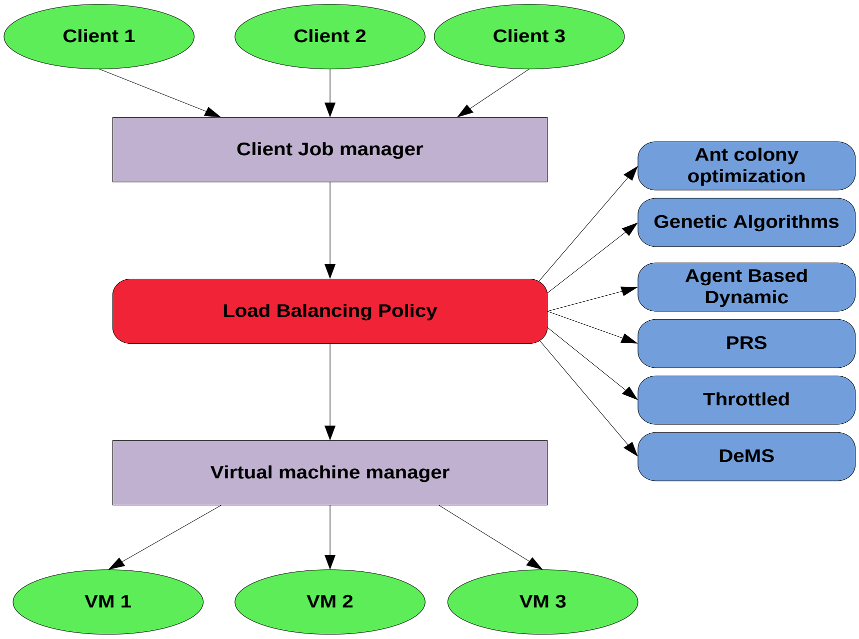


Fig. 3.1 Load balancing techniques in Cloud computing[8]

3.3 Comparison and discussion of load balancing techniques.

Load balancing in Cloud computing is distributing the workload among the all node and transfer the load from heavily node to idle node. It helps to improve response time, migration time, throughput, Resource utilization, scalability and overall performance.

Response time: It is the time interval between sending request and receiving response. This time should be minimum for increasing the performance.

Migration time: It is time taken to migrating the task or transfers the task from one node to another. Minimize the migration time will maximize the performance.

Throughput: This metrics is used to estimate the task, whose execution complete successfully. For increasing the performance, increase the value of this metrics.

Resource utilization: it is used to insure that the utilization of system resources. Better load balancing techniques give the better resource utilization.

Scalability: It determines the ability of the system to accomplish load balancing algorithm with a restricted number of nodes.

Performance: It represents the effectiveness of the system after performing the load balancing algorithm. When the above metrics satisfy optimally then the overall system performance will be increases.

Table 3.2 Comparison of different Load balancing techniques with various metrics.

Metrics/tech	**Nature**	**Environment**	**Response time**	**Migration time**	**Through put**	**Resource utilization**	**Scalability**	**Performance**
LBACO[4]	Dynamic	Decentralized	-	-	-	✓	✓	✓
GA[5]	Dynamic	Centralized	✓	-	✓	-	✓	✓
ABDLB[6]	Dynamic	Centralized	-	-	✓	-	-	✓
PRS[7]	Dynamic	Decentralized	✓	-	✓	✓	✓	✓
Throttled[8]	Dynamic	Decentralized	-	-	✓	✓	✓	✓
DeMS[9]	Dynamic	Centralized	-	✓	-	✓	-	-
Proposed	Dynamic	Centralized	-	✓	✓	✓	-	✓

Chapter – 4

Methodology and Expected Outcome

Topics covered

4.1 Overview of existing work:

Grover et al[6] proposed a system Architecture which consists ‘n’ numbers of clients connected with Cloud service providers via internet and service provider consists VM, managed unit and ‘m’ numbers of shared pool of resources which are considering as servers. In First walk it moves from first server to last server and gathers information from all servers, for making decision for Load Balancing and balances the host’s load on the basis of Standard deviation

4.2 Proposed Method

4.2.1 Proposed Architecture

Fig.4.1, Shows the proposed system Architecture which is divided into three layer, User layer, Resource layer and scheduling layer. It also consist n No. of clients connected with Cloud service providers via internet and Cloud service provider consists virtual machines, Management unit, and m number of shared pool of resources which we are considering as a Hosts. This Architecture also consist one master slave mechanism for Balancing the load and migrate the load from overloaded host to under loaded host. At the shared pool of Hosts, agent complete one cycle in two walks:

- In First walk it moves from first server to last server and gathers information from all servers, for making decision for Load Balancing and
- In Second walk it balances the host’s load on the basis of Standard deviation method

Here, our contribution has been illustrated by green boxes. We divide this architecture in three layers, User layer, Scheduling layer and Resource layer. Also we use the master slave mechanism for migrating the jobs from overloaded server to underloaded server and for managing a new arrival jobs. Agent store the information of all host into that Slave and at the end all the Slave Metadata is stored into the Master.

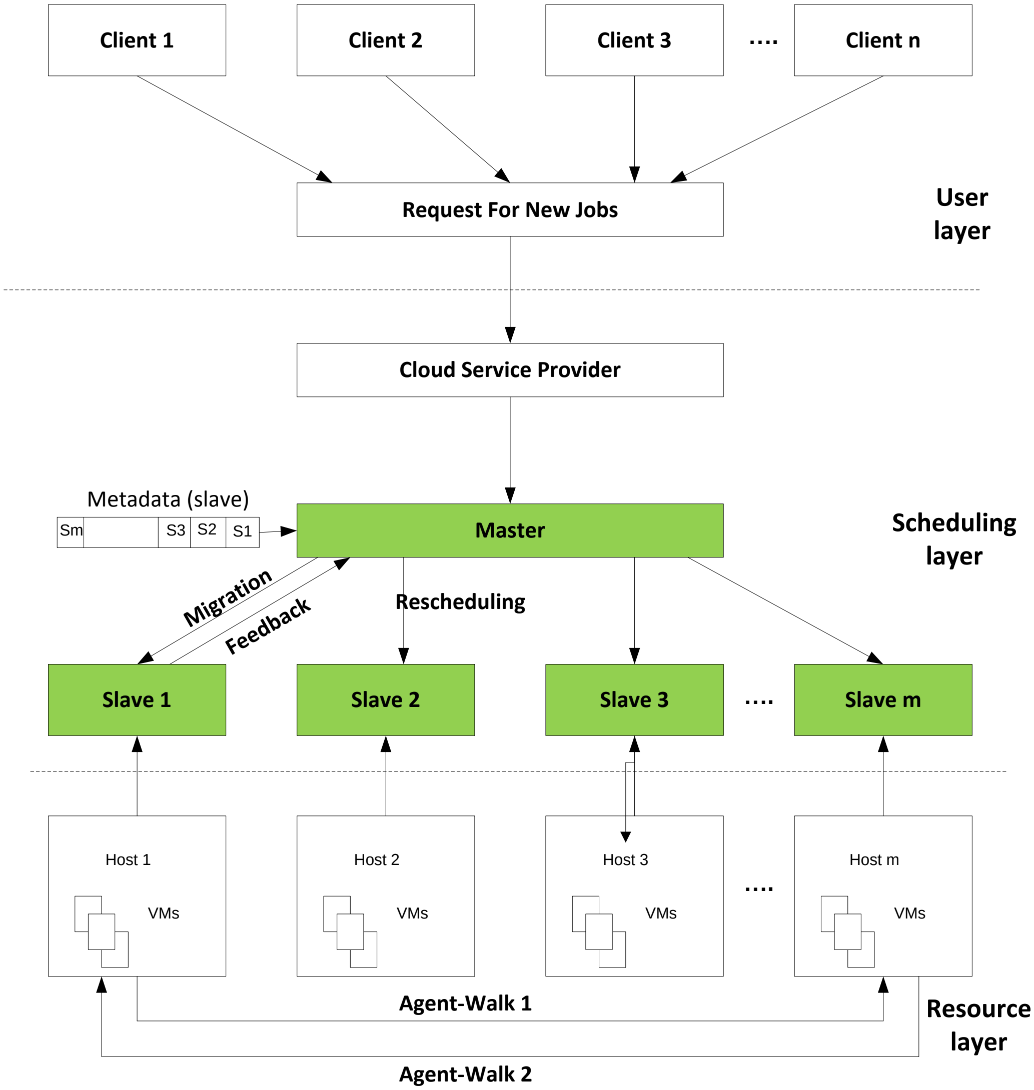


Fig. 4.1. System Architecture

4.2.2 Agent Walk1:

In the Agent walk1 Grover et al[6] describes the model in terms of flow chart. The working of the model can be explained in five steps.

Step1: Agent is activated at any random server and finds number of jobs in queue at that server.

Step2: Agent will repeat this process for all servers of shared pool.

Step3: After that it will calculate AVERAGE.

Step4: On the basis of AVERAGE, it will sense the server's status in terms of overloaded and underloaded.

Step5: Server's status will be decided as follows.

- AVERAGE, then transfer the server's status as overloaded.
- If the number of jobs at i^{th} server is less than the AVERAGE, then transfer the server's status as under loaded.

Here our contribution is shown in fig. 4.2. using shaded portion

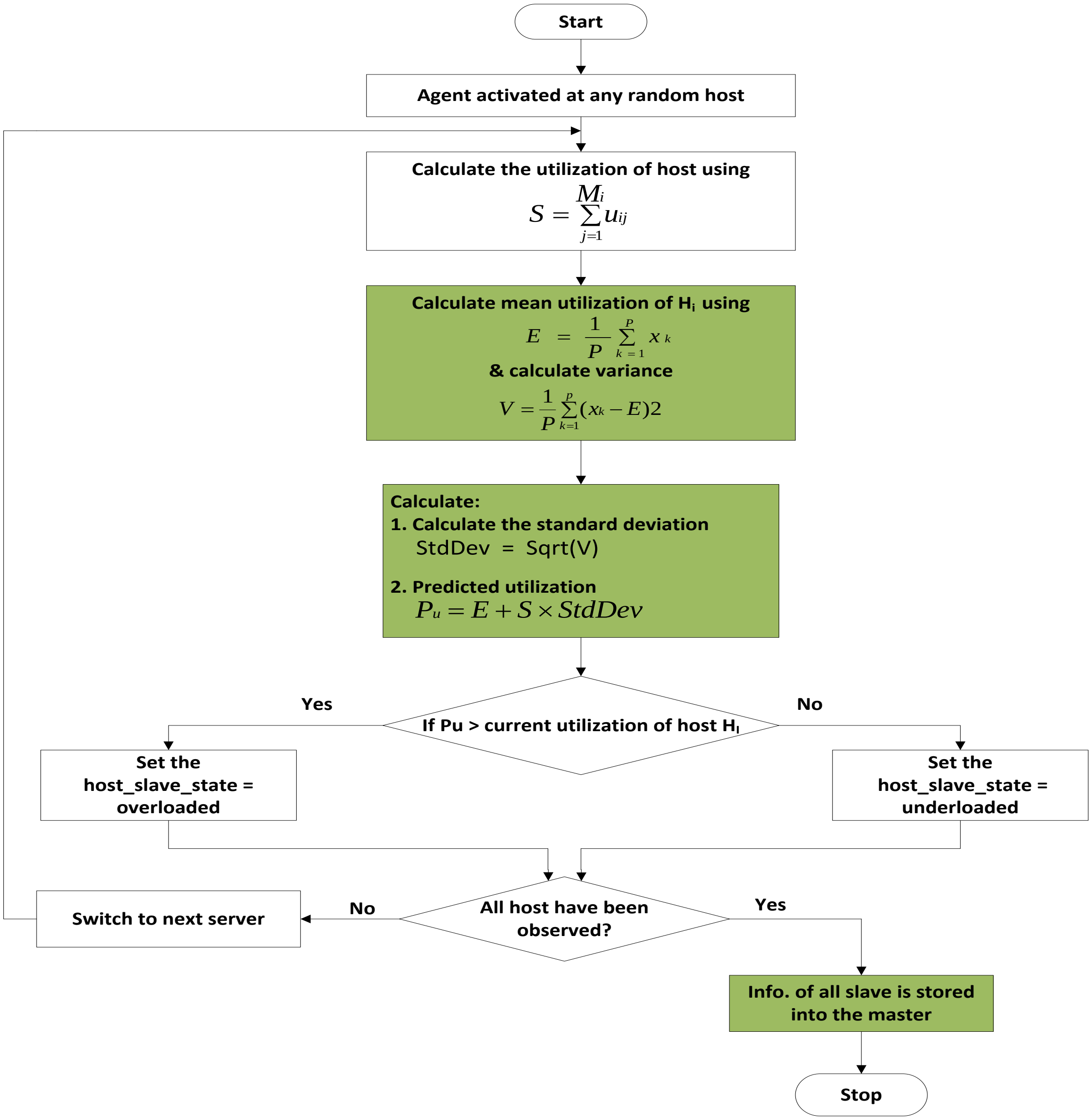

Start
Agent activated at any random host
Calculate the utilization of host using
$S = \sum_{j=1}^{M_i} u_{ij}$
Calculate mean utilization of H_i using
$E = \frac{1}{P} \sum_{k=1}^{P} x_k$
& calculate variance
$V = \frac{1}{P} \sum_{k=1}^{P} (x_k - E)2$
Calculate:
1. Calculate the standard deviation
StdDev = Sqrt(V)
2. Predicted utilization
$P_u = E + S \times StdDev$
If Pu > current utilization of host H_i
Yes
No
Set the host_slave_state = overloaded
Set the host_slave_state = underloaded
All host have been observed?
No
Yes
Switch to next server
Info. of all slave is stored into the master
Stop

Fig. 4.2 First walk of agent from first to last host for gathering load information

At the step1 we recommends to,

- Calculate, utilization Host using

$$S = \sum_{j=1}^{M_i} (u_{ij})$$

- Calculate, mean of utilization of Hi

$$E = \frac{1}{P} \sum_{k=1}^{\mathrm{p}} (X_k)$$

Whereas, X_k is sum of utilization of VMs on host Hi in time frame k.

