International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

An Overview of Load Balancing Techniques in Cloud Computing

Awwab Mohammad[1], M. Afshar Alam[2]
[1, 2]Department of Computer Science, SEST, Jamia Hamdard University, New Delhi, India-110062
Email address: awwab92[AT]hotmail.com

Abstract**—***"Cloud computing" is a term that includes virtualization, networking, distributed computing, software and web services. A cloud consists of numerous elements such as data centre, clients and distributed servers. It includes flexibility, fault tolerance, high availability, scalability, reduced overhead for users, reduced cost of ownership, on demand services etc. Fundamental to these issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, delay, memory capacity or network load. Load balancing is the process of allotting the load among various nodes of a distributed system to improve resource utilization as well as job response time while also avoiding a condition where some of the nodes are severely loaded while other nodes are idle or doing very little work. An important issue in cloud is scheduling of user requests i.e. how to allocate resources to these requests so that the requested tasks can be completed in a minimum time and the cost gained in the task should also be minimum. It is a type of load balancing and is not to be tangled with Domain Name System (DNS) load balancing. Whereas DNS load balancing uses hardware or software to achieve the purpose. Cloud load balancing uses facilities offered by various computer network companies. The main focus of paper is to study various load balancing algorithms and their applicability in cloud environment.*

Keywords**—** *Cloud computing, dynamic load balancing, load balancing, round robin, static load balancing.*

I. INTRODUCTION

The broad portrayal for Internet is cloud. Distributed computing is another and propelled innovation that is being utilized for business reason. Distributed computing [1] implies putting away and getting to information over the web rather than PC's hard drive. No requirement for vast interests in equipment or investing cash or energy in equipment is important while utilizing cloud. Rather we can give the exact size and sort of processing assets that we have to get our venture running and fit as a fiddle. We can get to unnumbered assets and pay for just those assets. Cloud giving focuses, for example, Microsoft Azure, Amazon and so on assume a crucial part. These focuses give utility figuring administration to programming focuses which can additionally give application administration to the end client through the web. For the clients' security insurance, the touchy information should be encoded before being transferred over the cloud. In a cloud based processing foundation, the assets are in another person's system and are remotely accessed by the cloud clients [2].

I.1 Cloud Components

A Cloud framework comprises of 3 noteworthy segments, for example, client computer, data centre, and distributed servers. Every component has a definite purpose and assumes a particular part.

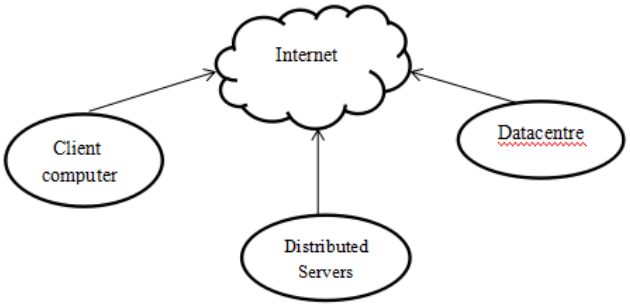

Fig 1. Cloud components.

II. LOAD BALANCING

Load Balancing is a strategy in which work is conveyed among every one of the servers, organize interfaces and figuring assets. Load adjusting in the cloud is totally not quite the same as the established observation on load adjusting usage by utilizing ware servers to play out the heap adjusting [3]. It is a procedure in which information is disseminated to different servers through various calculations and strategies to enhance the execution and asset use. It ensures that all the servers are being used and none is Idle. Load balancing is utilized to disperse a bigger preparing burden to lesser handling server for expanding the general execution.

It is a procedure of reassigning the aggregate load to the individual hubs of the aggregate framework to make asset use powerful and to enhance the reaction time of the occupation, all the while evacuating a condition in which a portion of the hubs are over stacked while some others are under stacked. A heap adjusting calculation which is dynamic in nature does not consider the past state or conduct of the framework, that is, it relies on upon the present conduct of the framework. Load adjusting calculation straightforwardly influences the impact of adjusting the server workloads. Its primary assignment is to choose how to pick the following server hub and exchange another association demand to it. Current principle load adjusting calculation is divided into static calculation and dynamic calculation [4].

