International Journal of Advance Research in Science and Engineering
Volume No. 11, Issue No. 10, October 2022
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Load Balancing in Cloud Computing

Deepak Kannaujaiya[1]

[1](*Department of Computer Science &Engineering, B.I.T, Gida, Gorakhpur, Utter Pradesh, India)*

ABSTRACT

The past few years have witnessed the emergence of a novel paradigm called cloud computing. Cloud computing is a structured model that defines computing services, in which data as well as resources are retrieved from cloud service provider via internet through some well-formed web-based tool and application. Cloud Computing is nothing but a collection of computing resources and services pooled together and is provided to the users on pay-as-needed basis. Sharing of the group of resources may initiate a problem of availability of these resources causing a situation of deadlock. One way to avoid deadlocks is to distribute the workload of all the VMs among themselves. This is called load balancing. Load unbalancing problem is a multi-variant, multi-constraint problem that degrades performance and efficiency of computing resources. Load balancing algorithms and job allocations are main research problems in areas of resource management of future internet. The goal of balancing the load of virtual machines is to reduce energy consumption and provide maximum resource utilization thereby reducing the number of job rejections. The aim of this paper is to discuss the concept of load balancing in cloud computing. This paper deals with the different load balancing algorithms such as static load balancing algorithms (SLBA) and dynamic load balancing algorithms (DLBA). The load balancing algorithm can be used to better utilize and better understand user needs.

Keywords— Cloud Computing, Load balancing, load balancer, static load balancing, dynamic load balancing algorithm, load balancing metrics.

I. INTRODUCTION

The US National Institute of Standards and Technology (NIST) characterizes cloud computing as pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. A cloud computing model is efficient if its resources are utilized in best possible way and such an efficient utilization can be achieved by employing and maintaining proper management of cloud resources. The problem of load unbalancing is an undesirable event in the Cloud service provider side that degrades the performance and efficacy of the computing resources along with guaranteed Quality of Service (QoS) on agreed Service Level Agreement (SLA) between consumer and provider. Under these circumstances there arises need for load balancing (LB) and is a peculiar topic of research interest among researchers. The load balancing in cloud computing can be done at physical machine level or VM level. Load balancing is a method that distributes the workload among diverse nodes in the given environment such that it ensures no node in the system is over loaded or sits idle for any instant of time. An efficient load balancing algorithm will make sure that every node in the system does more or less same volume of work. Load balancing is a new approach that assists networks and resources by

providing a high throughput and least response time. In cloud platforms, resource allocation (or load balancing) takes place majority at two levels.

- At first level: The load balancer assigns the requested instances to physical computers at the time of uploading an application attempting to balance the computational load of multiple applications across physical computers.
- At second level: When an application receives multiple incoming requests, each of these requests must be assigned to a specific application instance to balance the computational load across a set of instances of the same application. This paper is organized as follows: Section 2 contains an introduction of load balancing. Section 3 describes the existing classification of load balancing algorithms. Dynamic load balancing policies are analyzed in section 4. Different load balancing metrics are given in section 5 and final conclusion of the work in section 6.

II. LOADBALANCING

2.1 Overview of Load Balancing:

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine. Balancing the load of virtual machines uniformly means that anyone of the available machine is not idle or partially loaded while others are heavily loaded. In cloud computing, if users are increasing load will also be increased, the increase in the number of users will lead to poor performance in terms of resource usage. With the load balancer you can split the workload and balance it between two or more servers in the cloud. As a result, you can configure your infrastructure to maximize activity, optimize resource allocation, and provide a smallest amount of response time.If some good load balancing technique is implemented, it will equally divide the load (here term equally defines low load on heavy loaded node and more load on node with less load now) and thereby we can maximize resource
utilization.

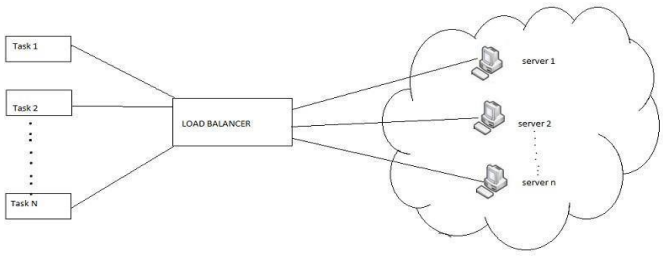

Fig 1. Diagram for load balancing

2.2 What is meant by load balancer?

The main aim of the load balancer helps to assign resources equally to the tasks for resource efficiency and user satisfaction at minimal expense, quality output, gripping rapid traffic blast sustain traffic on the website and elasticity which motivates us to identify problems in LB and to work

- Reduce waiting time,
- Reduce the response time,

- Increase the utilization of resources,
- Improve reliability ,
- Increase throughput,
- Load balancing enhance the performance of the system by managing every node.

