



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: B CLOUD AND DISTRIBUTED

Volume 19 Issue 1 Version 1.0 Year 2019

Type: Double Blind Peer Reviewed International Research Journal

Publisher: Global Journals

Online ISSN: 0975-4172 | Print ISSN: 0975-4350 | DOI: 10.17406

Measuring the Performance on Load Balancing Algorithms

By Saumendu Roy, Dr. Md. Alam Hossain, Sujit Kumar Sen, Nazmul Hossain
& Md. Rashid Al Asif

Jashore University of Science & Technology (JUST)

Abstract- Load balancing is an integrated aspect of the environment in cloud computing. Cloud computing has lately outgoing technology. It has getting exoteric day by day residence widespread chance in close to posterior. Cloud computing is defined as a massively distributed computing example that is moved by an economic scale in which a repertory of abstracted virtualized energetically. The number of clients in cloud computing is increasing exponentially. The huge amount of user requests attempt to entitle the collection for numerous applications. Which alongside with heavy load not far afield off from cloud server. Whenever particular (Virtual Machine) VMs are overloaded then there are no more duties should be addressed to overloaded VM if under loaded VMs are receivable. For optimizing accomplishment and better response or reaction time the load has to be balanced between overloaded VMs (virtual machines). This Paper describes briefly about the load balancing accession and identifies which is better than others (load balancing algorithm).

Keywords: *cloud computing, load balancing, virtualization, load balancing algorithms.*

GJCST-B Classification : *C.m*



Strictly as per the compliance and regulations of:



Measuring the Performance on Load Balancing Algorithms

Saumendu Roy^α, Dr. Md. Alam Hossain^σ, Sujit Kumar Sen^ρ, Nazmul Hossain^ω & Md. Rashid Al Asif[¥]

Abstract- Load balancing is an integrated aspect of the environment in cloud computing. Cloud computing has lately outgoing technology. It has getting exoteric day by day residence widespread chance in close to posterior. Cloud computing is defined as a massively distributed computing example that is moved by an economic scale in which a repertory of abstracted virtualized energetically. The number of clients in cloud computing is increasing exponentially. The huge amount of user requests attempt to entitle the collection for numerous applications. Which alongside with heavy load not far afield off from cloud server. Whenever particular (Virtual Machine) VMs are overloaded then there are no more duties should be addressed to overloaded VM if under loaded VMs are receivable. For optimizing accomplishment and better response or reaction time the load has to be balanced between overloaded VMs (virtual machines). This Paper describes briefly about the load balancing accession and identifies which is better than others (load balancing algorithm).

Keywords: cloud computing, load balancing, virtualization, load balancing algorithms.

I. INTRODUCTION

Cloud computing is extensively received IT-based service. However, there is a number of issues that have not been fully figured out such as load balancing, real-time scheduling, VM migrations and many more. The load balancer is a device that distributes application or network traffic across a bunch of servers. The load balancing enhances responsiveness and raises the appearance of its. A load balancer lies between the server farm and the client. This is responsible for receiving oncoming network traffic and application. Also distributing the traffic across

various backend servers using different techniques. By balancing the requests of application across numerous servers, a load balancer diminishes the load of the single server. Side by side it hinders any one application server from fitting an odd point of failure. Thus flourishing overall application responsiveness and availability.

Balancing of the load is a vital issue in the cloud that discuss with the skilled and scalable distribution of workload. Side by side it takes care that all the computing materials must be distributed. The number of requests and consumers for the services are increasing in cloud computing day by day. For this reason, load balancing is the significant research area for conducting the user's requests efficiently. The load can be network load, memory capacity, CPU load or delay. Load balancing is the process of allocating the work-load among different nodes of a distributed system. It enhances both the job response time and resource utilization. While also ignoring a situation where a number of the nodes are loaded while several nodes are inactive or doing the very tiny amount of work. Load balancing assures that every node or all the processor in the system of the network does nearly the equable number of work at any required time. This procedure can be a sender, receiver-initiated or symmetric type (a coalition of sender-initiated and receiver-initiated types). For skilled & fruitful management and behaviors of cloud service provider's resources, various algorithms of load balancing have mentioned. Additionally, the motives are to minimize resource usage, maximize throughput and reduce response time - two genres of load balancing algorithms that are static and dynamic. Static load balancing algorithms are easy to design and take very less execution time. This algorithm works on the base of the knowledge of tasks and resources of the system. But these kinds of algorithms can only be practical in the scenario, where most of the hosts in the cluster have identical processing capability. Examples of static load balancing algorithms are least connection scheduling, Round Robin, weighted round robin algorithm and many more. Dynamic load balancing algorithms are more supple and reliable. Which also capable of handling a great number of user's requests than static algorithms. These algorithms stand on the current state of the system and are the best suited for switching environment.

