ISSN 2277-3061
Volume 15 Number 04
INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

OPTIMIZED LOAD BALANCING STRATEGY IN CLOUD COMPUTING : A REVIEW

Parveen Kaur [(1)], Monika Sachdeva [(2)]
[(1)] Research Scholar, Department of Computer Science Engineering, SBSSTC, Ferozepur
peenajosan23@gmail.com
[(2)] Associate Professor, Department of Computer Science Engineering, SBSSTC, Ferozepur
monika.sal@rediffmail.com

ABSTRACT

Now a days every organization is migrating towards cloud computing as cloud computing is considered being more flexible and scalable as compared to other technologies. The technology simply means to provide the computing resources and services through a network. This paper discusses the existing approaches for scheduling algorithms that can maintain the load balancing and provides better improved strategies through efficient job scheduling and modified resource allocation techniques. 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 utilization 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. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time.

Keywords

Cloud Computing, Load Balancing, VM, Host, Datacenter, ESCE, Throttled.

INTRODUCTION

Cloud computing provide infrastructure, platform, and software as services. These services are using pay-as-you use model to customers, regardless of their location. Cloud computing is a cost effective model for provisioning services and it makes IT management easier and more responsive to the Changing needs of the business [1]. The access to the infrastructure incurs payments in real currency in cloud environment. Today network bandwidth, Less response time, minimum delay in data transfer and minimum data transfer cost are main challenging issues in cloud computing load balancing environment. Cloud sim is simulation based approach. The Simulation based approaches provide significant Benefits, as it allows researchers to test their proposed algorithms and protocols in a repeatable and controlled environment free of cost, and to find solution to the performance bottlenecks before deploying in the real cloud. Cloud computing involving distributed technologies to satisfy a variety of applications and user needs. Share resources, software, information via internet are the main functions of cloud computing to reduced cost, better performance and satisfy needs. To improve the response time of the job, distribute the total load of the collective system. By this removing a condition in which some of nodes are overloaded while some other are under loaded. Load balancing algorithms dose not taken the previous state or behaviour of the system, it depends upon the present behaviour of the system because it is dynamic in nature. Round robin algorithm process on circular order by handling the process without priority but equally spread current execution handle the process with priories. Throttled algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation. The architecture is complete formation for virtual machines, less response time and minimum delay to transfer. Therefore model estimated the virtual machine cost and low data transfer cost. This type of computational model promises to reduce the capital and operational cost of the client. The total execution time is estimated in three phases. 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.

LOAD BALANCING FOR CLOUD

Load balancing is one of the main issues related to cloud computing. The load can be a memory, CPU capacity, network or delay load. It is always required to share work load among the various nodes of the distributed system to improve the resource utilization and for better performance of the system. This can help to avoid the situation where nodes are either heavily loaded or under loaded in the network. Load balancing is the process of ensuring the evenly distribution of work load on the pool of system node or processor so that without disturbing, the running task is completed. The goals of load balancing [1] are to:

- Improve the performance
- Maintain system stability
- Build fault tolerance system
- Accommodate future modification.

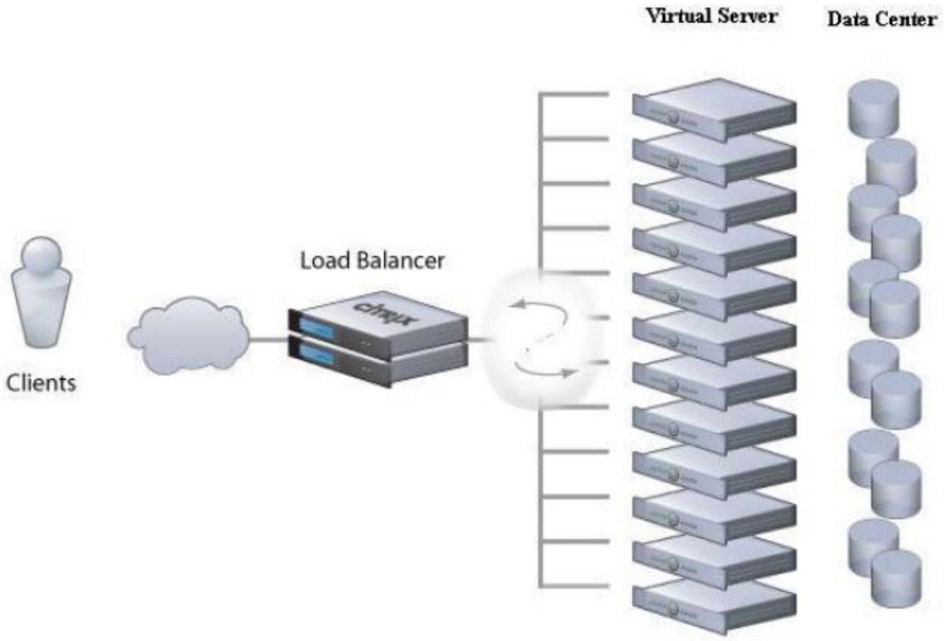

Figure 1. Load Balacing in Cloud Environment

STATIC ALGORITHM

In static algorithm the traffic is divided evenly among the servers. This algorithm requires a prior knowledge of system resources, so that the decision of shifting of the load does not depend on the current state of system. Static algorithm is proper in the system which has low variation in load.

