

# Improved Load Balancing Algorithm Based On Performance in Cloud Computing

Vrushali Lanjewar<sup>1</sup>, Prof. R.V. Dharaskar<sup>2</sup>, Dr. V. M. Thakare<sup>3</sup>

PG Student, Department of CSE, SGBAU, Amravati, India.<sup>1</sup>

Associate Professor, Department of CSE, SGBAU, Amravati, India.<sup>2</sup>

Associate Professor and Head, Department of CSE, SGBAU, Amravati, India.<sup>3</sup>

**ABSTRACT:** The main challenge in cloud computing is load balancing to distribute the dynamic workload across the multiple nodes to ensure that no single node is inundated which helps in optimal utilization of resources and hence enhances the system performance. In enjoin to achieve utmost profits with optimized load balancing algorithms, it is essential to utilize resources efficiently. The few existing algorithms can helps in maintaining load balances as well as enhanced strategies had provided through efficient scheduling a job and resource allocation technique. This paper discusses several existing load balancing algorithms in cloud computing and their challenges.

**KEYWORDS:** Cloud computing, CloudSim, CloudAnalyst, Load balance; Performance etc.

## I. INTRODUCTION

Cloud computing is a new infrastructure prototype that offers enhanced scalability, elasticity, business agility, faster startup time, reduced management costs, and just-in-time availability of resources. Distributed file systems are key building blocks for cloud computing applications. In general, in a distributed file system the load of a node is proportional to the number of file chunks that node possesses. In such file systems, nodes simultaneously serve computing and storage functions; a file is partitioned into a number of chunks allocated in distinct nodes. As the files in a cloud can be arbitrarily created, deleted, appended and nodes can be up-graded, replaced and added in the file system, the file chunks are not distributed as uniformly as possible among the nodes. Load balance among storage nodes is a critical function in clouds. In a load-balanced cloud, the resources can be well utilized and provisioned.

### I.I LOAD BALANCING IN CLOUD COMPUTING

The main objectives of load balancing are to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers. Load balancing is a method for distributing tasks onto multiple computers. For instance, distributing incoming HTTP requests for a web application onto multiple web servers; there are a few different ways to implement load balancing. Using multiple components with load balancing instead of a single component may increase reliability and availability through redundancy. Fig. 1 is illustrating the basic principle of load balancing. The major purpose of load balancing is to distribute the workload of an application onto multiple computers, so the application can process a higher work load.

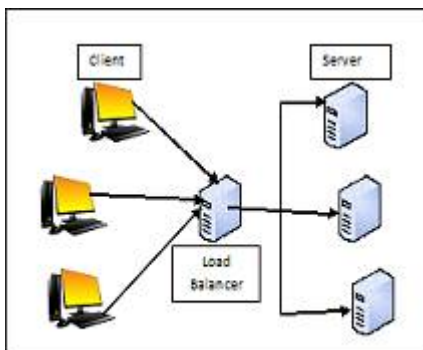


Fig.1 Load Balancing

A minor goal of load balancing is often to provide redundancy in any

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

application. That is, if one server in a cluster of servers fails, the load balancer can temporarily remove that server from the cluster, and divide the load onto the functioning servers. When an error happens and the tasks were moves from the failing server to a functioning server, this is called as failover. A set of servers running the same application in cooperation is typically referred to as a cluster of servers. The purpose of a cluster is usually both of the above two mentioned goals; to distribute load on to different servers and to provide redundancy/failover for each other.

## II. BACKGROUND

Prior load balancing algorithms introduce a performance bottlenecks, as a normal application may be unable to manipulate the unrequested files during the lengthy load rebalancing period. Honeybee Foraging Behaviour, Biased Random Sampling, Active Clustering, and the three representative algorithms were chosen for the comparative study. The first was directly based on a naturally occurring phenomenon, honey bee foraging, the second try to find a desired global outcome from biased random sampling, while the third used system rewiring termed Active Clustering to improve the load balancing performance [1]. By extending the classic Map Reduce model with an agent-aid layer and abstracting working load requests for data blocks as tokens, the agents can reason from previously received tokens about where to send other tokens in order to balance the working tasks and improve system performance[2]. Load-balancing algorithms based on DHTs have been extensively studied. However, most existing solutions are designed without considering both movement cost and node heterogeneity and may introduce significant maintenance network traffic to the DHTs [3]. The load rebalancing problem in distributed file systems specialized for large-scale, dynamic and data-intensive clouds. The proposal is comparable to the centralized algorithm in the Hadoop HDFS production system and dramatically outperforms the competing distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead. [4]. LBNP is assigning data to Reduce tasks based on the performance of Tasktracker nodes which is running the Reduce task. Hadoop is based on the mechanism of heart beating. Only if there are free Reduce slots, Hadoop will assign tasks when Tasktracker reports heartbeat. So, need to make data pre-allocation before the allocation of Reduce tasks. [5]

## III. PREVIOUS WORK DONE

M. Randles et al [1] presented a paper on a comparative study of three distributed load-balancing algorithms for Cloud computing scenarios. It was noted that current commercial offerings based on centralized allocation will cease to be scalable as demand outstrips attempts at apportioning load across all nodes in a system. Therefore a decentralized approach is required whereby load balancing emerges as a global outcome from local node interactions. Yang Xu, et al [2] puts forward a novel model to balance data distribution to improve cloud computing performance in data-intensive applications, such as distributed data mining. The key contribution lies in building an efficient token routing algorithm in spite of agents unknowing to the global state of data distribution in cloud. The load balancing model given by Gaochao, et al [3] in article is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations.

