e-ISSN: 2349-9745
p-ISSN: 2393-8161
Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Analysis of Load Balancing Algorithms In Cloud Environment

K.Karthika[1], S.Muthukumar[2]
[1]*PG Scholar, Dept of Computer Science and Engineering, Kathir College of Engineering/ Anna University, India*
[2]*Assistant Professor, Dept of Computer Science and Engineering, Kathir College of Engineering/ Anna University, India*

Abstract-The vast requirements of infrastructure and resources in IT environment supported by a new trend of technology are called Cloud Computing. The major vital feature of cloud computing environment is Load Balancing. By stipulation of resources to cloud user's basis on the demand to make sure the efficient load balancing for resource exploitation. Cloud users prioritization based on scheduling criteria in load balancing. In the user level the services will be provide. The cloud user submits their larger tasks and the task will perform by servers and then cloud user gets back the result. Cloud Computing environment have a several issues and challenges there are security ,availability, scalability ,resource allocation etc., Availability will be increased while increasing the load balancing in task on cloud. Load Balancing Techniques are need to augment the performance of the intact Cloud Environment. In cloud all nodes will get the load by using different Load balancing algorithms. These algorithms defined to distribute the load between nodes. Satisfaction of users and reliability is a major achievement in cloud using load balancing. This paper concentrates and compares the various load balancing techniques in the cloud.

Keywords - Availability, Cloud Computing, Load Balancing, Resource allocation, Security.

I. INTRODUCTION

Based on the scalability the huge and tiny business companies are moving to cloud environment Cloud Environment have a huge data centers for executing a job. These data centers contain thousands of blade servers. In cloud the jobs are executed by these servers. Service Oriented Architecture (SOA) is a term to define the Cloud Computing and the users get different types of services. Users infrastructure is not require for getting the service. It means, user does not know about the service origin and its infrastructure. The ease of the cloud is users can only pay the usage of the services in the cloud.

In our day-today life one of the best examples of Cloud is the Electric Current Supply. Large wind mills produced the power. Wires and transformers are used to transform the power to various areas. Users can use these power and pay for the consumed power. In our day to day life we don't know where its produced and how its transformed to our houses. We are only consumers.

Cloud environment consist of four different types. They are

- Public Cloud(Cost Free, access anyone)
- Private Cloud(for single organization people, Pay for only used)
- Hybrid Cloud(Combination of public & private Clouds)
- Community Cloud(For Communication purpose)

Based on the form of service user can access the cloud resources. Cloud Environment provides 3 basic services. They are,

- Platform as a Service (PaaS)

- Software as a Service (SaaS)
- Infrastructure as a Service (IaaS)

II. LOAD BALANCING

Cloud Environment is balance the load used a technique as Load Balancing [1]. It means balanced the load based on the transferring the heavily loaded nodes to low loaded nodes. As a result, no node should be heavily loaded. Thereby it will increase the availability of nodes.
Figure 1 shows Load Balancing Technique major work. The Load Balancer may be any software or hardware which receives jobs from different users in different locations. The received loads are distributed evenly across all the servers in Data Center.

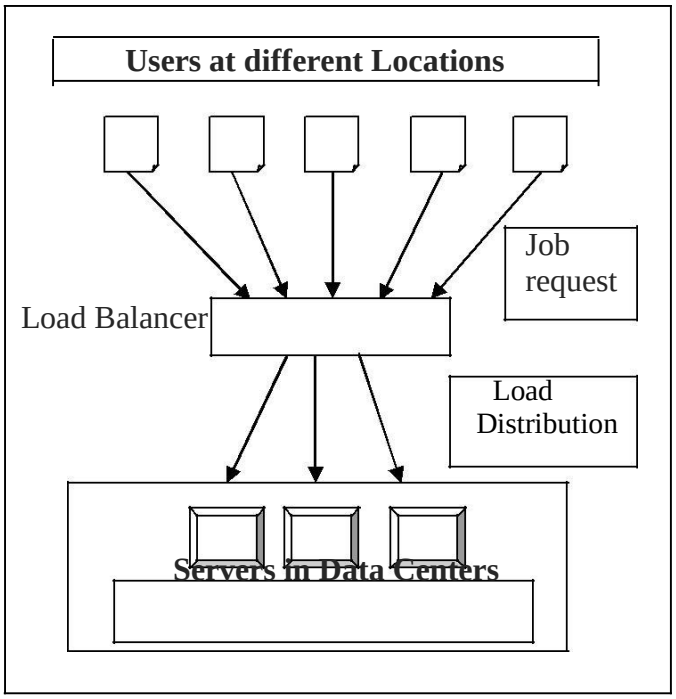

Fig 1: Load Balancing Technique

2.1 Need for Load Balancing

To avoid the overload need Load Balancing. If the single node gets all the jobs the load will be increase as well as its queue size is increased and it becomes overloaded. If balance the load across several nodes while every node is in running state but not in overloaded state.
The goals are as follows [2]:

