Load Balancing in Cloud Computing

Priya Bag[1] Rakesh Patel[2] Vivek Yadav[3]
[1,3]B.E. Student [2]Lecturer
[1,2,3]Department of Information Technology
[1,2,3]Kirodimal Institute of Technology, Raigarh(C.G.),India

Abstract— Load balancing is one of the interesting and prominent research topics in cloud computing, which has gained a large attention recently. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancers are used for assigning load to different virtual machines in such a way that none of the nodes gets loaded heavily or lightly. The load balancing needs to be done properly because failure in any one of the node can lead to unavailability of data. In this paper we have discussed many different load balancing techniques used to solve the issue in cloud computing environment. In this paper, an overall review of the current load balancing algorithms in the Cloud Computing environment is presented.

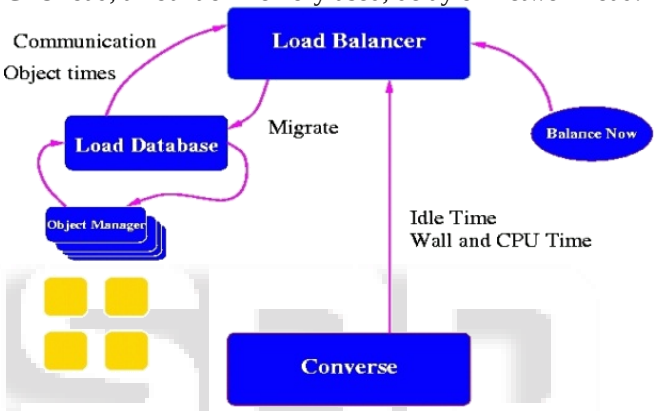
Key words: Human brain, Blue Brain, Artificial Brain

I. INTRODUCTION

Cloud computing is an internet computing in which the load balancing is the one of the challenging task. Load balancing is a methodology to distribute workload across multiple computers, or other resources over the network links to achieve optimal resource utilization, maximize throughput, minimum response time, and avoid overload. Load balancers can work in two ways: one is cooperative and non-cooperative. In cooperative, the nodes work simultaneously in order to achieve the common goal of optimizing the overall response time. In non-cooperative mode, the tasks run independently in order to improve the response time of local tasks. Load balancing algorithms, in general, can be divided into two categories: static and dynamic load balancing algorithm. A static load balancing algorithm does not take into account the previous state or behavior of a node while distributing the load. On the other hand, a dynamic load balancing algorithm checks the previous state of a node while distributing the load. To overcome this situation, many load balancing algorithms are proposed by researchers, with their own pros and cons.

A. Load Balancing:

Load Balancing is a technique to distribute the load evenly among all the nodes of the network. If any node is heavy i.e. have more load than required then its load is given to the node with less load. Hence load balancing helps the overloaded and under loaded nodes. Load balancing is a major challenge of cloud computing. The important things to consider while developing such algorithm are: estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones. This load considered can be in terms of CPU load, amount of memory used, delay or Network load.



B. Goals of Load Balancing

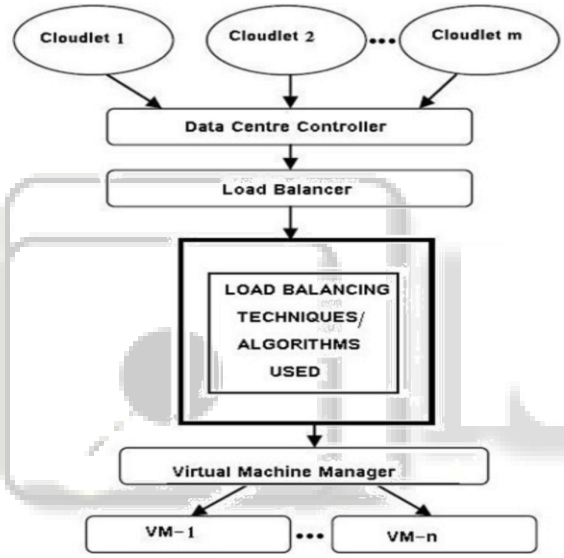
Goals of load balancing involve:

- To improve the performance substantially
- To have a backup plan in case the system fails even partially
- To maintain the system stability
- To accommodate future modification in the system
- Optimum resource utilization
- Maximum throughput
- Maximum response time
- Avoiding overload