II.1 Load Balancing in Cloud Computing

Cloud computing is a kind or virtualization. The three fundamental segments of cloud are: data centres accumulation of servers facilitating administrations, distributed servers and the customers. Distributed computing is generally used to host benefits for administration provisioning that run client server

Awwab Mohammad and M. Afshar Alam, "An overview of load balancing techniques in cloud computing," *International Research Journal of Advanced Engineering and Science*, Volume 2, Issue 3, pp. 108-111, 2017.

programming at remote area. The types of services a cloud environment provides are Software as a Service (SaaS), Platform as a service (PaaS), Infrastructure as a Service (IaaS), and Hardware as a Service (HaaS) [2]. In cloud environment, there are requests for services and platforms that arrive at different intervals of time thus necessitating the load balancing of servers. Load is the measure of amount of computational work that a system performs. The different types of loads are: CPU load, network delay load, memory used load etc. Thus load balancing is thus among the main issues of cloud computing.

A fundamental case of load balancing in our day by day life can be identified with sites. Without load balancing, clients could encounter delays, timeouts and conceivable long framework reactions. Load balancing arrangements for the most part apply excess servers which help a superior distribution of the communication traffic so that the site accessibility is indisputably settled [5]. In the range of cloud computing, the fundamental target of load balancing strategies is to enhance execution of computing in the cloud, backup plan in case of system failure, keep up scalability and stability for accommodating an expansion in substantial scale computing, decreases related expenses and response time for working in the cloud and furthermore increases the accessibility of resources [6].

II.2 Goals of Load Balancing

The goals [7] of load balancing are:

- To escalate the performance considerably.
- To have a backup capability if the system fails partly.
- To preserve the system reliability.
- To provide stability against variations to the system.
- Works as a driver rather than as a service.
- If we have two members in load balance group, with priority function we can send all the movement to one node and keep other one as a backup.
- Helps with disaster recovery.

III. LOAD BALANCING ALGORITHMS

Efficient algorithms are needed to guarantee the even distribution of load on servers. Tremendous need for load balancing is necessary in large and complex systems. For simplifying load balancing worldwide, techniques are used that would act at the modules of the clouds in a way that the load of the whole system is distributed. Load balancers implement specific type of algorithms to make load balancing decisions. The decision determines to which remote server to forward a new job. Some of the algorithms for load balancing are studied in this paper.

Load balancing algorithms can have three categories based on beginning of process as follows:

- Sender Initiated: This type of load balancing algorithm is initialized by the sender. Here, the sender sends a request message until it finds a receiver that can accept the load.
- Receiver Initiated: This type the load balancing algorithm is initiated by the receiver. Here, the receiver sends request message until it finds a sender that can get the load.
- Symmetric: It is the combination of both sender initiated and receiver initiated.

Depending on the existing state of the system, load balancing algorithms can be divided into 2 categories:

III.1 Static Load Balancing Algorithm

Static Load balancing algorithms [8] allot the tasks to the nodes just with respect to the capacity of the node to process new demands. The process is constructed exclusively on the bases of earlier information of the nodes' properties and abilities. These would incorporate the nodes processing power, memory and capacity limit, and latest known communication performance. In spite of the fact that they may incorporate information of the communication earlier executions, static algorithms do not consider dynamic changes of these traits at runtime. Also, these algorithms cannot adjust to load changes amid run-time.

In static load balancing, the execution of the processors is resolved toward the start of execution. Depending on their execution, the work load is allocated by the master processor. The slave processors compute their apportioned work and present their outcome to the master. A task is constantly executed on the processor to which it is assigned that is static load balancing strategies are non-pre-emptive. The objective of static load balancing strategy is to diminish the execution time, limiting the communication delays.

III.1.1 Brute-Force Load Balancing Algorithms

a. Round robin: This algorithm uses the time slicing mechanism. As the name suggests, work here takes place in rounds where each node is given a time slot and has to wait for their turn. The time is divided and the interval is allotted to every node. Each node is given a time slot within which they have to perform their tasks. An open source simulation executed the algorithm software known as “cloud analyst”. It is the default algorithm used in the simulation. Processors are allocated to each process in a circular order without any priority and thus there is no starvation [9]. By this approach, the traffic on servers will be depreciated easily and consequently it will lean the situation to an imperfection. Thus weighted round robin was introduced to improve this issue. In weighted round robin algorithm, each server is allotted a weight and according to this weight value, jobs are distributed. Processors with higher capacities are assigned larger values and hence would receive more tasks and those with lower capacities are assigned lower values and thus lower tasks. In a condition where all weights become equal, servers will obtain balanced traffic. In cloud computing system, precise prediction of execution time is not possible so static algorithm is not preferred. Hence, dynamic distinction of round robin has been proposed to tackle this issue more efficiently [10].

b. Opportunistic load balancing algorithm

This algorithm [11] tries to keep each node busy. Thus it does not consider the workload of each computer. OLB reports unexecuted tasks to currently available nodes in random, irrespective of the nodes’ current workload. Each job assigned to the node is in random order. It provides load balance schedule but results in a poor make-span. Since OLB does not calculate the execution time of a node, the task is

Awwab Mohammad and M. Afshar Alam, “An overview of load balancing techniques in cloud computing,” *International Research Journal of Advanced Engineering and Science*, Volume 2, Issue 3, pp. 108-111, 2017.

processed slower than normal and thus would cause bottlenecks although some of the nodes are free.

