



## A NOVEL APPROACH OF JOB ALLOCATION USING MULTIPLE PARAMETERS IN CLOUD ENVIRONMENT

Ashima <sup>(1)</sup>, Vikramjit Singh <sup>(2)</sup>

<sup>(1)</sup> Research Scholar, Department of Computer Engineering, NWIET, Moga  
roohashima@gmail.com

<sup>(2)</sup> Assistant Professor, Department of Computer Science and Engineering, NWIET, Moga  
vikram\_cse@northwest.ac.in

### ABSTRACT

Cloud computing is Internet ("cloud") based development and use of computer technology ("computing"). It is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. This research deals with the balancing of work load in cloud environment. Load balancing is one of the essential factors to enhance the working performance of the cloud service provider. Grid computing utilizes the distributed heterogeneous resources in order to support complicated computing problems. Grid can be classified into two types: computing grid and data grid. We propose an improved load balancing algorithm for job scheduling in the Grid environment. Hence, in this research work, a multi-objective load balancing algorithm has been proposed to avoid deadlocks and to provide proper utilization of all the virtual machines (VMs) while processing the requests received from the users by VM classification. The capacity of virtual machine is computed based on multiple parameters like MIPS, RAM and bandwidth. Heterogeneous virtual machines of different MIPS and processing power in multiple data centers with different hosts have been created in cloud simulator. The VM's are divided into 2 clusters using K-Means clustering mechanism in terms of processor MIPS, memory and bandwidth. The cloudlets are divided into two categories like High QOS and Low QOS based on the instruction size. The cloudlet whose task size is greater than the threshold value will enter into High QOS and cloudlet whose task size is lesser than the threshold value will enter into Low QOS. Submit the job of the user to the datacenter broker. The job of the user is submitted to the broker and it will first find the suitable VM according to the requirements of the cloudlet and will match VM depending upon its availability. Multiple parameters have been evaluated like waiting time, turnaround time, execution time and processing cost. This modified algorithm has an edge over the original approach in which each cloudlet build their own individual result set and it is later on built into a complete solution.

### KEYWORDS

Cloud computing, Load balancing, Virtual machine, Host, Datacenter, Datacenter Broker

### INTRODUCTION

Cloud could be a term used as a trope for the wide space networks (like internet) or several such giant networked atmosphere. It came partially from the cloud like image wont to represent the complexities of the networks within the schematic diagrams. It represents all the complexities of the network which can hold everything from cables, routers, servers, knowledge centers and every one such alternate devices. Cloud computing is an on demand service in which shared resources, information, software and other devices are provided according to the clients requirement at specific time. It's a term which is generally used in case of Internet. The whole Internet can be viewed as a cloud. Capital and operational costs can be cut using cloud computing [36]. With traditional desktop computing, we run copies of software programs on our personal computer. The documents we make are stored on our own pc. Although documents can be accessed from other computers on the network, they cannot be accessed by computers outside the network. This is PC-centric. By cloud computing, the software programs one use are not run from one's personal computer, but are quite stored on servers accessed via the Internet. If a computer crashes, the software is still available for others to use. Similar goes for the documents one create; they are stored on a collection of servers accessed through the Internet. Anyone with permission can not only access the documents, but can also edit and cooperate on those documents in real time.

### NECESSARY CHARACTERISTICS OF CLOUD COMPUTING

Cloud technology is in the news quite often these days, but it still seems to be mysterious and confusing to the non-techie crowd [7]. Cloud options are enticing various industries across the board, which is why it's important to know its essential characteristics as a software offering. Here are the five main characteristics that cloud computing offers businesses today.

