A Survey: Comparative Study of an Assortment of Load Balancing Algorithms

G Gayatri1 Dr. P K Vaishali2

[1] Assistant Professor, CSE, JITS, Karimnagar, Telangana State, India.
[2] Professor, CSE, JITS, Karimnagar, Telangana State, India.

Abstract- *Cloud computing refers to control, organizing, and retrieving the applications online. It offers online data storage, infrastructure and application. As Cloud Computing is one of the greatest platform which provides storage of data in very lower cost and obtainable for all time over the internet .Load balancing is one of the main challenges in cloud computing. The objective of load balancing is to dispense the dynamic workload across manifold nodes to certain that no single node is weighed down. To minimize the resource consumption this will further condense energy consumption and carbon emission rate. This paper presents various load balancing algorithms in different cloud environment. And also discussed various metrics and performance parameters that have been considered to judge against different techniques.*

Keywords-- *Load balancing, cloud environment, map reduce*

I. INTRODUCTION

Load balancing is one of the central issues in cloud computing. Load balancing in clouds is a mechanism that distributes the excess dynamic local workload squarely across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio making sure that no single node is snowed under. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource exploitation 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 hence civilizing the overall performance of the system[6]. Load balancing ensures that all the processor in the system or every node in the network does in the region of the equal amount of work at any instant of time that can help in utilizing the available resources optimally, thereby minimizing the resource consumption [10]. It also helps in executing miscarry-over, qualifying scalability, avoiding bottlenecks and over-running, dropping response time etc. Apart from the above-mentioned factors, load balancing is also required to attain Green computing in clouds which can be broken with the help of the following two factors:

• Reducing Energy Consumption- Load balancing helps in avoiding warmness by balancing the workload crosswise to all the nodes of a cloud, hence dropping the quantity of energy spent [7].

• ***Reducing Carbon Emission***- Energy consumption and carbon emission go in deliver. The more the energy frenzied, higher is the carbon mark out. As the energy consumption is reduced with the help of Load balancing, so is the carbon emission helping in achieving Green computing[8].

II. METHODOLOGY

A. Load Balancing on the origin of Cloud Environment

Cloud computing can have either static or dynamic environment built upon how developer organizes the cloud necessitated by the cloud supplier. Any algorithm with reference to load balancing is calculated based on the state or behavior of the system, which may be static or dynamic.

1) Static Environment

In static environment the cloud provider installs harmonized resources. Also the resources in the cloud are not bendable when environment is made static.

These algorithms do not depend upon the modern state of the system and have prior knowledge regarding system resources, nodes capacity, processing power, memory, performance and statistics of all user requirements and all tasks in an application. These kinds of algorithms face a major drawback in case of sudden failure of system resource and tasks. These user requirements are not subjected to any change at run-time. Although static environment is easier to simulate but is not well suited for heterogeneous cloud environment. Round

Robin algorithm is the best example that provides load balancing in static environment.

2) Dynamic Environment

In dynamic environment the cloud supplier mounts sorted resources. The resources are bendable in dynamic environment. These algorithms take decisions concerning load balancing based upon the current state of the system and don't need any prior knowledge about the system whereas it takes into account run-time statistics. Algorithm proposed to achieve load balancing in dynamic environment can easily adapt to run time changes in load i.e., requirements of the users are flexibility that may change at run time. This approach is an improvement over the static approach. The algorithms in this category are considered complex to be simulate, but have better fault tolerance and highly adaptable with cloud computing environment and overallperformance [3]. Load Balancing Min-Min (LBMM) algorithm and opportunistic load Balancing (OLB) algorithm are uses resource allocation in dynamic environment.

III. OVERVIEW OF LOAD BALANCING ALGORITHMS

B. Round Robin Algorithm:

Round Robin is a very famous load balancing algorithm, in which the processes are divided between all processors. The process allocation order is retained locally independent of the distributions from remote processors. In Round Robin, it send the requests to the node with the least number of connections, so at any fact of time some node may be deeply loaded and other persistent doled .This difficulty is compact by CLBDM. Central Load Balancing.

Decision Model (CLBDM): CLBDM is a central load balancing decision model, which is suggested based on session switching at the application layer. The improvement is that, in the cloud it calculated the connection time between the client and the node, and if that linking time surpasses a threshold then linking will be ended and chore will be forwarded to another node using the consistent Round Robin rules. Round Robin techniques have high throughput. High throughput is necessary for overall system performance. Overhead, Response time, Resource utilization, Scalability and performance of this technique are very high. Round robin technique doesn't have any fault tolerance to measure the capability of an algorithm to perform uniform load balancing in case of any failure. In this there is no migration time.