DYNAMIC ALGORITHM

In dynamic algorithm the lightest server in the whole network or system is searched and preferred for balancing a load. For this real time communication with network is needed which can increase the traffic in the system. Here current state of the system is used to make decisions to manage the load.

METRICS FOR LOAD BALANCING IN CLOUD

Various metrics considered in existing load balancing techniques in cloud computing are discussed below-

- **Scalability** is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.
- **Resource Utilization** is used to check the utilization of re-sources. It should be optimized for an efficient load balancing.
- **Performance** is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.
- **Response Time** is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.
- **Overhead Associated** determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and interprocess communication. This should be minimized so that a load balancing technique can work efficiently.

LOAD BALANCING ALGORITHMS

In order to balance the requests of the resources it is important to recognize a few major goals of load balancing algorithms:

a) **Cost effectiveness**: primary aim is to achieve an overall improvement in system performance at a reasonable cost.

b) **Scalability and flexibility**: the distributed system in which the algorithm is implemented may change in size or topology. So the algorithm must be scalable and flexible enough to allow such changes to be handled easily.

c) **Priority**: prioritization of the resources or jobs need to be done on before hand through the algorithm itself for better service to the important or high prioritized jobs in spite of equal service provision for all the jobs regardless of their origin.

Following load balancing algorithms are currently prevalent in clouds:-

Round Robin: In this algorithm [7], the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. Thi s algorithm is mostly used in web servers where http requests are of similar nature and distributed equally.

Connection Mechanism: Load balancing algorithm [8] can also be based on least connection mechanism which is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically to estimate the load. The load balancer records the connection number of each server. The number of connection increases when a new connection is dispatched to it, and decreases the number when connection finishes or timeout happens.

Randomized: Randomized algorithm is of type static in nature. In this algorithm [7] a process can be handled by a particular node n with a probability p. The process allocation order is maintained for each processor independent of allocation from remote processor. This algorithm works well in case of processes are of equal loaded. However, problem arises when loads are of different computational complexities. Randomized algorithm does not maintain determini stic approach. It works well when Round Robin algorithm generates overhead for process queue.

Equally Spread Current Execution Algorithm: Equally spread current execution algorithm [9] process handle with priorities. it distribute the load randomly by checking the size and transfer the load to that virtual machine which is light ly loaded or handle that task easy and take less time , and give maximize throughput. It is spread spectrum technique in which the load balancer spread the load of the job in hand into multiple virtual machines.

Throttled Load Balancing Algorithm: Throttled algorithm [9] is completely based on virtual machine. In this client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is give by the client or user. In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation.

A Task Scheduling Algorithm Based on Load Balancing: Y. Fang et al. [10] discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.

Min-Min Algorithm: It begins with a set of all unassigned tasks. First of all, minimum completion time for all tasks is found. Then among these minimum times the minimum value is selected which is the minimum time among all the tasks on any resources. Then according to that minimum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. Then again the same procedure is followed until all the tasks are assigned on the resources. But this approach has a major drawback that it can lead to starvation [12].

Max-Min Algorithm: Max-Min is almost same as the min-min algorithm except the following: after finding out minimum execution times, the maximum value is selected which is the maximum time among all the tasks on any resources. Then according to that maximum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines[12].

LITERATURE SURVEY

- [Wu et al.,2013b] proposed a task scheduling algorithm based on QoS-driven for cloud computing. Intially, it calculates the priority of the tasks based on various QoS parameters such as completion time, priviledges and then assign the task to a service which has minimum completion time.

- [Domanal and Reddy,2014] proposed a VM-assign load balancing algorithm which maintains an index table for the virtual machines. It finds the least loaded VM and then checks whether it is not the last used VM and assigns the load to the virtual machine. If it is the last used VM, then it finds the other least loaded VM from the index table.

- [Mesbahi et al., 2014] proposed a new multilevel cloud light weight architecture for load balancing in cloud computing. At the lower level lies the VMs. hosts and datacenters. At the middle level lies the VM manager and at the topmost level applies the Head Vm manager and ESB (Enterprise Level Bus). The algorithm proposed considers all the tasks of equal weight in terms of distributing workload and achieves load balancing as well as assures QoS to the users

- [Delavar and Aryan, 2014] proposed a hybrid heuristic algorithm to reduce the completion time. The algorithm computes the priority of tasks based on impact of tasks on each other in work flow graph. Then, it decides the suitable scheduling for these tasks and obtains early response time, load balancing and speedup ratio.

- [Sharma and Peddoju, 2014] proposed a load balancing algorithm based on the response time of the incoming requests. The algorithm only considers the response time of the incoming requests while scheduling the task to a virtual machine. It prevents extra computation and reduces communication cost.

