Load Balancing in Cloud Computing

Sukhpreet Kaur[#1]

[#1]*Assistant Professor, Department of Computer Science, Guru Nanak College, Moga, India*

[1]sukhpreetchanny50@gmail.com

Abstract:

Cloud computing helps to share data and provides many resources to users. In cloud computing paradigm, load balancing is one of the challenges. With tremendous increase in users and their demand of different services on the cloud computing platform, efficient usage of resources in the cloud environment became a critical concern. Load balancing helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Load balancing technique is used to distribute tasks from over loaded resources to under loaded or idle resources. Many algorithms are suggested to enhance the overall performance of the Cloud and provide the user more satisfying and efficient service. In this paper, we investigate the different algorithms proposed to resolve the issue of load balancing in cloud computing.

Keywords:

Cloud Computing, Load balancing, Load balancing Algorithms.

1. Introduction

Cloud Computing(CC) has become the essential requirement for the IT companies. In cloud computing the term cloud means 'Internet'. It has moved Cloud computing refers to hardware resources and software services provided over the internet rather than physically having the computing resources at the customer location. Cloud computing(CC) allows the end users to pay only what resources they have been used. Due to significant cost saving many smaller and medium sized organizations are also looking forward for using cloud services. The emerging demand for cloud services is driven by continuing globalization, consumer acceptance of technology, economic downturn and the growth of the extended enterprise. Cloud Computing [1] enables many organizations to limit the large capital investment that is associated with costly data centres and for the applications and transforming these costs into operating expenses paying for cloud resources only as required. Today there are many cloud services providers are available- Google Cloud, AWS, Amazon EMC2 , OpenStack, CloudStack, Open Nebula etc.

Load balancing is one of the central issue in cloud computing(CC). Load balancing is the mechanism of spreading the load among various nodes of a distributed system to improve both resource utilization and job response time. It also helps avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work [2]. It also ensures that all the processing in the system or every node in the network does approximately the equal amount of work at any instance of time. The basic reason for this requirement is the rapid increase in the number of users and their demand for cloud services. Load balancing helps to achieve a high user satisfaction and resource utilization.

The responsibility of load balancing algorithm is that to map the jobs which are set forth to cloud domain to the unoccupied resources so that the overall available response time is improved as well as provides efficient resource utilization.

Load balancing is achieved by using multiple resources that is , multiple servers that are able to fulfil a request or by having multiple paths to a resource. When one or more components of any service fail, load balancing facilitates continuation of the service by implementing fair-over, that is , it helps in provisioning and de-provisioning of instances of applications without fail. It also ensures that every computer resource is distributed efficiently and fairly. [3]. Load balancing enables scalability, avoids bottlenecks and also reduces time taken to give the respond. The following figure 1 shows the structure of load balancing in Cloud Computing(CC)[4]:

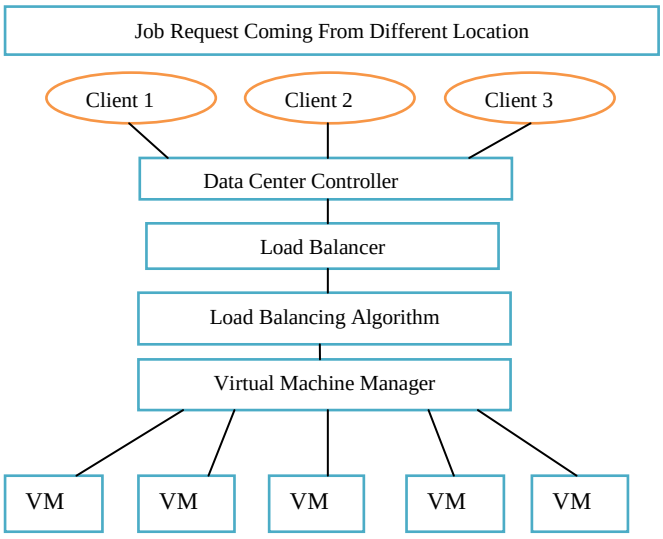

Figure 1: Load balancing in Cloud Computing

2. Classification of Load balancing Algorithms:

Load balancing algorithms can be broadly classified into two types:

A. Static algorithms:

The static load balancing algorithms doesn't depend on the current state of the system. Prior knowledge to the system is needed. In static load balancing algorithms the assignment of tasks to processor is done before program execution begins, at compile time. The process of scheduling is based on prior knowledge of node's properties and capabilities such as node's execution time, processing resources, memory and storage capacity etc[5]. Which are assumed to be known at compile time. These algorithms are used in the environment where there are few load variations. These algorithms cannot adopt to load changes during run time. The goal of static load balancing is to minimize the overall execution time.

