International Journal of Trend in Research and Development, Volume 3(3), ISSN: 2394-9333
www.ijtrd.com

A Holistic Approach to Load Balancing of Cloud Services

[1]E. Ravi Kumar and [2]K. Kotaiah Swamy,
[1]Associate Professor, [2]Assistant Prof (Sr Grade),
[1,2]Department of Information Technology, Vardhaman College of Engineering, Hyderabad, India.

Abstract: The cloud computing is a kind of Internet-based computing that makes available shared processing resources, data and storage space to computers and other devices on demand by pay as you go. The Load balancing in cloud computing environment has a great impact on the performance; Good load balancing makes cloud computing more efficient and improves user satisfaction. This paper (research) is going to address the issues like an optimal and feasible solution to cloud user by providing better utilization of the resources with less amount of time and fast throughput of the system.

Keywords: *Cloud Computing, Load Balancing.*

I. INTRODUCTION

Cloud computing is a new computing paradigm that involves a real-time communication network for large scale distributed computing environment, with extensive use of virtualization. Cloud computing does not mean the applications that are delivered over the internet, but also the underlying hardware and the software systems that are present in the data centers. The services being provided are termed under Software as a Service (SaaS), while the system software and the hardware make up the Cloud. If t he Cloud is made available on a pay-per-use basis, it is called a Public Cloud; the service being sold is Utility Computing. On the other hand, there are some internal data centers of some organizations that are not made available to the public.

In Cloud computing environment, different tasks may arrive randomly at different times and use CPU time randomly which increase the burden on one resource may heavily while the other resources may be free or very less utilized. This increases the load on one resource heavily such that at one point in time it may not handle the requests. A proper load balancing approach may solve these problems by fairly distributing load among different processing elements.

According to the Cloud Security Alliance, the top three threats in the cloud are Insecure Interfaces and API's, Data Loss & Leakage, and Hardware Failure—which accounted for 29%, 25% and 10% of all cloud security outages respectively.

II. RELATED WORK

We have plenty of load balancing algorithms for load balancing. Load balancing among different nodes of distributed systems can be done static algorithms and dynamic algorithms.

A. Static Algorithms

Static algorithms better suitable for those systems where load variation is low. In these algorithms, the load will be distributed evenly among the systems. Prior knowledge of system resources is required for these algorithms so that decision making is easy for load shifting does not depend on the current state. Some of the algorithms are discussed here.

Random: The random algorithm is static load balancing algorithm in nature[3]. A random number generator is used to select the nodes randomly[4]. Processes are handled by node n with a particular probability p [5]. The order of the process allocation is maintained for each processor. The algorithm works well with equally loaded processes but will not work if the loads are of different computational complexities. The algorithm does not follow a deterministic approach.

FCFS: The simplest load balancing algorithm is First Come First Serve algorithm [5] where each load balancer maintains a job queue in which job waits for its turn to get executed. The benefits of FCFS are being fast and simple. But FCFS results in a poor overall response time in case of small tasks have to wait for a longer time because of being at a later place in the queue.

B. Dynamic Algorithms

Dynamic algorithms are designed to deal with systems with irregular load variations. These algorithms try to search the servers with low weights so the dynamic requests can be assigned in the system. The load balancer will check the memory and the CPU usage. If the usage is very high then the client request will be transferred to the next available node.

Weighted RR: In Weighted RR method, a random weight is defined for each system. The number of connections that each machine receives over time is proportionate to the defined ratio weight. This algorithm works because whenever we define weighted assignments then requests are sent by the load balancer to machine for each request to the others. This algorithm works smoothly but has a problem because of the static definition of the weights in the beginning.

Dynamic Round Robin: DRR [4] is similar to Weighted Round Robin; the difference is that the servers are monitored continuously and the weights are kept on changing. It makes use of various aspects of real-time server performance analysis, like the current number of connections per node or the fastest node response time to distribute the connections. The drawback with this algorithm is that it is rarely available in a simple load balancer, because of its dynamic nature.

Throttled load balancing algorithm: In this algorithm [7], whenever a client request is received, the load balancer tries to find a suitable Virtual Machine to perform the required operation. The algorithm processes the client by maintaining a list of the all available VMs. In order to speed up the lookup process indexing is performed first. The request from a client is accepted if a match is found on the basis of size and availability of the machine. The VM is then allocated to the client. However, if a VM that matches the criteria is available, then the load balancer queues up the request.