III.1.2 Static Load Balancing Algorithms based on the completion Time of Tasks on Machines

a. Min min load balancing algorithm

This algorithm [12] initiate with a set of unassigned tasks. Firstly, minimum completion time for the entire task is found. Then among them, minimum value is chosen which is the minimum time among all the task on any resource. According to minimum time, the task is programmed on the corresponding machine. The execution time for all other tasks is updated on that machine and the task is removed from the list. This technique is monitored till all the tasks are assigned the resource. In cases where the number of small tasks is higher than the number of large tasks, this algorithm achieves better performance. However, this approach can lead to starvation which is its drawback. This algorithm does not consider high or low machine and task heterogeneity.

b. Max min load balancing algorithm

This algorithm [13] is similar to min min load balancing algorithm except that after finding out the minimum execution times, the maximum value is chosen which is the maximum time among all tasks on the resources. Then according to maximum time, a task is programmed on the corresponding machine. The execution time for all tasks is updated and assigned tasks are removed from the list of tasks that are to be assigned to the machines. The algorithm is expected to perform perfectly as all the requirements are known in advance.

c. Two phase load balancing algorithm

This algorithm [14] combines OLB and Min-Min scheduling algorithms to develop better load balancing systems with increased efficiency and maintenance. OLB keeps each node in a working state thus to achieve the goal of load balance and Min-Min scheduling algorithm is utilized to minimize the execution time of every task on the nearby node minimizing the overall completion time. This combined approach hence helps in an proficient utilization of resources and enhances the work efficiency.

III.2 Dynamic load balancing algorithm

In dynamic load balancing algorithms, work load is circulated among the processors at runtime. The master appoints new procedures to the slaves in view of the new data gathered [15, 16]. Dissimilar to static algorithms, dynamic algorithms allocate processes dynamically when one of the processors ends up under loaded. Rather, they are buffered in the line on the fundamental host and distributed dynamically upon requests from remote hosts.

Dynamic load balancers consistently screen the load processors, and when the load imbalance achieves some predefined level, the redistribution of work happens. But as this observing takes CPU cycles so care must be taken as when it ought to be summoned. This redistribution incurs additional overhead at execution time.

III.2.1 Honey bee foraging:

In [17] author has proposed an algorithm called the honey bee foraging algorithm. When honeybees go on a hunt for food, they do the exceptional dance called the "waggle" after obtaining the food they show the other members that they have found the food. The type of the dance, its character signifies the quality and quantity of food that they have found. The dance all tells the exact outdistance of food from beehive. Thus servers are sorted below the virtual servers with their own virtual waiting queue. Each server considering the requirement from its queue first computes the lucre which is corresponding to the lineament that bees display in waggle dance. In load balancing, this waggle dance which is the gain is matched to measure time that is needed to meet resources used to satisfy the hypothesis.

III.2.2 Active clustering

Active clustering [17] is considered in self-aggregation algorithm to renovate the network. This algorithm works on the principle of combination of similar nodes together and working on these groups. Many load balancing algorithms work well where the nodes are aware of "like" nodes and can delegate workload on them. The process involved is:

- A node starts the process and selects another node called the matchmaker node from its neighbour filling the criteria that it should be of different type than the previous one.
- The matchmaker node then forms a link between neighbours which is the same type as the initial node.
- The matchmaker node then detaches the link between the initial node and itself.

III.2.3 Biased random sampling

M. Randles et al. [18] examined a distributed and adaptable load balancing approach that utilizes random sampling of the framework area to accomplish self-association consequently balancing the load over all nodes of the framework. The execution of the framework is enhanced with high and comparative population of resources thus resulting in increased throughput by adequately using the expanded system resources. It is debased with an expansion in population diversity.

IV. RESEARCH METODOLOGY

This research is an outcome of a broad study of the existing load balancing algorithms. A number of research papers were reviewed and thus this paper is the outcome of the gained knowledge.

Network load is not always equally distributed in order to provide quicker access for all the devices that need the cloud computing service. It is difficult to decrease overhead involved while generating schedules for multiple workflows because there may be many users competing for common resources and decisions must be made in possible shortest time. Load Balancing is a method 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, concurrently removing a situation in which some of the nodes are over loaded while some others are under loaded. Thus, load balancing is a reasonable technique that facilitates networks and resources by providing a maximum throughput with minimum response time by dividing the traffic between servers.