P is the total time frames.

- Calculate, variance of utilization for host Hi

$$V = \frac{1}{P} \sum_{k=1}^{\mathrm{p}} (X_k - \mathrm{E})^2$$

- The standard deviation equals to the square root of V.

StdDev = $\sqrt{V}$

At the Step3 we contribute, instead of calculating the AVERAGE, we calculate the predicted utilization using standard deviation method[16]. Because of calculating AVERAGE of all job into the queue it's not enough for finding the server is overloaded or underloaded we use this standard deviation method. Understanding of my best calculating standard deviation is better than calculating only AVERAGE of queuing jobs.

PU = E + S * StdDev

At Step4 we use 'PU' for deciding the host is overloaded or Not overloaded. If 'PU' is greater than the current utilization of host than host status is overloaded otherwise under loaded and this information is stored into the slave.

At the last step5 Information of all Slaves is stored into the Master.

4.2.3 Agent Walk2:

In the Agent walk2 Grover et al[6] describes the model in terms of flow chart. The working of this model can be explained below.

Agent will start backtracking from last server to first server for balancing load of servers. At each server it will check the condition. If the status of server is overloaded than transfer the jobs to underloaded server otherwise receive the jobs from overloaded server. Continue this process until the first server.

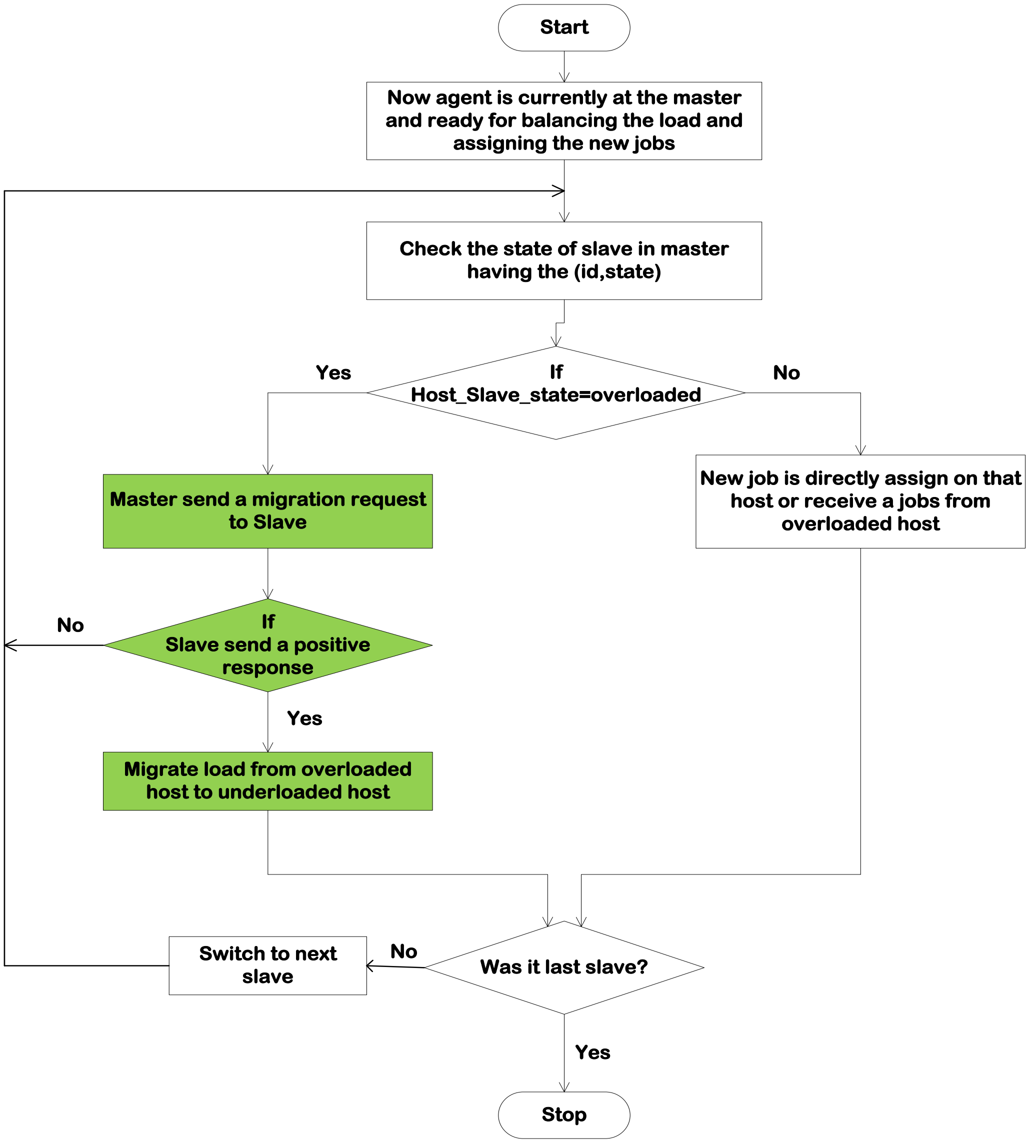

Start
Now agent is currently at the master and ready for balancing the load and assigning the new jobs
Check the state of slave in master having the (id,state)
Yes
If Host_Slave_state=overloaded
No
Master send a migration request to Slave
New job is directly assign on that host or receive a jobs from overloaded host
No
If Slave send a positive response
Yes
Migrate load from overloaded host to underloaded host
Switch to next slave
No
Was it last slave?
Yes
Stop

Fig. 4-3: Second walk for balancing load

Here, we recommend using master slave mechanism. Master have the all information about the slave. Agent is currently at the Master and ready for balancing the load and also assigning the new jobs. First, agent will check the state of slave in master having the id and state. For each host check the condition, if the host_slave_state equals to overloaded then master send a migration request to the slave. If slave send a positive response to the master then migrate jobs to underloaded host. If the host_slave_state equals to underloaded then Receive jobs from "overloaded" host or new job is directly assign on that host. Agent will perform this operation until it reaches at the first host with balancing all host's load is shown in fig. 4.3.

Chapter – 5

Implementation strategy

Topics covered

5.1 About Tool

5.2 Implementation Configuration

5.3 Simulation

5.1 About Tool

As there is several different simulation packages can be used for Cloud simulation, in which of the commonly used simulators: Green Cloud, MDCSim and CloudSim. Comparison of these simulators is as shown in table 5.1.

Table 5.1 Comparison of Simulators

Simulator	Cloudsim	Green Cloud	MDCSim
Platform	-	NS2	CSIM
Language	Java	C++/OTcl	C++/Java
Availability	Open Source	Open Source	Commercial
Graphical Support	Limited (Cloud Analyst)	Limited (Network animator)	None
Application models	Computation, Data transfer	Computation, Data transfer	Computation

After exploring CloudSim tool we found that due to its platform independent and open source nature, we can simulate our proposed technique easily. So, we are going to use CloudSim tool for the implementation.

CloudSim

We are going to use CloudSim toolkit for the implementation. Because our proposed approach is resource allocation mechanism and VM consolidation approach which have been implemented in CloudSim toolkit.

Figure 5.1 shows the CloudSim architecture and its components. For modeling and simulation of virtualized Cloud based data center environments support in simulation layer adding committed management interfaces for VMs, memory, storage, and bandwidth.

CloudSim is an extensible simulation toolkit that allows modeling and simulation of Cloud computing systems and provides the environments for application provisioning. It supports the system and behavior modeling of components such as data centers, virtual machines (VMs) and resource provisioning policies of Cloud system.[15]

Also Support energy-aware computational resources and dynamic insertion of simulation elements stop and resume of simulation. CloudSim toolkit also support for user-defined policies for allocation of hosts to virtual machines and allocation of host resources to virtual machines.

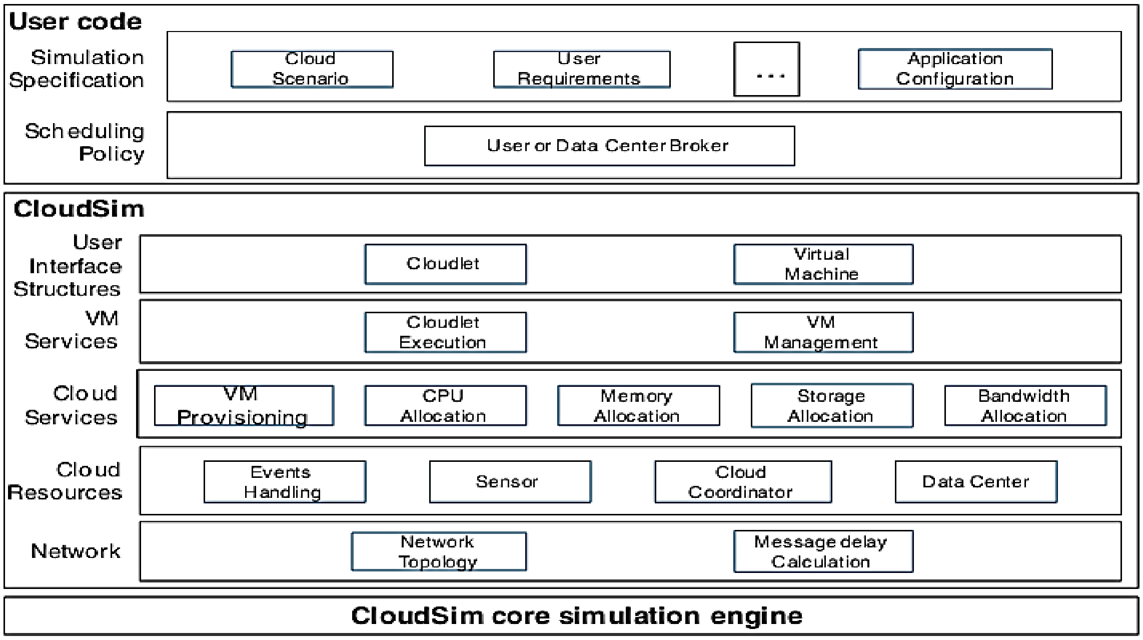


Fig. 5.1 CloudSim Architecture [15]

5.2 Implementation Configuration

- Core i3 processor
- Operating System : Windows 8 (64-bit)
- Simulator : CloudSim 3.0 toolkit
- RAM : 2 GB

Table 5.2 – Specifications of Hosts

	Name	MIPS	RAM (MB)	Bandwidth (Gb per sec)	Core / Processing Elements
Type 1	HpProLiantMl110G4Xeon3040	1860	4096	1	2
Type 2	HpProLiantMl110G5Xeon3075	2660	4096	1	2

Table 5.1 shows the specification of Host in Cloudsim. There are two types of host created in Cloudsim having different values of MIPS and same RAM with 2 core processing Element.

Table 5.3 – Specifications of Virtual Machines

	MIPS	Core / Processing Elements	RAM (MB)	Bandwidth (Mb per sec)
Type 1	2500	1	870	100
Type 2	2000	1	1740	100
Type 3	1000	1	1740	100
Type 4	500	1	613	100

Table 5.2 shows the Specification of virtual machines in Cloudsim. There are four types of VMs Created in Cloudsim having different values of MIPS and RAM with single processing element.

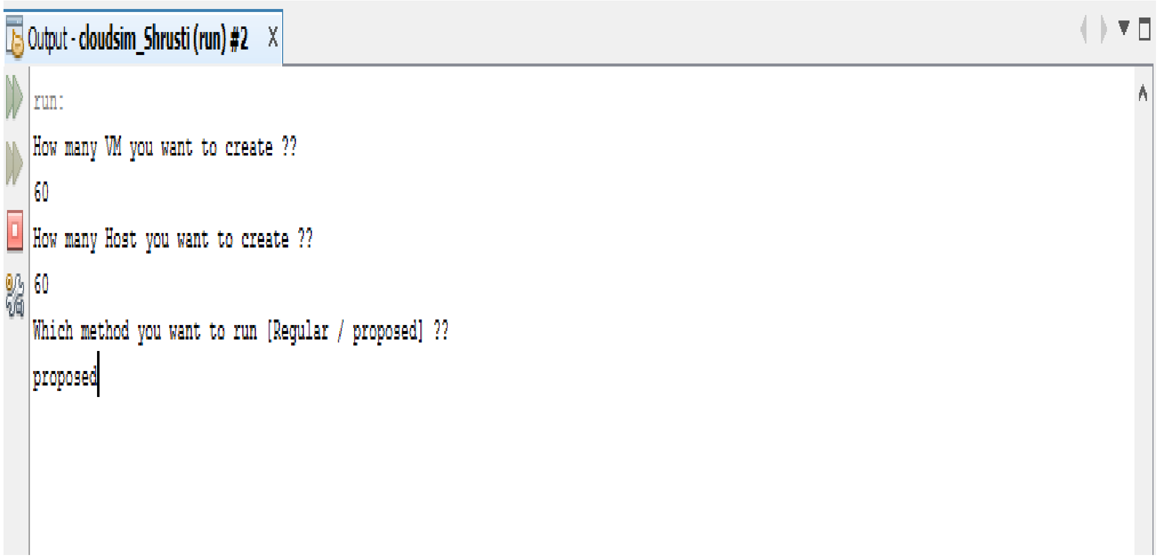
5.3 Simulation Result

In this section Result shows the Performance of proposed method as compared to Existing Load Balancing Technique based on number of parameter

like CPU time, Migration time, Number of migrated Host, No. of host to be shut down for Resource utilization and energy efficiency.