- To increase the availability
- To increase satisfaction of user
- To improve the ratio of resource utilization
- In queue to minimize the waiting time of job as well as to reduce job execution time
- To improve the overall performance of Cloud environment

2.2 Basic Types of Load Balancing Algorithms

Load Balancing algorithms is categorized into three types [3] based on the initiator of the algorithm,

Sender Initiated

Sender identifies that the nodes are plagued so that the sender initiates the execution of Load Balancing algorithm.

Receiver Initiated

The requirement of Load balancing circumstances can be known by the receiver/server in cloud and that server initiates the execution of Load Balancing algorithm.

Symmetric

It is the mixture of both the sender initiated and receiver initiated types. It takes advantages of both types.Based on the present state of the system, the two types of load balancing algorithms are used:

Static Schemes

Account does not have the current status node [6]. Predefine the nodes and their properties. Based on this earlier information, the algorithm works. Seeing as it does not use current system status information, it is less composite and it is simple to apply.

Dynamic Schemes

This kind of algorithm is based on the present system information [6]. The algorithm works according to the changes in the state of nodes. Dynamic schemes are expensive one and are very complex to implement but it balances the load in effective manner.

Status Table

Status table [1] is a data structure to maintain the current status of all the nodes in the cloud environment. This information can be used by some of the dynamic scheme algorithms to allocate jobs to the nodes that are not heavily loaded.

III. LOAD BALANCING ALGORITHMS

Quite a lot of Load Balancing algorithms were proposed. Some of algorithms are analyzed here.

3.1 Dynamic Round Robin Algorithm

It is a term extension for the Round Robin algorithm [7]. Generally, each physical machine has number of virtual machines. This algorithm mostly works on dropping the power utilization of physical machine. So that two rules used:

1. The virtual machine in one physical machine has completed its execution but all other virtual machines are unmoving running then if any new virtual machine is arrived means the parallel physical machine does not agree to the new virtual machine. Such physical machines are called the term as “retiring" state physical machines (i.e.), when all the other virtual machines has completed their execution then we can shut down that physical machine.

2.The second rule, if a physical machine is in retiring state for a lengthy time then in place of waiting, all the running virtual machines are migrated to other physical machine. After the successful immigration, we can shut down the physical machine.

This algorithm mechanism to accumulate more power by shut down the physical machines. Hence the charge for power expenditure is low. But it does not range up for large data centers.

3.2 Hybrid Algorithm

It is the mixture of both Dynamic Round Robin and First - Fit algorithms [1]. It is used to decrease the power expenditure by physical machines. The First - Fit algorithm is functioned during rush hours to

enlarge resource utilization of physical machines and Dynamic Round Robin algorithm is functioned for the duration of non-rush hours to combine virtual machines and shut down the physical machines. This algorithm is used for improving Resource allocation and Power Consumption. This algorithm does not balance up for large data centers.

3.3 Equally Spread Current Execution (ESCE) Algorithm

Each job size in the queue should be estimates by the Cloud Manager and seem to be for the availability of resources for that job [8]. If every one of the resources are obtainable for that job then instantly the job scheduler allocates that job to the resource. This algorithm is for increase response time and processing moment in time of a job. But it is not fault tolerant and it has the trouble of single point of disappointment.

3.4 Central Load Balancing Policy for Virtual Machine

This algorithm balances load consistently across the distributed systems and cloud computing environments [9]. This policy increases the whole performance of the system but does not consider the systems that are fault-tolerant.

3.5 Enhanced Equally Distributed Load Balancing Algorithm

This algorithm [9] works to allocate the load equally across all the nodes in cloud environment. It handles the requests with priorities. It is a distributed algorithm by which the load can be distributed not only in an evenhanded manner but also it assign the load systematically by checking the counter variable of each data center. After checking, it conveys the load accordingly (i.e.) least amount value of the counter variable will be elected and the request is handling simply and takes less time and gives maximum throughput. For every arrival of job, the counter variable is enlarged by 1 and for dispatching a job; the counter variable is reduce by 1. The algorithm is performed by a single central server node. In this method, the central server node gets overhead by getting more jobs. The algorithm assigns job support on the counter variable and the weight will not considered.