B. Dynamic algorithms :

Dynamic algorithms work on current state of nodes and distributes load among the nodes. No prior knowledge is needed. A dynamic algorithm redistributes the processes among processors during execution time[6]. This redistribution is performed by transferring tasks from the heavily loaded processor to the lightly loaded processors to improve the performance. Dynamic algorithms react to the system state that changes dynamically. Dynamic algorithms constantly checks the different properties of the nodes such as its capability , network bandwidth processing power, memory and storage capability and other parameters thereby assigning suitable weights to the processors[7]. However they are more accurate and could result in more efficient load balancing. Major drawback of Dynamic algorithms is the run-time overhead due to the transfer of load information among processor and decision-making for the selection of processes and communication delays associated with the task relocation itself. Dynamic load balancing algorithms can be centralized or distributed , depending on whether the responsibility for task of global dynamic scheduling should physically reside in the single processor (centralized) or the work involved in making decisions should be physically distributed among processor.

3. Performance Measurement of Load balancing in CC:

Following are the parameters used to assess various load balancing techniques to get improved distribution of recourses as per demands of the cloud users[8].

A. Throughput: It is used to calculate number of tasks whose execution has been completed in a given amount of time. The performance of any system is improved if throughput is high.

B. Response time: It is amount of time that is taken by a particular load balancing algorithm to response a task in a system. This parameter should be minimized for better performance of a system.

C. Fault Tolerance: It means recovery from failure. The load balancing should be a good fault tolerance technique.

D. Scalability: It is ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved for efficient load balancing.

E. Migration time: It is time to migrate the tasks or resources from one node to other nodes. It should be minimized in order to enhance the performance of the system.

4. Load Balancing Algorithms:

A. Round Robin Algorithm:

Round Robin algorithm uses the time slice mechanism. Here the time is divided into multiple slices and each node is given a particular time quantum or time interval and in this quantum the node will perform its operations. The resources of the service provider are provided to the client on the basis of this time quantum. This algorithm simply allots the job in round robin fashion which doesn't consider the load on different machines. As a result, at any moment some node may possess heavy load and other mat have no request.

In Round Robin Scheduling the time quantum play a very important role for scheduling, because if time quantum is very large then Round Robin Scheduling Algorithm is same as the FCFS Scheduling. If the time quantum is extremely too small then Round Robin Scheduling is called as Processor Sharing Algorithm and number of context switches is very high. Though the algorithm is very simple but there is an additional load on the scheduler to decide the size of quantum [3] and it has longer average waiting time, higher context switches higher turnaround time and low throughput.

B. Equally Spread Current Execution Algorithm (ESCE):

In spread spectrum technique load balancer makes effort to preserve equal load to all the virtual machines connected with the data centre. Load balancer maintains an index table of Virtual machines as well as number of requests currently assigned to the Virtual Machine (VM).

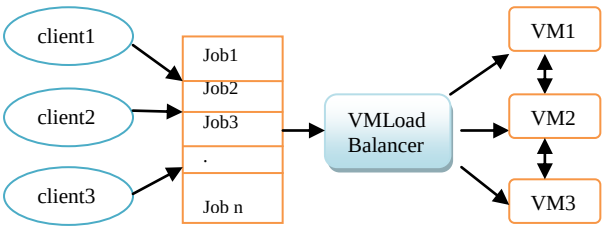

Figure 2: ESCE Algorithm

If the request comes from the data centre to allocate the new VM, it scans the index table for least loaded VM. In case there are more than one VM is found than first identified VM is selected for handling the request of the client/node, the load balancer also returns the VM id to the data centre controller. The data centre communicates the request to the VM identified by that id. The data centre revises the index table by increasing the allocation count of identified VM. When VM completes the assigned task, a request is communicated to data centre which is further notified by the load balancer. The load balancer again revises the index table by decreasing the allocation count for identified VM by one but there is an additional computation overhead to scan the queue again and again. The figure 2 shows the working of ESCE algorithm.

C. Min-Min Algorithm:

This algorithm starts with a task set which are initially not assigned to any of the nodes[10]. At first, the minimum completion time is calculated for all the available nodes. Then, the task with minimum expected completion time is selected and assigned to the node with minimum execution time. This task is then removed from the task-set. This process is repeated until all the tasks have been assigned to the equivalent nodes. The algorithm is better when the situation is like where the number of small tasks are greater than the number of larger tasks. The descriptive flow of the algorithm is presented in the below Figure 3.