Thus, this model divides the public cloud into several cloud partitions. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy. A novel load-balancing algorithm is proposed by Hasio, et al [4] to deal with the load rebalancing problem in large-scale, dynamic, and distributed file systems in clouds has been presented in this paper. Zhipeng Gao, et al [5] address the problem of how to assign data after Map phase to balance the execution time of each Reduce task by proposing a novel load balancing algorithm based on nodes performance (LBNP), in which the input data of poor performance nodes are decreased.

## III. EXISTING METHODOLOGY

Load balancing existing techniques are Honey-bee Foraging Behaviour, Biased Random Sampling, Active Clustering, Token-Based Heuristic Algorithm, Cloud Partitioning, Load Rebalancing algorithm, LBNP algorithm.

**Honeybee Foraging Behaviour:** A decentralized honeybee-based load balancing technique scrutinize that is a self-organizing, beehive-based load-balancing algorithm was used at the application layer for optimum resource allocation.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

It achieves global load balancing through local server actions.

**Biased Random Sampling:** Biased Random Sampling technique for load balancing in cloud to examine, a distributed and scalable load balancing approach was proposed that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system.

**Active Clustering:** Active Clustering technique for load balancing in cloud was proposed to investigate a self-aggregation load balancing technique that to optimize job assignments by connecting similar services using local re-wiring. [1].

**Token based load balancing:** A heuristic approach for token based load balancing and this approach allow fast, efficient routing decisions, without requiring accurate knowledge of the complete global state. Initially, agents have no knowledge of their neighbours working load distribution; therefore they have no idea where to forward the tokens. To assist their decisions, agents build their knowledge base. In this paper, we build agents' knowledge base solely from the tokens previously received. [2]

**Cloud partitioning:** Cloud partitioning concept for load balancing strategy is presented and the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition. [3]

**Load rebalancing algorithm:** A load-balancing algorithm presented in which each node has global knowledge regarding the system that leads to low movement cost and fast convergence. Then extend this algorithm for the situation that the global knowledge is not available to each node without degrading its performance. The main objective is to allocate the chunks of files as uniformly as possible among the nodes such that no node manages an excessive number of chunks. A file is partitioned into a number of chunks allocated in distinct nodes so that Map Reduce tasks can be performed in parallel over the nodes. For example, consider a word count application that counts the number of distinct words and the frequency of each unique word in a large file. In such an application, a cloud partitions the file into a large number of disjointed and fixed-size pieces (or file chunks) and assigns them to different cloud storage nodes (i.e., chunk servers). [4].

**Load balancing algorithm based on nodes performance:** A load balance algorithm based on nodes performance (LBNP) that uses historical information to estimate the performance of nodes and assigns the tasks according to the nodes performance. The paper contains a deep study of operational mechanism of Map Reduce, focused on the skewed running time of Reduce phase caused by uneven nodes performance, and present a load balance algorithm based on nodes performance (LBNP) to balance the execute time of Reduce phase. LBNP has four steps: data pre-allocation, adaptation of partition function, Reduce tasks allocation, and evaluation of nodes performance. [5]

## V. ANALYSIS AND DISCUSSION

Honeybee Foraging Behaviour is best suited for the conditions where the diverse population of service types is required. Performance of the system is enhanced with increased system diversity but throughput is not increased with an increase in system size. In Biased Random Sampling technique, the performance of the system is improved with high and similar population of resources thus resulting in an increased throughput by effectively utilizing the increased system resources. Active clustering technique enhances the performance of the system with high resources thereby increasing the throughput by using these resources effectively. It is degraded with an increase in system diversity. [1].

In Token-Based Heuristic Algorithm built a prototype of our system, and the experimental results show that the approach can significantly improve the efficiency of cloud computing.[2]. The load balancing strategy proposed cloud partitioning. When the environment is very large and complex, these divisions simplify the load balancing. [3]. The aim of Load Rebalancing algorithm is to reduce network traffic or movement cost caused by rebalancing the loads of nodes as much as possible to maximize the network bandwidth available to normal applications [4].