Awwab Mohammad and M. Afshar Alam, "An overview of load balancing techniques in cloud computing," *International Research Journal of Advanced Engineering and Science*, Volume 2, Issue 3, pp. 108-111, 2017.

V. CONCLUSION

Load balancing is among the main challenges in cloud computing. It requires distributing the workload consistently across all nodes to achieve satisfaction and resource utilization ratio by ensuring that each computing resource is distributed competently and equally. With appropriate load balancing, resource usage can be kept minimum which will reduce energy consumption. This study explains the concept of load balancing giving a brief idea about static and dynamic load balancing algorithms.

Scheduling in distributed operating systems has a substantial role in overall system performance and throughput. The scheduling in distributed systems is known as an NP-complete problem even in the best conditions. Various algorithms have been discussed in this paper

In future, we will be using Genetic algorithm as a solution to load balancing problem. It can be best to solve NP-complete problem. This algorithm studies multi objectives in its solution evaluation and solves the scheduling problem in a way that concurrently minimizes execution time and communication cost, and maximizes average processor utilization and system throughput.

REFERENCES

[1] K. Hashizume, D. G Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing"; Hashizume," *Journal of Internet Services and Applications*, vol. 4, issue 5, pp. 1-13, 2013.

[2] A. Mohammad, S. M. Kak, and A. Alam, "Cloud computing: Issues and challenges," *International Journal of Advanced Research in Computer Science (IJARCS)*, vol. 8, no. 2, pp. 26-28, 2017.

[3] Y. R. Kumar, M. M. Priya, and K. S. Chatrapati, "Effective distributed dynamic load balancing for the clouds," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, issue 2, pp. 1-6, 2013.

[4] Y. Kaushik and C. K. Jha, "Performance comparision of dynamic load balancing algorithm in cloud computing," *International Journal of Advanced Networking and Applications (IJANA)*, vol. 8, issue 1, pp. 2986-2990, 2016.

[5] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, "Availability and load balancing in cloud computing," *International Conference on Computer and Software Modeling, IACSIT Press*, Singapore, vol. 14, pp. 134-140, 2011.

[6] M. Armbrust, A. Fox, and R. Griffit., "A view of cloud computing," *Communications of the ACM*, vol. 53, no.4, pp. 50–58, 2010.

[7] A. M. Alakeel, "A guide to dynamic load balancing in distributed computer systems," *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, no.6, pp. 153-160, 2010.

[8] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. A. Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," *IEEE Second Symposium on Network Cloud Computing and Application*, 2012.

[9] S. S. Moharana, R. D. Ramesh, and D. Powar, "Analysis of load balancers in cloud computing," *International Journal of Computer Science and Engineering*, vol. 2, issue 2, pp. 101-108, 2015.

[10] P. Samal and P. 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, issue 3, pp. 416-419, 2013.

[11] C.-L. Hung, H.-H. Wang, and Y.-C. Hu, "Efficient load balancing algorithm for cloud computing network," *IEEE*, vol. 9, pp. 70-78, 2012.

[12] T. Kokilavani and D. I. George Amalarethinam, "Load balanced min-min algorithm for static meta-task scheduling in grid computing," *International Journal of Computer Applications*, vol. 20, no. 2, pp. 43-49, 2011.

[13] U. Bhoi and P. N. Ramanuj, "Enhanced max-min task scheduling algorithm in cloud computing," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, issue 4, pp. 259-264, 2013.

[14] S. C. Wang, K. Q. Yan, W. P. Liao, and S. S. Wang, "Towards a load balancing in a three-level cloud computing network," *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, Chengdu, China, pp. 108-113, 2010.

[15] S. Malik, "Dynamic load balancing in a network of workstation," 95.515 Research Report, 19th November, 2000.

[16] Y. T. Wang and Morris, R., "Load Sharing in distributed systems," *IEEE Transactions on Computors*, vol. C-34, issue 3, pp. 204-217, 1985.

[17] R. P. Padhy and P. G. P. Rao, "Load balancing in cloud computing systems," Thesis from National Institute of Technology, Rourkela-769 008, Orissa, India, 2011.

[18] T. R. V. Anandharajan and M. A. Bhagyaveni, "Co-operative scheduled energy aware load-balancing technique for an efficient computational cloud," *IJCSI International Journal of Computer Science Issues*, vol. 8, issue 2, pp. 571-576, 2011.

.