3.6 Decentralized Content aware Load Balancing Algorithm

It is based on Workload and Client Aware Policy (WCAP) [10]. It uses a Unique as well as Special Property (USP) to specify the distinct and special property of the requests as well as computing nodes. USP facilitate the scheduler to make a decision for the best suitable node for processing the requests. This strategy is implementing in a decentralized manner with low down overhead. By using the contented information to narrow down the investigate, this technique improves the penetrating performance of the system. It also helps in decreasing the still time of the computing nodes therefore increasing their utilization.

3.7 Join-Idle-Queue

This algorithm presents huge scale load balancing with spread dispatchers by, primary load balancing the still processors across dispatchers for the availability of still processors at each dispatcher and then, assigning jobs to the processors to reduce average queue length at each processor [10]. By eliminate the load balancing work from the vital path of request processing, it effectively decreases the system load, incurs no communication overhead at job arrivals and does not increase actual response time.

3.8 Honeybee Foraging Behavior Algorithm

It is a decentralized honeybee-based load balancing method that is a nature-inspired algorithm for self-organization [10]. It achieves worldwide load balancing through local server actions. Performance of the

system is improved with better system diversity but throughput is not improved with an increase in system size. It is best match for the conditions where the diverse population of service types is essential.

3.9 Min-Min Algorithm

Every job has the execution time and completion time [7]. The cloud manager identifies both the execution time & completion time of every impulsive job in queue. The job which has least completion time is identified and then allocates the job to the processor that has the ability to complete the job within its particular completion time. But larger task has to be waited for long period of time in the queue.

3.10 Max-Min Algorithm

It's mechanism as Min-Min algorithm. But it provides more priority to the larger tasks [11]. The jobs that have huge execution time or vast completion time are executed first. The major problem is that tiny jobs have to be waiting for lengthy time.

3.11 RASA Algorithm

It's mingled with Min-Min and Max-Min Algorithms [11]. The algorithm builds a matrix C where C_{ij} represents the completion time of the task T_i on the resource R_j. If the available resources is odd, the Min-Min Algorithm is applied to allocate the first task, otherwise the Max-Min algorithm is submit an application. The remaining tasks are assigned to their appropriate resources by one of the two strategies alternatively. Another exchange of Min-Min and Max-Min strategies results in consecutive execution of a small and a large task on different resources and hereby, the waiting time of the small tasks in Max-Min algorithm and the waiting time of the large tasks in Min-Min algorithm are ignored.

3.12 Improved Max-Min Algorithm

Max-Min Algorithm extension is Improved Max-Min Algorithm [12]. The Max-Min algorithm selects the task with the utmost finishing point time and allocate it to the resource on which achieve minimum execution time. The basic idea of a better version of Max-Min algorithm assign task with maximum execution time to resource produces minimum complete time rather than original Max-Min assign task with maximum completion time to resource with minimum execution time. It uses the advantages of Max-Min and also covers its disadvantages.

3.13 2-Phase Load Balancing Algorithm

Combined OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) Scheduling algorithms to make use of better executing efficiency and preserved the load balance of the system [7]. OLB scheduling algorithm keeps each node in functional state to complete the target of load balance and LBMM scheduling algorithm is utilized to decrease the execution of time of each task on the node thereby minimizing on the whole completion time. This algorithm works to develop the utilization of resources and increases the work efficiency.

3.14 Power Aware Load Balancing (PALB) Algorithm

This algorithm is applied in cluster controller [13]. There is a presence of Job Scheduler whose work is to simulate requests from users for virtual machine instances. The cluster controller maintains the utilization state of each active compute node and makes decisions on where to instantiate new virtual machines. This algorithm is mainly designed to reduce the power consumption. This algorithm is used to power off the physical machines that are in idle state rather than entering into low power state. But it does not scale up for large cloud data centers.

IV. COMPARISON OF LOAD BALANCING ALGORITHMS

TABLE 1 show the comparison of LB algorithms which were discussed above.