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

The main idea of LBNP is assigning data to Reduce tasks based on the performance of Task-tracker nodes which is running the Reduce task. The evaluation algorithm of nodes' performance in LBNP is simple but LBNP could be further improved [5]. Advantage and disadvantages of the methods are discussed as shown in Table1.

METHOD/ALGORITHM USED	ADVANTAGES	DISADVANTAGES
<b>HONEY-BEE FORAGING BEHAVIOUR</b>	Decentralized honey bee-based load balancing technique that is a nature-inspired algorithm for self-organization.	Performance of the system is enhanced with increased system diversity but throughput is not increased with an increase in system size.
<b>BIASED RANDOM SAMPLING</b>	Use a random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system.	It is degraded with an increase in population diversity.
<b>ACTIVE CLUSTERING</b>	Performance of the system is enhances with high resources thereby increase; throughput by using these resources effectively.	It is degraded with an increase in system diversity.
<b>TOKEN BASED HEURISTIC ALGORITHM</b>	Agents can intelligently pass tokens to maximize team efficiency with limited communication costs. Significantly improved efficiency of cloud computing.	In the random token passing, agents did not build any knowledge but tokens were forwarded in a random way.
<b>CLOUD PARTITIONING</b>	A simple method was used for partitioning idle state with a more complex method for the normal state.	The system status changed but the information will still be old. An erroneous load strategy choice & an erroneous nodes order.
<b>LOAD REBALANCING ALGORITHM</b>	1. This load-balancing algorithm exhibits a fast convergence rate. 2. Efficiency, effectiveness of this algorithm was validated.	Synthesis workloads stress test the load-balancing algorithms by creating a few storage nodes that are heavily loaded.
<b>LOAD BALANCING ALGORITHM BASED ON NODES PERFORMANCE (LBNP) ALGORITHM</b>	1. Job's execution time decreases using LBNP. 2. Evaluation algorithm of nodes performance in LBNP is simple.	Execution time is longer because of poor nodes performance.

**Table 1. Advantages and disadvantages**

### Performance Attributes for Load Balancing In Clouds:

Various performance attributes considered in existing load balancing techniques in cloud computing are discussed below:

**Throughput:** It is considered as number of tasks whose execution has been completed. For any load balancing system throughput decides the quality so it should be high.

**Response Time:** It is the total time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.

**Overhead:** It determines the amount of extra work required to be done to implement the load balancing algorithms. Overhead should be minimized to get the best results.

**Migration time:** It is the time of movement of job from the master system to the slave system and vice versa in case of results. Migration time is the overhead which cannot be removed but should be minimized.

**Resource Utilization:** It is the parameter that can be considered so that the resources are being fully utilized. The resource utilization must be very high since one of the main aspects of cloud computing is to use the resources at the maximum.

**Scalability:** It defines if the algorithm being implemented can be used for any number of nodes and any kind of

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

hierarchical structure in cloud. Scalability is one of the main concepts since clouds are expanding daily with increasing heterogeneity.

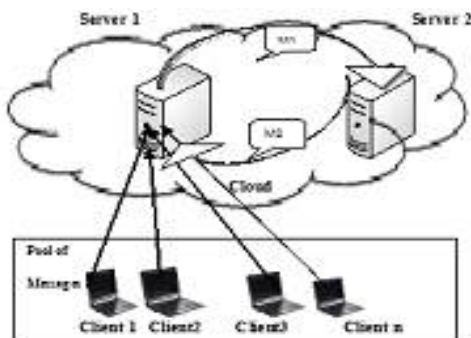
**Fault Tolerance system:** is a system in which the processing does not get affected because of failure of any particular processing device in the system. The load balancing should be fault tolerant.

**Performance:** determines the efficiency of the system. The overall performance of the system should increase by using a particular load balancing technique. Based on these performance attributes, the existing load balancing techniques have been compared as shown in Table 2.

ALGORITHM	THROUGHPUT	RESPONSE TIME	OVER HEAD	MIGRATION TIME	RESOURCE UTILIZATION	SCALABILITY	FAULT TOLERANCE	PERFORMANCE
HONEY-BEE FORAGING BEHAVIOUR	Y	X	X	X	X	Y	X	Y
BIASED RANDOM SAMPLING	Y	X	X	X	X	Y	X	Y
ACTIVE CLUSTERING	X	X	X	X	X	Y	X	Y
TOKEN BASED HEURISTIC ALGORITHM	Y	X	X	Y	Y	Y	X	Y
CLOUD PARTITIONING	X	Y	Y	X	Y	X	X	Y
LOAD REBALANCING ALGORITHM	Y	Y	X	X	Y	Y	Y	Y
LBNP ALGORITHM	X	Y	X	Y	X	X	X	Y

**TABLE 2-** Comparison of Existing Load Balancing Techniques Based On Various Metrics.

## VI. PROPOSED METHODOLOGY



**Fig.2** Load balancing in cloud

The proposed a load balancing algorithm which will transfer the load to another server in cloud when the current server is overloaded. When the multiple request are arrived to allocate the resource at server in cloud environment, the server gets overloaded at some instance.

