Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 11, Number 12 (2018) pp. 1065-1072

http://www.ripublication.com

Design of a Broker Policy for Resource Management in Cloud Data Centers

V.Umesh

Research Scholar Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore, India

Abstract

Since nearly all applications are being provided over the internet, the need for computing resources is shifting from the user's location to the service provider. The concept of services has gained become popular with the widespread use of the term "cloud computing", which is a new model that has been addresses user requests on a pay-per-use basis. With the biggest advantage being elasticity in terms of increase or decrease of computing resources like computation power, storage and bandwidth, cloud is giving better computing solutions to the users of its services. However the success of these solutions is determined by the use of efficient policies and algorithms that dictate the fundamental concept of cloud computing. These policies include service brokerage, load balancing, virtual machine management and service level agreements. In today's scientifically advanced and IT dominated era, cloud computing, its applications and uses in load balancing has increased to the maximum level. Routing a request to proper destination is a key consideration in distribution networks. Service broker policy is the essential logic, which functions with this intention. Choosing the right data center is the job of service broker policy. The simulation tool CloudAnalyst contains some service broker policies with which a request is directed to the appropriate data center. This paper focuses on the efficient usage of resources in cloud computing by implementing service broker policies for higher and better performance.

Keywords: Cloud computing, data center, service broker policies, load balancing algorithms.

1. INTRODUCTION

Cloud computing is the technology of the future to provide service on demand of the user that is “pay-per-use”. Architecture of the cloud includes as demand timely, repeatable and controllable methodologies for evaluating of algorithms, application and policies for development of cloud services. Cloud application is mostly related with two parameters virtual machine and data transfer to the user. Cloud computing provides real time flexible resources to the user. So the objective of the cloud computing is to efficiently allocate the resources to client in the form of cloud services. Cloud load balancing is to distribute workload and computing resources in a cloud computing environment. It allows managing the application or workload demands by allocating resources. Most important of the load balancing is a device that spreads network across multiple server or computer. Cloud uses large data center and powerful servers for the necessary interface as discussed in [1],[2]. Cloud computing manages the data center and building blocks. Major task of server is to control and allocate resources, data storage and various services by server and service provider.

3. SERVICE BROKER POLICY

A service broker determines which data center should provide the service to the requests coming from each user. Therefore, service broker controls the traffic routing between user and data centers. So it can be called as data center selection policy.

CloudAnalyst consists of three service broker policies:

1. Closest Data Center Policy
2. Optimal Response Time Policy
3. Dynamically Reconfigurable Routing with Load Balancing

3.1 CLOSEST DATA CENTER POLICY

This policy selects the nearest data center in terms of network latency for a user. A list of data centers is kept where the data center with the least network latency is at the top of the list. User requests are completed by selecting the topmost data center from this list. If two or more data centers with the same latency are near, then the choice will be made randomly to balance the load.

3.2 OPTIMAL RESPONSE TIME POLICY

This policy initially finds the closest data center using the previous policy. But when the performance of the closest data center starts to decrease, this policy calculates the current response time for each data center. Then the data center that has the lowest response time is chosen. However, there is a half chance for the selection of the quickest and nearest data center.

3.3 DYNAMICALLY RECONFIGURABLE ROUTING WITH LOAD BALANCING

This policy is an extension of Closest Data Center Policy which utilizes current execution load in order to scale the application deployment. The number of virtual machines (VMs) is increased or decreased accordingly. This will be done by considering the current processing time and best processing time ever attained. This policy is still being researched so it gives ineffectual results.