Author α: Lecturer at the department of Computer Science & Engineering in Nothern University of Business & Technology Khulna, Bangladesh. e-mail: saumocse3j6@gmail.com

Author σ: Assistant Professor at the department of Computer Science & Engineering in Jashore University of Science & Technology (JUST), Jashore, Bangladesh. e-mail: alamcse_iu@yahoo.com

Author ρ: Student at the department of Computer Science & Engineering in Jashore University of Science & Technology (JUST), Jashore, Bangladesh. e-mail: sujit2j8@gmail.com

Author ω: Assistant Professor at the department of Computer Science & Engineering in Jashore University of Science & Technology (JUST), Jashore, Bangladesh. e-mail: n.hossain@just.edu.bd

Author ¥: Lecturer at the department of Computer Science & Engineering in Barisal University, Barisal. e-mail: rashid.al.asif@gmail.com

II. LOAD BALANCING

Cloud computing is the outbound internet based procedure which strengthens commercial computing. Cloud is a podium providing dynamic pool resources and virtualization. It also simplifies the scalable IT resources such as infrastructure, services and, applications. These materials work with internet on pay-per-use criteria. Which aids in the coordination of retention in a quick way. The cloud computing steps both data and computing from portable desktop and PCs to the data centers. Cloud computing can synthesize on need changes. Which in turn decreases the expense of capital required in hardware and software. Thus, cloud computing helps to provide a structure for compatible access to computing resources and that too in on-demand style. By excellence of cloud computing, resources can be accepted and shifted very firstly. As well as the interaction with less service provider. Cloud computing also enhances the availability of resources. The key point on cloud computing is scheduling (RAS) and resource allocation which is gained by using its strategies and algorithms. These have a straight outcome on cloud performance and cost. Load balancing is one of the vital concerns on cloud computing [2], i.e. When one or more elements of any job failed, it (load balancing) helps in the regularity of the services. By accomplishing provisioning and de-provisioning of examples of applications without any fail. Thus, Load Balancing is a technique for distributing the dynamic local workload evenly across all the nodes in the entire cloud. It will also ignore the condition where some nodes are loaded. While others are inactive or doing very little amount of work. Load balancing maximizes the overall performance of the system alongside with its resource utilization characteristics. This working theory of load balancing helps to gain high user contentment. By enhancing the overall system performance of the skilled distribution of every resource can be done. In cloud computing outlay of resources and preservation of energy are not considered. But their insertion along with exact load balancing helps in reducing it and forming enterprises greener. Another important criterion is the scalability in cloud computing that is enabled by load balancing. All these criteria raise resource utilization in such a way that will minimize energy outgoing and carbon foot prints, resulting in acquiring green computing.

III. WORKING PROCESS OF LOAD BALANCING

Load balancing in clouds is a procedure that allocates the interperence dynamic local workload evenly across all the nodes. It is used for gaining a service provisioning and resource utilization rate and therefore improving the overall system performance for

incoming tasks. Which are coming from various positions are accepted by the load balancer and then allotted to the data center, for the accurate load distribution.

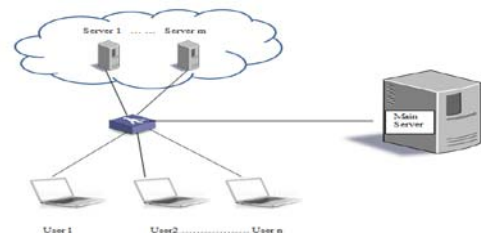


Fig. 1: Load Balancing

The target of load balancing is as follows:

- To improve service availability
- To raise the user pleasure
- To maximize resource utilization
- To minimize the waiting time and the time of task coming from a various position
- To increase the performance
- To maintain the stability of the system
- To build system fault tolerance
- To accommodate future modification

a) Central Load Balancing Decision Module (CLBDM)

To give valued information and force the decision-making method of a load balancer, thus sustaining satisfactory load balancing in cloud environments. It is not sufficient to give information from the networking part of the computer system or from the exterior load balancer. Including weighted round-robin algorithm and session-switching. So that it can insist sessions on one of the web application servers. Which can catch all data for that session attainable until it is concluded manually or it just timeouts. CLBDM organizes information from virtualized environments and ends user experience. To be able to proactively force load balancing decisions or reactively switch it in handling critical conditions.

CLBDM solves the problem depicted in the weighted round-robin algorithm and session-switching. In a fashion that introduces a central module that is forcing decisions made by load balancers.