- i. **On-demand self-service:**-A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- ii. **Broad network access:** - Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- iii. **Resource pooling:** -The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

- iv. **Rapid elasticity:** - Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- v. **Measured service:** - Cloud systems automatically control and optimize resource use by leveraging a metering capability<sup>1</sup> at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

## LOAD BALANCING

It is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded. It is used to achieve a high user satisfaction and resource utilization ratio, hence improving the overall performance of the system. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. It also helps in implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning, reducing response time etc. Load Balancing is a method to distribute workload across one or more servers, network interfaces, hard drives, or other computing resources [13]. Typical datacenter implementations rely on large, powerful (and expensive) computing hardware and network infrastructure, which are subject to the usual risks associated with any physical device, including hardware failure, power and/or network interruptions, and resource limitations in times of high demand. Load balancing in the cloud differs from classical thinking on load-balancing architecture and implementation by using commodity servers to perform the load balancing. This provides for new opportunities and economies-of-scale, as well as presenting its own unique set of challenges. Load balancing is used to make sure that none of your existing resources are idle while others are being utilized. Modern high-traffic websites must serve hundreds of thousands, if not millions, of concurrent requests from users or clients and return the correct text, images, video, or application data, all in a fast and reliable manner. To cost-effectively scale to meet these high volumes, modern computing best practice generally requires adding more servers [28]. A load balancer acts as the “traffic cop” sitting in front of your servers and routing client requests across all servers capable of fulfilling those requests in a manner that maximizes speed and capacity utilization and ensures that no one server is overworked, which could degrade performance. If a single server goes down, the load balancer redirects traffic to the remaining online servers. When a new server is added to the server group, the load balancer automatically starts to send requests to it.

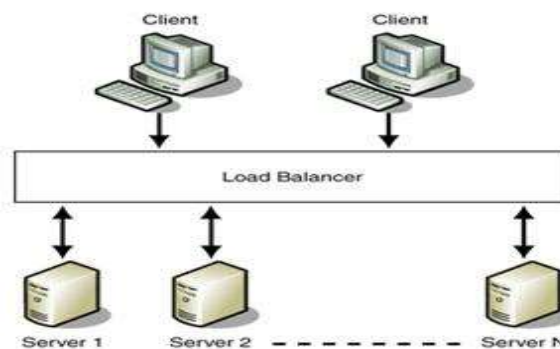


Figure 1. Model of Load Balancing [28]

In order to distribute the necessary tasks, load balancers go through a series of steps. First, the load balancer will query the available servers to ensure their availability. The load balancer pings a server, and if the expected response occurs, it will be included in the available list. If the server fails to respond, it will not be used until another test is performed and it returns with the appropriate response.

## RELATED WORK

We have studied and analyze many load balancing techniques for cloud computing. After our detailed literature review we are able to discover theme of our work including the techniques we have incorporated in various phases, to implement it and to evaluate our results and conclusion.

Reena et al. (2015) discussed an efficient Dynamic Load Management Algorithm where load is managed by the server by considering the present status of present VMs for request assignment sharply. That distribution of the entire incoming request among the virtual machines effectively. Hiren et al. (2015) presented a Flexible load sharing algorithm (FLS) which introduce the third function. The third function makes partition the system in to domain. This function is helpful for the selection of other nodes which are present in the same domain. The third partition function will make the partition in to each domain so that using the remote location the information will be transferred to another domain. Such that the node will exchange the information from one node to another node. B. Kruekaew et al. (2014) applies artificial Bee Colony (ABC) to optimize the scheduling of Virtual Machine (VM) on Cloud computing. The main contribution of work is to analyze the difference of VM load balancing algorithm and to reduce the make span of data processing time. Nagamani H et al. (2014)