[1] Input VMs and host

Fig. 5.2 shows the how many VMs and Host you want to create.

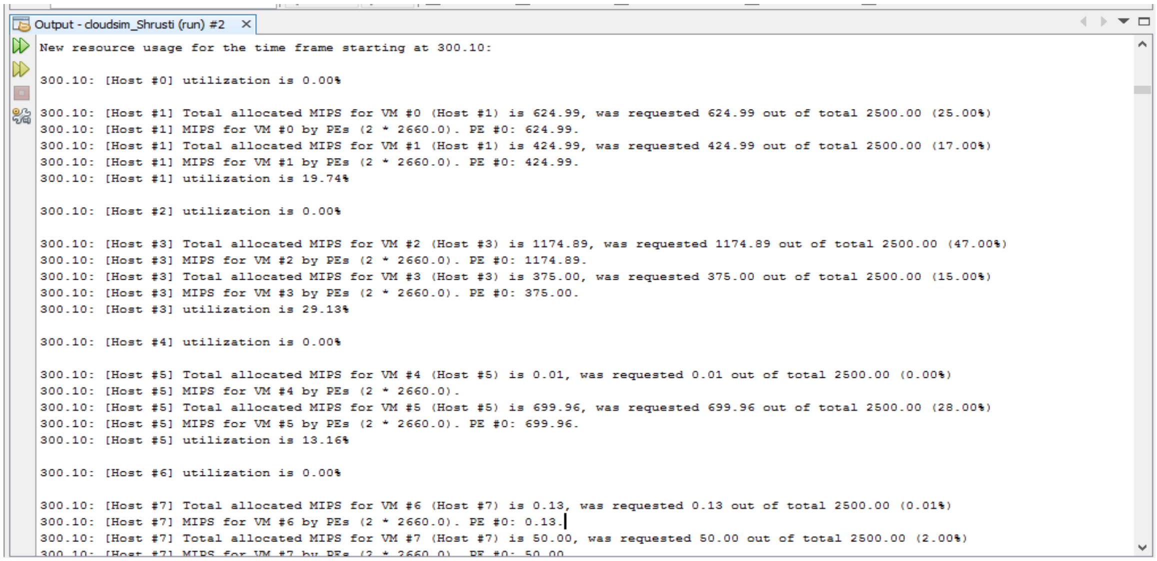


```
run:
How many VM you want to create ??
60
How many Host you want to create ??
60
Which method you want to run [Regular / proposed] ??
proposed
```

Fig. 5.2 Input For Host and VMs.

[2] Utilization Of All VMs and Host

Fig. 5.3 shows the Utilization of all VMs and Host and also shows the requested MIPS by VM form the total capacity of that Host.

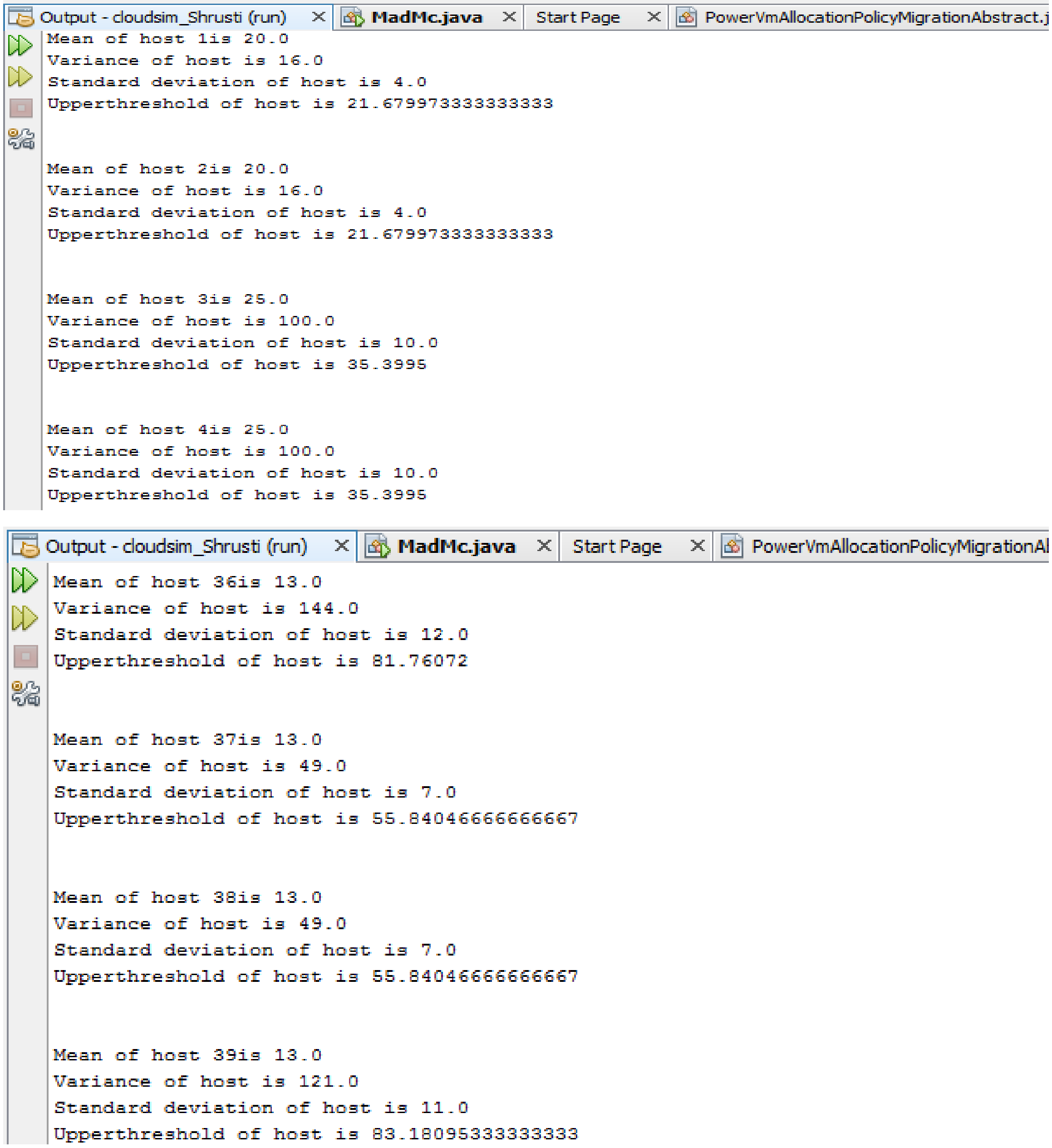


```
New resource usage for the time frame starting at 300.10:

300.10: [Host #0] utilization is 0.00%

300.10: [Host #1] Total allocated MIPS for VM #0 (Host #1) is 624.99, was requested 624.99 out of total 2500.00 (25.00%)
300.10: [Host #1] MIPS for VM #0 by PEs (2 * 2660.0). PE #0: 624.99.
300.10: [Host #1] Total allocated MIPS for VM #1 (Host #1) is 424.99, was requested 424.99 out of total 2500.00 (17.00%)
300.10: [Host #1] MIPS for VM #1 by PEs (2 * 2660.0). PE #0: 424.99.
300.10: [Host #1] utilization is 19.74%

300.10: [Host #2] utilization is 0.00%

300.10: [Host #3] Total allocated MIPS for VM #2 (Host #3) is 1174.89, was requested 1174.89 out of total 2500.00 (47.00%)
300.10: [Host #3] MIPS for VM #2 by PEs (2 * 2660.0). PE #0: 1174.89.
300.10: [Host #3] Total allocated MIPS for VM #3 (Host #3) is 375.00, was requested 375.00 out of total 2500.00 (15.00%)
300.10: [Host #3] MIPS for VM #3 by PEs (2 * 2660.0). PE #0: 375.00.
300.10: [Host #3] utilization is 29.13%

300.10: [Host #4] utilization is 0.00%

300.10: [Host #5] Total allocated MIPS for VM #4 (Host #5) is 0.01, was requested 0.01 out of total 2500.00 (0.00%)
300.10: [Host #5] MIPS for VM #4 by PEs (2 * 2660.0).
300.10: [Host #5] Total allocated MIPS for VM #5 (Host #5) is 699.96, was requested 699.96 out of total 2500.00 (28.00%)
300.10: [Host #5] MIPS for VM #5 by PEs (2 * 2660.0). PE #0: 699.96.
300.10: [Host #5] utilization is 13.16%

300.10: [Host #6] utilization is 0.00%

300.10: [Host #7] Total allocated MIPS for VM #6 (Host #7) is 0.13, was requested 0.13 out of total 2500.00 (0.01%)
300.10: [Host #7] MIPS for VM #6 by PEs (2 * 2660.0). PE #0: 0.13.
300.10: [Host #7] Total allocated MIPS for VM #7 (Host #7) is 50.00, was requested 50.00 out of total 2500.00 (2.00%)
300.10: [Host #7] MIPS for VM #7 by PEs (2 * 2660.0). PE #0: 50.00.
```

Fig. 5.3 Utilization Host and VMs.

[3] Mean utilization, Variance, Standard deviation, and Upper Threshold Of Host.

Fig. 5.4 shows the Mean utilization of all Host, Also calculating the variance and standard deviation, finally calculate the upper threshold or also we can say that predicted utilization for the host for finding the over utilized node.


Output - cloudsim_Shrusti (run)
MadMc.java
Start Page
PowerVmAllocationPolicyMigrationAbstract.j

```
Mean of host 1is 20.0
Variance of host is 16.0
Standard deviation of host is 4.0
Upperthreshold of host is 21.679973333333333


Mean of host 2is 20.0
Variance of host is 16.0
Standard deviation of host is 4.0
Upperthreshold of host is 21.679973333333333


Mean of host 3is 25.0
Variance of host is 100.0
Standard deviation of host is 10.0
Upperthreshold of host is 35.3995


Mean of host 4is 25.0
Variance of host is 100.0
Standard deviation of host is 10.0
Upperthreshold of host is 35.3995
```

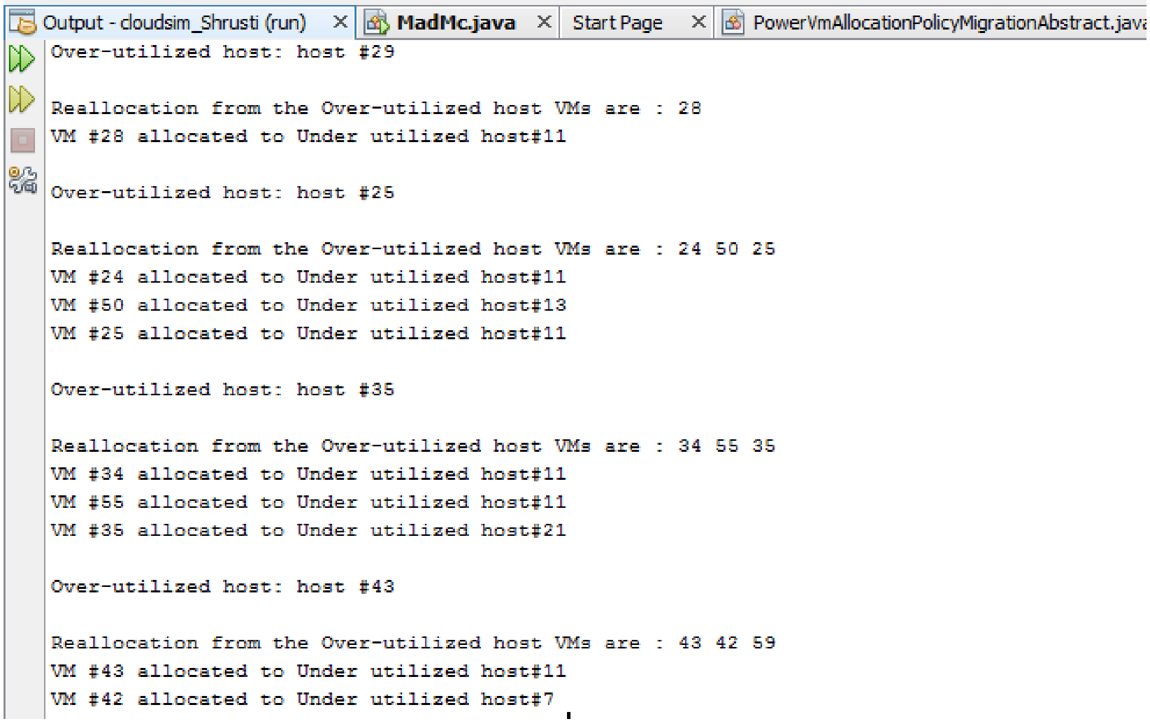
Output - cloudsim_Shrusti (run)
MadMc.java
Start Page
PowerVmAllocationPolicyMigrationAl

```
Mean of host 36is 13.0
Variance of host is 144.0
Standard deviation of host is 12.0
Upperthreshold of host is 81.76072


Mean of host 37is 13.0
Variance of host is 49.0
Standard deviation of host is 7.0
Upperthreshold of host is 55.84046666666667


Mean of host 38is 13.0
Variance of host is 49.0
Standard deviation of host is 7.0
Upperthreshold of host is 55.84046666666667


Mean of host 39is 13.0
Variance of host is 121.0
Standard deviation of host is 11.0
Upperthreshold of host is 83.18095333333333
```

Fig. 5.4 Mean utilization, Variance, Standard deviation, and Upper Threshold Of Host

[4] Over Utilized Node

Fig. 5.5 shows the Over Utilized node and also generate a migration map for migration of VMs from Over Loaded Host to Under loaded.