II. ALGORITHMS IN LOAD BALANCING

A. Weighted Active Monitoring Load Balancing Algorithm

The 'Weighted Active Monitoring Load Algorithm' is implemented; modifying the Active Monitoring Load Balancer by assigning a weight to each VM as discussed in Weighted Round Robin Algorithm of cloud computing in order to achieve better response time and processing time

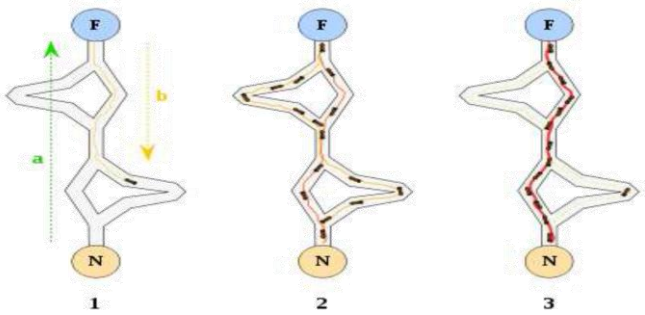


B. Dynamic Load Balancing Algorithms

In dynamic algorithm the lightest server in the whole network or system is searched and preferred for balancing a load For this real time communication with network is needed which can increase the traffic in the system. Here current state of the system is used to make decisions to manage the load.

The three methods are:

- Simulated Annealing: We directly minimize the above cost function by a process analogous to slow physical cooling
- Orthogonal Recursive Bisection: A simple method which cuts the graph into two by a vertical cut, then cuts each half into two by a horizontal cut, then each quarter is cut vertically, and so on.
- Eigenvector Recursive Bisection: This method also cuts the graph in two then each half into two, and so on, but the cutting is done using an eigenvector of a matrix with the same sparsity structure as the adjacency matrix of the graph.

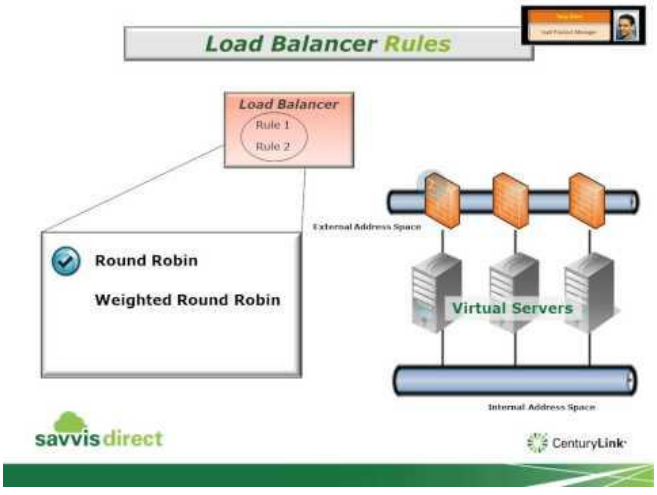


III. STATIC LOAD BALANCING ALGORITHMS

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more imperfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic.

A. Round-Robin Load Balancer

It is a static load balancing algorithm, which does not take into account the previous load state of a node at the time of allocating jobs. It uses the round robin scheduling algorithm for allocating jobs. It selects the first node randomly and then, allocates jobs to all other nodes in a round robin manner. Since the running time of any process is not known prior to execution, there is a possibility that nodes may get heavily loaded. This algorithm will not be suitable for cloud computing because some nodes might be heavily loaded and some are not.



B. Min-Min

It is a static load balancing algorithm. So, all the information related to the job is available in advance. Some terminology related to static load balancing:

- ETC: If a job is not executable on a particular node, the entry in the ETC matrix is set to infinity. The expected running time of the job on all nodes are stored in an ETC (expected time of compute) matrix.

- OLB: It provides load balance schedule but it results in very poor make-span. It is an opportunistic Load Balancing, in which each job is assigned to the node in an arbitrary order, irrespective of the ETC on the nodes.
- MET: In this, each job is assigned to the node which has the smallest execution time as mentioned in ETC table, regardless of the current load on that processor. MET tries to find the best job-processor pair, but it does not take into consideration the current load on the node: It is a Minimum Execution Time algorithm.
- MCT: It is a Minimum Completion Time algorithm which assigns jobs to the node based on their minimum completion time.