proposed a new technique to achieve load balancing called Load Balancing with Optimal Cost Scheduling Algorithm. The user selects a list of services available from the service pack. The scheduler processes these tasks to the virtual machine (VM) based on the configuration and computing power of the VM. This task is achieved with minimum execution cost which is a profit for the service provider and minimum execution time which is an advantage for both service provider and the user. [30] M. Ajit et al. (2013) described Cloud load balancing helps to enhance the overall cloud performance. This paper presents the analysis of three contemporary algorithms in cloud analyst tool to resolve the issue of cloud load balancing as a preparation phase for new load balancing technique. This research work done into three parts: calculation and assignment of load assignment factor to hosts, provisioning and balancing. Desai et al. (2013) discusses about the emerging technology i.e. a new standard of large scale distributed computing and parallel computing. It provides shared resources, information or other resources as per clients' requirements at specific times. For better management of available good load balancing techniques are required. Babu et al. (2013) proposed a Honey Bee Behavior inspired Load Balancing [HBB-LB] technique which helps to achieve even load balancing across virtual machine to maximize throughput. It considers the priority of task waiting in queue for execution in virtual machines. Al-Rayis et al. (2013) explains that basically, load balancers can be deployed based on three different architectures. The centralized load balancing architecture which includes a central load balancer to make the decision for the entire system regarding which cloud resource should take what workload and based on which algorithm(s). In the hierarchical load balancing architecture, a main load balancer (parent) receives all job requests, and then it spreads them to other connected load balancers (children) where each load balancer in the tree may use a different algorithm.

## PROBLEM FORMULATION

- VM's are categorized only on a single parameter which is MIPS. Multiple parameters like RAM and Bandwidth should also be considered for allocation of cloudlets to VM
- Extra overhead time is involved for processing the mid-point of the cloudlet list and virtual machine list.
- While computing midpoint, a single virtual machine can be assigned with multiple tasks of higher instruction size.
- No suitable criteria has been defined for handling the faulty virtual machines and task migration at that particular time.

## OBJECTIVES

The primary purpose of the cloud system is that its client can utilize the resources to have economic benefits. A resource allocation management process is required to avoid underutilization or overutilization of the resources which may affect the services of the cloud.

- To implement and study the performance of existing load balancing algorithms.
- To compute the capacity of VM based on multiple parameters like MIPS, RAM and Bandwidth.
- To optimize the overhead time required for processing task list and Vm list.
- To implement the fault tolerance mechanism for Virtual machines.
- To develop the proposed algorithm and compare the performance of proposed algorithm with existing algorithm.

## RESEARCH METHODOLOGY

The proposed algorithm is implemented through the simulation software like cloud sim used. For the implementation of the application, java language is used. This research work considers Datacenter, VM, host and Cloudlet components from CloudSim for implementation of a proposed algorithm.

- Initialize the Cloud Sim in Java by creating different data centers having multiple number of hosts of different configurations.
- Each Host in the datacenter will have the different numbers of Virtual machines of different capacities.
- Then we will create the Cloudlets of varying length and size of different users.
- Create the multiple number of user cloudlets of different requirements and size.
- The Datacenter broker inside the cloud simulator will retrieve the number of cloudlets and number of virtual machines created by the user.
- Divide the VM's into 2 clusters: High Processing VM's and Low Processing VM's
- Clustering is done at the VM level on the basis of Vm capacity in terms of processor, memory, bandwidth by using K-Means Clustering mechanism.
- Create the multiple number of user cloudlets of different requirements and size. Cloudlets are divided into categories: high processing and low processing tasks.
- The decision is based on length of the cloudlets. The cloudlet whose task size is greater than the threshold value will enter into high processing tasks and cloudlet whose task size is lesser than the threshold value will enter into low processing tasks.
- Submit the job of the user to the datacenter broker. DCB will first find the suitable VM according to the requirements of the cloudlet and will match VM depending upon its availability.
- Create a backup task of each cloudlet which is going to execute by the VM.
- If Task  $T_1$  executes successfully on VM, then back up of task  $T_1$  is no longer required and therefore, backup task is deleted successfully.

- If VM which is executing the current task has become faulty, then the backup task of  $T_1$  will be executed by another VM depending upon its availability.
- The VM which is failed is added to the blacklisted table so that no other cloudlet is further assigned to the faulty VM.
- Change the status of that virtual machine from idle to busy.
- Dispatch our cloudlet to that virtual machine and we will modify the rating of that particular virtual machine.
- Repeat the same procedure for all the remaining cloudlets. .