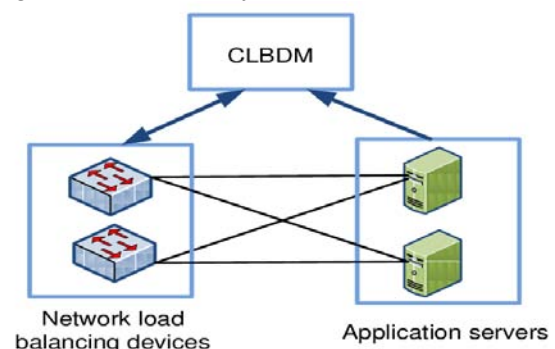


Fig. 2: CLBDM-Placement in-a-computer

The benefit of this module is to interact (monitor) whole parts of our computer system, with application servers and load balancers. Then, based on the gathered information and interior computation, CLBDM will interference forwarding decisions on load balancers. Facilitated designed is shown in Figure-2, which also delineates the direction of data flooding to and from CLBDM.

CLBDM is submitted with other data to fine tune of a user experience sensor. Which can be accomplished as an unresisting taping device or software on the application servers. This sensor is controlling the session of users including whole requests sent and received by end the user. And metering time wanted for every action to finish. If the tendency of response time for end user on an individual application server will launch to rise, this is a crystal clear signal that something is running incorrectly. When collected information rises above inartificial numbers, CLBDM is getting action and steps load to other application servers.

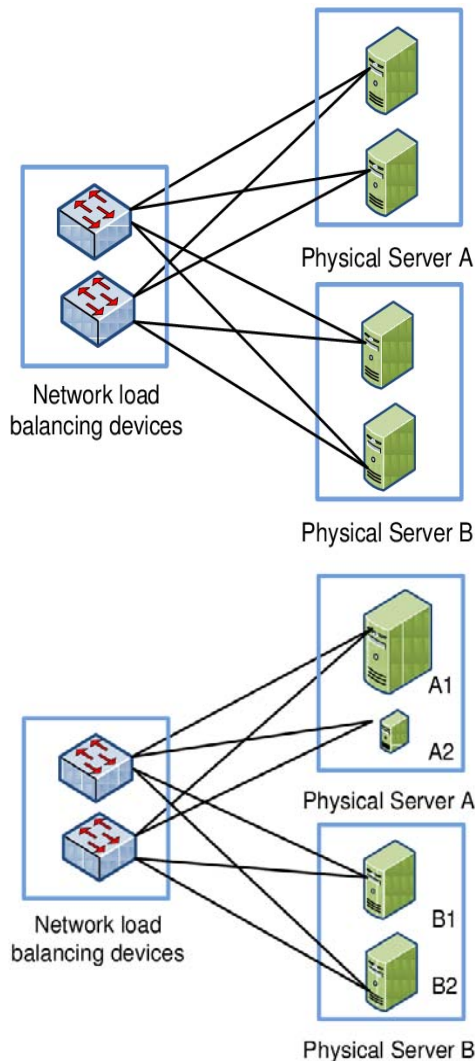


Fig. 3: Dividing & Reallocation-of-resources on physical-servers-in-several-smaller-virtualized-servers

Figure 3, can severely reduce the number of potentially attacked users. The idea here is to raise the number of application servers by using virtualization technology; it would be contingent on increasing amount of available application servers from two to four. This would instantly decrease the number of potentially attacked users from 50% to 25%. Virtualization technology that gives a dynamical allocation of resources. Which is used on virtualized servers with live migration. A move that CLBDM can make when it reveals issue with specific virtual server. It would dynamically allocate resources from one virtual server (A2) to another (A1). It is thus giving the possibility to server A1 to still complete action that has used all of its resources. But, at this time CLBDM will do even more activities. For example, it would point to load balancers that the server A1 is in trial and that it cannot take new user sessions. This thought will dynamically be returned if CLBDM gets information from server A1 that the load has been normalized.

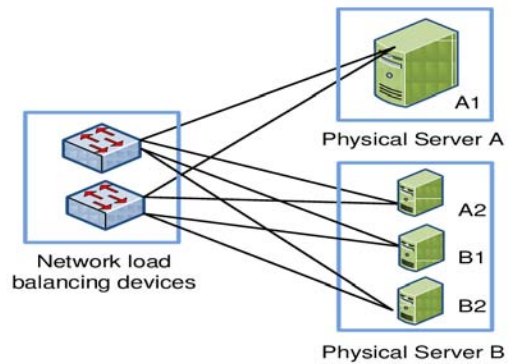


Fig. 4: Virtualization technology to live migrate virtual Server A2 to Physical Server B

Figure 4, exhibits an even further development of the formerly depicted condition. Where the load on Server A1 is still growing, now CLBDM has a choice that allows virtualized servers to be shifted from one physical server to another. It is accurately that here has occurred in a way that Server A2 has been live migrated (without any pause) to Physical Server B. While the Server A1 now has full resources of Physical Server A, giving it even more probabilities to complete jobs presently running on it.

b) Drawbacks of CLBDM Model

One of probable fact is that CLBDM can turn in a way, that it can enter unexpected loops and start to spray its decision between nodes. It is resulting in low performance and end-user experience. This kind of facts can be reduced if not ignored by cautiously designing CLBDM decision algorithms. But there should also be ignored if it is possible to find such an unwanted state.

