A Review on Load Balancing Techniques in Cloud Computing

[1]Jigna Acharya, [2]Chirag Patel
[1]PG Student, [2]Associate Professor
[1 & 2] Government Engineering College
Modasa, India,

***Abstract* - Cloud computing is the new computing paradigm in IT industry now a days where platform, software, applications and some other IT services are available over the internet as per user's demand. Users pay for resources which they require. Load balancing for the cloud computing has become a very interesting research area. Many algorithms were suggested for load balancing. These algorithms are used to improve the overall performance of the cloud environment. In this paper the different algorithm are discussed which are used for solving the issues of load balancing in cloud computing.**

***Index Terms* - cloud computing, load balancing**

I. INTRODUCTION

Now a day's cloud computing is a emerging[1] as a new paradigm in distributed computing. Cloud computing is an on demand services in which shared resources, information, software and other services are provided according to the clients requirement. The main objective of cloud computing is to provide the satisfactory level of performance. Load balancing is the main issues in a cloud computing. Load balancing is the techniques to divide the workload between many computers equally in order to improve the performance. Many algorithms are proposed through which load can be dispatched equally. In this paper our main objectives to provide comparative study of different existing load balancing techniques. The overall paper is arranged in a planned way as follows: section 2 provides the types of load balancing algorithm. Section 3 provides literature survey of load balancing algorithm. Section 4 gives the comparative study. Section 5 concludes the paper [2].

II. TYPES OF LOAD BALANCING ALGORITHM

Depending who initiated the process, load balancing algorithm can be classified into three categories.

- Sender Initiated: - If the load balancing algorithm is initiated by server.
- Receiver Initiated: - If the load balancing algorithm is initiated by receiver.
- Symmetric: - It is combination of both sender and receiver.

Depending on the current state of system it can be classified into two types.

- Static Algorithm: - In this type the assign of task to processor is done before program execution is begin. It does depend on current state of system. Static algorithms are non preventive its main aim is to minimize the overall execution time.
- Dynamic Algorithm: - It is based on the redistribution of process among the processors during execution time. Its main aim to improve the performance of application to transfer task from heavily loaded processor to lightly loaded processor. No prior knowledge is needed. So it is better than static approach.

In a cloud computing, dynamic load balancing can be classified in two different ways: centralized or distributed. In distributed system the dynamic load balancing algorithm is executed by all nodes present in the system.

III. LITERATURE SURVEY OF LOAD BALANCING ALGORITHM

The cloud computing is a network of [2] distributed data centers, each consisting of hundred servers when user submit a task to data center controller. Data center controller use VM Load balancer to decide which VM should be assign to the new request for processor. The VM Load balancer can use the following algorithm for cloud computing.

A. Honey bee Foraging Algorithm [3]

This algorithm is dynamically decentralized load balancing technique for self-organization. It is derived from the behavior of honey bees for finding and reaping food. Forager bees searches for [2] food source, upon finding one they come back to beehive to advertise this using waggle dance. This dance gives ideas of quality and quantity of food as well as gives distance of food source from hive. Harvester bee then follows the forager to the location of food and then began to reap it.

In this technique the web servers are grouped under virtual server (VS) each virtual server having its own virtual service queues. Each VS processing a request from its own queue and calculates a profit, which is same to the quality that bees show in their dance. One measure of this reward can be the time spends by CPU on processing request. The dance floor in case of honey bee is analogous to an advert board which is used to advertise the profit of the entire colony.

Each of the servers takes the role of either a forager or harvester [2]. After processing the server, a request can post their profit with a probability pr. A server can randomly choose a VS's queue with probability px. A server processing a request, calculates

its profit and compare it with the colony profit and then sets its px. If this profit was high, then the server stays at the current virtual server. If it was low, then the server returns to the idle / waiting behavior.

B. Biased Random Sampling [3]

In this technique the load on a server is represented by a virtual graph. Each server is a treated as a node in the graph, with each degree directed to the free resource of the server. Whenever node executes a job, it deletes an incoming edge, which indicates reduction in availability of free resources. After completion of a job, the node creates an incoming edge, which indicates increase in availability of free resources.

The addition and deletion of processes is performed by the process of random sampling. The walk starts at any one node and at every step a neighbor is chosen randomly. The last node is selected for allocation for load.

A node upon receiving a job, will execute it only if its current walk length is equal to or greater than threshold value. Else, the walk length of the job under consideration is incremented and another neighbor node is selected randomly. When, a job is executed by a node in the graph, an incoming edge of that node is deleted. After completion of the job, an edge is created from the node initiating the load allocation to the process to the node which was executing the job.

C. Active Clustering[3]

Active clustering works on the principle of grouping similar nodes together and working on these groups [1]. The process involves following steps:

- A node initiates the process and selects another node called the matchmaker node from its neighbors satisfying the criteria that it should be of different type than the former one [1].
- The so called matchmaker node then forms a connection between neighbors of it which is of the same type as the initial node.
- The matchmaker node then detaches the connection between itself and initial node.