## K-MEANS CLUSTERING

K-Means clustering aims to partition,  $n$  (number of observations) quantities into  $k$  (assume  $k$  clusters) clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume  $k$  clusters) fixed a priori. The main idea is to define  $k$  centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point re-calculate  $k$ - new centroids as barycenter of the clusters resulting from the previous step. After having these  $k$  new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop it may be noticed that the  $k$  centers change their location step by step until no more changes are made or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function known as the squared error function given by:

$$J(V) = \sum_{i=1}^c \sum_{j=1}^{c_i} (\|x_i - v_j\|)^2$$

Where,

$\|x_i - v_j\|$  is the Euclidean distance between  $x_i$  and  $v_j$ .

$c_i$  is the number of data points in  $i^{th}$  cluster.

$c$  is the number of cluster centers.

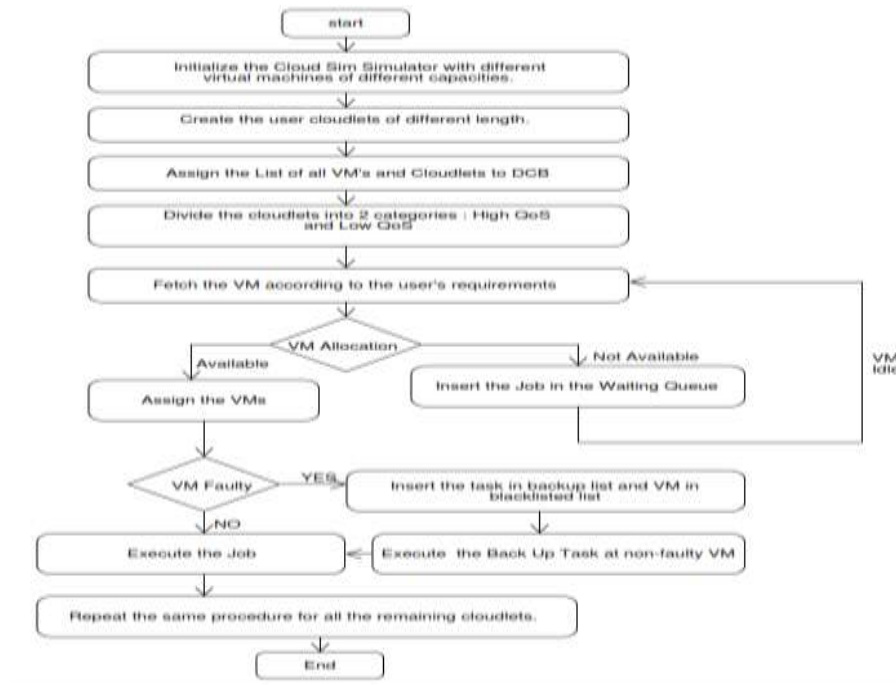


Figure 2. Flow Chart of Proposed Methodology

## SIMULATION RESULTS



In the following tables, we have taken multiple cloudlets ranging from 5 to 20000. To compute the results of proposed work we are taking 10 virtual machines with MIPS = 100, RAM = 256 MB and Bandwidth = 300 MB/second to calculate the parameters. The results of the proposed work have been mentioned in table 1.

**Table1. Results of the proposed work**

S. No.	Number of Cloudlets	Total Processing Time (seconds)	Total Waiting time (seconds)	Total Processing Cost (dollars)
1	5	0.0	0.0	0.0
2	10	0.0	0.0	0.1
3	50	0.1	0.4	0.4
4	100	0.3	1.7	0.8
5	150	0.4	4.0	1.2
6	200	0.5	7.2	1.6
7	300	0.8	16.6	2.4
8	400	1.0	29.6	3.2
9	500	1.3	46.5	3.9
10	600	1.6	67.4	4.7
11	700	1.8	91.7	5.5
12	800	2.1	119.8	6.3
13	900	2.3	152.2	7.1
14	1000	2.6	187.7	7.9
15	1200	3.1	271.1	9.5
16	1400	3.6	368.9	11.1
17	1600	4.1	482.3	12.6
18	1800	4.7	611.3	14.2
19	2000	5.2	754.5	15.8
20	2500	6.5	1179.9	19.8
21	3000	7.8	1701.2	23.7
22	3500	9.1	2314.8	27.7
23	4000	10.4	3025.1	31.6
24	4500	11.7	3830.7	35.6
25	5000	13.0	4728.8	39.5
26	5500	14.3	5723.6	43.5
27	6000	15.5	6813.4	47.4
28	6500	16.8	7994.5	51.4
29	7000	18.1	9273.7	55.3



