TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 13, No. 3, March 2015, pp. 568 ~ 573
DOI: 10.11591/telkomnika.v13i3.7237

Performance Analysis of Load Balancing Techniques in Cloud Computing Environment

V Ravi Teja Kanakala*, V.Krishna Reddy
Research Scholar, KL University, Vaddeswaram, India
*Corresponding author, e-mail: raviteja.kanakala@gmail.com

Abstract

Cloud computing is a forerunner among the technologies emerging today in the IT world. It has already been some years after the emergence of cloud technology but still the world didn't get the complete fruits of cloud computing. Out of the many revolutionary results expected as outcomes from the cloud computing technology there were very few achieved and the rest of the expected are still under research. Two of the main obstacles in the usage of cloud computing are Cloud Security and Performance stability. Load Balancing is one of the elements that shows impact on the performance stability of cloud computing. In this paper we discuss about load balancing and different algorithms that are proposed for distributing the load among the nodes and also the parameters that are taken into account for calculating the best algorithm to balance the load.

Keywords: *load balancing, cloud computing, cloud security, performance stability*

1. Introduction

Cloud computing provides the availability of IT resources which are at different parts of the world to users who wants to access those resources from their work place in the form as a service through an optimized and reliable service provider maintaining convenience and ubiquity [1]. The cloud computing technology intends to provide “computing as a utility” [2] in near future. Cloud computing provides its services online on-demand and pay-as-you-go basis. Cloud computing has changed the way the IT companies way of trading and designing their products. Cloud computing has minimized the payment of premium costs for products with its pay as you go basis which increased the traffic in the IT services making Load Balancing a central point of research. To balance the load among multiple resources in a cloud there are several algorithms proposed but till now no algorithm was able to balance the load in a cloud without performance degrading. Many researchers are working on the issue of load balancing and there are many algorithms being proposed day by day as outcome of their research. In this paper we survey some of the optimistic algorithms which had shown some improvement in load balancing and increased the level of performance.

2. Definition

Load balancing is a performance improving method applied in the area of Networking to distribute the work load across multiple resources that are involved in the computation of a networking task. Here Load can be memory, processor capacity, network load etc. Load balancing optimizes the use of resources, reduces response time, avoids overload on any one system by distributing the load to multiple components [3]. Cloud computing can serve complicated task that requires huge computational resources by using distributed resources in a decentralized manner. In a network with high computational requirements we can balance and distribute the load evenly across all the nodes contentedly by using appropriate scheduling algorithms. There are several algorithms proposed earlier in the area of operating systems to schedule resources to the processes. But in cloud computing to balance the load there are some issues like security, service speed, reliability etc. So to develop an optimized load balancing algorithm one must try to create an environment in which total load of a system can be reassigned to multiple components that work collectively in that system so that there will be

Received October 28, 2014; Revised December 25, 2014; Accepted January 10, 2015

no overloaded and under loaded nodes in a system which reduces the overall response time of the system improving the speed, security and reliability. While developing a load balancing algorithm we must consider some important things like estimating the load appropriately, monitoring the performance and stability of the system while performing a task, and selection of proper nodes according to the nature of work.

3. Challenges in Cloud Computing Load Balancing

Finding a solution for problems in load balancing is never an easy process there will be many challenges to be faced while developing a solution. Here in this section we will discuss some of the common challenges that might be faced while developing a solution for a problem of load balancing in cloud computing.

Distribution of Cloud Nodes: There are many algorithms being proposed for load balancing in cloud computing. Among them some algorithms might produce efficient results with small networks or a network with closely located nodes. Such algorithms are not suitable for large networks because those algorithms cannot produce the same efficient results when applied to larger networks. There are many reasons that affect the efficiency in larger networks like speed of the network, distance between the clients and server nodes and also the distance between all the nodes in the network [2]. So while developing a load balancing algorithm one should try for better results in spatially distributed nodes balancing the load effectively reducing network delays.

Migration Time: In cloud computing the service-on-demand method will be followed which means when there is a demand for a resource the service will be provided to the required client. So while serving the client on his demands sometimes we need to migrate resources from long distances due to unavailability in near locations. In such cases the time of migration of the resources from far locations will be more which will affect the performance of the system. While developing an algorithm one should note that resource migration time is an important factor that greatly affects the performance of the system.

Performance of the System: It doesn’t mean that if the complexity of an algorithm is high then the performance of the system will be high. Any time load balancing algorithm must be simple to implement and easy to operate. If the complexity of algorithm is high then the implementation cost will also be more and even after implementing the system performance will be decreased due to more delays in the functionality of the algorithm.