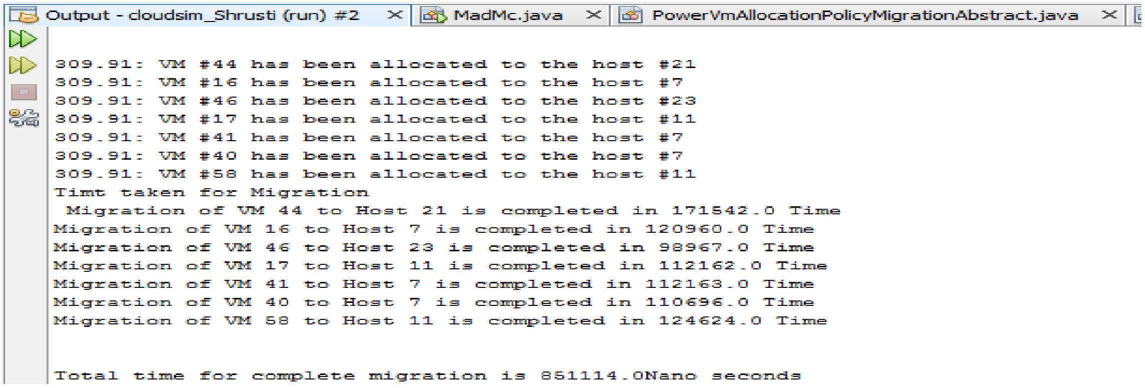
Output - cloudsim_Shrusti (run)
MadMc.java
Start Page
PowerVmAllocationPolicyMigrationAbstract.java

```
Over-utilized host: host #29

Reallocation from the Over-utilized host VMs are : 28
VM #28 allocated to Under utilized host#11

Over-utilized host: host #25

Reallocation from the Over-utilized host VMs are : 24 50 25
VM #24 allocated to Under utilized host#11
VM #50 allocated to Under utilized host#13
VM #25 allocated to Under utilized host#11

Over-utilized host: host #35

Reallocation from the Over-utilized host VMs are : 34 55 35
VM #34 allocated to Under utilized host#11
VM #55 allocated to Under utilized host#11
VM #35 allocated to Under utilized host#21

Over-utilized host: host #43

Reallocation from the Over-utilized host VMs are : 43 42 59
VM #43 allocated to Under utilized host#11
VM #42 allocated to Under utilized host#7
```

Fig. 5.5 Over utilized Host and Migration map

[5] Migration time for VMs

Fig. 5.6 shows the Migration time for migrating a VMs from Overloaded Host to under loaded Host.



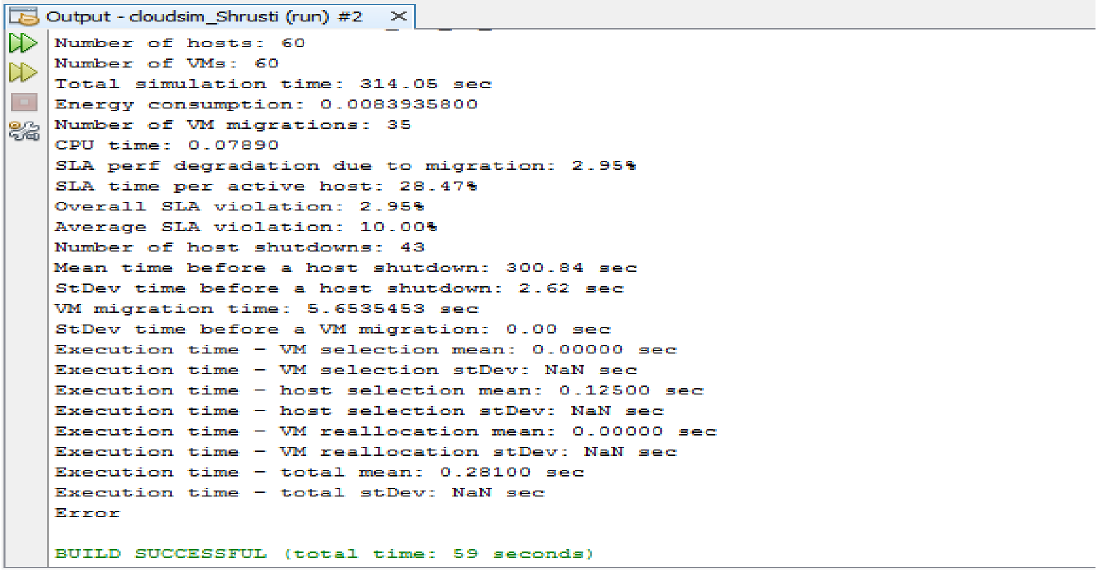
Output - cloudsim_Shrusti (run) #2
MadMc.java
PowerVmAllocationPolicyMigrationAbstract.java

```
309.91: VM #44 has been allocated to the host #21
309.91: VM #16 has been allocated to the host #7
309.91: VM #46 has been allocated to the host #23
309.91: VM #17 has been allocated to the host #11
309.91: VM #41 has been allocated to the host #7
309.91: VM #40 has been allocated to the host #7
309.91: VM #58 has been allocated to the host #11
Timt taken for Migration
 Migration of VM 44 to Host 21 is completed in 171542.0 Time
Migration of VM 16 to Host 7 is completed in 120960.0 Time
Migration of VM 46 to Host 23 is completed in 98967.0 Time
Migration of VM 17 to Host 11 is completed in 112162.0 Time
Migration of VM 41 to Host 7 is completed in 112163.0 Time
Migration of VM 40 to Host 7 is completed in 110696.0 Time
Migration of VM 58 to Host 11 is completed in 124624.0 Time


Total time for complete migration is 851114.0Nano seconds
```

Fig. 5.6 Migration time

[6] Overall Result of Proposed System

Fig. 5.7 shows the Overall result of proposed system that shows the Energy consumption, No of migrated VMs, Migration time for VMs, No. of Host shutdown and CPU time.

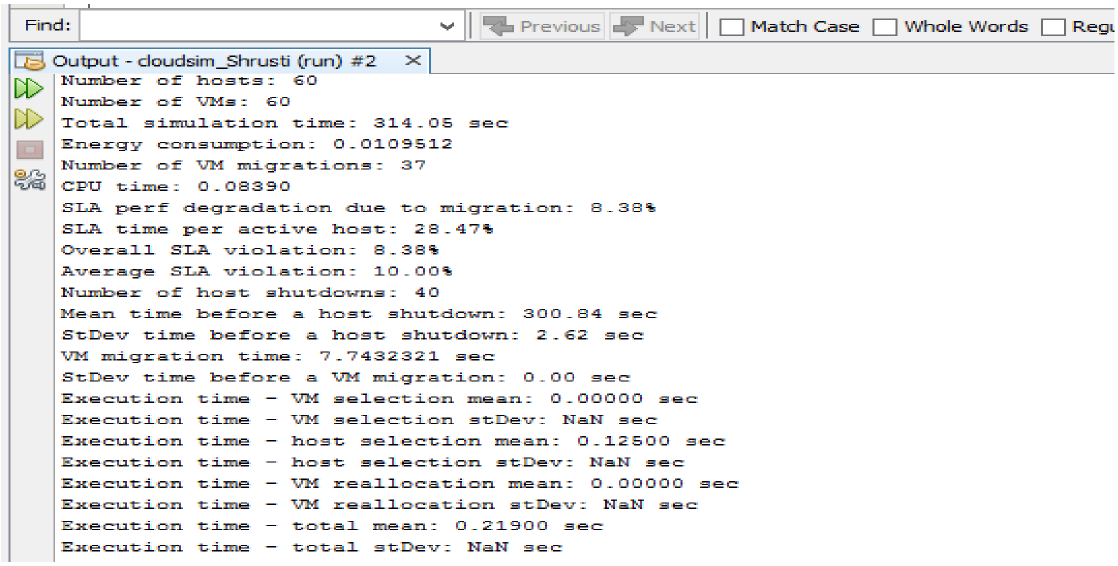


```
Number of hosts: 60
Number of VMs: 60
Total simulation time: 314.05 sec
Energy consumption: 0.0083935800
Number of VM migrations: 35
CPU time: 0.07890
SLA perf degradation due to migration: 2.95%
SLA time per active host: 28.47%
Overall SLA violation: 2.95%
Average SLA violation: 10.00%
Number of host shutdowns: 43
Mean time before a host shutdown: 300.84 sec
StDev time before a host shutdown: 2.62 sec
VM migration time: 5.6535453 sec
StDev time before a VM migration: 0.00 sec
Execution time - VM selection mean: 0.00000 sec
Execution time - VM selection stDev: NaN sec
Execution time - host selection mean: 0.12500 sec
Execution time - host selection stDev: NaN sec
Execution time - VM reallocation mean: 0.00000 sec
Execution time - VM reallocation stDev: NaN sec
Execution time - total mean: 0.28100 sec
Execution time - total stDev: NaN sec
Error

BUILD SUCCESSFUL (total time: 59 seconds)
```

Fig. 5.7 Output (Proposed System)

[7] Overall Result of Existing System

Fig. 5.8 shows the Overall result of Existing system that shows the Energy consumption, No of migrated VMs, Migration time for VMs, No. of Host shutdown and CPU time.



```
Number of hosts: 60
Number of VMs: 60
Total simulation time: 314.05 sec
Energy consumption: 0.0109512
Number of VM migrations: 37
CPU time: 0.08390
SLA perf degradation due to migration: 8.38%
SLA time per active host: 28.47%
Overall SLA violation: 8.38%
Average SLA violation: 10.00%
Number of host shutdowns: 40
Mean time before a host shutdown: 300.84 sec
StDev time before a host shutdown: 2.62 sec
VM migration time: 7.7432321 sec
StDev time before a VM migration: 0.00 sec
Execution time - VM selection mean: 0.00000 sec
Execution time - VM selection stDev: NaN sec
Execution time - host selection mean: 0.12500 sec
Execution time - host selection stDev: NaN sec
Execution time - VM reallocation mean: 0.00000 sec
Execution time - VM reallocation stDev: NaN sec
Execution time - total mean: 0.21900 sec
Execution time - total stDev: NaN sec
```

Fig. 5.8 Output (Existing System)

5.3.1 Simulation Results and Charts

[1] CPU time

Fig. 5.9 shows the result of CPU time shown in Table 5.4 for Same Host and Different VMs on it. The results shows the proposed system's, CPU time is less as compared to Existing method.

Table 5.4 CPU time

	VM	CPU Time(ms)	
		Existing	**Proposed**
Host=60	60	83.90	78.90
	70	115.60	95.90
	80	143.20	115.60
	90	193.78	176.10

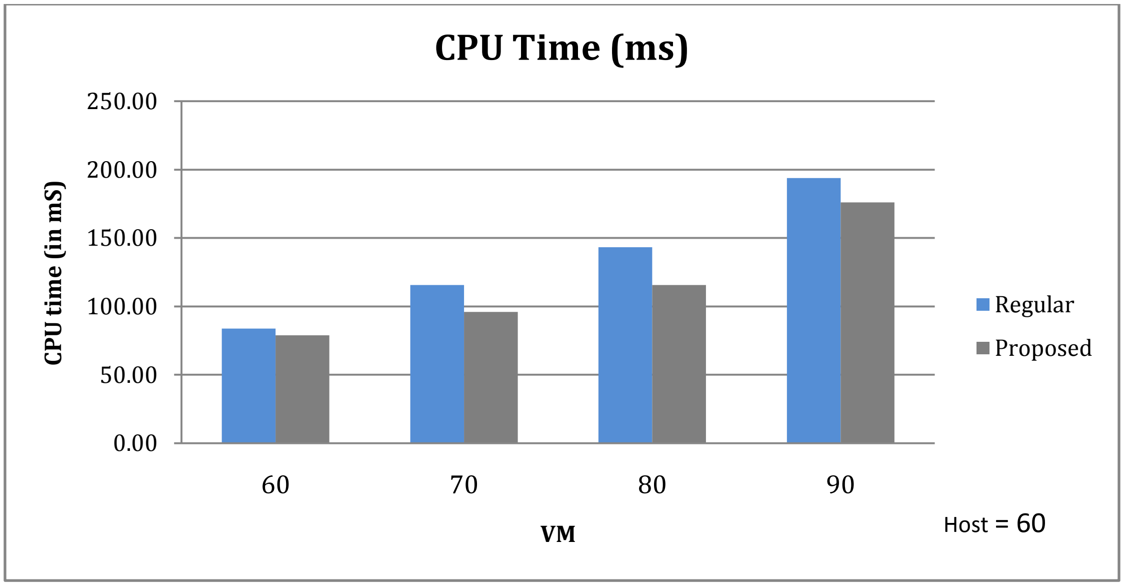


Fig. 5.9 CPU Time chart

[2] Migration VMs

Fig. 5.10 shows, Result shown in table 5.5 for number of VMs Migrated from Overloaded Host to Underloaded Host.