30	7500	19.4	10648.6	59.3
31	8000	20.7	12113.6	63.2
32	8500	22.0	13675.9	67.2
33	9000	23.3	15336.9	71.2
34	9500	24.6	17085.7	75.1
35	10000	25.9	18933.6	79.1
36	15000	38.9	42617.9	118.6
37	20000	51.8	75765.9	158.1

## PARAMETERS USED

### TOTAL PROCESSING TIME:-

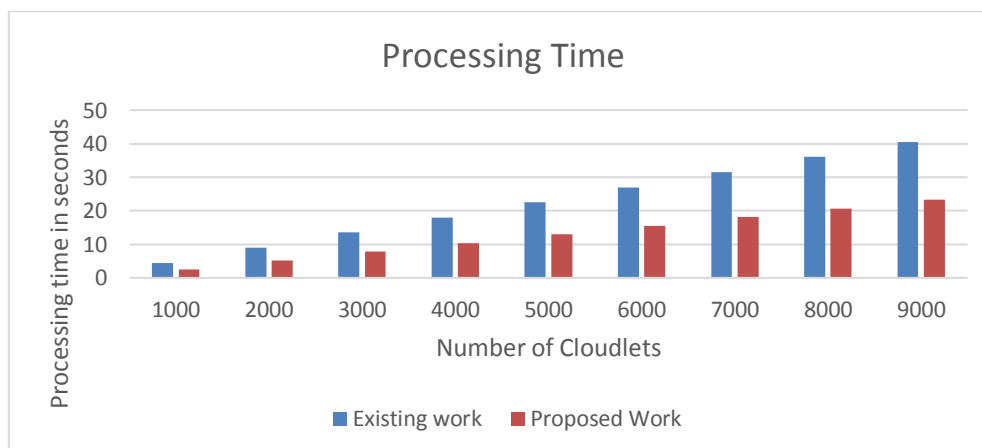
It is defined as the time interval between the request sent and the response received by the cloud user/ consumer. Overall processing time is calculated as given as follow:-

$$\text{Response Time (RT)} = \text{FT} - \text{ST}$$

Where, FT=finish time of execution

ST=start time of execution

The data, thus collected was used to plot a graph consisting of number of jobs of the existing work and the proposed work.



**Figure 3. Total Processing Time**

For the above bar chart, it is clear that total processing time has been reduced

### TOTAL PROCESSING COST

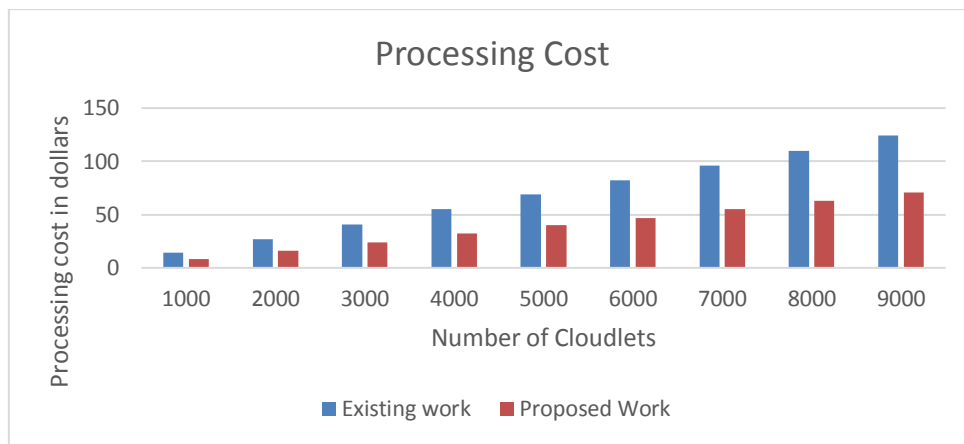
It is obtained by addition of cost per storage, cost per memory, cost per processor \* Processing time.

