Static Load Balancing Algorithms In Cloud Computing: Challenges & Solutions

Nadeem Shah, Mohammed Farik

Abstract: Cloud computing provides on-demand hosted computing resources and services over the Internet on a pay-per-use basis. It is currently becoming the favored method of communication and computation over scalable networks due to numerous attractive attributes such as high availability, scalability, fault tolerance, simplicity of management and low cost of ownership. Due to the huge demand of cloud computing, efficient load balancing becomes critical to ensure that computational tasks are evenly distributed across servers to prevent bottlenecks. The aim of this review paper is to understand the current challenges in cloud computing, primarily in cloud load balancing using static algorithms and finding gaps to bridge for more efficient static cloud load balancing in the future. We believe the ideas suggested as new solution will allow researchers to redesign better algorithms for better functionalities and improved user experiences in simple cloud systems. This could assist small businesses that cannot afford infrastructure that supports complex & dynamic load balancing algorithms.

Index Terms: Cloud Computing, Load Balancing, Static Load Balancing Algorithms

1 INTRODUCTION

CLOUD computing is a technology that hosts computing services in centralized datacenters and provides access to them through the Internet. According to Katyal & Mishra [1], the cloud is a pool of heterogeneous resources. Cloud computing is very much a utility, like electricity: sold on demand, instantly scalable to any volume, and charged by use, with the service provider managing every aspect of the service except the device used to access it [2]. Cloud load balancing refers to distributing client requests across multiple application servers that are running in a cloud environment [3]. Fig. 1 illustrates a basic cloud load balancing scenario. In this paper, we have identified the existing static algorithms used for simple cloud load balancing and have suggested a hybrid algorithm for developments in the future.

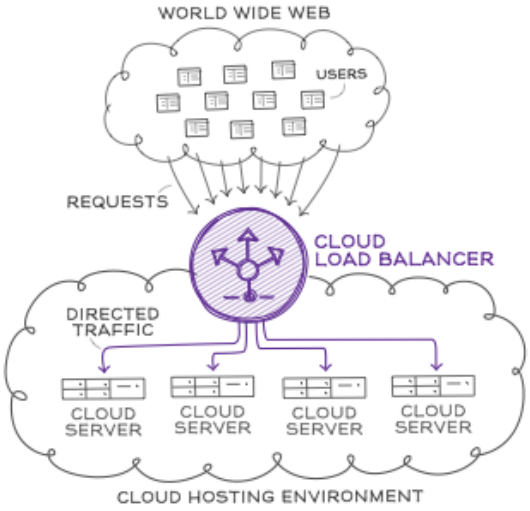

Fig.1 *An example of Cloud Load Balancing [4]*

In section 2 of this paper, we provide an overview of related work in terms issues and goals in load balancing, in section 3, we classify load balancing algorithms, and in section 4, we discuss the strengths and weaknesses of existing static load balancing algorithms. Furthermore, in section 5, we suggest a possible new solution to bridge the gaps in static load balancing algorithms, and conclude in section 6.

2 LOAD BALANCING IN THE CLOUD

In a cloud-based environment, where request for services and platforms can arrive at variable time periods, it is necessary to balance the load on the servers [5]. Load is a measure of the amount of computational work that a system performs. The different types of load are CPU Load (the sum of number of processes that are currently running and the number that are waiting to run), amount of memory used and the network delay load (the time it takes for a bit of data to travel across the network from one node to another). Load balancing is thus one of the key issues in the realm of cloud computing. Load balancing is also a process of distributing processing and communication activities evenly across a computational

- *Nadeem Shah is currently pursuing the post graduate diploma program in information technology in the School of Science and Technology at The University of Fiji.*
- *Mohammed Farik is a Lecturer in Information Technology in the School of Science and Technology at The University of Fiji.*
- *Email: mohammedf@unifiji.ac.fj*

network so that no single device is overwhelmed. Scalability, one of the most important features of cloud computing , is also enabled by load balancing [6]. It is especially important for networks where it is difficult to predict the number of requests that will be issued to a server. Busy web sites typically employ two or more web servers in a load balancing scheme. If one server starts to get swamped, requests are forwarded to another server with more capacity. The goal of load balancing is improving performance by balancing the load among various resources (network links, central processing units, disk drives.) to achieve optimal resource utilization, maximum throughput, maximum response time, and avoiding overload. To distribute load on diverse systems, different load balancing algorithms are used[7]. Various issues exist when dealing with load balancing in a cloud computing environment [8]. Each load balancing algorithm must design to achieve the desired goals i.e. achieving higher throughput, minimum response time and maximum resource utilization. Most existing static algorithms are just a trade-off between all these metrics and are not able to achieve all the required goals.