Least connections: In this dynamic load balancing method, the load balancer records the number of connection for each server, increasing the number when a new connection is dispatched to it, and decreasing the count when connection finishes or the timeout happens [3]. Using this algorithm, the system passes a new connection to the server that has the least number of current connections. As this method is dynamic in nature, it distributes connections based on various aspects of real-time server performance analysis, like the current number of connections per node or the fastest node response time. Least Connections method [4] works best in the environments where the servers or other equipment that are being load balanced have similar capabilities. This Application Delivery Controller method is rarely available in a simple load balancer.

III. PROPOSED METHODOLOGY

In this paper, a new methodology has been proposed by combining the existing algorithms to produce a new load balancing algorithm which can deal with dynamic load balancing with low cost, high throughput.

Round Robin with Active Clustering (RRAC) Algorithm: RRAC is a dynamic load balancing algorithm which is a combination of round robin and active clustering algorithms. The Round Robin algorithm [3] allocates the nodes to fulfill the requests in a round-robin manner for a definite time slice, i.e. according to the process allocation order that is maintained locally. This serves the advantage of the fast response in case of equal workload distribution amongst the processes. However, the job processing time for different processes is not the same. So, some nodes may be heavily loaded while some others may remain idle. Active clustering is an enhanced method of random sampling, where this algorithm works on the principle of grouping similar nodes together and start working on these group nodes [6]. This method uses the resources efficiently thereby increases the throughput and performance of the system by using high resources. In this, approach, a technique called matchmaker is introduced. When an execution starts in a network, the process gets initiated and searches for the next matching node said to be match-maker which should satisfy the criteria that it should be the different one from the former one. Once the matchmaker is found the process gets initiated and as soon as the process gets over the matchmaker gets detached from the network. Thus, this is an iterative process in the network to balance the load efficiently. This method may be particularly useful in environments where servers are distributed across different logical networks.

RRAC Algorithm

Assigning ranks:

1. for each process in DAG calculate average execution time on all processors
2. if a process is a final process
3. Rank of process = average of the process
4. Else if
5. Rank of process = average of process + max(rank(predecessors)) + channel weight
6. end if
7. end for

Mapping Logic:

1. for each process in the list of ordered processes
2. if the process is the first process in the list then
3. map process to the processor with minimum execution time
4. else
5. if the process and its predecessor on the same processor p_j
6. comm_time = 0
7. else
8. comm._time=communication time between two nodes
9. end if
10. for each processor
11. process _execution_time = excution_time of process on processor + comm_time + predecessor_excution time
12. end for
13. end for
14. map the process to the processor with minimum total_execution_time

Clustering Method

1. Put each task in the cluster
2. Generate the list of all the triplets
3. Sort the triplets by decreasing count and by decreasing

Amount of communication

4. for each triplet do
5. if the geometric or temporal criterion is satisfied then
6. combine the two clusters
7. end if
8. end for

CONCLUSION

As a part of my research work, I have gone through the different methodologies from literature for the given proposal and my experimental results provide the better utilization of cloud resources (throughput, process time).I am going to implement the proposed system by experimenting with software tools like MAT-Lab, NS-Tool. The future work will be going to have an Experimental setup and results comparisons.

References

[1] Ms.Shilpa D.More, "Reviews of Load Balancing Based on Partitioning in Cloud Computing", International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3965-3967.

[2] B. Naresh Kumar Reddy and N.Venktram, "An Efficient Data Transmission by using Modern USB Flash Drive," International Journal of Electrical and Computer Engineering, ISSN: 2088-8708 Vol. 4, Number 5, pp. 730-740, 2014.

[3] Analysis of Load Balancing Techniques in Cloud Computing. K, Sidhu A and Kinger, Supriya. 2, Fatehgarh Sahib : International Journal of Computers & Technology, April 2013, International Journal of Computers & Technology, Vol. 4. ISSN 2277-3061.

[4] B. Naresh Kumar Reddy and Narasimha, "An Efficient Online Mileage Indicator by Using Sensors for New Generation Automobiles," IEEE Bangalore Section technically co-sponsor on 2nd International Conference on

Advanced Computing, Networking and Security, pp. no. 198-203, Dec, 2013

[5] A Review of the Load Balancing Techniques at Cloud Server. Bala, Kiran, et al. 1, Chandigarh : International Journal of Advances in Computer Science and Communication Engineering, March 2014, International Journal of Advances in Computer Science and Communication Engineering, Vol. 2. ISSN 2347-6788.

[6] B. Naresh Kumar Reddy, M.H.Vasantha, Y.B.Nithin Kumar and Dheeraj Sharma, "A Fine Grained Position for Modular Core on NoC," IEEE International Conference on Computer, Communication and Control (IC4), pp. 1-4, 2015

[7] B. Naresh Kumar Reddy, M.H.Vasantha, Y.B.Nithin Kumar and Dheeraj Sharma, "Communication Energy Constrained Spare Core on NoC," 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-4, 2015.