IV. VIRTUALIZATION

Virtualization means that do not exist in, but it distributes everything as actual or real. The software implementation of a machine is Virtualization that will run different programs as an actual machine. By virtualization, the end user can use the various services or applications of the cloud. For this reason, this is the master part of the cloud environment. There is a number of kind's virtualization that is used in the cloud.

Two kinds of virtualization. They are:

- a. Full Virtualization
- b. Para virtualization

a) Full Virtualization

Full virtualization which means that a machine is installed on another machine. This machine (virtual) gives all the functionalities that exists on the original machine. It facilitates when real machine not independent then the user uses the virtual machine.

b) Para virtualization

Para virtualization, the hardware accommodates many operating systems to run the only machine. It also allows skilled use of resources such as processor and memory.

V. BASIC TYPES OF LOAD BALANCING ALGORITHMS

There is an excessively large necessity for load balancing in large distributed and complicated systems. Load balancer gets a decision to shift the job to the distant server for balancing. The load balancer which can work in two ways: Cooperative and non-cooperative are the two ways. A cooperative way, to gain the satisfactory response times, entire nodes work together. A non-cooperative way, response time is incremented by the freely running the tasks. Some load balancing algorithms are smeared in this paper.

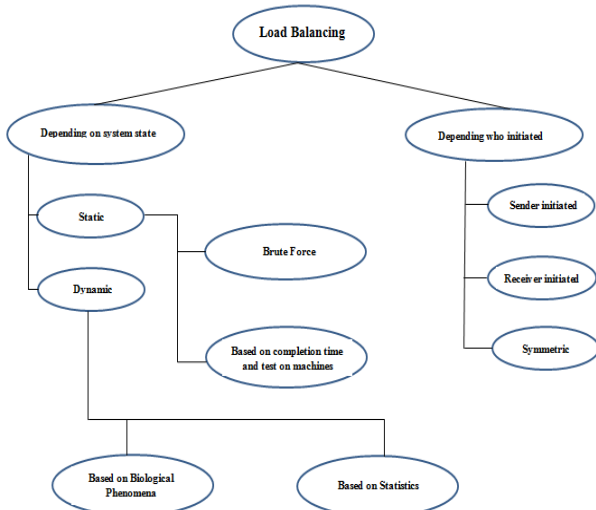


Fig. 5: Basic Types of the algorithm

- Based on the recent status of the system, the algorithms of load balancing can be categorized in two ways:
- Static algorithm: The present situation of the node is not taken into estimation [3]. All the nodes and their features are known in advance. Based on this earlier sense, the algorithm works. Since it does not use recent system status information, it is simple to implement.
- Dynamic algorithm: This type of algorithm is the basis for the recent situation of the system [3]. The algorithm works following to the dynamic changes in the instance of nodes. Status Table controls the status of the entire nodes in the cloud environment. Dynamic algorithms are not easy to implement, but it balances the load in a useful manner.

The initiator of algorithms, Load Balancing algorithms can be divided into three types:

- Sender-Initiated: Sender recognizes that the nodes are in number so that the sender initiates the perfections of load balancing algorithm.
- Receiver Initiated: The necessity of Load balancing condition can be recognized by the receiver/server in the cloud. And that server initiates the accomplishment of Load Balancing algorithm.
- Symmetric: It is the summation of both types that means the initiator of sender and receiver.