- [Ren et al., 2011] proposed a load balancing algorithm for a cloud computing platform based on weighted least connection algorithm. The proposed algorithm is based on prediction and uses the historical data and uses the smoothing factor to make the recent data have greater impact than long term data. The algorithm is effective in reducing the load on real servers.

- [Wu et al., 2013a] proposed an elastic load balancing algorithm that uses prediction based on historical data. The algorithm is an improvement over traditional algorithm which does not respond to dynamic changes. This algorithm applies for the virtual machines in advance based on heuristic data and revises prediction results to handle the current load effectively.

PROBLEM DESCRIPTION

The algorithm used in Base paper is Active VM algorithm which is also called Equally spread current execution.

- ESCE load algorithm makes an effort to equally spread the execution load on different VM's. ESCE uses allocation table and reservation table.
- Reservation Table keeps detail of number of requests allocated to each VMs.
- Allocation table keeps detail of all virtual machine
- Use of allocation table and reservation table to keep detail of number of requests allocated to each VMs.
- Allocation of VM to any cloudlet using min_count(allocation_count + reservation_count) formula.

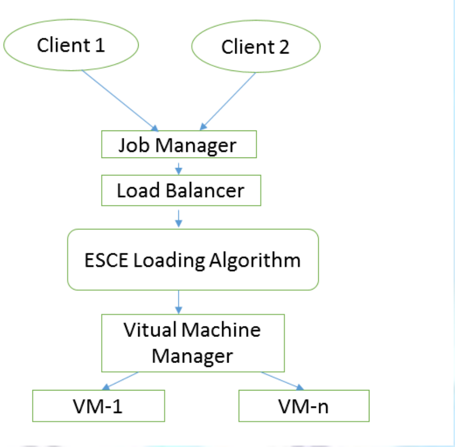

Figure 2. Load Balancing Algorithm

RESEARCH GAP

- The algorithm used in Base paper is ESCE which never considers the capacity of a virtual machine. This is can lead to improper load balancing.
- It does not include instruction size of each cloudlet before allocating to VM.
- Both above problems effects the overall performance of the system.
- The waiting time / processing time / processing cost are all affected because of the above said problems.

OBJECTIVES

- To study the performance of existing load balancing algorithms.
- To enhance the performance of overall system when request frequency is high during peak hours.
- To reduce the response time of user.
- To enhance the processing time of the cloudlet.
- To decrease the overall cost of the system.
- To enhance availability of VM to cloudlets during peak hours.
- To implement the proposed algorithm in cloudSim simulator and compare the performance of the proposed work with the existing algorithms.

CONCLUSION

This paper is based on cloud computing technology which has a very vast potential and is still unexplored. The capabilities of cloud computing are endless. Cloud computing provides everything to the user as a service which includes platform as a service, application as a service, infrastructure as a service. One of the major issues of cloud computing is load balancing because overloading of a system may lead to poor performance which can make the technology unsuccessful. So there is always a requirement of efficient load balancing algorithm for efficient utilization of resources. Our paper focuses on the various load balancing algorithms and their applicability in cloud computing environment.

REFERENCES

[1] Wu, H.-S., Wang, C.-J., and Xie, J.-Y. (2013a). Terascaler elb-an algorithm of predictionbased elastic load balancing resource management in cloud computing. In Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on, pages 649-654. IEEE.

[2] Wu, X., Deng, M., Zhang, R., Zeng, B., and Zhou, S. (2013b). A task scheduling algorithm based on qos-driven in cloud computing. Procedia Computer Science, 17:1162-1169.

[3] Sharma, A. and Peddoju, S. K. (2014). Response time based load balancing in cloud computing. In Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on, pages 1287-1293. IEEE.

[4] Ren, H., Lan, Y., and Yin, C. (2012). The load balancing algorithm in cloud computing environment. In Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on, pages 925-928. IEEE.

[5] Raju, R., Amudhavel, J., Kannan, N., and Monisha, M. (2014). A bio inspired energy-aware multi objective chiropteran algorithm (eamoca) for hybrid cloud computing environment. In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on, pages 1-5. IEEE.

[6] Mesbahi, M., Rahmani, A. M., and Chronopoulos, A. T. (2014). Cloud light weight: A new solution for load balancing in cloud computing. In Data Science & Engineering (ICDSE), 2014 International Conference on, pages 44-50. IEEE.

[7] Domanal, S. G. and Reddy, G. R. M. (2013). Load balancing in cloud computingusing modified throttled algorithm In Cloud Computing in Emerging Markets (CCEM), 2013 IEEE International Conference on, pages 1-5. IEEE.

[8] Domanal, S. G. and Reddy, G. R. M. (2014). Optimal load balancing in cloud computing by efficient utilization of virtual machines. In Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on, pages 1-4. IEEE.

[9] Delavar, A. G. and Aryan, Y. (2014). Hsga: a hybrid heuristic algorithm for work flow scheduling in cloud systems. Cluster computing, 17(1):129-137.