3 Classification of Load Balancing Algorithms

Load balancers implement type specific algorithms to make load balancing decisions. The decision determines to which remote server to forward a new job [5]. Few of the algorithms for load balancing are studied in this section. Depending on system state, load balancing algorithms can be divided into two types as static and dynamic [9]. A static load balancing algorithm does not take into account the previous state or behavior of a node while distributing the load [10]. On the other hand, a dynamic load balancing algorithm checks the previous state of a node while distributing the load, such as CPU load, amount of memory used, delay or network load, and so on [9].

3.1 Static Algorithm

Static algorithms are appropriate for systems with low variations in load [11]. In static algorithm the traffic is divided evenly among the servers. This algorithm requires a prior knowledge of system resources the performance of the processors is determined at the beginning of the execution, therefore the decision of shifting of the load does not depend on the current state of system [12]. However, static load balancing algorithms have a drawback in that the tasks are assigned to the processor or machines only after it is created and that tasks cannot be shifted during its execution to any other machine for load balancing [13].

3.2 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 [11].

Dynamic algorithms respond to the actual current system state in making load transfer decisions. Since current state of the system is used to make dynamic load balancing decisions, processes are allowed to move from an over utilized machine to an under utilized machine in real time dynamically [13].

4 Strengths & Weaknesses of Static Load Balancing Algorithms

Following are the static load balancing algorithms that are currently common in simple cloud computing environments.

4.1 Round-Robin Load Balancing Algorithm

The round-robin load balancing algorithm uses the round-robin scheme for allocating jobs [5][14]. It selects the first node randomly and then allocates jobs to all other nodes in a round robin fashion. Processors are assigned to each process in a circular order without any sort of priority and hence there is no starvation. This serves the advantage of fast response in the case of equal workload distribution amongst processes [15]. However, different processes have different processing times, therefore at any point of time some nodes may be heavily loaded while others remain idle and under-utilized[6].

4.2 Weighted Round-Robin Load Balancing Algorithm

Weighted round-robin was developed to improve the critical issues with round robin algorithm [5]. In weighted round robin algorithm, each server is assigned a weight and according to the values of the weights, jobs are distributed. Processors with greater capacites are assigned a larger value. Hence the highest weighted servers will receive more tasks. In a situation where all weights become equal, servers will receive balanced traffic.

4.3 Opportunistic Load Balancing Algorithm

Opportunistic load balancing algorithm attempts to keep each node busy [5]. Therefore it does not consider the present workload of each computer. OLB dispatches unexecuted tasks to currently available nodes in a random order regardless of the node's current workload. Since OLB does not calculate the execution time of the node, the task to be processed will be processed in a slower manner resulting in bottlenecks despite some of the nodes being free [5].

4.4 Min-Min Load Balancing Algorithm

Min=min load balancing algorithm begins by finding the minimum completion time for all tasks. Then among these minimum times, the minimum value is selected which is the minimum time amongst all tasks on any resource [14]. According to that minimum time, the task is then scheduled on the corresponding machine. The execution time for all other tasks is updated on that machine and that task is removed from the list. This procedure is followed until all the tasks are assigned the resource. In scenarios where the number of small tasks is more than the number of large tasks, this algorithm achieves better performance [5]. However, this approach can lead to starvation [15].

4.5 Max-Min Load Balancing Algorithm

Max-min load balancing algorithm is similar to the min-min algorithm except the following: after finding out the minimum execution times, the maximum value is selected which is the maximum time amongst all tasks on the resources [14]. Then according to the maximum time, the task is scheduled on the corresponding machine. The execution time for all other tasks is updated on that machine and the assigned task is removed from the list of tasks that are to be assigned to the machines. Since the requirements are known beforehand, this algorithm is expected to perform well.