VI. EXISTING LOAD BALANCING ALGORITHMS

Following load balancing algorithms are currently prevalent in cloud.

a) Round Robin

In this algorithm [6] the round robin technique pursues a time plot or slice while processing the data. Every process is going to perform in the time slice and then change to other and follow on ring the path. In round robin, until all processes completed their task, a balancing procedure is followed so that it can balance the process in a group. The process is going to happen in round robin until all processes finish their task. Such that a balance method is implemented to equal the process in a group pattern. This algorithm is extensively used in web servers where HTTP requests are very analogous and distributed equally.

b) Min-Min

The manager of the cloud recognizes the minimum completion and execution time of the unassigned each task. Which are waiting to execute and store in a queue. Min-min is static load balancing algorithm. So the metrics concerned to the job are known in advance. In these types of the algorithm, the cloud manager firstly concerns with the jobs having minimum performance time. By bestowing those to the corresponding processors with the capability of

complete the job in fixed finishing time. The jobs have to wait for the un-specific period because of maximum execution time. In the processor, until all the tasks are assigned, the assigned tasks are updated in the processors. As well as the task is cleared and the process is periodic till all the tasks are mapped in the waiting queue. Min-min starts by a set of all unassigned jobs and found each task minimum finishing time. Amongst these least times, the minimum value is chosen. That is the minimum time between all the tasks on any resources. Then following to that minimal time, the task is cadastral on the corresponding processor. After then the perfection time for all other jobs is updated on that machine. It is done by adding the execution time of the assigned task to the execution times of other task on that machine. And finally, assigned task is dismissed from the list of the jobs that are to be assigned to the machines. Then again the similar method is followed until all the tasks are assigned on the resources. But this way has a drawback that it can lead to starvation.

c) *Opportunistic Load Balancing Algorithm (OLB)*

It is a static load balancing algorithm. For this reason, it does not consider the recent workload of the Virtual Machines. Its effort to repose each node busy. This algorithm deals rapidly with the unexecuted tasks in indiscriminate order to the recently available node. Every task is assigned to the node lamely. It gives a load balance scheme without good results. The task will process sluggishly because it does not compute the present execution time of the node. Advantages that it endeavors' to keep each node engaged. This algorithm deals rapidly with the unexecuted tasks in random order to the recently available node. Every task is assigned to the node randomly. Disadvantages that it provides load balance scheme with poor results. The task will process in a slack manner because it does not compute the current execution time of the node.

d) *Max Min algorithm*

Max Min algorithm which works as similar to the Min-Min algorithm without the following: after discovering the minimum execution time, the cloud manager handles with tasks having maximum execution time. The assigned task is dismissed from the list of the tasks. Which are to be assigned to the processor and the working time for all other tasks is updated on that processor. Because of its static way the necessity are informed in advance then the algorithm completed well. An improved rending of max-min algorithm is mentioned in [7]. It is the basis of the cases, where meta-tasks include kindred tasks of their execution and completion time. Development in the ability of the algorithm is gained by increasing the scope of concurrent execution of tasks on resources. Advantages of Max-min technique which resolves the rigidity of Min-min. By giving Precedence to a large number of tasks. The Max-

min algorithm picks the task with the Maximum completion time and assigns it to the resources on which it acquires minimum execution time. It is crystal clear that the Max-min feels better option. Whenever the number of little tasks is much more than large ones, one of the properties of the Max-min method is that selects large tasks to be completed quickly. Which in turn little task stays for high time and it is the Disadvantages of Max-Min.

e) *Honeybee Foraging load balancing Algorithm*

The idea astern the BCO (Bee Colony Optimization) is to make the multi-agent system (colony of artificial bees) able to effectively solve stiff combinatorial optimization problems. A colony of honey bee can enhance itself over long ranges as to trace multiple food sources. Like as flower patches and then these bees harvest nectar or faring from these sources. A little amount of the colony discovers the environment looking for new flower patches when the food source is encountered the scout bees. Going in the field that surrounds the hive and control for quality beneficial. When they back to the hive, the scouts gather the food harvested. There is a location in the hive named as the "dance floor." Where waggle dance is completed by the bees that found a very profitable food. By the dance (waggle) a scout bee goes through the location of its search to idle viewers, which aids in the using of the flower patch. Here the term of the dance is following to the scout's rating of the food source. To harvest the best-rated flower patches, more foragers get recovered. When the dance is finished, the scout back to the food source. It found to see more food. Till the food is profitable, food sources will be posted by the scouts when they back to their hive. Foragers who are recovered currently may waggle dance as well. Those will step-up the recruitment for profitable flower patches. This method will go on to discover the most profitable flower patches. The Honey Bee nature stimulated load balancing algorithm was mentioned, which aims to gain well-balanced load cross VMs to maximize the throughput. And side by side, to balance the precedence of tasks on the VMs. So, the amount of waiting time of the tasks on the list is minimum. Using this algorithm reduction in waiting time and average execution time of tasks on the list was improved. This algorithm works for heterogeneous type of systems. And for balancing non-preemptive independent tasks. Some advantages that it has low overhead, less migration time, less response time and optimum resource utilization. On the other side, it has some disadvantages that it has fewer throughputs, less fault tolerant and less scalability.