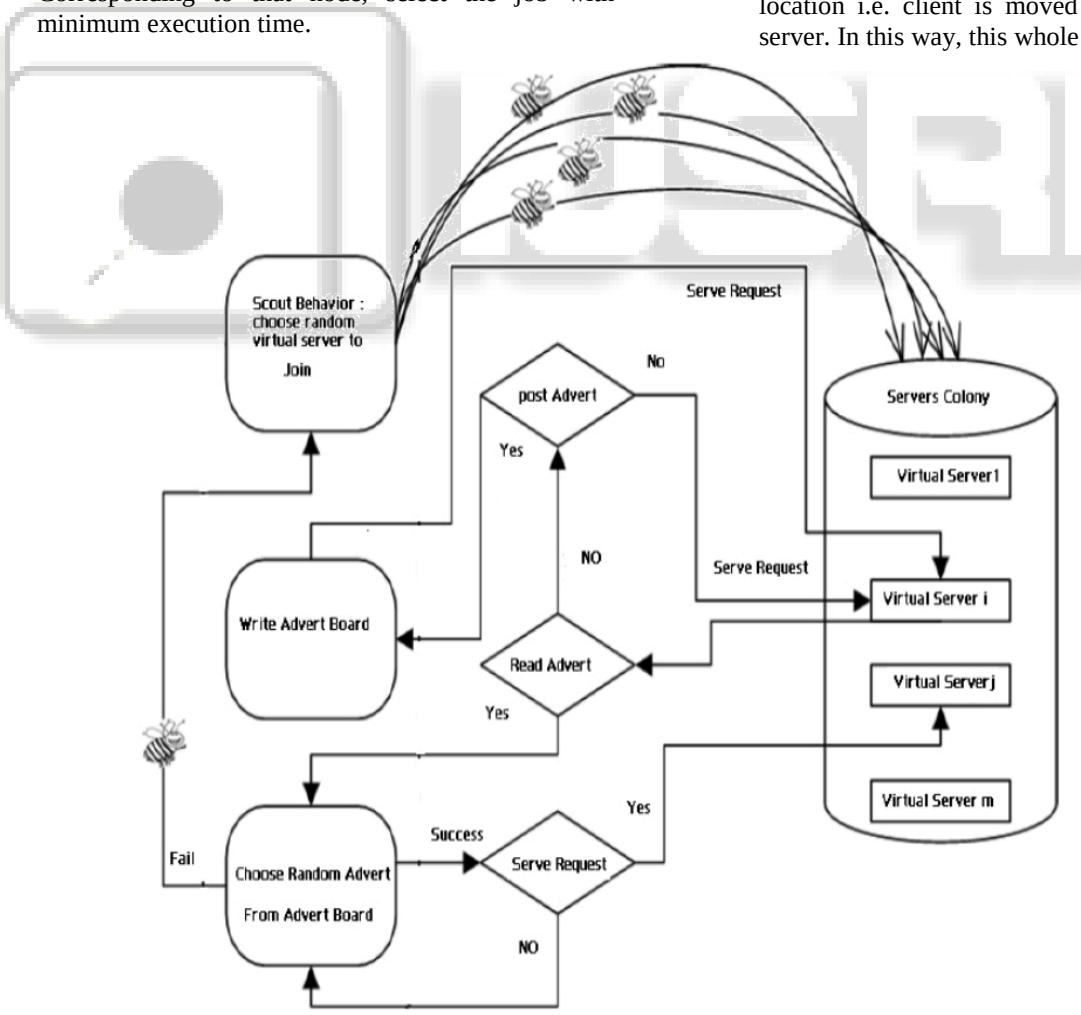
C. Load Balance Min-Min

LBMM is a static load balancing algorithm. This algorithm implements load balancing among nodes by considering it as a scheduling problem. The main aim of this algorithm is to minimize the make-span, which is calculated as the maximum of the completion times of all the jobs scheduled on their respective resources. This algorithm performs the following steps for scheduling the jobs on the nodes.

- Execute the Min-Min scheduling algorithm and calculate the make-span.
- Select the node with the highest make-span value.
- Corresponding to that node, select the job with minimum execution time.
- The completion time of the selected job is calculated for all the resources.
- Maximum completion time of the selected job is compared with the make-span value. If it is less, the selected job is allocated to the node, which has the maximum completion time. Else, the next maximum completion time of the job is selected and the steps are repeated.
- The process stops if all the nodes and all the jobs are assigned.