Failure of controller: Definitely centralized load balancing algorithms (Having one controller) can provide efficient results while balancing the load than the distributed algorithms. But in centralized load balancing algorithms when the controller fails the whole system will be halted, in such cases there will be a huge loss for both client and service provider. So, the load balancing algorithms must be designed in a decentralized and distributed fashion so that when a node acting as a controller fails the system will not halt [5]. In such cases the control will be given to other nodes and they will act as controllers of the system.

Energy Management: A load balancing algorithm should be designed in a way such that the operational cost and the energy consumption of the algorithm must be low. Increase in the energy consumption is one of the main problems that cloud computing is facing today. Even though by using energy efficient hardware architectures which slows down the processor speed and turn off machines that are not under use the energy management is becoming difficult. So, to achieve better results in energy management a load balancing algorithm should be designed by following Energy Aware Job Scheduling methodology [6].

Security: Security is one of the problems that cloud computing has in its top most priority. The cloud is always vulnerable in one or the other way to security attacks like DDOS attacks etc. While balancing the load there are many operations that take place like VM migration etc at that time there is a high probability of security attacks. So an efficient load balancing algorithm must be strong enough to reduce the security attacks but should not be vulnerable.

4. Classification of Load Balancing Algorithms

In General load balancing algorithms can be categorized into two different sections called static load balancing algorithms and dynamic load balancing algorithms.

Static Load Balancing: In a static load balancing algorithm while assigning tasks to the nodes it will not check the state and functionality of the node in previous tasks [4]. The process of assigning the tasks will be purely based on the system's prior knowledge on the properties and the capabilities of the node like processing power, storage capacity and memory availability. Even though the above listed properties of a node are considered before assigning a task they cannot adapt to the dynamic changes in the attributes and the allotted load on the node during runtime [5].

Round Robin Algorithm (RR): This is a static load balancing algorithm because before assigning a task to a node it will not take into account the previous state and functionality of that node. To allocate the jobs the first node will be selected randomly and then the remaining nodes are allocated jobs in a round robin manner. This way of scheduling the load will create problems because while allocating the jobs one node may be heavily loaded and one may be lightly loaded irrespective of their capacity. To solve this inequality in load distribution "weighted round robin algorithm was proposed" in this algorithm every node will be assigned weights respective of their capacity then according to that measure the load will be assigned to the nodes [5]. Even though the load is distributed equally it is not possible to predict the execution time of a process. So, this algorithm is not suitable for efficient load balancing.

Central Load Balancing Decision Model (CLBDM): This algorithm is a development for round robin algorithm configured with weight and session-switching in which we can't find the connection time between the process and the node. In central load balancing decision model algorithm the connection time between the client and the node in the cloud is calculated by implementing a software module called Central Load Balancing Decision Module. This module will interact with all parts of the system and collects information regarding the load balancers and servers etc. After collecting the data a sensor will be implemented, this monitors the performance of a node and measures the required time for a task to be completed by the respective node. In this way CLBDM algorithm produces better results than the weighted and session switch configured algorithm.

Ant Colony Optimization Algorithm (ACO): Ant Colony Optimization algorithm simulates the ant foraging behavior. In this algorithm the behavior of ants is used for gathering information from different nodes in the system. When the execution begins the ant and its pheromone will get initiated from the head node and moves to the next node. If the ant finds any of the nodes under loaded it will move forward to another node and if that node is overloaded it will come back to the previous node [8], in this way the information about different nodes is gathered by the system. Due to the moving of ants forward and backward there will be some delay in the traffic many researchers proposed solutions like exiting the ant instead of moving backward in the path.

Map Reduce Algorithm (MR): Map-Reduce is a programming model designed for handling large volumes of data by dividing the huge tasks into small and independent ones. In this algorithm there are two tasks called Map and Reduce, Map function is used to map the tasks and partitions the tasks into independent ones and then compares each task with every other task and then the Reduce function groups the similar tasks and reduces the results of the tasks. Here in this algorithm the problem is the Map function reads multiple tasks simultaneously which will become an overhead on the Reduce function to minimize the results.

Load Balance Min-Min Algorithm (LB Min-Min): In the traditional Min-Min algorithm among all the available resources a resource with minimum execution time will be opted out and then a task with minimum load is assigned to the respective resource hence the name of the algorithm is min-min. But while implementing this algorithm we are facing problems like load imbalance and unconcern of user priority. To solve these problems there are improvements proposed for the Min-Min algorithms called User-Priority guided load balancing algorithm by Huankai in [9]. In the User-Priority guided load balancing algorithm the improvements are task reassignment while selecting smallest sized task according to the resource make span and while dividing the tasks according to the user priority demand.