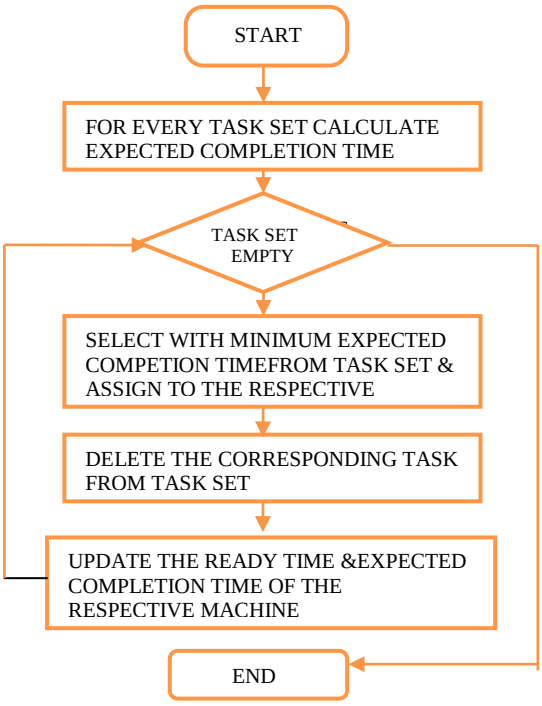

Figure 3. Flowchart for Min-Min algorithm

D. Max-Min Algorithm:

The max-min algorithm is much same as min-min algorithm. Max-Min algorithm starts with the set of all the submitted tasks in the task-set that are initially unassigned to any node[11]. At first, the minimum completion time for all available tasks is calculated. Then the task with maximum completion time is selected and assigned to the resource with minimum execution time. This task is removed from the task-set and process is repeated until the task-set is empty.

This algorithm outperforms the Min-Min algorithm where short tasks are in high numbers as compared to long ones. For example: if in a task set only a single long task is presented then the Max-Min algorithm runs short tasks concurrently along with long task. The algorithm suffers from starvation where the tasks having the maximum completion time will get executed first while leaving behind the tasks having the minimum completion time.

E. Throttled Algorithm:

This algorithm works by finding the appropriate virtual machine for assigning a particular task. In this algorithm the load balancer maintains an index table of virtual machines as well as their states(Available or Busy)[11]. The client first makes a request to Data Centre to find a suitable virtual machine to perform required operation. The Data Centre receives the request from client for the allocation of Virtual Machine. Then, Data Centre queries the load balancer for allocation of VM. The load balancer scans the index table from top until the first available VM is found or index table is scanned fully.

If VM is found, the VM id is send to the Data Centre. Then the Data Centre communicates the request to the VM identified by the id. Further, the Data Centre acknowledges the load balancer of the new allocation and the Data Centre revises the index table accordingly.

During processing the request of the client, if VM is not found, the load balancer returns -1 to the Data Centre. The Data Centre queues the request until the next availability of VM. When the VM finishes processing request, it sends results to the Data Centre and acknowledges load balancer for de-allocation of VM. Load balancer updates the allocation table by decreasing the allocation for VM by 1.

The total execution time is estimated in three phases[12]. In the first phase the formation of the virtual machines and they will be idle waiting for the scheduler to schedule the jobs in the queue, once jobs are allocated, the virtual machines in the cloud will start processing, which is the second phase, and finally in the third phase the cleanup or the destruction of the virtual machines. The throughput of the computing model can be estimated as the total number of jobs executed within a time span without considering the virtual machine formation time and destruction time The proposed algorithm will improve the performance by providing the resources on demand, resulting in increased number of job executions and thus reducing the rejection in the number of jobs submitted.

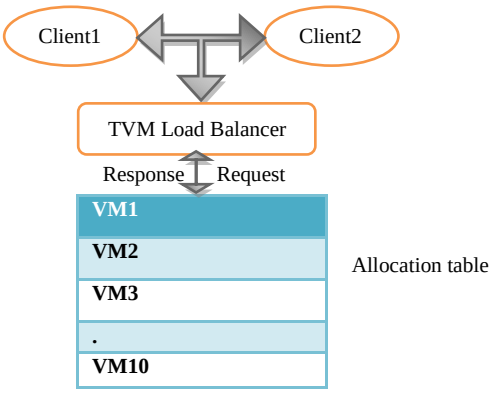

Figure 4:Throttled Algorithm

F. Ant Colony Optimization Algorithm:

ACO algorithm is based on the behaviour of real ants [13]. It is based on the ability of ants to find an optimal path from nest to source. In ACO algorithm, when the request is initiated, the ant starts its movement. Ants originate from the root node and moves from one node to next node and check whether the node is overloaded or under loaded. When ants move over the network, they update the pheromone table which stores the information about each node's utilization. Ant move in two ways:

Forward Movement: The ants continuously move in the forward direction from one node to another node over the network and check whether it is overloaded or under loaded, if ant find an overloaded node it will continuously moving in the forward direction and check every node.

Backward Movement: If an ant finds an overloaded node in its movement when it has previously encountered an under loaded node then it will go backward to the under loaded node to check if it is still under loaded or not and if it finds it still under loaded then it will redistribute the work to the under loaded node.