Here, server1 will receive more request than its capacity when the request pool of messages reached at optimal. Then incoming requests are directly forwarded to server 2. For maintaining the request pool of messages DHT is used. So, without degrading its performance the request will transfer to another server as soon as the server maximum capacity. Then this server will moves request to another server. While transferring data to another server Kerberos protocol is used to ensure the security of data. The Kerberos security system protects electronic transmissions that get sent across the network. It does this by scrambling the information so that only the computer that's supposed to receive the information can



# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

unscramble it. In addition, it makes sure that your password never gets sent across the network, only a scrambled "key" to your password. Kerberos is necessary because there are people who know how to tap into the lines between the computers and listen out for passwords.

## V. SIMULATION TOOLS USED

The CloudSim toolkit supports system and behavior modeling of cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. CloudSim is a self-contained platform which can be used to model data centers, service brokers, scheduling and allocation policies of a large scaled Cloud platform. It provides a virtualization engine with extensive features for modelling the creation and life cycle management of virtual engines in a data center.

CloudAnalyst separates the simulation experimentation exercise from a programming exercise. It also enables a modeller to repeatedly perform simulations and to conduct a series of simulation experiments with slight parameters variations in a quick and easy manner. Cloud Analyst can be applied to examining behavior of large scaled Internet application in a cloud environment. Several simulators have been developed specifically for performance analysis of cloud computing environments including CloudSim, Cloud Analyst; these are used for performance evaluation of different load balancing algorithms for different parameters such as throughput, resource utilization, maintenance and cost, overhead etc.

## VI. RESULTS & DISCUSSIONS

### 1.1 CLOUDSIM

CloudSim project has found widespread use in both research-based cloud development and industry-based cloud development. It provides a generalized framework to apply experiments, according to specific design issues that can be seamlessly set to the offered generalized framework.

With its earlier versions based on GridSim, CloudSim was developed by the GRIDS laboratory of the University of Melbourne to model and simulate several cloud computing infrastructures. Conceptually, CloudSim on one side offers classes representing data centers, physical hosts, virtual machines, services to be executed in the data centers, users of cloud services, internal data center networks, and energy consumption of physical hosts and data centers elements. On the other side, CloudSim supports dynamic insertion of simulation elements and provides message-passing application and data center network topology.

CloudSim experiment

The following scenario was assumed to test the capability of CloudSim for modeling and simulation. The objective of this scenario is to calculate the power consumption (watt) for a cloud data center. We assumed that the simulation environment consists of one data center with two hosts (physical machines). Each host has one CPU core (with computing capacity 1000 MIPS), 2 GB of RAM memory, and 1 TB of storage.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

Simulation completed.

OUTPUT						
Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	160	0.1	160.1
1	SUCCESS	2	1	719.98	0.1	720.08
2	SUCCESS	2	1	719.98	0.1	720.08
3	SUCCESS	2	1	719.98	0.1	720.08
4	SUCCESS	2	1	719.98	0.1	720.08
5	SUCCESS	2	1	719.98	0.1	720.08
6	SUCCESS	2	1	719.98	0.1	720.08
7	SUCCESS	2	1	719.98	0.1	720.08
8	SUCCESS	2	1	719.98	0.1	720.08
9	SUCCESS	2	1	719.98	0.1	720.08

```

*****Datacenter: Datacenter_0*****
User id      Debt
3            224.8
  
```

**Fig.3 Data Center Output By CloudSim**

In this simulation, we have 10 VMs running on the host, with the following constraints: one CPU core, with 500 MIPS computing capacity, 512 MB RAM memory, and 10 GB of storage. A default policy allocates each VM to the first available host with the maximum number of free processing elements (Pes) based on FCFS. Each host dynamically distributes/shares its capacity of a core among all VMs running concurrently (time-shared VM scheduler). The environment is composed of ten cloudlets (task units), where upon each instance of cloudlet requires 400,000 million instructions, or about 6 min to complete execution on a host. Figure 3 shows the data center results of the simulation produced by CloudSim.

In addition, CloudSim supports the modeling and simulating of different cloud computing environments, as well as simulation of many data centers. Furthermore, evidence in literature suggests that CloudSim is capable of instantiating 100,000 machines in less than 5 min. It further enables studying on allocating policies and migrations of virtual machines (VMs) rendering its very useful in the research field for many researchers.