Load Balance Max-Min Algorithm (LB Min-Max): In the Max-Min load balancing algorithm large tasks will be having highest priority. Here in this algorithm the process begins with calculating the information about the execution time of all the resources and then among all the resources a node with largest execution time will be selected. After selecting a node with highest execution time a task with completion time suitable to the resource is assigned. O. M. Elzeki proposed in [10] that in this algorithm when resources that are capable of faster execution are assigned small tasks and resources with slow execution are assigned large tasks will improve the performance of the algorithm because during the execution of a large task on a slow performing resource there will be many small tasks executed on the fast performing resource.

Dynamic Load Balancing: A dynamic load balancing algorithm checks the previous state and behavior of a node while achieving a task. It will take the different runtime properties of the nodes while processing the tasks into account and will assign the tasks based on the properties collected on the node in the runtime [4]. The advantage of dynamic load balancing algorithm is when a node in the system fails it will not stop the whole system it will only affect the performance of the system. A dynamic load balancing algorithms requires constant monitoring of the functionality of a node and is very difficult in implementation. A dynamic Load balancing algorithm can be implemented in two forms distributed and non-distributed. In distributed type of algorithm all the nodes in the system will interact with each other and the task is distributed among the nodes but in non-distributed type of algorithm all the nodes work independently for achieving the task.

Throttled Load Balancer Algorithm: This is a simple algorithm mainly used for load balancing in virtual machines. The operation of this algorithm begins with load balancer; it collects the indexing values of all the virtual machines in the system and stores in indexing table. When a request is received by the load balancer for resource allocation it parses the indexing table and allocates resource according to the requirements of the client [5]. After allocation the load balancer updates the indexing table. After the completion of task the resource de-allocation starts. This algorithm helps in achieving better performance and high utilization of resources.

Honeybee Foraging Algorithm: This algorithm is inspired from the behavior of honeybees. Honeybees first go out and search for honey sources and then reap the honey from the sources. After that they come back to the honey comb and calculates the food left, if there is a sufficient amount of food they will stay in the comb otherwise they will go out in search of more honey. Inspired from this M.Randles in [11] proposed a decentralized self organization algorithm. In this algorithm all the resources are grouped as virtual resources. Each virtual resource maintains a process queue and accepts requests from the queue and processes those requests. After processing each request every resource will calculate its profit, if it is high the resource stays else it moves to the forage. That is why this algorithm is named as Honeybee foraging algorithm. Maintaining a separate queue for each node and computation of profit after processing requests will become overhead for the system and also there is no improvement in the throughput of the system.

Biased Random Sampling Algorithm: In this algorithm the method of random sampling the system domain is followed for achieving self organization to balance the load among the nodes in the system. This algorithm works with the help of a virtual directed graph which is constructed based on the connectivity of nodes in the system. In the graph each node represents a vertex. When the load balancer receives a request it compares the walk length (traversal between nodes) of the request and the threshold value of the node in the system and if the walk length is equal or greater than the threshold value then the request will be processed at that node otherwise the request will be forwarded to other nodes in the graph [7].

Exponential Smooth Forecast based on Weighted Least Connection: This algorithm ESWLC (shortly) is an improvement for WLC proposed by REN in [13]. In WLC algorithm the load will be distributed on the resources based on the number of connections that resource is having. When the load balancer receives a request from a client it gathers information about the number of connections for all nodes and from that a node with least number of connections will be assigned the task. So, here the drawback for WLC algorithm is it will not check the node's capabilities like Processing power, Disk space availability, and memory. In ESWLC the selection of a node will be done after considering all the above mentioned capabilities [12].

Index name Server Algorithm: This algorithm is developed aiming to reduce the duplication and redundancy of data. It integrates the process of de-duplication and access point selection optimization. To optimize the access point many operations like downloading the hash code block of data and allocation of sufficient bandwidth to download from a resource by establishing connections with the resource are to be done.

Join Idle Queue Algorithm (JIQ): This algorithm is suitable for large scale distributed systems and dynamically scalable web services. This algorithm is an improvement proposed for a basic load balancing algorithm that works with distributed dispatchers. In the basic algorithm the ideal processors has to inform about their idleness to the dispatcher without the knowledge of job requests which removes the load balancing task from the critical path [14]. To solve this problem in JIQ algorithm an improvement was proposed to first load balance the processors on the dispatchers and then load balance the job queue length at each processor.