Load balancing is also needed for achieving Green computing in clouds. The factors responsible for it are:

1. Limited Energy Consumption: Load balancing can reduce the amount of energy consumption by avoiding over hearting of nodes or virtual machines due to excessive workload.

2. Reducing Carbon Emission: Energy consumption and carbon emission are the two sides of the same coin. Both are directly proportional to each other. Load balancing helps in reducing energy consumption which will automatically reduce carbon emission and thus achieve Green Computing.

2.4 Significance of Load Balancing:

i. ***Better Performance:***

Load balancing methods are smaller and easier to implement than their counterparts. Organizations can work their customers' applications more quickly and deliver better performance at relatively low cost.

ii. ***Maintain Website Traffic:***

Cloud balancing provides scalability to manage your site traffic. With efficient load balancing, you can easily manage user traffic at a high level with the presence of servers and network devices.

iii. Handle Sudden Traffic Burst: The **request generator** generates user requests which are user tasks that need computing resources for their execution. **Data center controller** is in-charge of task management. The load balancer checks which VM to assign for a given user task. The **first level load balancer** balances the given workload on individual **Physical Machines** by distributing the workload among its respective associated **Virtual Machines**. The **second level load balancer** balances the workloadacrossdifferent Virtual Machines of different Physical Machines. Scheduling and allocating tasks to VMs based on their requirements constitute the cloud computing workload. The load balancing process involves the following activities:

2.5.1. Identification of user task requirements: This phase identifies the resource requirement of the user tasks to be scheduled for execution on a VM.

2.5.2. Identification of resource details of a VM: This checks the status of resource details of a VM. It gives the current resource utilization of VM and the unallocated resources. Based on this phase, the status of VM can be determined as balanced, overloaded or under-loaded with respect to a threshold.

2.5.3. Task scheduling:

Once resource details of a VM are identified the tasks are scheduled to appropriate resources on appropriate VMs by a scheduling algorithm.

2.5.4. Resource allocation:

The resources are allocated to scheduled tasks for execution. A resource allocation policy is being employed to accomplish this. While, scheduling is required for speeding up the execution, allocation policy is used for

proper resource management and improving resource performance. The strength of the load balancing algorithm is determined by the efficacy of the scheduling algorithm and the allocation policy.

2.5.5. Migration:

Migration is an important phase in load balancing process in cloud and latter is incomplete without the former. Migration is of two kinds in cloud based on entity taken into consideration- **VM migration** and **task migration**. VM migration is the movement of a VM from one physical host to another to get rid of the overloading problem and is categorized into types as **live VM migration** and **non-live migration**. Likewise task migration is the movement of tasks across VMs and is of two types: **intra VM task migration** and **inter VM task migration**. An efficient migration technique leads to an efficient load balancing. From the extensive survey it has been concluded that task migration process is more time and cost effective than VM migration and the trend has shifted from VM to task migration.

III. CLASSIFICATION OF LOAD BALANCING ALGORITHMS

In cloud computing, cloud servers should always be balanced, to use the resources with their full capacity. Sometimes it happens that some servers are heavily loaded while the other servers are under loaded or in idle state. To overcome this problem load balancing algorithms are used. These algorithms help in allocating every single task by monitoring load on each server. According to the Deepak Mahapatra, the balancing algorithm is defined as "The load balancing in clouds may be among physical hosts or VMs. This balancing mechanism distributes the dynamic workload evenly among all the nodes (hosts or VMs). The load balancing in the cloud is also referred to as load balancing as a service (LBaaS)".

Based on process orientation load balancing algorithms are classified as:

a) Sender Initiated: In this sender initiates the process; the client sends request until a receiver is assigned to him to receive his workload.

b) Receiver Initiated: The receiver initiates the process; the receiver sends a request to acknowledge a sender who is ready to share the workload.

c) Symmetric: It is a combination of both sender and receiver initiated type of load balancing algorithm.