## 1.2 CLOUDANALYST:

CloudAnalyst includes new powerful features such as an easy-to-use GUI, the ability to separate a simulation from the programming code, quick simulation setup, and an enhanced graphical results display, featuring useful formats such as tables and charts. Installing CloudAnalyst is very simple and time efficient.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

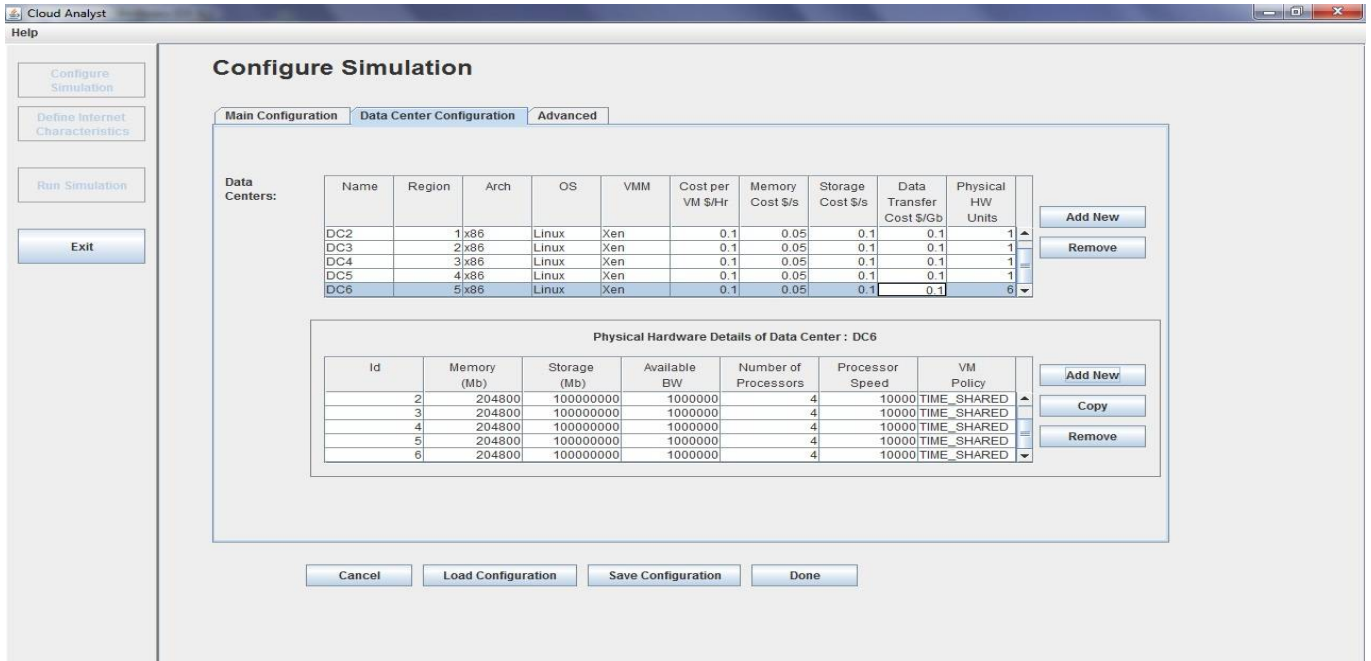


Fig. 3 CloudAnalyst Configuration Panel

The main window of CloudAnalyst lets the user configure the simulation by determining data centers, hosts, VMs, and user base that represent the number of users (e.g., 1000 users), as shown in fig. 3. The user base sends requests that are executed according to a routing policy in a specific data center.