f) *Active Clustering*

The idea of clustering in cloud computing is active Clustering is a clustering based algorithm. By making a bunch or cluster of nodes, the performance of

an algorithm can be increased. Every cluster can be occupied as a group. The main idea behind an active clustering is to group the same nodes together and then work on these groups. The method of forming a cluster revolves around the idea of the node. In this technique, the first node points a neighbor node named the matchmaker node which is of a variant type. This matchmaker node creates a connection with its surroundings, which is of similar types as the earlier node. At last the matchmaker node gets separated. This technique is followed repetitively. The system performance is increased with handsome availability of resources. Thereby enhancing the throughput. This increment in throughput is due to the proficient utilization of resources.

g) *Ant colony optimization*

The main idea of ACO is to copy the foraging characteristics of ant colonies. When ant groups try to find for the food, they use an exceptional type of chemical to link with each other. That, chemical is named as pheromone. At the very fast ants begin to search their foods loosely. Once the ants find a way to the food source, they consent pheromone on the path. To sense pheromone on the ground, an ant can pursue the trails of the other ants to the food source. As this method continues, maximum ants attract to select the shortest path as there have been a massive amount of pheromones congealed on this path. This united pheromone is saving and pheromone following characteristic of ants turn the illuminative source of ACO.

h) *OLB+LBMM*

Opportunistic Load Balancing and Load Balance Min-Min are referred as OLB and LBMM algorithms to use better-executing efficiency. And control the system of load balancing. This linked way helps in a skilled utilization of resources and increases work efficiency. It provides a better output than the above-argued algorithms. It is the summation of Opportunistic Load Balancing and Load Balance Min-Min scheduling algorithms to use better execution efficiency. And control the load balancing of the system. OLB scheduling algorithm stays every node in active state to acquire the aim of load balance. And also the algorithm of LBMM scheduling is used to reduce the completion time of every task on the node. Thereby reducing the total completion time. This algorithm employs to increase the utilization of resources and increases the work efficiency.

i) *Biased Random Sampling load balancing Algorithm*

Biased Random Sampling Load Balancing Algorithm is a dynamic approach. The network is presented in the form of a virtual graph. Each server is taken as a vertex of the node and the in degree represents the available free resources the nodes have.

By the in degree, the load balancer distributes the job to the node. The nodes have at least one in degree then load balancer distributes the job to that node. When the job is allocated to the node then the in degree is decrement by one. And it's get incremented again when the job gets executed. Random sampling technique is used in the addition and deletion of the processes. The processes are centralized by the threshold value, which points the maximum traversal from one node to the destination node. The length of traversal is known as walk length. Surrounding the node of the present node is selected for the traversal. After getting the request, load balancer selects a node loosely and compares the present walk length with the threshold value. If the recent walk length is greater than or equal to the threshold value, the job is completed at that node. Otherwise, the walk length of the job is increased and another neighbor node is selected randomly. The performance decreases as the number of servers increase.

j) *Generalized Priority Algorithm*

In this algorithm [19] the tasks are prioritized according to the size of the tasks. Such that the task with the highest size gets the highest priority in the system and execute first at its best. Also, the virtual servers are prioritized according to their million instructions per second (MIPS) value in the virtual server distribution system. Such that the Server with the highest MIPS value gets the highest priority. Hence the load balancing is done accordingly and, it gets the maximum utilization of the resources according to the data size in progress.

k) *Join-Idle-Queue*

Load balancing algorithm of Join-Idle-Queue proposed by Y. Lua et al. [20] which is an algorithm for web services and systems. It simplifies the large scale load balancing with distributed dispatchers. In each dispatch firstly load balancing algorithm idles the processors for the availability. And then does allotment of the task to processors in such a way that reduces the queue length at each server. This algorithm removes the load balancing work from the troublesome path of request processing, which helps in effective reduction of the system load.

l) *Genetic Algorithm based Load Balancing*

It is introduced by KousikDasgupta and BrototiMandal [21]. This algorithm prospers to balance the load of the cloud infrastructure at the time of trying to shorten span of a job. Genetic-based approach follow some rules and randomization according to the network load effectively.

m) *Stochastic Hill Climbing Technique*

KousikDasgupta and BrototiMandal proposed [23] a novel load balancing technique by utilizing the algorithm of Stochastic Hill Climbing. It selects randomly

to form the uphill moves with effective possibility-author of a local optimization way Stochastic Hill climbing utilize the resources. And the algorithm is used for distribution of incoming jobs to the virtual machines (VMs) or servers.