Based on the current state of the system load balancing algorithms are classified as:

1. Static Load Balancing

In the static load balancing algorithm the decision of shifting the load does not depend on the current state of the system. It requires knowledge about the applications and resources of the system. The performance of the virtual machines is determined at the time of job arrival. The master processor assigns the workload to other slave processors according to their performance. The assigned work is thus performed by the slave processors and the result is returned to the master processor.

Static load balancing algorithms are not preemptive and therefore each machine has at least one task assigned for itself. This algorithm has a drawback that the task is assigned to the processors or machines only after it is created and that task cannot be shifted during its execution to any other machine for balancing the load.

2. Dynamic Load Balancing

In this type of load balancing algorithms the current state of the system is used to make any decision for load balancing, thus the shifting of the load depends on the current state of the system. It allows for processes to move from an over utilized machine to an underutilized machine dynamically for faster execution.

This means that it allows for process preemption which is not supported in Static load balancing approach. An important advantage of this approach is that its decision for balancing the load is based on the current state of the system which helps in improving the overall performance of the system by migrating the load dynamically.Traditional types of algorithms come under the static ones and the metaheuristic algorithms come under the dynamic algorithms.

3.1 Static Load Balancing Algorithms:

Round-robin algorithms: presently it is available such as (1) round robin, (2) weighted round robin.

3.1.1 Round Robin:

Round robin use the time slicing mechanism. The name of the algorithm suggests that it works in the round manner where each node is allotted with a time slice and has to wait for their turn. The time is divided and interval is allotted to each node. Each node is allotted with a time slice in which they have to perform their task. The complicity of this algorithm is less compared to the other two algorithms. An open source simulation performed the algorithm software know as cloud analyst, this algorithm is the default algorithm used in the simulation. This algorithm simply allots the job in round robin fashion which doesn't consider the load on different machines.

Time-based algorithms: are like (1) Max–Min algorithm, (2) Min–Min algorithm;

3.1.3 Min-Min load balancing algorithm:The Algorithm take up with a task set which are initially not assigned to any of the nodes. Initially the minimumcompletion time is calculated for all the available nodes. Once this calculation gets completed the task having the completion time minimum is chosen and assigned to the respective node. The execution time of all other tasks which are currently available in that machine is updated and the task gets discarded from the available task set. The routine is done time after time until all the tasks have been assigned to the equivalent machines. The algorithm works better when the situation is like where the small tasks are greater in number of than the large tasks. The algorithm has a disadvantage that it leads to starvation.

Min-Min is a simple and fast algorithm capable of providing improved performance. Min-Min schedules the ideal tasks at first which results in best schedules and improve the overall make span. Assigning small task first is its drawback. Thus, smaller tasks will get executed first, while the larger tasks keeps on in the waiting stage, which will finally results in poor machine use. Min-Min exhibits minimum completion time for jobs which are unassigned (similar to MCT), and later allocating the jobs with minimum completion time

The max-min algorithm is much the same as to min-min algorithm. At first for all the available tasks are submitted to the system and minimum completion time for all of them are calculated, then among these tasks the one which is having the completion time, maximum is chosen and that is allocated to the corresponding machine. This algorithm outperform than Min-Min algorithm where when short tasks are in high numbers when compared to that of long ones. For e.g. if in a task set only a single long task is presented then ,Max Min algorithm runs short tasks concurrently along with long task. The make span focus on how much small tasks

will get executed concurrently with the large ones. Max-Min is almost identical to Min-Min, except it selects the task having the maximum completion time and allocates to the corresponding machine. The algorithm suffers from starvation where the tasks having the maximum completion time will get executed first while leaving behind the tasks having the minimum completion time. Architectural description of Max Min algorithm is presented below:

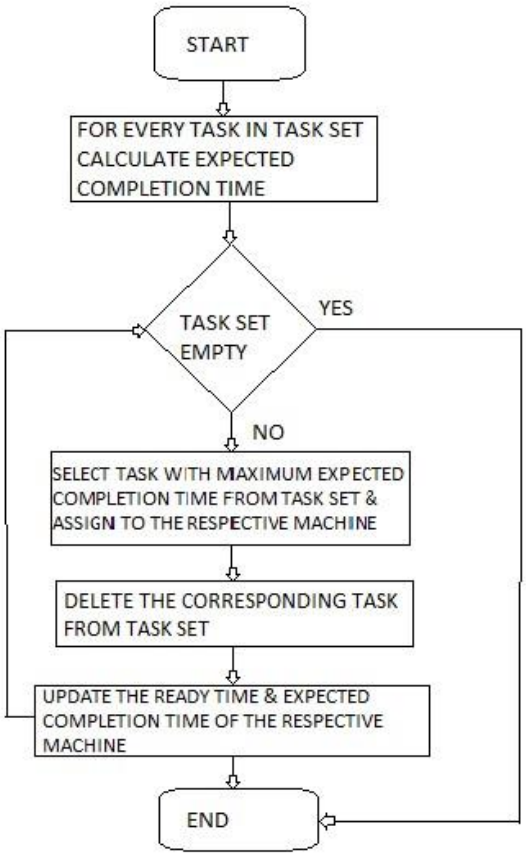

Fig .2 Architectural description of max-min algorithm.

Agent-based algorithm includes algorithms such as Opportunistic Load Balancing.

3.1.5 Opportunistic Load Balancing (OLB):

It keeps all the servers busy and never considers the load of the task which is currently running on the server. Besides the current task running on the server it randomly allocated another task.

Connection-based algorithms: are available such as (1) least connection, (2) weighted least connection.

3.1.6 Least Connection:

In this algorithm it considers currently running tasks also which is not taken by the above mentioned algorithms. According to which other tasks are assigned, these lead to the least number of active sessions in current time.

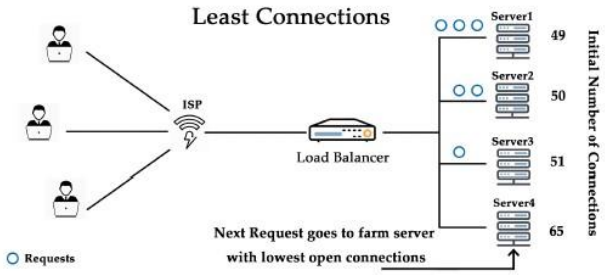

Fig 3. Least Connection Algorithm

3.1.7 Weighted Least Connection:

This algorithm is similar to the least connection algorithm. As seen in Weighted Round Robin where weights are assigned to the server likewise in Weighted Least Connection also weights are assigned in numerical form.

3.2 Dynamic Load Balancing Algorithms:

Optimization algorithms: are available such as (1) honey bee foraging, (2) ant colony optimization, and (3) hybrid genetic based host load and also in particle swarm optimization.

3.2.1 Honey bee foraging:

This dynamic algorithm is derived from the honey bees; the detailed study is done on their behavior of how they search for food and reap food. Scout is a class of bees which finds food and informs others through the dance called vibration. This gives the estimate about the quality and quantity of the food. This technique is used in load balancing to inform the underloaded and overloaded VMs. The tasks from the overloaded machines are shifted to the underloaded machine. Similarly, the tasks from the overloaded VMs are considered as the honey bees. Submitting tasks to the underloaded VM, the task will update the number of tasks and load of that particular VM to all other waiting tasks.

Based on the load and the priority tasks choose VMs.

Whenever a high priority task has to be submitted to other VMs, it should consider the VM that has a minimum number of high priority tasks so that the particular task will be executed at the earliest. Since all VMs will be sorted in ascending order based on load, the task removed will be submitted to under loaded VM. In essence, the tasks are the honey bees and the VMs are the food sources. Loading of a task to a VM is similar to a honey bee foraging a food source.

3.2.2 Ant colony optimization:

ACO is a technique of problem-solving inspired by the behavior of ants in searching the optimal paths from the nest to food; they all work together and search new sources of food while some ants parallel works on shifting food from source to nest. Many researchers are inspired by this behavior of ants and helped them to solve real life problems in different fields. In this method, a pheromone table of a node is maintained and an ant updates the entries of the node from source to destination. The other routing ants reference the pheromone table and calls that have it as their destination. However, for asymmetric networks, the costs from to and from to may be

different. Hence, in this approach for updating pheromone is only appropriate for routing in symmetric networks.

3.2.3 Genetic Algorithms (GA):

GA is one of the most used algorithms which solves the NP (Non-polynomial) -complete problems. It is derived from the soft computing method. It comes under the heuristic search process. GA is inspired by natural evolution from the human mind and genetics.