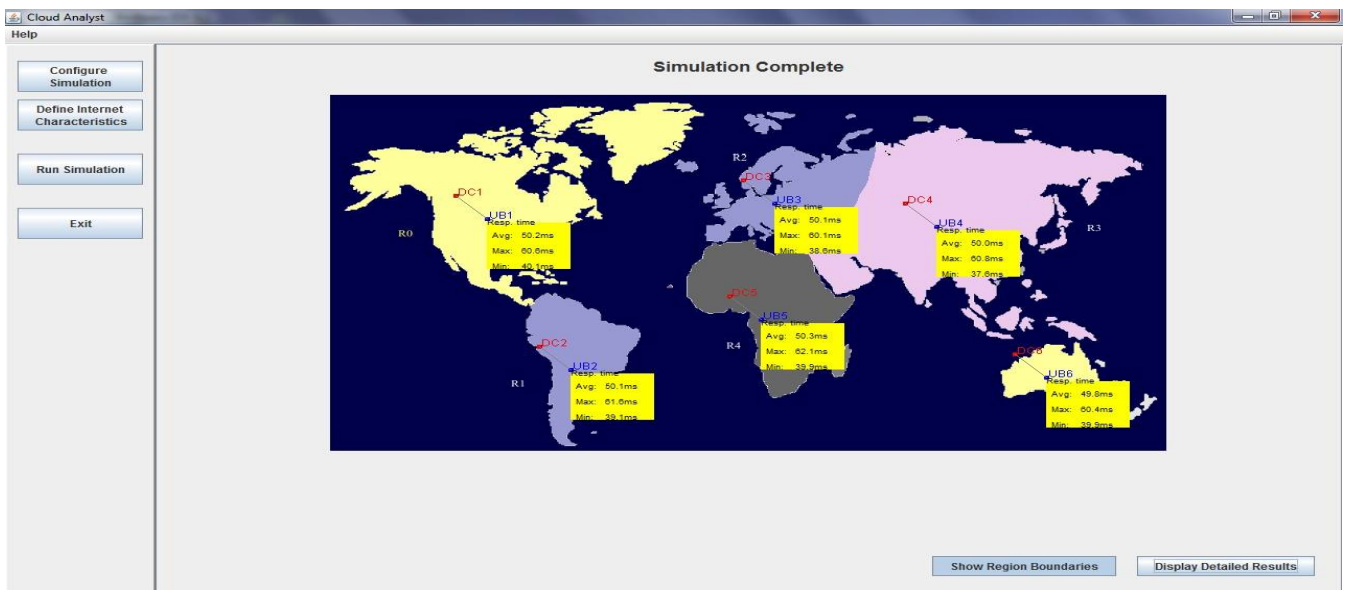


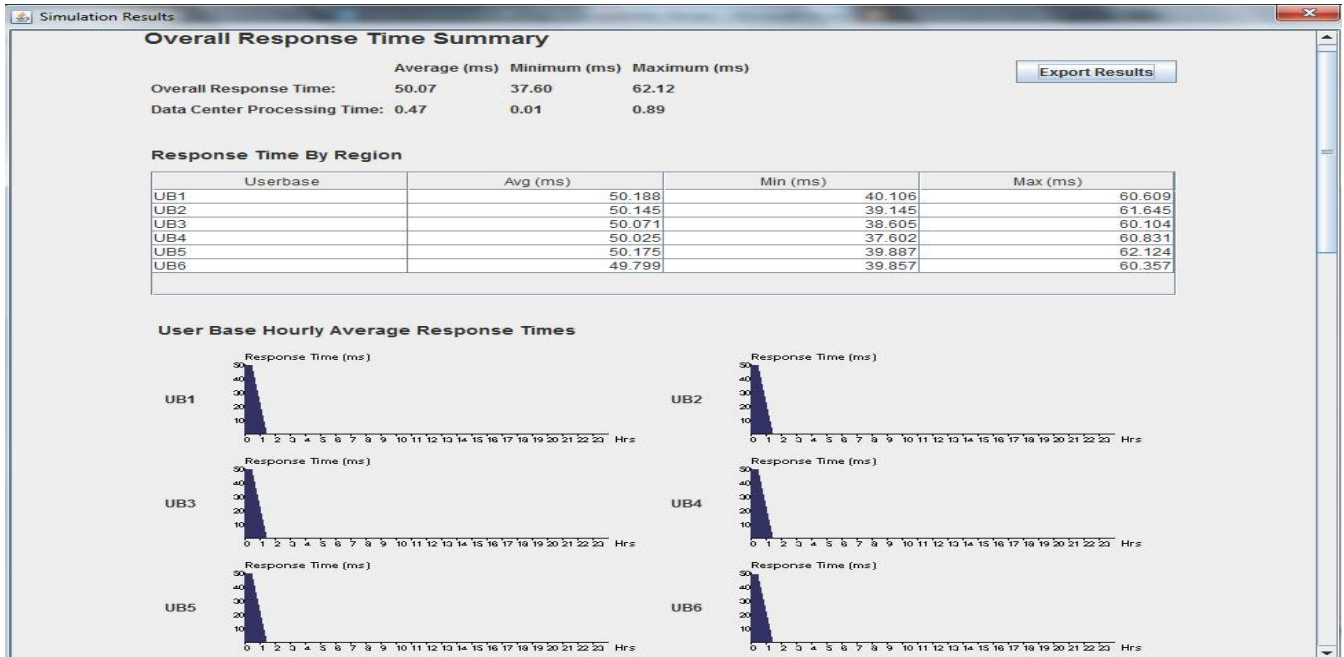
Fig. a)