Table 5.5 No. of VMs Migrated

	VM	VMS Migrated	
		Existing	Proposed
Host=60	60	37	35
	70	36	30
	80	33	28
	90	28	21

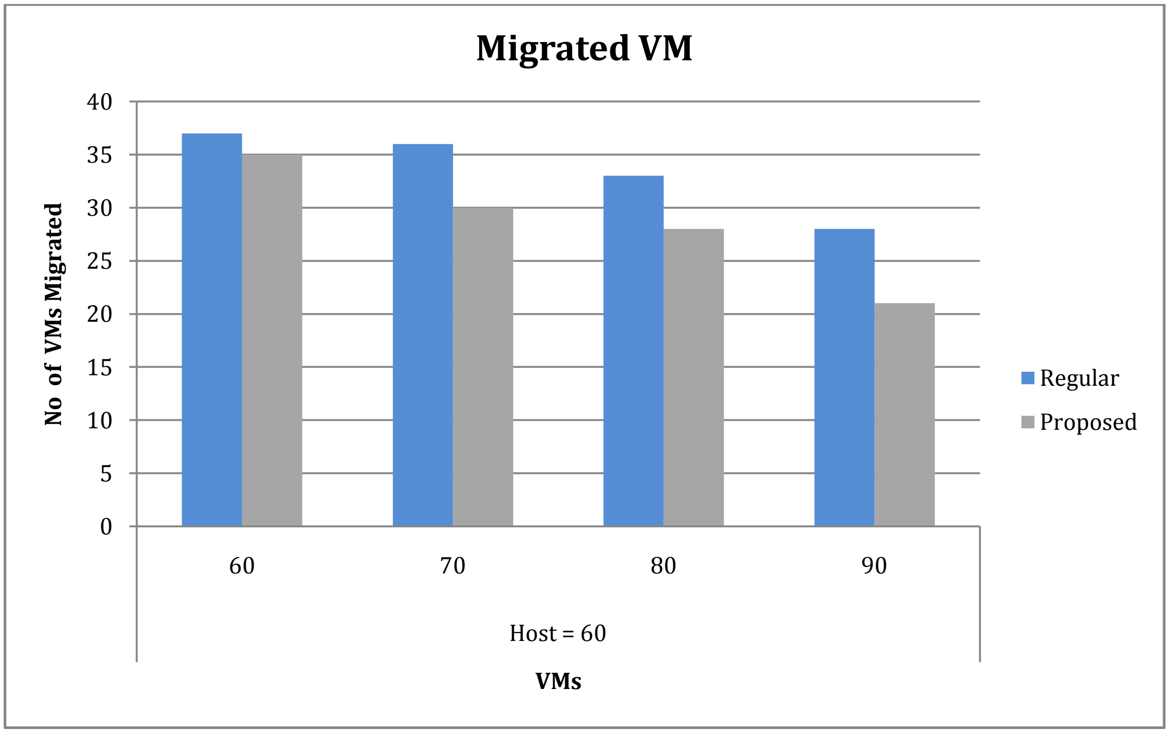


Fig. 5.10 No. of VMs Migrated chart

[3] Migration Time

Fig. 5.11 shows the total migration time for all VMs which is migrated from Overloaded host to underloaded host.

Table 5.6 Migration time

	VM	VMs Migration Time(ms)	
		Existing	Proposed
Host=60	60	7743.2	565.35
	70	844.45	725.64
	80	856.56	796.89
	90	974.84	818.95

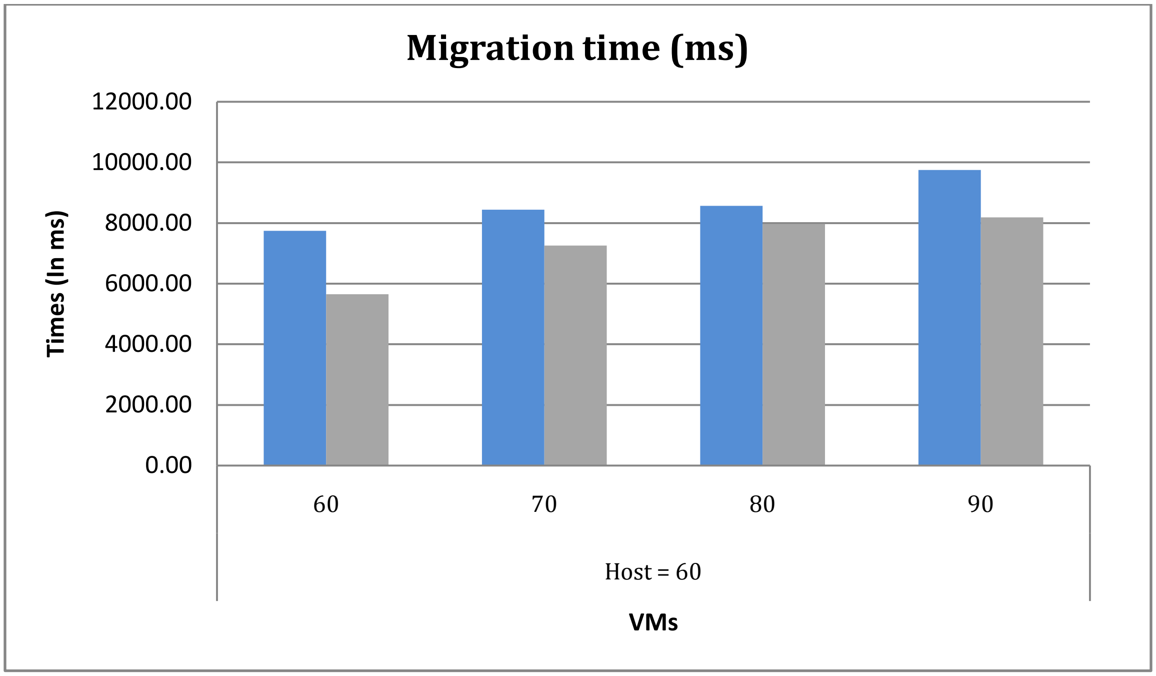


Fig. 5.11 Migration Time Chart

[4] Host shutdown

Fig. 5.12 shows the total Host shut down. When the number of host shutdown increases then resource utilization will increases.

Table 5.7 Total host shutdown

	VM	**Host shut down**	
		Existing	**Proposed**
Host=60	60	40	41
	70	36	38
	80	33	36
	90	25	27

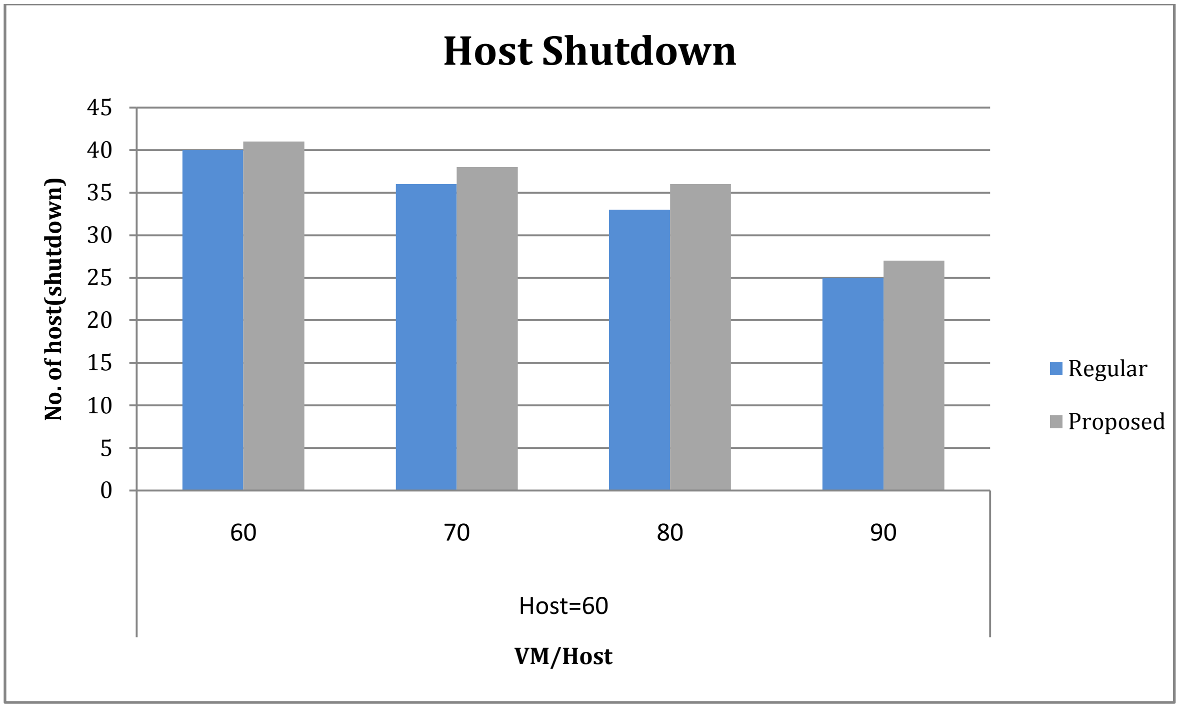


Fig. 5.12 Host shut down Chart

Chapter – 6

Conclusion and Future Work

Topics covered

6.1 Conclusion

6.1 Conclusion

In this post-graduate dissertation, we study the various load balancing schemes and the issues of load balancing in Cloud computing. Improper workload in Cloud environment results into over or under utilization of computing resources and that affects the performance of overall system. It help to achieve the user satisfaction by improving the metrics like, response time, migration time, throughput, resource utilization and performance.

In our proposed work, we modify the agent based dynamic load balancing algorithm by adding the standard deviation method to decide whether the host is overloaded or not. To provide a better load balancing in terms of improved performance, reduced CPU time, increased resource utilization.

We implement this method in Cloudsim toolkit 3.0 and generated results of proposed method are compared with existing load balancing algorithm. Result shows the overall performance of proposed method has been improved as compared to the existing load balancing method.

In future work, one can extend the work by implementing the same on a real world Cloud environment like Xen or other.

References

[1] Mell P, Granc T, “A NIST definition of Cloud computing”, September 2011, http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145Clouddefinition.pdf

[2] Nuaimi KA, Mohamed N, Nuaimi MA And Al-Jaroodi J. “A survey of load balancing in Cloud computing: challenges and algorithms”, Second IEEE Symposium in Network Cloud Computing and Applications (NCCA), 2012 , 137-142.

[3] Sreenivas V, Prathap M And Kemal M. “ Load balancing techniques: Major challenge in Cloud Computing-a systematic review”. International Conference on IEEE in Electronics and Communication Systems (ICECS), 2014 , 1-6.

[4] Li K, Xu G, Zhao G, Dong Y, And Wang D. “Cloud task scheduling based on load balancing ant colony optimization”. Sixth Annual IEEE Symposium in Chinagrid Conference, 2011, 3-9.

[5] Dasgupta K, Mandal B, Dutta P, Mandal JK, And Dam S. “A genetic algorithm (ga) based load balancing strategy for Cloud computing”. Procedia Technology, 2013, 340-347.

[6] Grover J, And Katiyar S. “Agent based dynamic load balancing in Cloud Computing”. International Conference on IEEE in Human Computer Interactions (ICHCI), 2013, 1-6.

[7] Chen H, Zhu X, Guo H, Zhu J, Qin X, And Wu J. “Towards energy-efficient scheduling for real-time tasks under uncertain Cloud computing environment”. In Journal of Systems and Software,2011, 20-35.

[8] Chen H, Zhu, X., Guo, H., Zhu, J., Qin, X., & Wu, J. (2015). Towards energy-efficient scheduling for real-time tasks under uncertain Cloud computing environment. Journal of Systems and Software, 99, 20-35.

[9] Liu Y, Zhang C, Li B, & Niu, j. “DeMS: A hybrid scheme of task scheduling and load balancing in computing clusters”. In Journal of Network and Computer Applications, 2015.

[10] Alakeel M. "A Guide to dynamic Load balancing in Distributed Computer Systems", International Journal of Computer Science and Network Security, 2010, 10, 153-160.

[11] Mata-Toledo R and Gupta P, "Green datacenter: how green can we performance", In Journal of Technology Research, Academic and Business Research Institute, 2010, 2, 1-8.

[12] Lee R. and Jeng B, "Load-balancing tactics in Cloud". International Conference on IEEE on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011, 447-454.

[13] HZ M, Veeravalli B. "Requirement-aware strategies for scheduling real-timedivisible loads on clusters". In J. Parallel Distrib. Comput. Sinha, P. K.. Distributed operating systems: concepts and design, 73, 1083–109, 1998.

[14] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010

[15] Calheiros R N, Ranjan R, Beloglazov A, Rose C A F D, and Buyya R, "CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, Wiley Press, New York, USA, 2010.

[16] Z Cao and dong S, " Dynamic VM consolidation for energy-aware and SLA violation reduction in Cloud computing. "13th International Conference in IEEE on Parallel and Distributed Computing, 2012, 363-369.