Table 1. Representation of the performance of some popular Load Balancing Algorithms in Cloud Computing

	CLBDM	ACO	Honeybee Foraging	JIQ	ESWLC	MR	LB Min-Min	LB Min-Max
Throughput	High	Low	Low	Moderate	High	High	Moderate	High
Speed	Moderate	Low	Moderate	Low	High	Moderate	Low	High
Complexity	Moderate	Moderate	High	Low	Low	High	High	Moderate
Fault Tolerance	Low	High	Moderate	Low	High	Moderate	Low	High
Network overhead	High	High	High	Moderate	Moderate	High	High	Moderate
Response time	Moderate	High	High	High	Low	High	Low	Low
Resource Utilization	High	Moderate	High	Low	High	High	Moderate	High
Performance	Moderate	Low	Low	Low	High	Moderate	Low	High
Migration time	High	Low	Low	High	Low	High	Moderate	Moderate

5. Conclusion

In this paper we discussed the significance of load balancing in cloud computing and also we discussed various challenges that occur while balancing load in a cloud computing network. In this paper the classification of load balancing algorithms was discussed in core and the existing algorithms for balancing the load in cloud computing are briefed along with the improvements proposed to improve their performance. Performance of the algorithms which were being implemented for balancing the load in cloud computing was not up to the requirements of cloud. There are different areas involved in achieving the load balancing of a cloud. The problem that we are facing with the existing load balancing algorithms is they are not able to perform well in all the required areas of load balancing. For example, consider Load Balance Max-Min Algorithm, it is good in the throughput but its complexity is high which degrades the performance of the cloud. So, in our future work we will implement changes to algorithms like Exponential Smooth Forecast based on Weighted Least Connection algorithm and Load Balance Max-Min Algorithm and improve their performance to meet the requirements of load balancing in cloud computing and to increase the performance of cloud.

References

[1] NIST The NIST definition of cloud computing. Available: http://csrc.nist.gov/groups/SNS/cloud computing/ cloud-def-v15.doc. 2009.

[2] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi, Jameela Al-Jaroodi. A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms. *IEEE Second Symposium on Network Cloud Computing and Applications*. 2012.

[3] wikipedia.org, http://en.wikipedia.org/wiki/Load_balancing_(computing).

[4] V Krishna Reddy, Srikanth Reddy. A Survey of Various Task Scheduling Algorithms in Cloud Computing. *i-manager's Journal on Computer Science (JCOM).* 2013; 1(1).

[5] Divya Chaudhary, Rajender Singh Chhillar. A New Load Balancing Technique for Virtual Machine Cloud Computing Environment. *International Journal of Computer Applications.* 2013; 69(23): 0975 – 8887.

[6] Vasic N. Making cluster applications energy-aware. Proc of automated ctrl for datacenters and clouds. 2009.

[7] Shanti swaroop moharana, Rajadeepan d Ramesh, Digamber powar. Analysis Of Load Balancers In Cloud Computing. *International Journal of Computer Science and Engineering (IJCSE).* ISSN 2278-9960. 2013; 2(2).

[8] Zhang Z, X Zhang. *A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation.* In proc. 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), IEEE. 2010; 2: 240-243.

[9] Huankai Chen, Proff Frank Wang, Dr Nahelian, Gbola Akanmu. *User-Priority Guided Min-Min Scheduling algorithm for Load Balancing in Cloud Computing.* proc. National conference on Parallel Computing Technologies (PARCOMPTECH), IEEE. 2013; 1-8.

[10] V Krishna Reddy, B Thirumala Rao, Dr LSS Reddy, P Sai Kiran. Research Issues in Cloud Computing. *Global Journal of Computer Science and Technology.* 2011; 11(11).

[11] OM Elzeki , MZ Reshad, MA Elsoud. Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications*. 2012; 50(12): 0975 – 8887.

[12] M Randles, D Lamb, A Taleb-Bendiab. *A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing.* Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia. 2010; 551-556.

[13] TY, WT. Lee, YS Lin, YS Lin, HL Chan, JS Huang. *Dynamic load balancing mechanism based on cloud storage*. proc. Computing, Communications and Applications Conference (ComComAp), IEEE. 2012: 102-106.

[14] Ren X, R Lin, H Zou. *A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast.* proc. International Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE. 2011; 220-224.

[15] Yi Lua, Qiaomin Xiea, Gabriel Kliotb, Alan Gellerb, James R Larusb, Albert Greenbergc. *Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services.* Proc. The 29th International Symposium on Computer Performance, Modeling, Measurements and Evaluation, Elsevier. 2011.