IV. EXISTING LOAD BALANCING TECHNIQUES

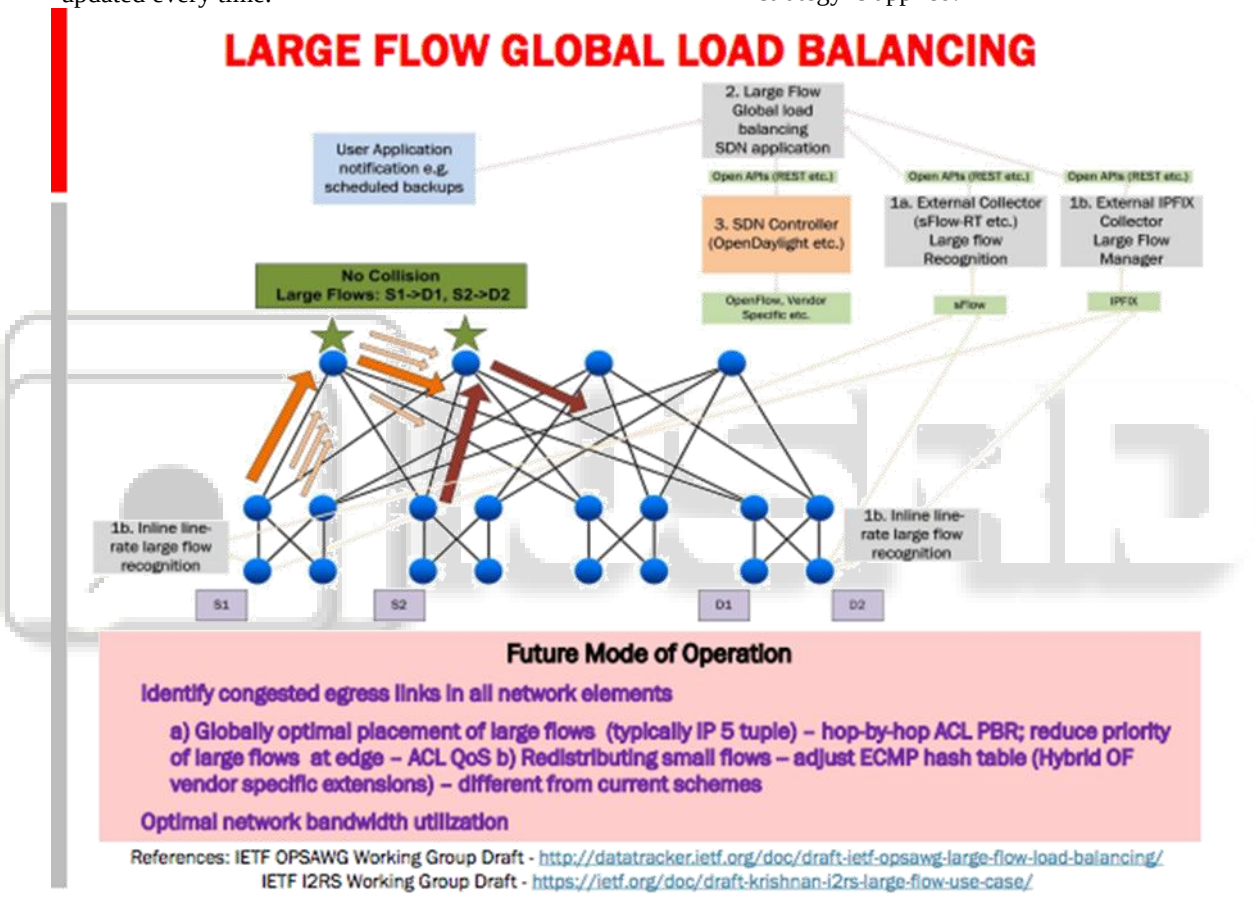
1) Honey Bee Foraging Algorithm: This whole algorithm is based on the process of honeybees finding the food and alarming others to go and eat the food. First forager bees go and find their food. After coming back to their respective beehive, they dance. After seeing the strength of their dance, the scout bees follow the forager bees and get the food. The more energetic the dance is, the more food available is. So this whole process is mapped to overloaded or under loaded virtual servers. The server processes the requests of the clients which is similar to the food of the bees. As the server gets heavy or is overloaded, the bees search for another location i.e. client is moved to any other virtual server. In this way, this whole technique works.



2) Task Scheduling Algorithm based on Load Balancing: Y. Fang et discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing

by first mapping tasks to virtual machines and then virtual machines to host resources.