Appendix A

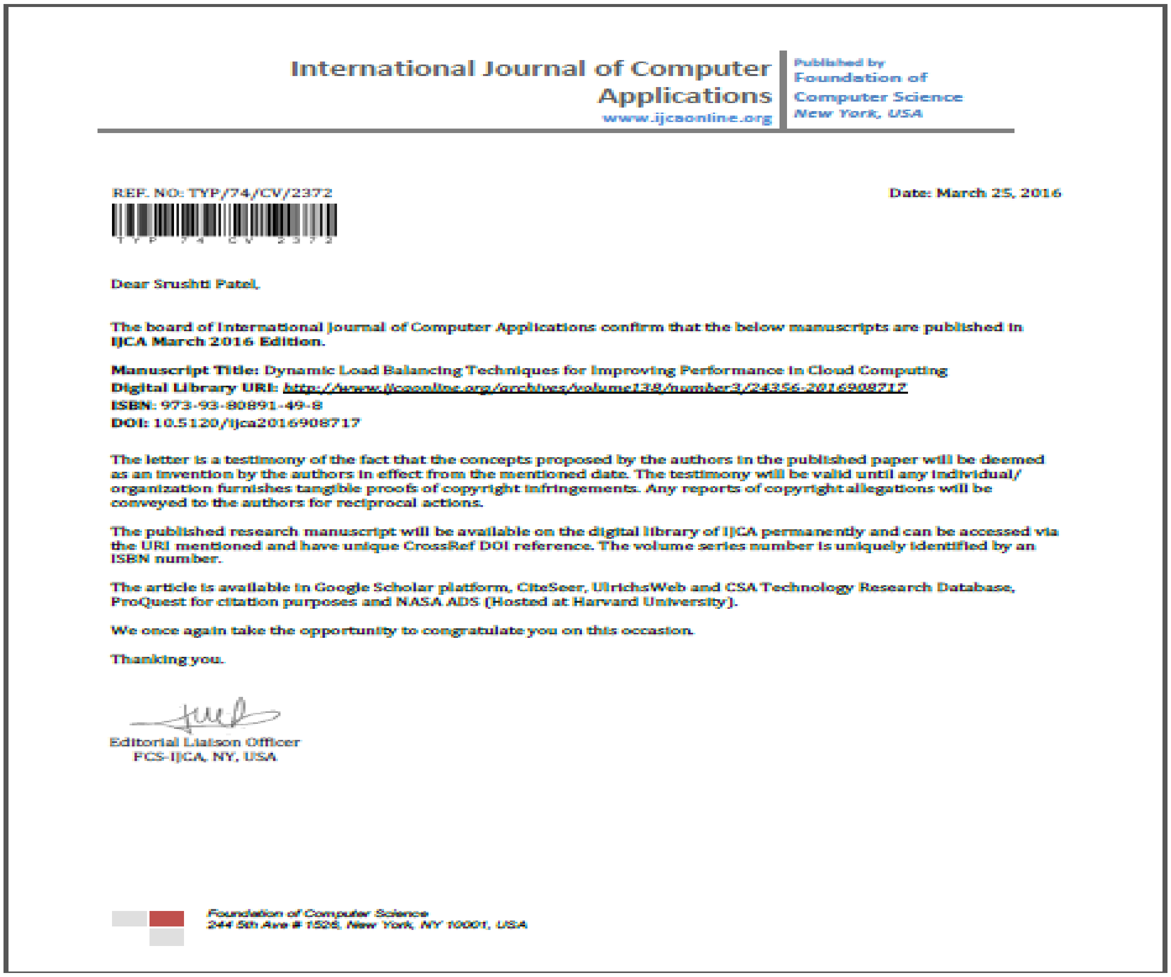
Abbreviations

ABDLB:	Agent based Dynamic Load Balancing
CPU:	Control processing Unit
DVFS:	Dynamic voltage frequency Scaling
GA:	Genetic Algorithm
IAAS:	Infrastructure as a Service
LBACO:	Load Balancing Ant Colony Optimization
MIPS:	Millions of instructions per second
PAAS:	Platform as a Service
PRS:	Partial Re-Scheduling
PU:	Predicted Utilization
QMT:	Query migrating task
RAM:	Read access memory
SAAS:	Software as a Service
STM:	Staged task migration
VMs:	Virtual machines

Appendix B

Paper Publication Certificate

Srushti Patel, Hiren Patel, Nimisha Patel, The Paper titled "**Dynamic Load Balancing Techniques For Improving Performance In Cloud Computing**" in International Journal of Computer Applications (IJCA), March 18, 2016. (http://www.ijcaonline.org/archives/volume138/number3/24356-2016908717)



International Journal of Computer Applications
www.ijcaonline.org

Published by
Foundation of
Computer Science
New York, USA

REF. NO: TYP/74/CV/2372

Date: March 25, 2016

Dear Srushti Patel,

The board of International Journal of Computer Applications confirm that the below manuscripts are published in **IJCA March 2016 Edition.**

Manuscript Title: Dynamic Load Balancing Techniques for Improving Performance in Cloud Computing
Digital Library URI: http://www.ijcaonline.org/archives/volume138/number3/24356-2016908717
ISBN: 973-93-80891-49-8
DOI: 10.5120/ijca2016908717

The letter is a testimony of the fact that the concepts proposed by the authors in the published paper will be deemed as an invention by the authors in effect from the mentioned date. The testimony will be valid until any individual/ organization furnishes tangible proofs of copyright infringements. Any reports of copyright allegations will be conveyed to the authors for reciprocal actions.

The published research manuscript will be available on the digital library of IJCA permanently and can be accessed via the URI mentioned and have unique CrossRef DOI reference. The volume series number is uniquely identified by an ISBN number.

The article is available in Google Scholar platform, CiteSeer, UlrichsWeb and CSA Technology Research Database, ProQuest for citation purposes and NASA ADS (Hosted at Harvard University).

We once again take the opportunity to congratulate you on this occasion.

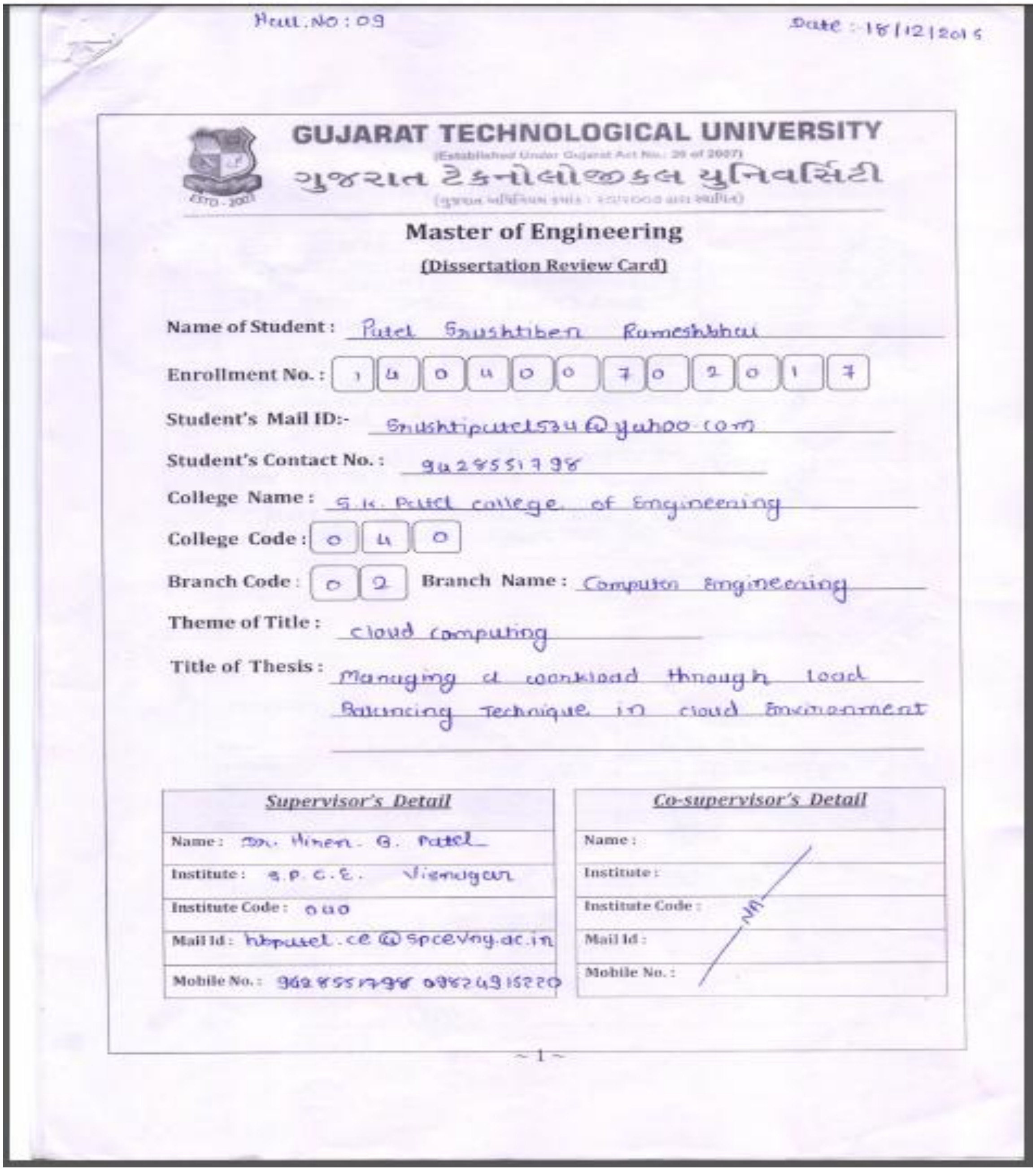
Thanking you.

Editorial Liaison Officer
FCS-IJCA, NY, USA

Foundation of Computer Science
244 5th Ave # 1526, New York, NY 10001, USA

Appendix C

Review Comment Cards



Hall No : 09 Date : 18/12/2015

GUJARAT TECHNOLOGICAL UNIVERSITY

(Established Under Gujarat Act No.: 20 of 2007)

ગુજરાત ટેકનોલોજીકલ યુનિવર્સિટી

[illegible]

Master of Engineering

(Dissertation Review Card)

Name of Student : Patel Srushtiben Rameshbhai

Enrollment No. : 1 4 0 4 0 0 7 0 2 0 1 7

Student's Mail ID:- srushtipatel534@yahoo.com

Student's Contact No. : 9428551798

College Name : S.K. Patel college of Engineering

College Code : 0 4 0

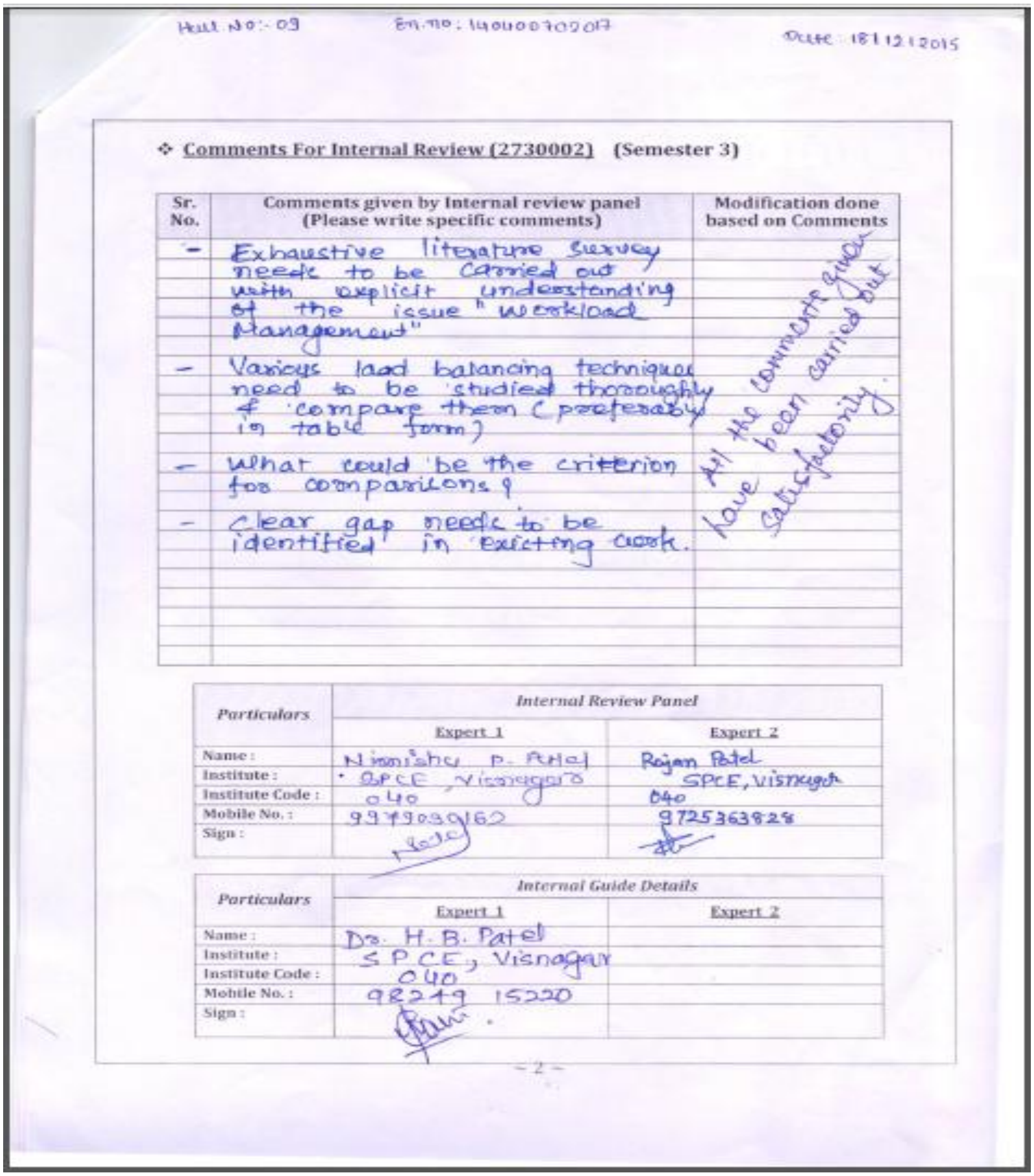
Branch Code : 0 2 Branch Name : Computer Engineering