C. Map Reduce-based Entity Resolution

Map Reduce is a calculating model and an correlated implementation for processing and making large datasets. Map chore and decrease chore are two main chore in this model which are produced by the user, Map takes an input duo and yields a set of transitional value duo or pair and Reduce chore or task accepts an transitional key and a set of standards for that key and combines these values or standards to form a reduced set of value. Map task or Map chore deliver objects in parallel and method them, this willets on the reduce chore or task to be burdened[4].

D. Ant colony optimization (ACO)

In ACO algorithm when the appeal in started the ant start its movement. Movement of ant is of two techniques:

Forward Movement: Forward Movement means the an tin uninterruptedly moving from one burdened or overloaded node to alternative node and plaid it is overloaded or under loaded, if ant discover an over loaded node it will uninterruptedly moving in the forward direction and plaid or check each nodes.

Backward Movement: If an ant discover an over loaded or burdened node the ant will use the back ward movement to get to the earlier node, if ant discovers the mark node then ant will commit suicide, this algorithm reduced the unnecessary back ward movement, overwhelmed heterogeneity, is outstanding in fault tolerance.

E. Load balancing of virtual machine resources :

Recommended a scheduling policy on load balancing of VM resources that usages past data and current state of the system. This approach attains the finest load balancing and condensed dynamic migration by using a genetic algorithm. It aids in determining the matter of load-imbalance and high cost of relocation thus attaining better resource utilization.

F. Index Name Server Algorithm (INS)

The INS algorithm suggested to discover an algorithm to diminish the data repetition and redundancy. INS is able to hold the load balancing dynamically INS have some parameters which aid in computing the best possible selection point like that Hash Code of the block of data to be downloaded, the location of the server, the changeover quality, the maximum bandwidth[2]. Another calculation point whether the linking can handle additional nodes or not. They categorised the busy levels B (a), B (b), and B(c). B (a)means that link or connection is very busy and cannot holder any additional connection. B (b) means connections is not busy and can handle additional links. B(c) means that the connection is limited.

G. 3Opportunistic Load Balancing (OLB)

OLB is a static load balancing algorithm that has the objective of custody for each node in cloud full. Though OLB does not compute the execution time of the node, owing to this the chores to be deal with in an unhurried manner and will reason bottlenecks since needs might be pending for nodes to be permitted.

H. Load Balancing Min-Min Algorithm (LBMM):

LBMM has a three level load balancing outline. In first level LBMM architecture is the appeal manager which is accountable for getting the task and handover it to service manager, when the service manager accepts the request; it splits it into subtask and allocates the subtask to a service node based on node accessibility, outstanding memory and the transmission rate which is liable for execution the task.

I. Dual Direction Downloading Algorithm(DDFTP)

DDFTP is a dual direction downloading algorithm from FTP server. This algorithm can be also applied for Cloud Computing load balancing. This is a fast and well-organized simultaneous technique for downloading large files from FTP server in a cloud environment.

J. Exponential Smooth Forecast-based on Weighted Lest Connection (ESBWLC)

The algorithm is a dynamic load balancing algorithm for cloud computing. ESBWLC form the conclusion of allocating a definite task to a node later having a number of task allotted to that service node and receiving to know the node's CPU power, memory, number of links and the quantity of disk space now in used, then ESBWLC guesses which node is to be selected based on exponential levelling.

K. A Lock-free multiprocessing solution for LB -X

A lock-free multiprocessing load balancing solution that evades the usage of shared memory in difference to other multiprocessing load balancing answers which usage shared memory and lock to preserve a user session. It is attained by changing Linux kernel. This way out helps in improving the global performance of load balancer in a multi-core environment by running various load-balancing procedures in one load balancer.

L. Honeybee Foraging Behaviour

A spread out honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. It attains global load balancing over local server actions. Performance of the system is improved with greater than before system diversity but throughput is not increased with an increase in system size. It is best apt for the conditions where the varied inhabitants of service types are essential. There is no throughput, overhead, fault tolerance, migration time, response time, scalability and performance of system that means these are very low. Resource utilization is very extraordinary like other methods.

IV. METRICS FOR LOAD BALANCING IN CLOUDS

The current load balancing techniques in clouds, deliberate numerous parameters like performance, response time, scalability, throughput, resource utilization, fault tolerance, migration time and associated overhead. But, for an energy-efficient load balancing, metrics like energy consumption and carbon emission should also be considered[5].