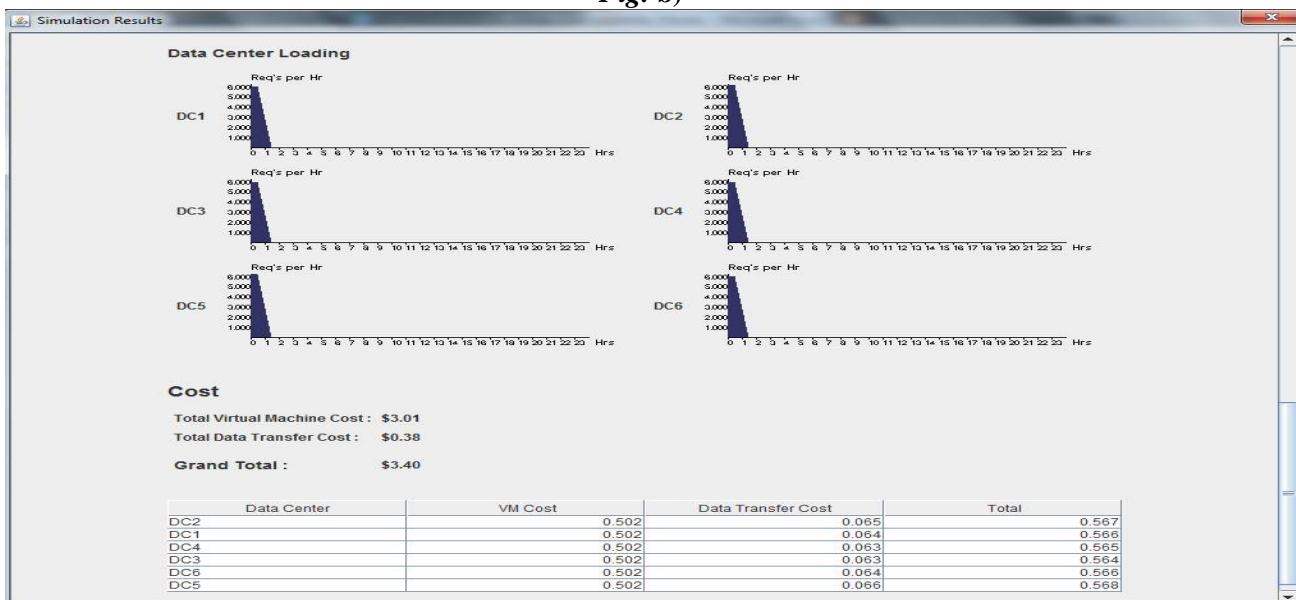
# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016



**Fig. b)**



**Fig. c)**

**Figure a) Cloud Analyst Simulation Window, b), c) detailed simulation result window**

When the simulation completes, the output panel displays the response time for each user base. The simulator provides a detailed screen with results that include response time for each user, request servicing time by each data center, and a number of all requests serviced by the data center. All of these results are displayed in tables and charts to simplify the analysis as shown in fig. 5.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

### Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	50.11	37.61	61.06
Data Center processing time:	0.49	0.01	0.90

### Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	50.19	37.61	60.86
UB2	50.01	38.64	60.89
UB3	49.94	38.88	61.06
UB4	50.21	39.14	59.64
UB5	50.20	37.66	60.66
UB6	50.10	38.86	60.61

### Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.46	0.02	0.86
DC2	0.50	0.03	0.89
DC3	0.49	0.02	0.88
DC4	0.50	0.02	0.88
DC5	0.50	0.03	0.90
DC6	0.47	0.01	0.86

### Cost

Total Virtual Machine Cost (\$): 3.01

Total Data Transfer Cost (\$): 0.38

Grand Total: (\$) 3.40

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC2	0.50	0.07	0.57
DC1	0.50	0.06	0.57
DC4	0.50	0.06	0.57
DC3	0.50	0.06	0.56
DC6	0.50	0.06	0.57
DC5	0.50	0.07	0.57

**Fig 5. Tables representing detailed Result of simulation**

### CloudAnalyst experiment

In this experiment, we assume that there is an internet application that is located in three data centers around the world. The cloud environment consists of six user bases which are geographically distributed and three data centers with 25, 10, and 30 VMs as shown in fig. 6 below. The user base traffic is routed to the data center according to the service proximity policy that selects the closest data center in terms of the lowest network delay. The duration of this simulation is 1 h.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

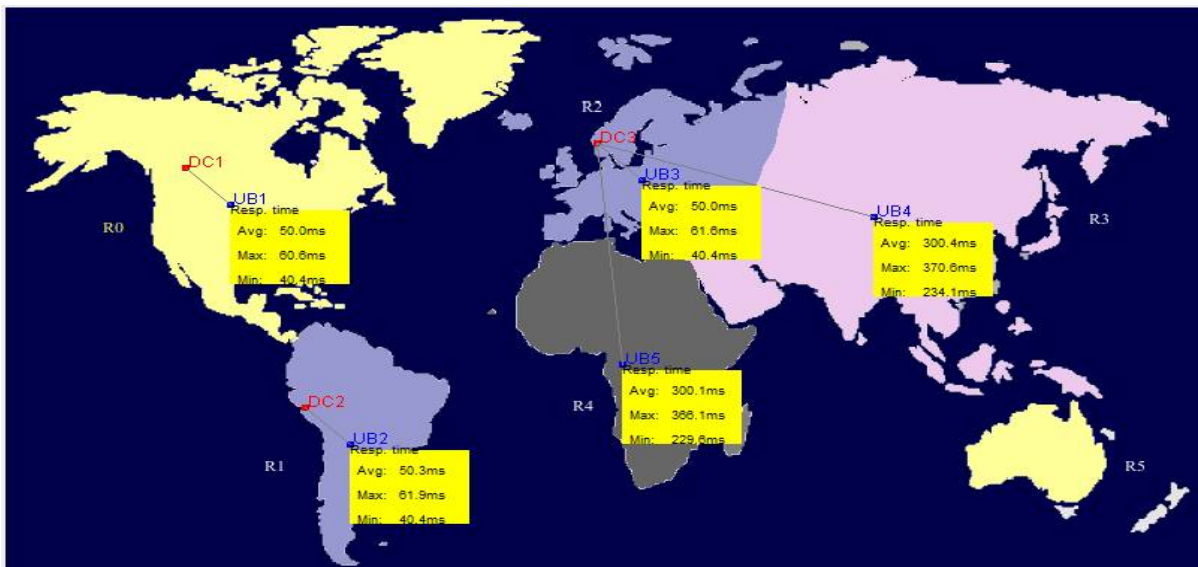
User bases:		Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
		UB1	0	60	100	3	9	1000	100
		UB2	1	60	100	3	9	1000	100
		UB3	2	60	100	3	9	1000	100
		UB4	3	60	100	3	9	1000	100
		UB5	4	60	100	3	9	1000	100