Processing cost=RT\* unit cost.

Where,

RT= response time

Unit cost = cost per unit time



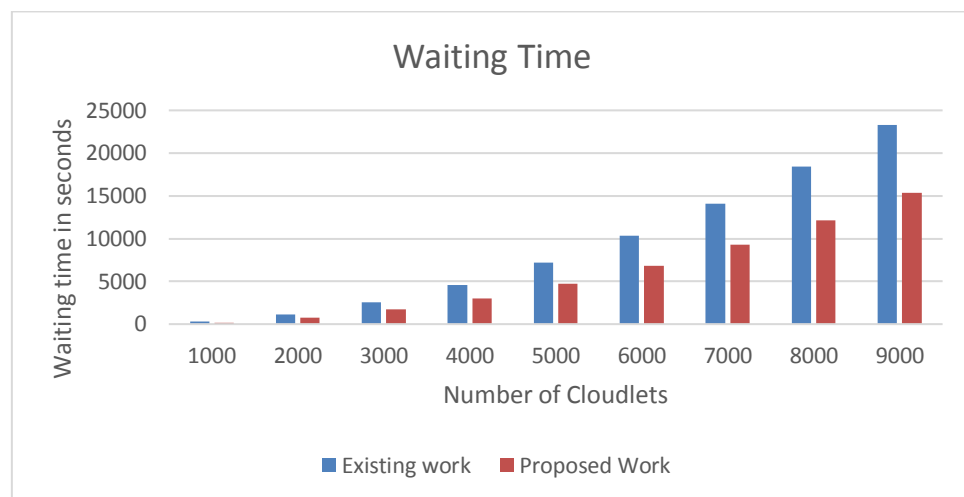
**Figure 4. Total Processing cost**

For the above bar chart, it is clear that total processing cost has been reduced.

### TOTAL WAITING TIME

Waiting time=Allocation Time-generation Time

Figure 5 illustrates that waiting time for the base work and proposed work



**Figure 5. Total waiting Time**

For the above bar chart, it is clear that total waiting has been reduced.

### CONCLUSION

The performance of improved load balancing algorithm has been studied in the research work. In order to evaluate performance of the proposed model simulation study has been put through various test conditions. It has been found that the model works well in ensuring an even distribution of the workload. In the current work, it has been assumed that all the incoming requests are independent to each other. The experiment conducted is compared with previous algorithm. The result indicates that the approaches outperform to previous algorithm in terms of total processing time, total processing cost and waiting time. To obtain a better solution, the model should be made more realistic by considering issues related to load balancing such as data locality, communication cost and flow time and results can be tested in real Cloud environment. However, within a computational cloud environment high throughput is of great interest rather than the load balancing. To achieve this, we have proposed the new dynamic load balancing algorithm. The experimental results obtained by applying the new proposed algorithm in the Cloud Sim Simulator, shows that the new work has outperformed the existing scheduling algorithms in large scale distributed systems.

### REFERENCES

- [1] S. Yakhchi, S. Ghafari, M. Yakhchi, M. Fazeli and A. Patooghy, "ICA-MMT: A Load Balancing Method in Cloud Computing Environment," IEEE, 2015.
- [2] S. Kapoor and D. C. Dabas, "Cluster Based Load Balancing in Cloud Computing," IEEE, 2015.