5 Possible New Solution

Enhanced results can be obtained if a hybrid static algorithm is developed by combining the features of the weighted round

robin algorithm and the max-min load balancing algorithm to bridge the gaps of both. In this case, the weight assignment feature of weighted round robin algorithm could be utilized to calculate the capacity of the server's resources before load assignment is done, combined with features of the max-min algorithm whereby the minimum and maximum execution times are calculated and the maximum time value is used to schedule tasks to the corresponding machines. Since the maximum time for task completion is known before scheduling of tasks is done, this information can be utilized to schedule heavier tasks (tasks with maximum time among all tasks) on heavier weighted machines as calculated with weighted round robin algorithm for faster task completion. Also starvation will be avoided as all tasks will be circulated in a round robin fashion.

6 CONCLUSION

Load balancing is necessary in cloud computing if efficient and maximum utilization of resources needs to be achieved. In this paper, we have discussed the existing static load balancing schemes available for cloud computing. We have also identified the gaps in current static load balancing algorithms and have suggested a possible bridging method for the two algorithms in particular by suggesting the development of a hybrid static algorithm that combines the features of both. This hybrid static algorithm could capitalize on the features of each when combined in one static load balancing algorithm as each fills in the gap of the other.

REFERENCES

[1] M. Katyal and A. Mishra, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment," vol. 1, issue 2, pp. 5-14, 2013. (2)

[2] Citrix Net Scaler - White Paper. Press release "Gartner EXP Worldwide Survey of Nearly 1,600 CIOs Shows IT Budgets in 2010 to be at 2005 Levels," pp. 1-9, 2010. (1)

[3] Inc. [Online]. http://www.citrix.com. [Accessed 02 September 2015].

[4] "Cloud Load Balancing," © NGINX Inc. [Online]. https://www.nginx.com/resources/glossary/cloud-load-balancing/. [Accessed 22 September 2015]. (14)

[5] "Cloud Load Balancing Image,".[Online].

[6] https://www.rackspace.co.uk/

[7] [Accessed 17 September 2015]. (15)

[8] A. Agarwal, Manisha G, R. N. Milind & Shylaja S. S. “A Survey of Cloud Based Load Balancing Techniques,” pp. 9-13, 2014. (3)

[9] N. J. Kansal and I. Chana, "Existing Load Balancing Techniques in Cloud Computing: A Systematic Review," vol. 3, issue. 1, pp. 87-91, 2012. (11)

[10] A. K. Sidhu and S. Kinger "Analysis of Load Balancing Techniques in Cloud Computing," vol. 4, no. 2, pp. 737-741, 2013. (7)

[11] N. S. Raghava and D. Singh, "Comparative Study on Load Balancing Techniques in Cloud Computing," vol. 1, no. 1, pp. 18-25, 2014. (9)

[12] Y. Sahu and R. K. Pateriya, "Cloud Computing Overview with Load Balancing Techniques," vol. 65, no. 24, pp. 40-44, 2013. (8)

[13] Amandeep, V. Yadav and F. Mohammad, "Different Strategies for Load Balancing in Cloud Computing Environment: A Critical Study," vol. 3, issue. 1, pp. 85-90, 2014. (10)

[14] T. Desai and J. Prajapati, "A Survey of Various Load Balancing Techniques and Challenges in Cloud Computing," vol. 2, issue 11, pp. 158-161. (5)

[15] Shiny, "Load Balancing in Cloud Computing: A Review," vol. 15, issue. 2, pp. 22-29, 2013. (13)

[16] R. R. Malladi, "An Approach to Load Balancing in Cloud Computing," vol. 4, issue 5, pp. 3769-3777, 2015. (6)

[17] R. Kaur and P. Luthra, "Load Balancing in Cloud Computing," DOI: 02.ITC.2014.5.92, pp. 374-381. (12)

[18] H. S. Brar, V. Thapar and K. Kishor, "A Survey of Load Balancing Algorithms in Cloud Computing," vol. 2, issue 3, pp. 103-106, 2014. (4)