Application Deployment Configuration:		Service Broker Policy: <span style="border: 1px solid black; padding: 2px;">Closest Data Center</span>			
Data Center	# VMs	Image Size	Memory	BW	
DC1	25	10000	512	1000	
DC2	10	10000	512	1000	
DC3	5	10000	512	1000	

**Fig 6. User base and Data center configuration**

After the simulation is completed, CloudAnalyst shows the regions with response to time results. This is shown in fig. 7.



**Fig. 7 A view of geographical response time result**

It is also possible to analyze the results by exporting a .pdf file that includes detailed charts and tables. Tables 1 and 2 are illustrative screenshots of a request servicing time for each data center and cost details, respectively.

### Data Center Request Servicing Time

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.471	0.016	0.856
DC2	0.475	0.032	0.866
DC3	0.385	0.013	0.855

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

## Cloud Environment Cost Results

Data Center	VM Cost	Data Transfer Cost	Total
DC2	1.004	0.065	1.069
DC1	2.509	0.064	2.573
DC3	0.502	0.192	0.693

The goal of this research was to determine whether the literature on load balancing techniques in cloud computing provides a standardized and precise base. Then an extensive systematic selection process was carried out to identify papers describing load balancing techniques in cloud computing. The results presented here thus give a good picture of the existing load balancing techniques in cloud computing.

## VII. CONCLUSION

This article gave a preface to cloud computing then explained the challenges that need to be focused in the coming future. This paper also contains the description of already available solutions for the load balancing problem. This script explains Load Balancing problem as the main problem in this article. Cloud Computing offers a very large number of opportunities of using IT infrastructure as a utility with many possibilities depending upon the needs of the organization. Simulation based approaches become popular in industry and academic world to estimate cloud computing systems, application behaviors and their security.

## VIII. FUTURE SCOPE

This paper proposed a load balancing approach for load balancing effect and it has employed an idea of Kerberos protocol with Key Distribution Center. A larger scale of simulation experiments on CloudSim and prototype experiments are one of our next research works.

## REFERENCES

- [1] Martin Randles, David Lamb, A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, pp.551-556, DOI: 10.1109 / WAINA.2010.85. 978-0-7695-4019-1/10\$26.00©2010IEEE.
- [2] Xu Y, Wu L, Guo L, Chen Z, Yang L, Shi Z. "An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing", In Workshops at the Twenty-Fifth AAAI on Artificial Intelligence, pp. 27-32, 2011 Aug 24.
- [3] Xu, Gaochao; Pang, Junjie; Fu, Xiaodong, "A Load Balancing Model Based on Cloud Partitioning for The Public Cloud", 2013 IEEE Transaction on Cloud Computing, Tsinghua Science and Technology, DOI: 10.1109/TST.2013.6449405, Vol.18, No.1, pp.34-39, Feb. 2013.
- [4] Hung-Chang Hsiao, Member, IEEE Computer Society, Hsueh-Yi Chung, Haiying Shen, Member, IEEE, and Yu-Chang Chao, "Load Rebalancing for Distributed File Systems in Clouds", IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 5, May 2013.
- [5] Zhipeng Gao; Dangpeng Liu; Yang Yang; Jingchen Zheng; Yuwen Hao, "A Load Balance Algorithm Based On Nodes Performance In Hadoop Cluster," in Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific, DOI:10.1109/APNOMS .2014.6996555, Vol., No., pp.1-4, 17-19 Sept. 2014
- [6] B. Wickremasinghe, R. N. Calheiros and R. Buyya, "CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications," 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, 2010, pp. 446-452.
- [7] P. Humane and J. N. Varshapriya, "Simulation of cloud infrastructure using CloudSim simulator: A practical approach for researchers," Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on, Chennai, 2015, pp. 207-211.