A simple GA consisting of triple processes: (1) availability, (2) Genetic and (3) replacement operations. GA's core is the creation of offspring via mutations & crossover methods, with the assumption that either binary coding, tree coding or numerical coding depends on the type of the chromosome.

3.2.4 Heuristic Algorithm:

Traditional methods of solving problems were very slow and not compatible for solving NP-complete problems. Heuristic algorithms solve problems in a faster and efficient way than the traditional way and designed to solve decision problems. Example is the Traveling Salesmen Problem where there is a list of cities to visit and the distance between the two cities are given and the optimal solution is to visit all the cities by travelling a minimum distance.

3.2.5 Particle Swarm Optimization (PSO) Algorithm: PSO was first introduced by Kennedy and Eberhart; it is one type of a meta-heuristics method. The PSO algorithm is similar to other population-based algorithms like GA but, there is no direct recombination of individuals of the population. PSO follows a model or a pattern for the social interaction and creates the communication between them, for example, fish schools and bird flock. PSO algorithm is very similar to the swarm intelligence optimization algorithms. It mainly focuses on reducing the total cost of computation of an application on the cloud computing environment.

3.2.6 Hill climbing:

A Hill climbing is the simple and iterative method based optimization algorithm; it moves towards the best solution step by step. It selects the increasing value and moves towards the peak or uphill. Peak is the top of the hill or the place where no neighboring value is more than the peak. The algorithm stops once it reaches the peak or to the stopping criteria. Sometimes it can reach a local optimum solution instead of global optimum. For the load balancing this algorithm maintains an index table with the list of VMs and their states (BUSY/AVAILABLE). When the new task request arrives randomly generates VM id and allocates the task to the VM if the state is available otherwise generates another VM id randomly. It updates the index table accordingly. Once the task is completed, de-allocates the VM and updates the table.

3.2.7 Equally Spread Current Execution (ESCE): ESCE is a dynamic load balancing algorithm, which handles the process with priority. It determines the priority by checking the size of the process. This algorithm distributes the load randomly by first checking the size of the process and then transferring the load to a virtual

machine, which is lightly loaded. The load balancer spreads the load on different nodes, and hence, it is known as spread spectrum technique.

3.2.8 Throttled Load Balancer (TLB): Throttled load balancer is a dynamic load balancing algorithm in which the client first requests the load balancer to find a suitable virtual machine to perform the required operation. In Cloud computing, there may be multiple instances of virtual machine. These virtual machines can be grouped based on the type of requests they can handle. Whenever a client sends a request, the load balancer will first look for that group, which can handle this request and allocate the process to the lightly loaded instance of that group.

IV. DYNAMICLOADBALANCINGPOLICIES

The different policies are described as follows:

1. *Location Policy:* The policy used by a processor or machine for sharing the task transferred by an over loaded machine is termed as Location policy.
2. *Transfer Policy:* The policy used for selecting a task or process from a local machine for transfer to a remote machine is termed as Transfer policy.
3. *Selection Policy:* The policy used for identifying the processors or machines that take part in load balancing is termed as Selection Policy.
4. *Information Policy:* The policy that is accountable for gathering all the information on which the decision of load balancing is based is referred as Information policy.
5. *Load estimation Policy:* The policy which is used for deciding the method for approximating the total work load of a processor or machine is termed as Load estimation policy.
6. *Process Transfer Policy:* The policy which is used for deciding the execution of a task that is it is to be done locally or remotely is termed as Process Transfer policy.
7. Priority Assignment Policy: The policy that is used to assign priority for execution of both local and remote processes and tasks is termed as Priority Assignment Policy.
8. Migration Limiting Policy: The policy that is used to set a limit on the maximum number of times a task can migrate from one machine to another machine.

V. LOADBALANCINGMETRICS

After studying the dynamic load balancing algorithms, we have compared all the algorithms on the bases of some predefined metrics. These metrics are as follows:

Throughput: Throughput is used to calculate the number of jobs whose execution has been completed. It should be high to improve the performance of the system.

Overhead: It determines the amount of overhead involved while implementing a load balancing algorithm. Overhead should be minimized so that a load balancing technique can work efficiently.

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

Migration time: Migration is the time of movement of job of 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.

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

Resource Utilization: It is used to check the utilization of resources. It should be optimized for an efficient load balancing.

Scalability: It is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.