The above set of processes is followed iteratively.

D. Double Thresold Energy Aware Load Balancing Algorithm [4]

In this algorithm there are three basic sections. The first section is balancing section which is responsible for determining where virtual machine (VM) will be instantiated. If all the nodes have utilization between 25%-75% then it searches the least loaded node to assign a new VM.

The second section is upscale section which is used to power on additional compute nodes during peak period[1]. It does this if all currently active compute nodes have utilization over 75%.

The last section is down scale section which is responsible for powering down or switching off idle compute nodes.

The load balances load efficiently but when the system load is very low there may be nodes having utilization is less than 25%. So the algorithm will start migrating VM of these nodes to another nodes. It may happen that after migrating on destination node still have utilization less than25% again these node migrate VM to another node. So this algorithm may initiate unnecessary migration of VM which may degrade the performance and may lead to high operational cost.

E. Join Ideal Queue [5]

The Author had proposed a novel load balancing algorithm known as Join Ideal Queue algorithm for dynamically scalable web services [6]. This algorithm provides large scale load balancing with distributed dispatchers by, first load assigning jobs to dispatchers and then, assigning jobs to processors to reduce average queue length at each processors [1]. The author analyzed this algorithm in large system limit and found the effective results. This algorithm incurs no communication overhead at job arrivals and does not increase actual response time.

F. Dual Direction Downloading Algorithm From FTP Server (DDFTP) [7]

This is a dynamically load balancing technique. It works by splitting a file size m into m/2 partitions. Then each server node starts processing the task assigned for it. For example, one server will start from block 0 and keeps downloading incrementally while another server starts from block m and keeps downloading in a decrimental order. Both work independently. At the end of downloading the whole file to client in best possible time. The algorithm is used for reduces the network communication overhead. Attributes like node load, network load, network speed are automatically taken into consideration.

G. Index Name Server (INS) [8]

The goal of this algorithm is to find and to minimize the data duplication and data redundancy. It integrates access point selection optimization. There are various parameters considered in the process of calculating the optimum selection point. These parameters are node performance, position of server, hash code of block data, weight judgment chart, bandwidth of required server and required path parameter. When new connection is added into the node then busy level of node calculation is required. Busy level can be classified into three category B(a), B(b) and B(c).B(a) is used for connection is busy and no any new connection is added. B(b) used for connection is not busy and additional connections can be added. B(c) used for connection is limited and further study needs to be done to know more about connection. B(b) is further divided into three category b(b1) which used for INS must analyzed and backup should be established, B(b2) which means that INS must send the request for backup nodes and B(b3) which used for highest level of efficiency is required.

H. Exponential Smoothing Forecast based on Weighted Least Connection (ESBWLC)[9]

This is dynamic load balancing algorithm based on weighted least connection (WLC). In WLC algorithm perform the comparison of sum of connections of each node in the cloud. The task is assigned to the node that has minimum number of connection.

Attributes like storage capacity, processing speed, band width are not considered in WLC algorithm. So the new proposed algorithm is called Exponential Smoothing Forecast based on Weighted Least Connection (ESBWLC). This algorithm improves WLC by taking into consideration for all parameters. ESBWLC assign the task to a node after knowing capabilities of the node. ESBWLC takes decision based on the node's CPU power, memory, number of connections and the amount of disk space currently being used. ESBWLC then predicts which node is to be selected based on exponential smoothing.

I. Ant Colony Optimization Algorithm [6]

This algorithm is static load balancing algorithm to solve the problem of load balancing in cloud computing. This algorithm chooses the behavior of the ants during search of foods. Ant posses with very intelligent way for finding the food by the method of shortest distance. In proposed algorithm once a request is send from user the ants are initiated and the ants start to move from the head node to end node. A node move from node to another node check if it is overloaded or not. If the ant finds an under loaded node, it will continue to move to check the next node. If next node is an overloaded the ant will use the backward movement to get to the previous node. When final node will be find the ant will suicide which avoids unnecessary backward movements.

J. Central Load Balancing Decision Model (CLBDM)[10]

CLBDM is an improvement algorithm of Round Robin algorithm which is based on session switching at the application layer. In round robin algorithm, it sends the requests to the node with the least number of connections. In CLBDM the connection time between the client and node in the cloud is calculated, and if that time is exceeds a threshold then there is an issue. If an issue is found then connection will be terminated and task will be forwarded to another node using administrator. It is again static load balancing algorithm.

K. Randomized [1]

This algorithm randomly assigns the selected job to the available VM. This algorithm does not consider that VM is overloaded or under loaded. This may result in the selection of a VM under heavy load and the job may require a long waiting time before being serviced.

IV. COMPARATIVE STUDY OF ALGORITHM

In this section, the different algorithms are discussed and compare these algorithms based on [11] the pros and cons.