Theme of Title : cloud computing

Title of Thesis : Managing a workload through load Balancing Technique in cloud Environment

Supervisor's Detail	Co-supervisor's Detail
Name : Dr. Hiren. B. Patel	Name :
Institute : S.P.C.E. Visnagar	Institute :
Institute Code : 040	Institute Code : NA
Mail Id : hbpatel.ce@spcevng.ac.in	Mail Id :
Mobile No. : 9428551798 09824915220	Mobile No. :

~ 1 ~



Roll No:- 09 En.no: 140400702017 Date: 18/12/2015

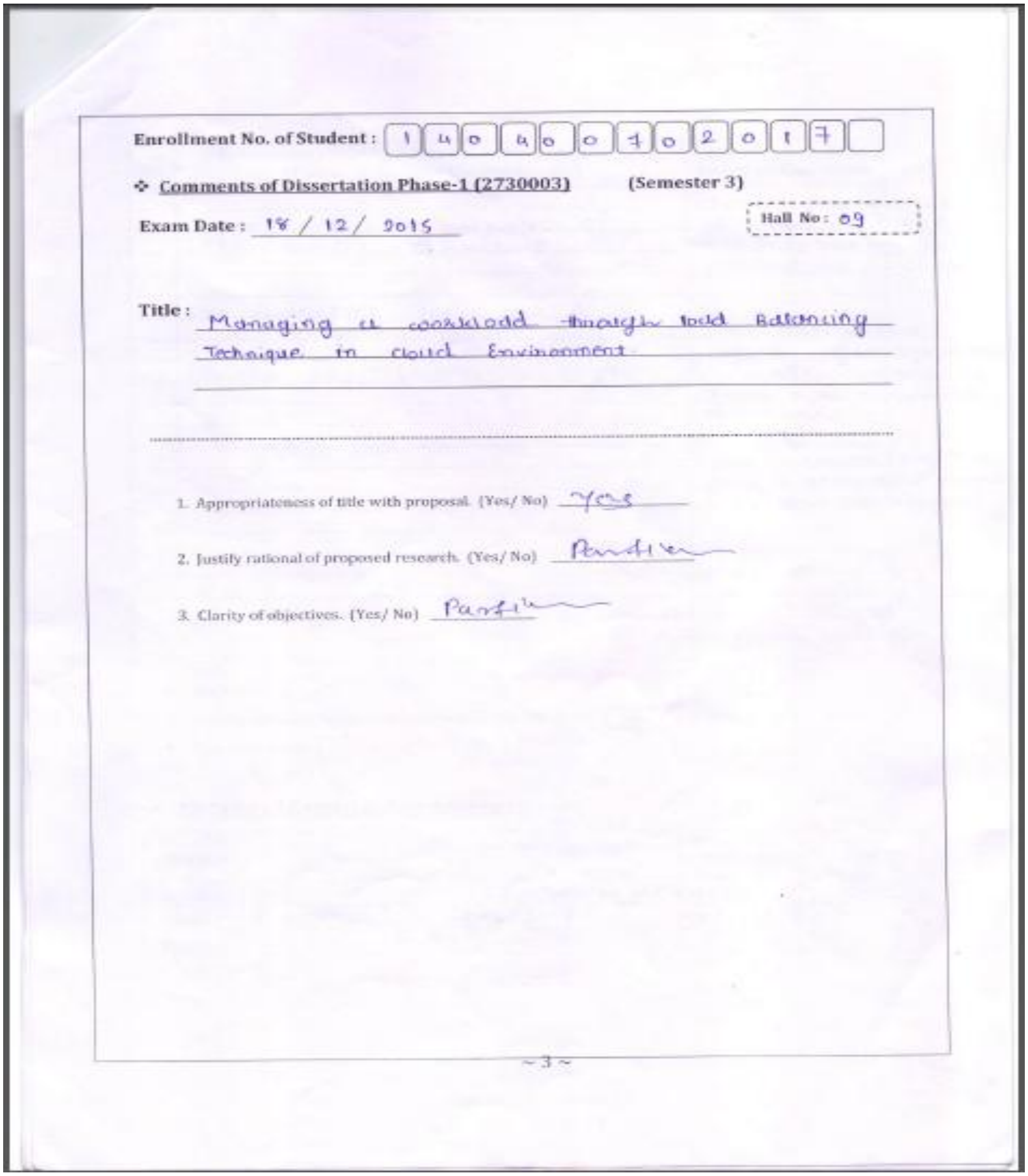
❖ Comments For Internal Review (2730002) (Semester 3)

Sr. No.	Comments given by Internal review panel (Please write specific comments)	Modification done based on Comments
-	Exhaustive literature survey needs to be carried out with explicit understanding of the issue "workload Management"	All the comments given have been carried out satisfactorily.
-	Various load balancing techniques need to be studied thoroughly & compare them (preferably in table form)	
-	What could be the criterion for comparisons?	
-	Clear gap needs to be identified in existing work.	

Particulars	Internal Review Panel	
	Expert 1	Expert 2
Name :	Nimisha P. Patel	Rajan Patel
Institute :	SPCE, Visnagar	SPCE, Visnagar
Institute Code :	040	040
Mobile No. :	9979030162	9725363828
Sign :		

Particulars	Internal Guide Details	
	Expert 1	Expert 2
Name :	Dr. H. B. Patel	
Institute :	SPCE, Visnagar	
Institute Code :	040	
Mobile No. :	98249 15220	
Sign :		

- 2 -



Enrollment No. of Student : 1 4 0 4 0 0 1 0 2 0 1 7

❖ **Comments of Dissertation Phase-1 (2730003)** **(Semester 3)**

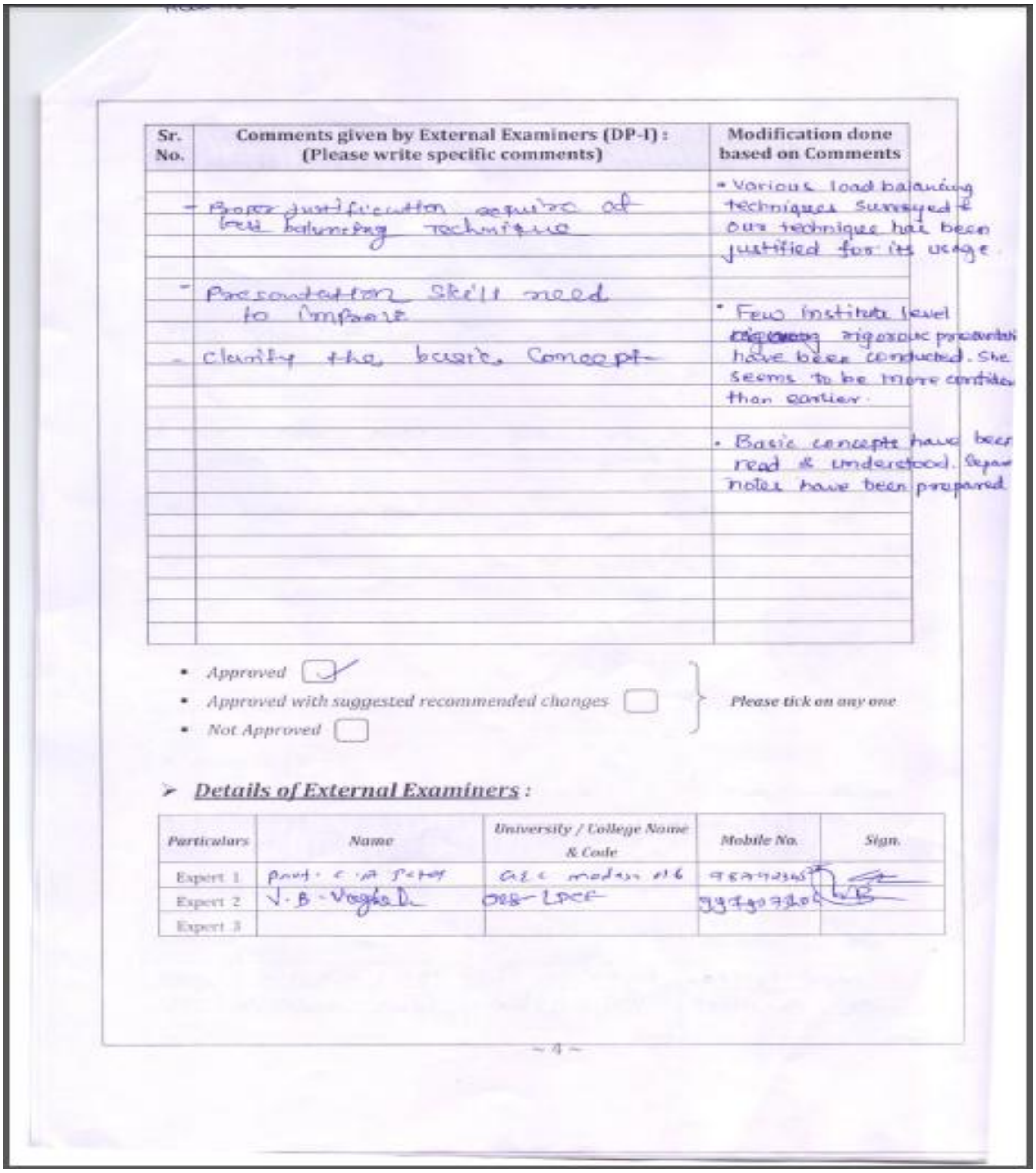
Exam Date : 18 / 12 / 2015

Hall No : 09

Title : Managing a workload through load Balancing Technique in cloud Environment

1. Appropriateness of title with proposal. (Yes/ No) Yes
2. Justify rational of proposed research. (Yes/ No) Partial
3. Clarity of objectives. (Yes/ No) Partial

~ 3 ~



Sr. No.	Comments given by External Examiners (DP-I) : (Please write specific comments)	Modification done based on Comments
	- Proper justification require of load balancing Technique	• Various load balancing techniques surveyed & our technique has been justified for its usage.
	- Presentation Skill need to improve	• Few institute level ~~[illegible]~~ rigorous presentation have been conducted. She seems to be more confident than earlier.
	- clarify the basic Concept	• Basic concepts have been read & understood. Separate notes have been prepared

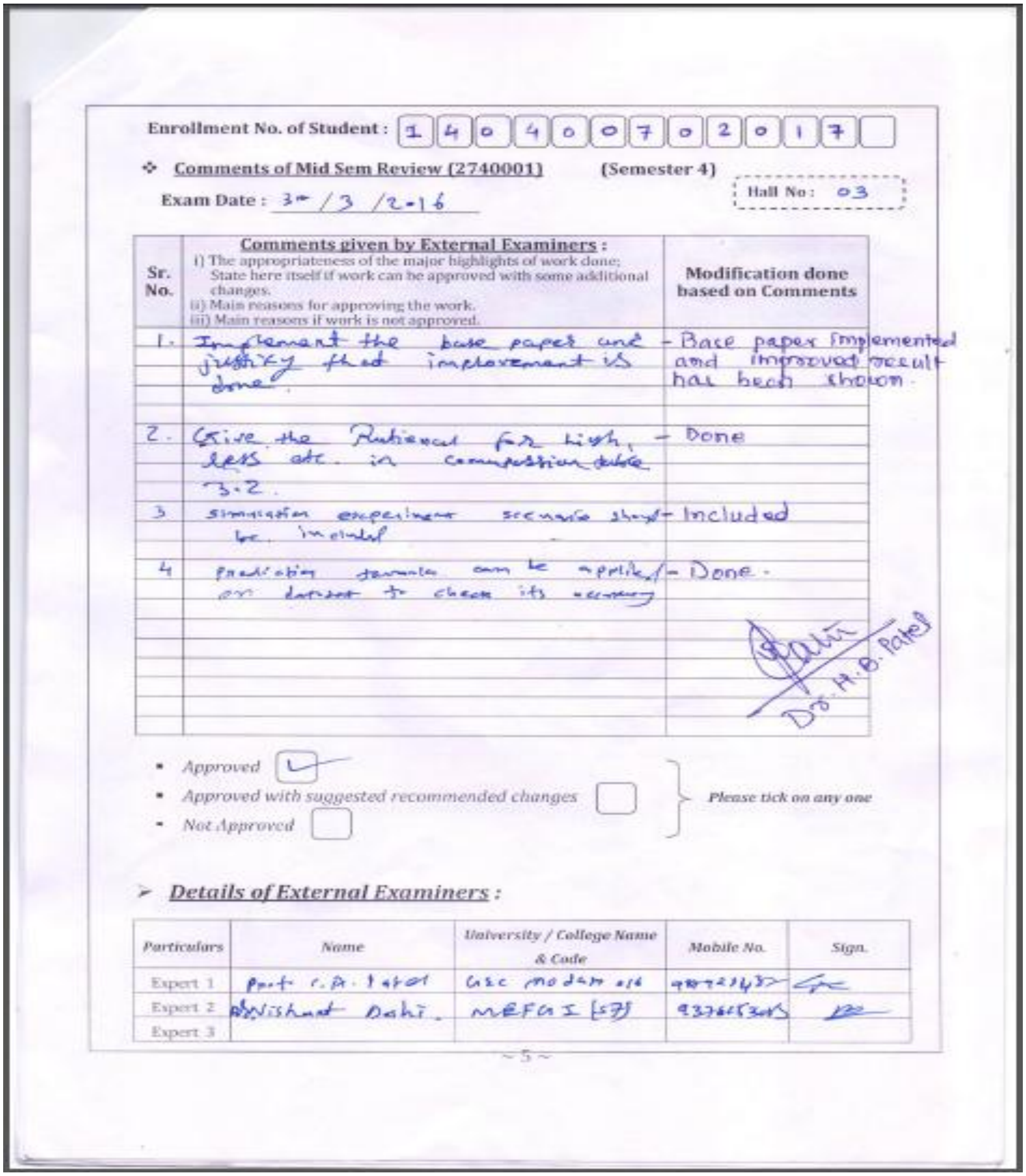
- *Approved* ☑
- *Approved with suggested recommended changes* □
- *Not Approved* □

Please tick on any one

➢ ***Details of External Examiners*** **:**

Particulars	*Name*	*University / College Name & Code*	*Mobile No.*	*Sign.*
Expert 1	Prof. C. A. Patel	GEC Modasa 016	9879423457	
Expert 2	V. B. Vaghela	028- LDCE	9979079209	VB
Expert 3				

~ 4 ~



Enrollment No. of Student : 1 4 0 4 0 0 7 0 2 0 1 7

- **Comments of Mid Sem Review (2740001)** (Semester 4)

Exam Date : 3rd / 3 / 2016

Hall No : 03

Sr. No.	**Comments given by External Examiners :** i) The appropriateness of the major highlights of work done; State here itself if work can be approved with some additional changes. ii) Main reasons for approving the work. iii) Main reasons if work is not approved.	**Modification done based on Comments**
1.	Implement the base paper and justify that improvement is done.	- Base paper implemented and improved result has been shown.
2.	Give the Rationale for high, less etc. in comparison table 3.2.	- Done
3	Simulation experiment scenario should be included	- Included
4	Prediction formula can be applied on dataset to check its accuracy	- Done.