3) Throttled Load Balancing Algorithm: This algorithm makes use of identity of virtual machines. Client requests the ID of virtual machine. Throttled load balancing algorithm returns that ID to the user.
4) Ant Colony Optimization Technique: In this technique, a pheromone table was being designed which was updated by ants as per the resource utilization and node selection formulae. Ants move in forward direction in search of the overloaded or under loaded node. As the overloaded node is traversed, then ants move back to fill the recently encountered under loaded node, so a single table is updated every time.
5) Role Based Access Control (RBAC): RBAC is a technique used to reduce the load of the cloud. In this, a role is assigned to each user so that limited applications of the cloud can be accessed by their respective number of users. So by this approach, the resources are restricted to the users.
6) Resource Allocation Scheduling Algorithm (RASA): In this algorithm, virtual nodes are created first. Then the expected response time of each virtual node is found. Then according to the least loaded node criteria, efficient virtual node is found and ID of that node is returned to the client. In this, Min-Min and Max-Minstrategies are followed. If number of resources available are odd, then Min-Min strategy is applied else Max Min strategy is applied.



V. CHALLENGES FOR LOAD BALANCING

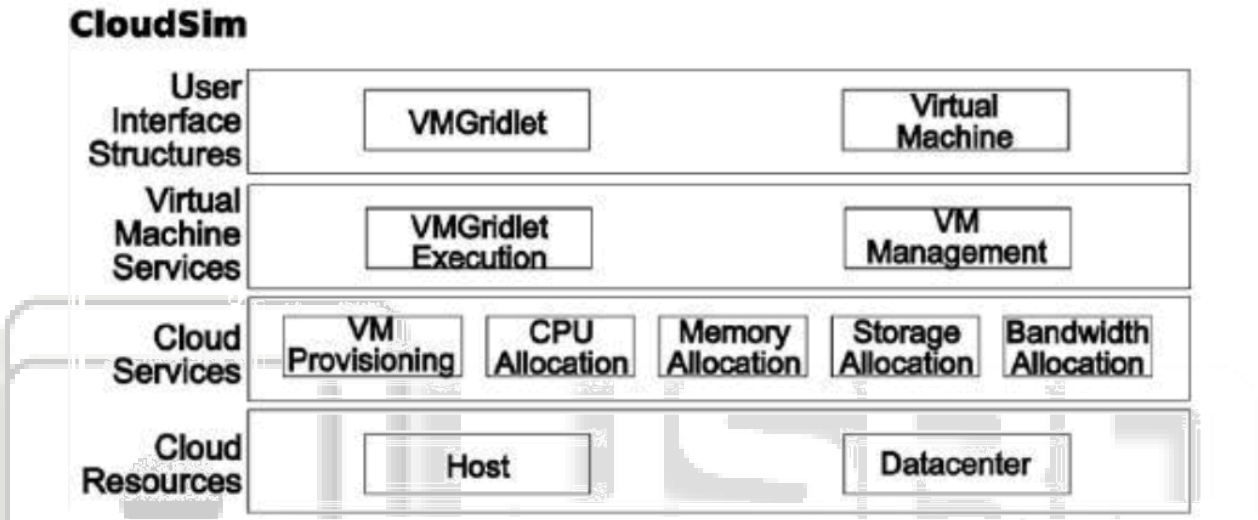
- Throughput: It is the total number of tasks that have completed execution for a given scale of time. It is required to have high through put for better performance of the system.
- Associated Overhead: It describes the amount of overhead during the implementation of the load balancing algorithm. It is a composition of movement of tasks, inter process communication and inter processor. For load balancing technique to work properly, minimum overhead should be there.
- Fault tolerant: We can define it as the ability to perform load balancing by the appropriate algorithm without arbitrary link or node failure. Every load balancing algorithm should have good fault tolerance approach.
- Migration time: It is the amount of time for a process to be transferred from one system node to another node for execution. For better performance of the system this time should be always less.
- Response time: In Distributed system, it is the time taken by a particular load balancing technique to respond. This time should be minimized for better performance.
- Resource Utilization: It is the parameter which gives the information within which extant the resource is utilized. For efficient load balancing in system, optimum resource should be utilized.
- Scalability: It is the ability of load balancing algorithm for a system with any finite number of processor and machines. This parameter can be improved for better system performance.

– Performance: It is the overall efficiency of the system. If all the parameters are improved then the overall system performance can be improved.

VI. CLOUD SIMULATOR- CLOUDSIM

Simulation environment allows customers or users to tune the performance bottlenecks or evaluates different kinds of features under varying load distributions. Different kinds of functionalities of CloudSim are presented in the following.