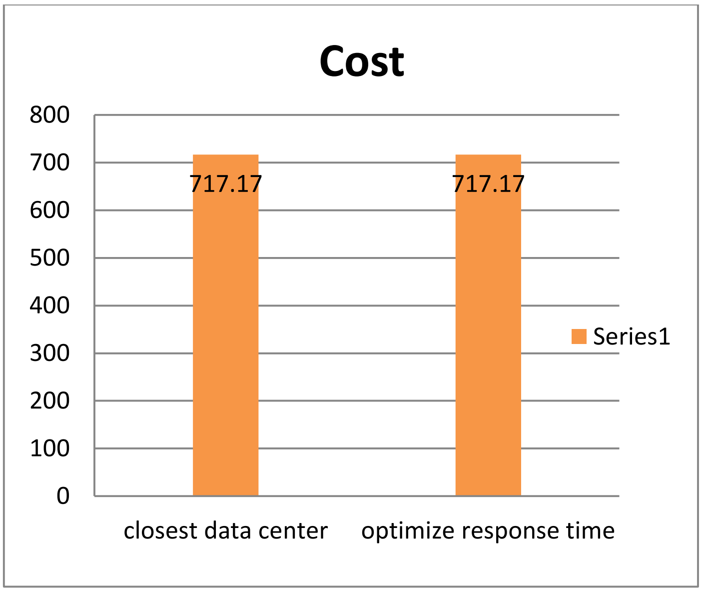

Fig.1. Graph of cost (comparison of policies)

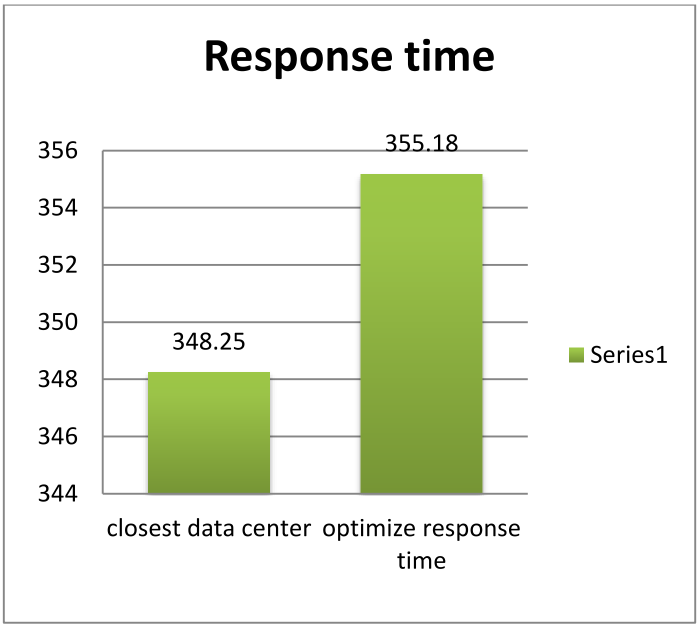

Fig.2. Graph of response time (comparison of policies)

3.4 BENEFITS OF SERVICE BROKER POLICY

1. Service Broker policy using between user and data center is controlled by the traffic routing.
2. It chooses which data center that should support the requests from every client. It means it gives flexible mapping of services to the available resources.
3. An efficient service broker policy ensures that the later tasks to proceed the request that will be done efficiently and in least response time.

4. LOAD BALANCING ALGORITHMS

Cloud computing research has used various load balancing algorithms in research field but only three algorithms which exist in cloud analyst simulators has been focused.

4.1 ROUND ROBIN LOAD BALANCING

This algorithm works in round robin manner. A Data Center Controller gets a request from a client. It notifies the round robin load balancer to allocate a new virtual machine (VM) for processing. RRLB chooses a VM randomly from the group and returns the VM id to Data Center Controller for processing. In this way the subsequent requests are processed in a circular order. The proposed algorithm uses the whole weight that is assigned alternatively. The job is located to all the nodes in a circular manner. Every server has same numerical rating as discussed in figure 3.