These are the challenges in cloud computing**.**

1) Overhead Associated- defines the amount of overhead intricate while executing a load-balancing algorithm. It is collected of overhead owing to drive of tasks, inter-processor and inter-process Communication. This should be diminished so that a load balancing technique can work capably.

2) Throughput - is used to compute the no. of chores whose calculation has been accomplished. It should be in height to progress the performance of the system.

3) Performance– is used to check the competence of the system. It has to be progress at a faithful cost e.g. diminish response time while keeping suitable delays.

4) Resource Utilization- is used to check the utilization of resources. It should be elevated for a well-organized load balancing.

5) Scalability- is the capacity of an algorithm to accomplish load balancing for a system with any determinate number of nodes. This metric should be enriched.

6) Response Time - is the amount of time taken to respond by a specific load balancing algorithm in a

distributed system. This parameter should be minimized.

7) Fault Tolerance- is the capability of an algorithm to perform uniform load balancing despite arbitrary node or link failure. The load balancing should be a good fault-tolerant method.

8) Migration time- is the time to migrate the jobs or resources from one node to other. It should be minimized in order to improve the performance of the system.

9) Point of Failure: Control the load balancing and collect data of different nodes and the system planned in a way that evades the single point of failure in the algorithms. For example centralized algorithms, if one controller fails, then the entire system would miscarry. Any load balancing algorithm must be intended in order to overwhelmed this challenge.

10) Energy Consumption (EC)- defines the energy consumption of all the resources in the system. Load balancing aids in evading overheating by balancing the workload crossways all the nodes of a Cloud, later dropping energy consumption.

11) Carbon Emission (CE) - computes the carbon emission of all the resources in the system. As energy consumption and carbon emission go hand in hand, the more the energy consumed, advanced is the carbon footmark. So, for an energy-efficient load balancing way out, it should be reduced.

12) Based on the above metrics, the existing load balancing techniques have been measured.

V. CONCLUSION

In this paper, we have surveyed and discussed the essentials of Load balancing in the cloud and proportional study of various load balancing algorithms in the cloud computing environment and also metrics or encounters of cloud. Median to these issues is the topics of load balancing, that is required to dispense the surplus dynamic local workload directly to all the nodes in load balancing. Whole node in the complete cloud to accomplish a high user gratification and resource utilization ratio. It also confirms that every computing resource is spread efficiently and abstemiously. Current Load Balancing techniques that have been purposely emphasis on dropping overhead, facility response time and improving performance etc., but no one of these techniques has considered the energy consumption and carbon emission factors. Therefore, there is a necessity to widen an Energy-efficient load balancing technique that can develop the performance of cloud computing the length of maximum resource utilization, in turn dipping energy consumption as well as carbon emission to possibility that will help to attain Green Computing[9]. In view we will hub on scheming algorithms which will reserve a better trade-of among all performance parameters.

REFERENCES

1. Vaquero, Luis M., Luis Rodero-Merino, and Rajkumar Buyya. "Dynamically scaling applications in the cloud." ACM SIGCOMM Computer Communication Review 41.1 (2011): 45-52.
2. Ferris, James Michael. "Load balancing in cloud-based networks." U.S. Patent No. 8,849,971. 30 Sep. 2014.
3. Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities." High Performance Computing & Simulation, 2009. HPCS'09. International Conference on. IEEE, 2009.
4. Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services." Algorithms and architectures for parallel processing. Springer Berlin Heidelberg, 2010. 13-31.
5. Karger, David R., and Matthias Ruhl. "Simple efficient load balancing algorithms for peer-to-peer systems." Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures. ACM, 2004.
6. Randles, Martin, David Lamb, and A. Taleb-Bendiab. "A comparative study into distributed load balancing algorithms for cloud computing." Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on. IEEE, 2010.
7. Congdon, Paul, and Sundar Subramaniam. "Load Balancing Algorithms." (1998).
8. Randles, Martin, David Lamb, and A. Taleb-Bendiab. "A comparative study into distributed load balancing algorithms for cloud computing." Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on. IEEE, 2010.
9. Kansal, Nidhi Jain, and Inderveer Chana. "Cloud load balancing techniques: a step towards green computing." IJCSI International Journal of Computer Science Issues 9.1 (2012): 238-246.
10. Li, Li Erran, and Thomas Woo. "Dynamic load balancing and scaling of allocated cloud resources in an enterprise network." U.S. Patent Application No. 12/571,271.