- [3] S. Garg, R. Kumar and H. Chauhan, "Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing," 1st International Conference on Next Generation Computing Technologies (NGCT-2015), pp. 77-80, 2015.
- [4] R. Panwar and D. B. Mallick, "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm," IEEE, pp. 773-778, 2015.
- [5] M. Belkhouraf, A. Kartit, H. Ouahmane, H. K. Idrissi, Z. Kartit and M. E. Marraki, "A secured load balancing architecture for cloud computing based on multiple clusters," IEEE, 2015.
- [6] L. Kang and X. Ting, "Application of Adaptive Load Balancing Algorithm Based on Minimum Traffic in Cloud Computing Architecture," IEEE, 2015.
- [7] N. K. Chien, N. H. Son and H. D. Loc, "Load Balancing Algorithm Based on Estimating Finish Time of Services in Cloud Computing," ICACT, pp. 228-233, 2016.
- [8] H. H. Bhatt and H. A. Bheda, "Enhance Load Balancing using Flexible Load Sharing in Cloud Computing," IEEE, pp. 72-76, 2015.
- [9] S. S. MOHARANA, R. D. RAMESH and D. POWAR, "ANALYSIS OF LOAD BALANCERS IN CLOUD COMPUTING," International Journal of Computer Science and Engineering (IJCSE) , pp. 102-107, 2013.
- [10] M. P. V. Patel, H. D. Patel and . P. J. Patel, "A Survey On Load Balancing In Cloud Computing," International Journal of Engineering Research & Technology (IJERT), pp. 1-5, 2012.
- [11] R. Kaur and P. Luthra, "LOAD BALANCING IN CLOUD COMPUTING," Int. J. of Network Security, pp. 1-11, 2013.
- [12] Kumar Nishant, , P. Sharma, V. Krishna, Nitin and R. Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization," IEEE, pp. 3-9, 2012.
- [13] Y. Xu, L. Wu, L. Guo,, Z. Chen, L. Yang and Z. Shi, "An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing," AI for Data Center Management and Cloud Computing: Papers from the 2011 AAAI Workshop (WS-11-08), pp. 27-32, 2011.
- [14] A. K. Sidhu and S. Kinger, "Analysis of Load Balancing Techniques in Cloud Computing," International Journal of Computers & Technology Volume 4 No. 2, March-April, 2013, ISSN 2277-3061, pp. 737-741, 2013.
- [15] O. M. Elzeki, M. Z. Reshad and M. A. Elsoud, "Improved Max-Min Algorithm in Cloud Computing," International Journal of Computer Applications (0975 – 8887), pp. 22-27, 2012.
- [16] B. Kruekaew and W. Kimpan, "Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony," Proceedings of the International Multi Conference of Engineers and Computer Scientists 2014 Vol I,IMECS 2014, 2014.
- [17] R.-S. Chang, J.-S. Chang and P.-S. Lin, "An ant algorithm for balanced job scheduling in grids," Future Generation Computer Systems 25 (2009) 20–27, pp. 21-27, 2009.
- [18] Z. Chaczko, V. Mahadevan, S. Aslanzadeh and C. Mcdermid, "Availability and Load Balancing in Cloud Computing," International Conference on Computer and Software Modeling IPCSIT vol.14 (2011) © (2011) IACSIT Press, Singapore, pp. 134-140, 2011.
- [19] R. K. S, S. V and V. M, "Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud," Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications - ACCTHPCA, June 2012, pp. 31-35, 2012.
- [20] Kumar Nishant, P. Sharma, V. Krishna, N. and R. Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization," IEEE, pp. 3-9, 2012.
- [21] Ankit Kumar, Mala Kalra, " Load Balancing in Cloud Data Center Using Modified Active Monitoring Load Balancer", IEEE pp. 1-5, 2016.
- [22] Saraswathi AT, Kalaashri.Y.RA, Dr.S. Padmavathi, "Dynamic Resource Allocation Scheme in Cloud Computing", ELSEVIER, pp. 30-36, 2015