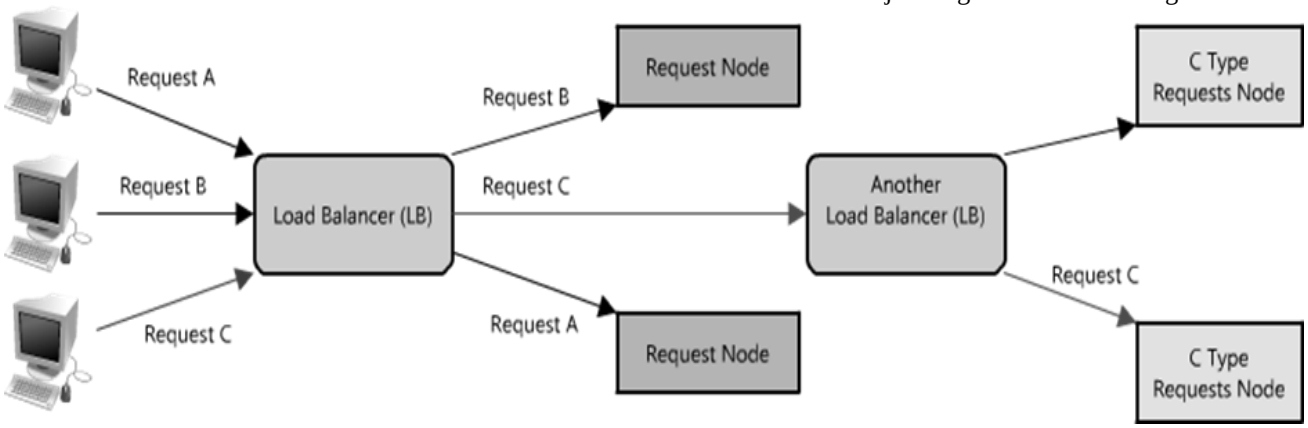
– support for modelling and simulation of large scale cloud computing data centers
– support for modeling and simulation of virtualized server hosts, with customizable policies for provisioning host resources to virtual machines
– support for modeling and simulation of energy-aware computational resources
– support for modeling and simulation of datacentre network topologies and message-passing applications
– support for modeling and simulation of federated clouds
– support for dynamic insertion of simulation elements, stop and resume of simulation
– support for user-defined policies for allocation of hosts to virtual machines and policies for allocation of host resources to virtual machines Besides these above-mentioned functionalities, while developers or researchers, uses CloudSimfeatures, need not to think about the lower level details of cloud based infrastructure and services. The architecture of CloudSim comprises of four layers, as shown in fig.



VII. GOALS OF LOAD BALANCING ALGORITHM

In order to balance the requests of the resources it is important to recognize a few major goals of load balancing algorithms:

– Cost effectiveness: primary aim is to achieve an overall improvement in system performance at a reasonable cost.
– Scalability and flexibility: the distributed system in which the algorithm is implemented may change in size or topology. So the algorithm must be scalable and flexible enough to allow such changes to be handled easily.
– Priority: prioritization of the resources or jobs need to be done on beforehand through the algorithm itself for better service to the important or high prioritized jobs in spite of equal service provision for all the jobs regardless of their origin.



VIII. CONCLUSION

As the Cloud Computing is an alluring concept in the present and upcoming time, the researchers had developed various techniques to cope up with the load balancing problem being faced while working with cloud computing. This paper gives an overall description of various distributed load balancing algorithms that can be used in case of clouds. The algorithms used in the cloud computing for load

balancing, this information might be useful in the research associated with cloud computing. One of the major issues of cloud computing is load balancing because overloading of a system may lead to poor performance which can make the technology unsuccessful. So there is always a requirement of efficient load balancing algorithm for efficient utilization of resources. Our paper focuses on the various load balancing algorithms and their applicability in cloud computing environment.

REFERENCE

[1] Load Balancing In Cloud Computing Systems By Ram Prasad Padhy ,P Goutam Prasad Rao.
[2] Load Balancing Algorithms In Cloud Computing By Doddini Probhuling L.
[3] Analysis Of Load Balancers In Cloud Computing By Shanti Swaroop Moharana1, Rajadeepan D. Ramesh2 & Digamber Powar3
[4] Execution Analysis Of Load Balancing Algorithms In Cloud Computing Environment By Soumya Ray And Ajanta De Sarkar
[5] Comparison Of Load Balancing Algorithms In A Cloud By Jaspreet Kaur
[6] Comparative Study On Load Balancing Techniques In Cloud Computing By N. S. Raghava* And Deepti Singh