G. Honey Bee Algorithm:

Honey Bee algorithm is based on the foraging behaviour of honey bees. The artificial bee colony contains three groups of bees: scout bees, Forager bees and onlooker bees [14]. Scout bees are sent for search of suitable food source, when they found the source, they return to the hive and advertise it in the form of "Waggle Dance". The suitability of the food source and its distance from the hive is communicated through the waggle dance display. Now, the forager bees follow the scout bees back to the discovered food source and begin to harvest it. After collecting the food they return to the hive and the remaining quality of the food available at the source is shown in the form of Waggle dance to decide that the remaining bees should sent to the same source or to search the new suitable source of food.

Table 1: Mapping of Honey Bee Behavior to a Cloud Environment

Honey bee Hive	Cloud Environment
Honey bee	**Task (also called as cloudlet)**
Source of the food	**Virtual Machine**
Honey bee foraging a food source	**Task is being loaded to a VM**
Honey bee reduction at food source	**VM is overloaded**
Finding new food source	**Removed task scheduling to under loaded VM**

First of all we calculate the capacity and load of all VMs and then group them into three categories as under:-

1. Overloaded Virtual Machine(OVM)
2. Under loaded Virtual Machine(UVM)
3. Balanced Virtual Machine(BVM)

If there is more than one under loaded VM in UVM list, select a VM in such a way that it has minimum objective function value so that the currently selected task will get executed faster.

5. CONCLUSION:

In this paper, we surveyed multiple algorithms for load balancing in cloud computing. Cloud computing has widely been adopted by the industry, through there are many existing issues like Load balancing, Virtual machine Migration, Server Consolidated, Energy Management, etc. Central issue to these is the issue of load balancing, that is required to distribute the excess dynamic local workload evenly to all the nodes in the whole cloud to get higher user satisfaction and resource utilization ratio. In this paper, We have described and compared different static and dynamic load balancing algorithms for cloud computing such as, round robin, Min-Min, Max-Min, Throttled Algorithm etc. considering the characteristics like throughput, fault tolerance, overhead, speed and complexity.

REFERENCES:

[1] Suruchee V.Nandgaonkar, Prof. A. B. Raut, "A Comprehensive Study on Cloud Computing", International Journal of Computer Science and Mobile Computing, Vol.3 Issue.4, April- 2014, pg. 733-738

[2] Shanti Swaroop moharana, Rajadeepan d. Ramesh & Digamber Powar , " Analysis of load balancers in cloud computing" International Journal of Computer Science and Engineering (IJCSE), Vol. 2, Issue 2, May 2013, 101-108.

[3] Tanveer Ahmed, Yogendra Singh, "Analytic Study Of Load Balancing Techniques Using Tool Cloud Analyst", International Journal of Engineering Research and Applications, Vol. 2, Issue 2,Mar-Apr 2012.

[4] Dharmesh Kashyap, Jaydeep Viradiya, " A Survey Of Various Load Balancing Algorithms In Cloud Computing" , International Journal of Scientific & Technology Research, Volume 3, issue 11, November 2014.

[5] Shikha Gupta, Suman Sanghwan, "Load Balancing in Cloud Computing: A Review", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 6, June 2015.

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

[7] Dr.S. Suguna and R. Barani, "Simulation of Dynamic Load Balancing Algorithms" , Bonfring International Journal of Software Engineering and Soft Computing, Vol. 5, No.1, July 2015

[8] Namrata Swarnkar, Asst. Prof. Atesh Kumar Singh, Dr. R. Shankar, "A Survey of Load Balancing Techniques in Cloud Computing", International Journal of Engineering Research & Technology ,Vol. 2 Issue 8, August - 2013

[9] MR.Manan D. Shah, MR.Amit A. Kariyani ,MR.Dipak L. Agrawal, "Allocation Of Virtual Machines In Cloud Computing Using Load Balancing Algorithm", IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555 Vol. 3, No.1, February 2013.

[10] Gaurav R. et al. "Comparative Analysis of Load Balancing Algorithms in Cloud Computing." International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1, No. 3, pp.120-124, May 2012.

[11] Foram F Kherani, 2Prof.Jignesh Vania , "Load Balancing in cloud computing" , International Journal of Engineering and Research, Volume 2, Issue 1 , 2014.

[12] Durgesh Patel , Mr. Anand S Rajawat , "Efficient Throttled Load Balancing Algorithm in Cloud Environment" , International Journal of Modern Trends in Engineering and Research , Volume 02, Issue 03, [March - 2015]

[13] Shagufta K., Niresh S, "Ant Colony Optimization for Effective Load Balancing in Cloud Computing" , Volume 2, Issue 6, November-December 2013, pp 78.

[14] Walaa Hashem, Heba Nashaat, and Rawya Rizk, "Honey Bee Based Load Balancing in Cloud Computing", Ksii Transactions On Internet And Information Systems Vol. 11, No. 12, Dec. 2017.