n) *Decentralized Content Aware Load Balancing*

A different content animate load balancing method known as Workload and Client Aware Policy (WCAP) that is proposed by H. Mehta et al. [24] which explain the identical and exceptional feature of the requests. As well as computing nodes by an identical and exceptional feature (USP). USP aids the scheduler in deciding the best and fit resources to execute the process. This technique had less overhead and implemented in a decentralized manner. This technique enhances the searching performance by using the kernel or content information. It also improves the utilization of resources by shortening the lazy time of the computing nodes.

o) *Server-Load Balancing for Internet Distributed Services*

A. M. Nakai et al. [25] presented a distributed server-based technique for web servers. It facilitates the reduction in service response time by using a protocol that bounds the redirection of requests to the closest remote servers without overloading them. Middleware is used in this technique to implement this protocol. To endure overload, the web server uses Heuristic. Heuristic scheme provides a surety to get the balance of load based on the job size. And also guarantee to not get a repetition of same size job in a single node.

VII. TESTING PARAMETERS

Existing load balancing procedures have been measured by the metrics that are discussed below:

a) *Throughput*

This metric is used to calculate the whole number of tasks, whose completion has been finished successfully. For overall system performance high throughput is needed.

b) *Overhead*

Overhead combined with any load balancing algorithm prefaces the extra cost engaged in implementing the algorithm. Overhead Associated specifies the amount of overhead involved while completing a load-balancing algorithm. It covers overhead due to movement of tasks, inter processor and inter-process contact. It should be as low as possible.

c) *Fault Tolerance*

Measuring the ability of an algorithm to execute identical load balancing in case of any shortcoming. It must be highly faulted.

d) *Migration Time*

It is described as, the entire time needed in migrating the resources or jobs from one node to another. It should be reduced or minimized.

e) *Response Time*

Response time can be determined as, the time interim between sending a request and getting its response. To boost the overall performance, it should be minimized.

f) *Resource Utilization*

It is used to assure the accurate utilization of all those resources, which covered the entire system. This issue must be optimized to have an efficient load balancing algorithm.

g) *Scalability*

It is the ability to perform uniform load balancing in a system with the rise in the number of nodes, according to the requirements. Higher scalability is preferred.

h) *Performance*

It is used to investigate, how efficient the system is. This has to be uplifted at a reasonable cost, e.g., reducing the response time though keeping the receivable delays.

VIII. COMPARISON AMONG DIFFERENT ALGORITHMS

Table 1: Parameters Performance among various algorithms

	Throughput	Response Time	Resource Utilization	Overhead	Fault Tolerance	Scalability	Performance	Migration Time
Round Robin	Yes	Yes	Yes	Yes	No	No	Yes	No
Min Min	Yes	Yes	Yes	Yes	No	No	Yes	No
OLB	No	No	Yes	No	No	No	Yes	No
Max Min	Yes	Yes	Yes	Yes	No	No	Yes	No
Honey Bee	Yes	No	No	No	No	Yes	Yes	No
Active Clustering	No	No	Yes	Yes	No	No	No	Yes
Ant Colony	Yes	No	Yes	No	No	Yes	Yes	Yes
OLB+LEMM	Yes	Yes	Yes	Yes	No	No	Yes	No
Biased Random Sampling	Yes	No	No	No	No	No	Yes	No
Generalized Priority	Yes	No	Yes	No	No	No	No	Yes
Join-Idle-queue	No	No	No	Yes	No	No	Yes	No
Genetic Algorithm Based Load Balancing	No	No	Yes	No	No	No	Yes	No
Stochastic Hill Climbing	Yes	Yes	Yes	No	No	No	Yes	No
Decentralized Content Aware Load Balancing	No	Yes	Yes	Yes	No	Yes	Yes	No
Server Load Balancing for Internet Distributed Services	No	Yes	No	No	No	Yes	Yes	No

IX. RESULT & DISCUSSION

From the analysis of various load balancing algorithm, it is very easy to say that OLB + LBMM is better than others. In this paper, we compare load balancing algorithms on several performances measuring metrics. Which helps us to find a conclusion. OLB + LBMM are better than others because it has a great throughput, response time, resource utilization and overhead which determines its performance. On the other hand round robin, max-min, min-min is almost the same as OLB+LBMM. But there is the cute difference between them to OLB+LBMM, which is the percentage of throughput, response time and so on. For this reason, OLB+LBMM is very suitable load balancing algorithm than others. OLB+LBMM have some limitations that it has scalability problems, less capability of fault tolerance and also less migration time.