There are various approaches for balancing the load for various situations. The static algorithms are very efficient in stable environment while the dynamic algorithms continuously monitor the resources at run time. Dynamic algorithm offers much better solution to adjust the load dynamically at run-time.

Table I shows a comparison of various algorithm based on their pros and cons.

Table I Pros and Cons of Load Balancing Algorithm [11]

Sr.No.	Algorithm	Description	Pros	cons
1	Honey Bee Foraging	Decentralized honey bee based nature inspired load balancing techniques	work good under heterogeneous environment	increases in resources does not improve throughput
2	Biased Random Sampling	The sampling walk starts at a specific node in graph and randomly choose neighbor node.	It is suitable for large network system	Not suitable for dynamic environment
3	Active Clustering	It works on the principle of grouping similar nodes together and working on them	it efficiently balances load	it does not perform very well in heterogeneous environment.
4	Double Threshold Energy Aware load balancing	It has three section: Balancing section check whether VM will be initialized, Upscale section powers on additional compute nodes and downscale section switches off idle nodes.	Balances load efficiently	It may initiate unnecessary migration of VMs and communication delay is not considered.
5	Join Idle Queue	Load balancing idle processors across dispatcher and then assigning jobs to processors to reduce average queue length at each processor.	No communication overhead	More power consumption
6	DDFTP	It work by splitting of file size m into m/2.	Reliable download of files	Full replication of data files that requires high storage in all nodes
7	INS	It finds and minimizes data redundancy and data duplication. It integrates access point selection optimization	Initially proved to handle some sort of dynamic load balancing	complicated in terms of implementation and no forecasting algorithm to identify the future behavior of the nodes.

8	ESBWLC	Predicts assignment of task based on the experience of the node's CPU power, memory, number of connections and the amount of disk space currently being used.	Capabilities of each node are considered	Prediction algorithm requires existing data and has long processing time
9	Ant Colony	Ants behavior is used for to gather information about nodes	Decentralized, no single point failure Ants can collect the information faster	Network Overhead Points of initiation of ants and number of ants are not clear
10	CLDBM	It has a human administrator which used to control the system	Solves issues of Round robin algorithm	Inherits Round robin issues such as not taking into consideration node
11	Randomized	It randomly assign job to the available VM	Very simple algorithm	Current load is not considered

V. CONCLUSION

In this paper we reviewed multiple algorithms for load balancing in cloud computing and discussed their pros and cons. Then existing algorithms are compared on the basis of their challenges which are present in cloud environment. Load balancing has two meanings: first, it puts a large number of concurrent accesses or data traffic to multiple nodes respectively to reduce time for response; second, it put the calculation from single heavy load to the multiple nodes to improve the resource utilization of each node. In this paper algorithm described not only balance the load but also help in utilization of resources, increase throughput and decrease response time. All these will reduce operational cost.

REFERENCES

[1] Subhadra Bose Shaw, Dr.A.K.Singh, "A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment," 2014 5th International Conference on Computer and Communication Technology (ICCCT)

[2]Priyadarshini Adyasha Pattanaik, Sharmishta Roy, Prasant Kumar Pattanaik, "IEEE 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)

[3] Martin Randles, David Lamb, A.Taleb-Bendiab , " A Comparative Study into Distributed Load Balancing Algorithm For Cloud Computing " , IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, 2010, PP. 551-556

[4] Jayant Adhikari, Prof. Sulabha Patil, "Double Thresold Energy Aware Load Balancing in Cloud Computing", 4th ICCCNT, July 2013

[5] Y.Lua, Q.Xiea, G.Kliotb, A.Gellerb, J. R. Lasurb, and A.Greenber, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", An international Journal on Performance evaluation, August 2011.

[6] Abhinav Hans, Sheetal Kalra, " 2014 International Conference on Medical Imaging, m-Health and Emerging Communication System" .

[7] AI-Jaroodi, J. and N. Mohamed. "DDFTP: Dual-Direction FTP," in proc. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid computing (CCGrid), IEEE,pp:504-503, May 2011.

[8] T-Y., W-T. Lee, Y-S.Lin, Y-S.Lin, H-LChan and J-S. Huang, "Dynamic load balancing mechanism based on cloud storage" in proc.Computing, Communications and Applications Conference (ComComAp), IEEE, pp:102-106, January 2012.

[9] Ren, X.R. Lin and H. Zou, " A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast" in proc. International Conference on. Cloud Computing and Intelligent System (CCIS), IEEE, pp:220-224,September 2011.

[10] Radojevic, B. and M. Zagar, " Analysis of issues with load balancing algorithms in hosted (cloud) environments." In proc.34th International Convention on MIPRO, IEEE, 2011.

[11] Bhushan Ghutke, Urmila Shrawankar, " Pros and Cons of Load Balancing Algorithms for Cloud Computing",IEEE,2014