Dr. H. B. Patel

- Approved ☑
- Approved with suggested recommended changes □
- Not Approved □

} *Please tick on any one*

➢ *Details of External Examiners :*

Particulars	*Name*	*University / College Name & Code*	*Mobile No.*	*Sign.*
Expert 1	Prof. C. R. Patel	GEC Modasa 016	[illegible]	
Expert 2	Nishant Doshi.	MEFGI [57]	9376653045	
Expert 3				

~5~

Compliance Report

Details of Comments Given by Internal Review Panel

1. Exhaustive Literature Survey needs to be carried out with explicit understanding of the issues “ Workload Management”.

   I have carried out exhaustive literature survey

2. Various load balancing technique need to be studied thoroughly and compare them (preferably in table form).

   We studied various techniques for load balancing n also compared them in table 3.2
3. What could be the criterion for comparisons?

   This criterion also shown in table 3.2
4. Clear gap needs to be identified in “Existing work”.

   Identified the gap.

Details of Comments Given by Dissertation Phase – I

1. Proper Justification require of load balancing technique.

   I have justify this technique.
2. Presentation skill need to be improved

   I have improved presentation skill.
3. Clarify the basic Concept

   Basic concept has been done

Details of Comments Given by Mid Semester Review

1. Implement the base paper and justify that implement is done

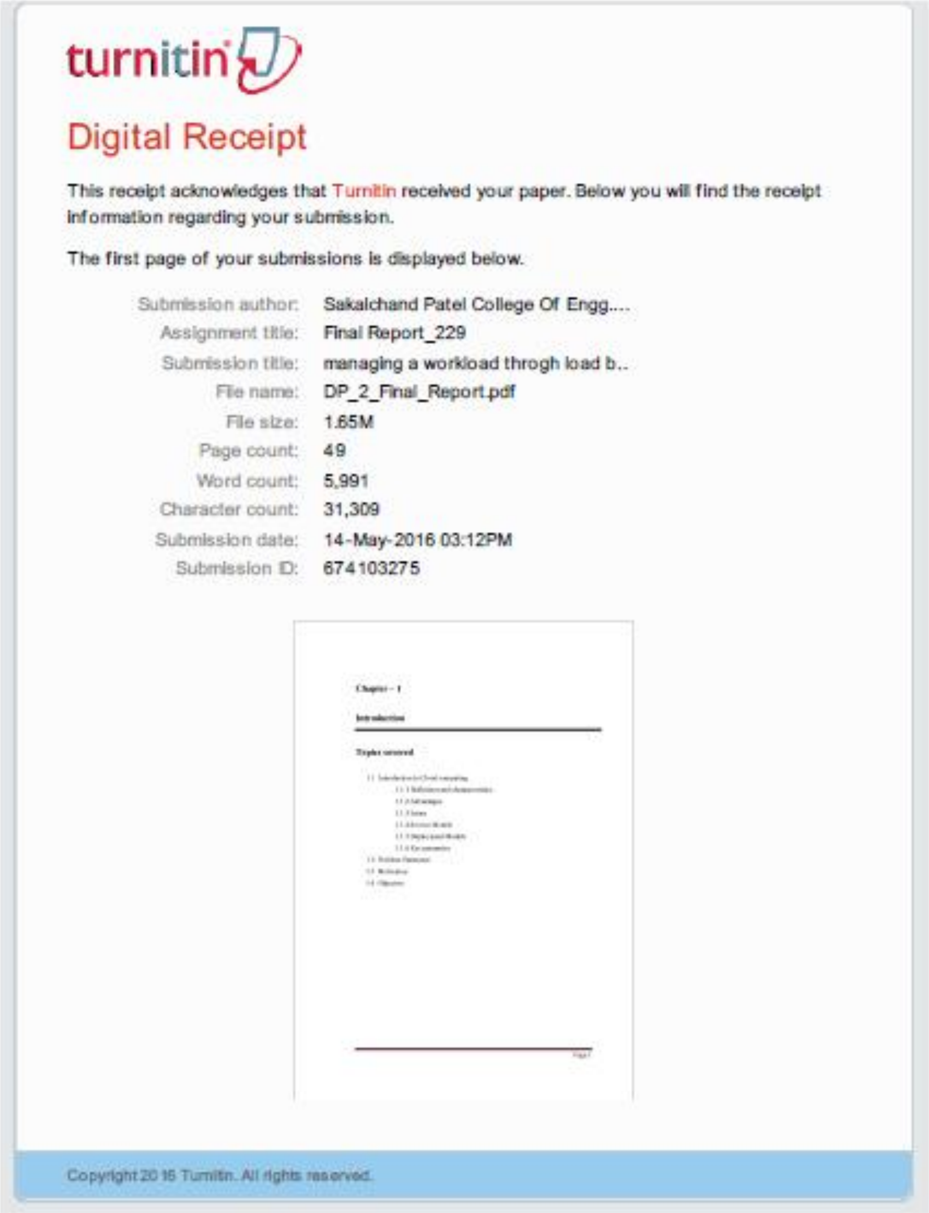
   Base paper implemented and improved result has been shown.
2. Give the rational in comparison table

   I have Done
3. Simulation experiment should be included

   Included
4. Prediction Formula can be applied on data to check its accuracy.

Appendix D

Plagiarism Report



turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

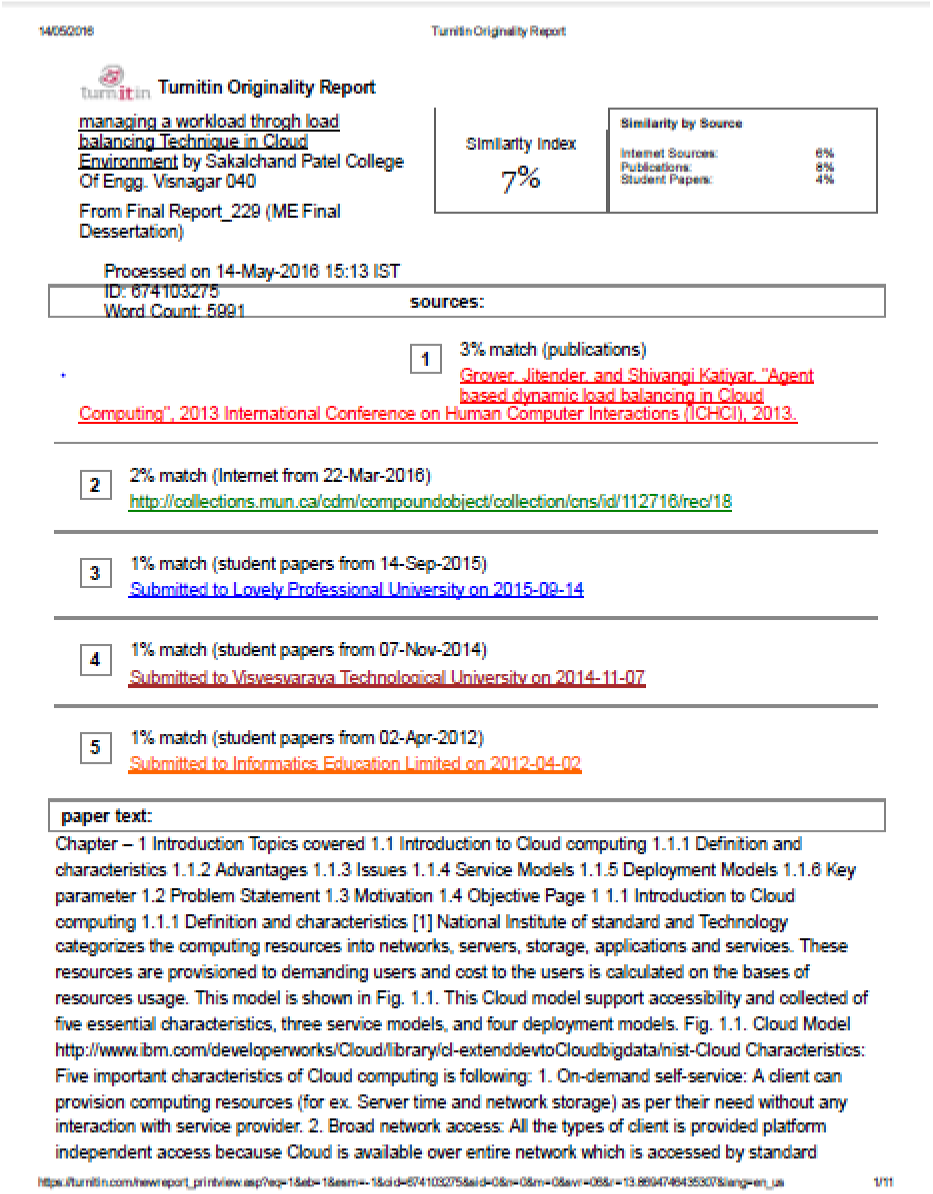
Submission author:	Sakalchand Patel College Of Engg....
Assignment title:	Final Report_229
Submission title:	managing a workload throgh load b..
File name:	DP_2_Final_Report.pdf
File size:	1.65M
Page count:	49
Word count:	5,991
Character count:	31,309
Submission date:	14-May-2016 03:12PM
Submission ID:	674103275

Chapter – 1

Introduction

Topics covered

Copyright 2016 Turnitin. All rights reserved.



14/05/2016 Turnitin Originality Report

Turnitin Originality Report

managing a workload throgh load balancing Technique in Cloud Environment by Sakalchand Patel College Of Engg. Visnagar 040

From Final Report_229 (ME Final Dessertation)

Processed on 14-May-2016 15:13 IST
ID: 674103275
Word Count: 5991

Similarity Index	Similarity by Source	
7%	Internet Sources:	6%
	Publications:	8%
	Student Papers:	4%

sources:

1 3% match (publications)
Grover, Jitender, and Shivangi Katiyar. "Agent based dynamic load balancing in Cloud Computing", 2013 International Conference on Human Computer Interactions (ICHCI), 2013.

2 2% match (Internet from 22-Mar-2016)
http://collections.mun.ca/cdm/compoundobject/collection/cns/id/112716/rec/18

3 1% match (student papers from 14-Sep-2015)
Submitted to Lovely Professional University on 2015-09-14

4 1% match (student papers from 07-Nov-2014)
Submitted to Visvesvaraya Technological University on 2014-11-07

5 1% match (student papers from 02-Apr-2012)
Submitted to Informatics Education Limited on 2012-04-02

paper text:

Chapter – 1 Introduction Topics covered 1.1 Introduction to Cloud computing 1.1.1 Definition and characteristics 1.1.2 Advantages 1.1.3 Issues 1.1.4 Service Models 1.1.5 Deployment Models 1.1.6 Key parameter 1.2 Problem Statement 1.3 Motivation 1.4 Objective Page 1 1.1 Introduction to Cloud computing 1.1.1 Definition and characteristics [1] National Institute of standard and Technology categorizes the computing resources into networks, servers, storage, applications and services. These resources are provisioned to demanding users and cost to the users is calculated on the bases of resources usage. This model is shown in Fig. 1.1. This Cloud model support accessibility and collected of five essential characteristics, three service models, and four deployment models. Fig. 1.1. Cloud Model http://www.ibm.com/developerworks/Cloud/library/cl-extenddevtoCloudbigdata/nist-Cloud Characteristics: Five important characteristics of Cloud computing is following: 1. On-demand self-service: A client can provision computing resources (for ex. Server time and network storage) as per their need without any interaction with service provider. 2. Broad network access: All the types of client is provided platform independent access because Cloud is available over entire network which is accessed by standard

https://turnitin.com/newreport_printview.asp?eq=1&eb=1&esm=-1&oid=674103275&sid=0&n=0&m=0&svr=08&r=13.869474643530785&lang=en_us 1/11