Performance: It is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

VI. CONCLUSION

The cloud computing has the dynamic nature and due to which cloud network has various issues like security, quality of service and fault occurrence etc. The load balancing is the major issue of cloud network which reduce its efficiency. Load balancing is a main task in cloud computing for efficient utilization of resources. The main goal of load balancing is to achieve higher client satisfaction, maximize resource utilization and increase the performance of the cloud system thereby reduction in the energy consumption and carbon emission rate. This research described the numerous algorithms for LB such as static load balancing algorithms and dynamic load balancing algorithms. Although different dynamic and static algorithms are proposed and developed in recent time each one has own advantages and disadvantages. In this manner, the objective is to balance the audience traffic of the cloud infrastructure while enhancing the execution, reducing the overall response time, increasing the throughput for a particular number of jobs, and proficiently handle resource usage.

VII. REFERENCES

[1] JianzheTai,JueminZhang,JunLi,WaleedMeleis and NingfangMi “A R A: Adaptive Resource Allocation for Cloud Computing Environments under Bursty Workloads” 978-1-4673-0012-4/11 ©2011 IEEE.

[2] Ali M Alakeel, “A Guide To Dynamic Load Balancing In Distributed Computer Systems”, International Journal of Computer Science and Network Security, Vol. 10 No. 6, June 2010.

[3] Abhijit A Rajguru, S.S. Apte, “A Comparative Performance Analysis of Load Balancing Algorithms In Distributed Systems Using Qualitative Parameters”, International Journal of Recent Technology and Engineering, Vol. 1, Issue 3, August 2012.

[4] Nidhi Jain Kansal, Intervener Chana, “Cloud Load Balancing Techniques: A Step Towards Green Computing”, IJCSI, Vol. 9, Issue 1, January 2012.

[5] R. Shimonski, Windows 2000 And Windows Server 2003, Clustering and Load Balancing Emeryville, McGrow-Hill Professional Publishing, CA, USA, 2003.

[6] David Escalnte and Andrew J. Korty, “Cloud Services: Policy and Assessment”, EDUCAUSE Review, Vol. 46, July/August 2011.

[7] Parin. V. Patel, Hitesh. D. Patel, Pinal. J. Patel, “A Survey on Load Balancing in Cloud Computing” IJERT, Vol. 1, Issue 9, November 2012.

[8] R. Mata-Toledo, and P. Gupta, “Green data center: how green can we perform”, Journal of Technology Research,Academic and Business Research Institute, Vol. 2, No. 1,May 2010, pages 1-8.

[9] S. K. Garg, C. S. Yeob, A. Anandasivamc, and R. Buyya,“Environment-conscious scheduling of HPC applications ondistributed Cloud-oriented data centers”, Journal of Parallel and Distributed Computing, Elsevier, Vol. 70, No. 6, May 2010, pages 118.

[10] Jitendra Bhatia, Tirth Patel, Harsha Trivedi, VishrutMajmudar," HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud",18,Dec2 012,Pages 15 -20 IEEE.

[11] Dr. Hemant S. Mahalle, Prof. Parag R. Kaveri ,Dr.Vinay Chavan," Load Balancing On Cloud Data Centres" , International Journal of Advanced Research in Computer Science and Software Engineering,Vol.3,Issue 1,January 2013.

[12] Cisco (2009) Cisco visual networking index: Forecast and methodology, 2009-2014.White paper.

[13] Weiss A (2007) Computing in the clouds. Networker 11: 16-25.

[14] Hayes B (2008) Cloud computing. Commun ACM 51: 9-11.

[15] Mell P, Grance T (2009) Draft NIST Working Definition of Cloud Computing. Nat Inst Standards Technol

[16] Rimal, Prasad B, Choi E, Lumb V (2009) A taxonomy and survey of cloud computing systems. Proceedings of 5th International Joint Conference on INC, IMS and IDC, IEEE .

[17] Sinha PK (1997) Distributed operating Systems Concepts andDesign. IEEE Computer Society Press.

[18] https://kemptechnologies.com/in/load-balancer/load-balancingalgorithms-techniques/

[19] Wang SC, Chen CW, Yan KQ, Wang SS (2013) The Anatomy Study of Load Balancing in Cloud Computing Environment. The Eighth International Conference on Internet and Web Applications and Services 230-235

[20] Gulati A, Chopra RK (2013) Dynamic Round Robin for Load Balancing in a Cloud Computing, International Journal of Computer Science and Mobile Computing .