X. CONCLUSION & FUTURE WORK

In this review, Different types of the algorithm of load balancing are analyzed. Side by side honey bee and round-robin algorithms are properly implemented but, others are not. Various issues are also discussed which must be apprehended into account during designing of new load balancing algorithms. The piece, existing load balancing algorithms are discussed and, comparative analysis is performed by different metrics parameters. Like performance, throughput, scalability, resource utilization, fault tolerance, response time, migration time. In future, we plan to improve this algorithm by consideration of Ant colony optimization and two-phases (OLB + LBMM). Other factors by exploring new effective load balancing algorithm DLB3M which can maintain better balance among parameters and also helps to acquire green computing.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems", Volume 25, Number 6, Pages: 599-616, ISSN: 0167-739X, Elsevier Science, Amsterdam, The Netherlands, June 2009.
2. Ram Prasad Padhy and P Goutam Prasad Rao, "Load balancing in cloud computing systems", Department of Computer Science and Engineering, National Institute of Technology, Rourkela Rourkela-769008, Orissa, India May, 2011.
3. Venubabu Kunamneni, "Dynamic Load Balancing for the cloud", International Journal of Computer Science and Electrical Engineering, 2012.
4. Saroj Hiranwal and Dr. K.C. Roy, "Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice" International Journal of Computer Science And Communication July-December 2011, Vol. 2, No. 2, Pp. 319-323.
5. Jasmin James and Dr. Bhupendra Verma, "Efficient VM load balancing algorithm for a cloud computing environment", International Journal on Computer Science and Engineering (IJCSE), 09 Sep 2012.
6. Nusrat Pasha, Dr. Amit Agarwal and Dr.Ravi Rastogi, "Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment" International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 5, May 2014.
7. Upendra Bhoi, Purvi N. Ramanuj, - Enhanced Max-min Task Scheduling Algorithm in Cloud Computing|| International Journal of Application or Innovation in Engineering &Management (IJAEM), Volume 2, Issue 4, April 2013.
8. Fulsoundar, Pritam, and Rajesh Ingle. "Prediction of Performance Degradation in Cloud Computing."
9. Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, and Shun-Sheng Wang, "Towards a Load Balancing in a Three-Level Cloud Computing Network,".
10. Hein Nguyen Van, Frederic Dang Tran, and Jean-Marc Menaud, "Performance and Power Management for Cloud Infrastructures," in 2010 IEEE 3rd International Conference on Cloud Computing, 2010.
11. Rahmeh OA, Johnson P, Taleb-Bendiab A.,||A Dynamic Biased Random Sampling Scheme for scalable and reliable Grid Networks||, The INFOCOMP Journal of Computer Science, vol. 7, 1-10.
12. Ram Prasad Padhy, P Goutam Prasad Rao, —Load balancing in Cloud Computing Systems||, National Institute of Technology, Rourkela, India, 2011.
13. Tushar Desai, Jignesh Prajapati, || A Survey of Various Load Balancing Techniques and Challenges in Cloud Computing || International Journal of Scientific & Technology Research, Volume 2, Issue 11, November 2013.
14. Maria Spinola, — An Essential Guide to Possibilities and Risks of Cloud Computing: a Pragmatic Effective and Hype Free Approach for Strategic Enterprise Decision Making||. (white paper) 2009.
15. Ratan Mishra and Anant Jaiswal, — Ant Colony Optimization: A solution of Load Balancing in Cloud||, International Journal of Web & Semantic Technology (IJWeST), April 2012.
16. Venubabu Kunamneni, "Dynamic Load Balancing for the cloud", International Journal of Computer Science and Electrical Engineering, 2012.
17. Pooja Samal, Pranati Mishra, ||Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing|| (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (3) , 2013, 416-419.

18. Che-Lun Hung¹, Hsiao-hsi Wang² and Yu-Chen Hu², Efficient Load Balancing Algorithm for Cloud Computing Network. IEEE Vol. 9, pp: 70-78, 2012.
19. Dr. Amit Agarwal, Saloni Jain "Efficient optimal algorithm of task scheduling in cloud computing environment" International Journal of computer Trends and Technology (IJCTT). V9 (7):344-349, March 2014. ISSN: 2231-2803.
20. Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mandal, Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", Elsevier (CIMTA) 2013.
21. Anamika Jain, Ravinder Singh, "Review of Peer to Peer Grid Load Balancing Model Based on Ant Colony Optimization with Resource Management" Volume 3, Issue 4, April 2013 IJARCSSE.
22. Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach", Elsevier (C3IT) 2012.
23. H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), pp.370-375, February 2011.
24. A. M. Nakai, E. Madeira, and L. E. Buzato, "Load Balancing for Internet Distributed Services Using Limited Redirection Rates", 5th IEEE Latin-American Symposium on Dependable Computing (LADC), pp.156- 165 2011.
25. Xi. Liu, Lei. Pan, Chong-Jun. Wang, and JunYuan. Xie, "A Lock-Free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), pp.1-4 2011.