Algorithm	Description	Advantages
Dynamic Round Robin Algorithm	1. Uses two rules to save the power consumption 2. Works for consolidation of VM	Reduce the power consumption
Hybrid Algorithm	1. Combination of Dynamic Round Robin and First-Fit Algorithm 2. Applied in non-rush hours and rush hours	1. Improved Resource Utilization 2. Reduced Power Consumption
ESCE Algorithm	Estimate the size of job and look for availability of resources	Improved response time and processing time
Central Load Balancing policy for VM	Balances the load evenly	Improves overall performance
Enhanced Equally Distributed Load Balancing Algorithm	Based on the counter variable, the job is allocated by Central Server	1. Computing Resource is distributed efficiently and fairly 2. Reduces request to response ratio
Decentralized Content Aware Load Balancing Algorithm	1. Uses Unique and Special Property(USP) of nodes 2. Uses content information to narrow down the search	1. Improves the searching performance hence increasing overall performance 2. Reduces idle time of nodes

Join-Idle Queue Algorithm	1. Assigns idle processors to dispatchers for the availability of idle processors 2. Then assigns jobs to processors to reduce average queue length	1. Reduces system load 2. Less communication overhead
Honeybee Foraging Behavior	Achieves global load balancing through local server actions	Improved scalability
Min-Min Algorithm	1. Estimates minimum execution time and minimum Completion time 2. Jobs having minimum completion time is executed first	Smaller tasks are executed quickly
Max-Min Algorithm	1. Same as Min-Min 2. Gives more priority to larger tasks than smaller one	Larger tasks are executed quickly and efficiently
RASA Algorithm	Combination of both Min-Min and Max-Min Algorithms	1. Efficient resource allocation 2. Minimum execution time
Improved Max-Min Algorithm	1. Improved version of Max-Min Algorithm 2. Assigns task with minimum execution time	Scheduling jobs effectively
2-Phase Load Balancing Algorithm	1. Uses OLB to keep each node busy 2. Uses LBMM to achieve minimum execution time of each job	1. Efficient utilization of resources 2. Enhances work efficiency

PALB Algorithm	1. Implemented in Cluster Controller 2. Use Job Scheduler to simulate requests from users for virtual machine instances	Physical Machines that are in idle state are move to power off state to conserve energy

Table 1. Comparison of LB Algorithms

V. CONCLUSION

Although there are quite a few issues in cloud environment, it has been broadly accepted by many organizations and industries. Researchers are doing many works to get resolution for those issues. For Load Balancing issue, the solution is to build up suitable algorithms that balance the load across the cloud environment. The new algorithm should decrease the server overhead, improve throughput, improve performance, decrease the server power consumption and also give out the load across the entire nodes.

REFERNECES

[1] Gaochao Xu, Junjie Pang and Xiaodong Fu, "A Load Balancing Model Based on Cloud Partitioning for the Public Load", IEEE Transactions on Cloud Computing, 2013.
[2] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid,"Availability and Load Balancing in Cloud Computing",International Conference on Computer and Software modeling IPCSI, 2011.
[3] Ratan Mishra and Anant Jaiswal, "Ant Colony Optimization:A solution of Load Balancing in Cloud", International Journal of Web & Semantic Technology (IJWesT), April 2012.
[4] Venubabu Kunamneni, "Dynamic Load Balancing for the cloud", International Journal of Computer Science and Electrical Engineering, 2012.
[5] Karanpreet Kaur, Ashima Narang, Kuldeep Kaur, "Load Balancing Techniques of Cloud Computing", International Journal of Mathematics and Computer Research, April 2013.
[6] Dr.Hemant S.Mahalle, Prof. Parag R. Kaveri, Dr. Vinay Chavan, "Load Balancing on Cloud Data Centers", International Journal of Advanced Research in Computer Science and Software Engineering, January 2013.
[7] Shreyas Mulay, Sanjay Jain, "Enhanced Equally Distributed Load Balancing Algorithm for Cloud Computing" International Journal of Research in Engineering and Technology, June 2013.
[8] Nidhi Jain Kansal, Inderveer Chana, "Cloud Load Balancing Techniques: A Step Towards Green Computing", International Journal of Computer Science, January 2012.
[9] S.Mohana Priya, B.Subramani, "A New Approach for Load Balancing in Cloud Computing", International Journal of Engineering and Computer Science, May 2013.
[10] O.M.Elzeki, M.Z.Reshad, M.A.Elsoud, "Improved Max-Min Algorithm in Cloud Computing", International Journal of Computer Applications, July 2012.
[11] Jeffrey M. Galloway, Karl L.Smith, Susan S. Vrbsky, "Power Aware Load Balancing for Cloud Computing", World Congress on Engineering and Computer Science, 2011.