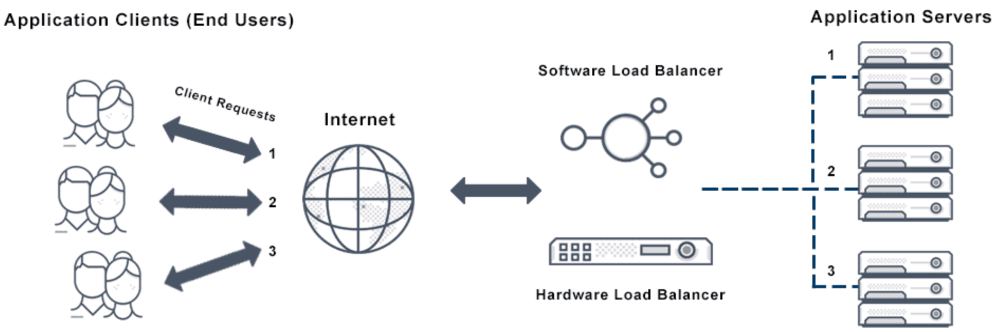

Fig.3. Round Robin Load Balancing (RRLB) Algorithm

The main benefit of round-robin load balancing is that it is extremely simple to implement. However, it does not always result in the most accurate or efficient distribution of traffic, because many round-robin load balancers assume that all servers are the same: currently up, currently handling the same load, and with the same storage and computing capacity. The following variants to the round-robin algorithm take additional factors into account and can result in better load balancing:

1. **Weighted Round Robin** – A weight is assigned to each server based on criteria chosen by the site administrator; the most commonly used criterion is the server's traffic-handling capacity. The higher the weight, the larger the proportion of client requests the server receives. If, for example, server A is assigned a weight of 3 and server B a weight of 1, the load balancer forwards 3 requests to server A for each 1 it sends to server B.
2. **Dynamic Round Robin** – A weight is assigned to each server dynamically, based on real-time data about the server's current load and idle capacity.

4.2 ACTIVE MONITORING LOAD BALANCING

This load balancing attempts to maintain equal workloads on all the available VMs as discussed in figure 4. The steps of this algorithm are:

1. Active Vm Load Balancer maintains an index table of VMs and the number of allocations assigned to each VM. At the start all VMs have 0 allocations.
2. When a request to allocate a new VM from the Data Center Controller arrives, it scans the table and identifies the least loaded in VM.
3. Active Vm Load Balancer then returns the VM id to the Data Center Controller.
4. The Data Center Controller sends the request to the VM. It's identified by that id.
5. Data Center Controller notifies the Active Vm Load Balancer of the new allocation.
6. Active Vm Load Balancer updates the allocation table increasing the allocation count for that VM by 1.
7. When the VM finishes processing the request, and the Data Center Controller receives the response to the cloudlet, it notifies the Active Vm Load Balancer of the VM de-allocation.
8. The Active Vm Load Balancer updates the allocation table by decreasing the allocation count for that VM by 1.

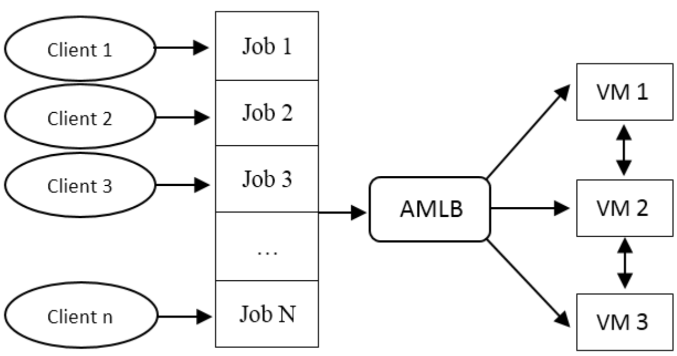

Fig.4. Active Monitoring Load Balancing (AMLB) Algorithm

4.3 THROTTLED LOAD BALANCING

This algorithm implements a throttled load balancer (TLB) to monitor the loads on each VM. Here each VM is assigned to only one task at a time and can be assigned another task only when the current task has completed successfully. Throttled load balancing is described in the flow diagram of figure 5. The steps of this algorithm are:

1. Throttled Vm Load Balancer maintains an index table of VMs and the state of the VMs (BUSY/AVAILABLE). At the start all VMs are available.
2. Data Center Controller receives a request to allocate a VM by the client.
3. Data Center Controller queries the Throttled Vm Load Balancer for the next allocation.
4. Throttled Vm Load Balancer scans the allocation table from top until the first available VM is found.

   If found:

   a. The Throttled Vm Load Balancer returns the VM id to the Data Center Controller.
   b. The Data Center Controller sends the request to the VM identified by that id.
   c. Data Center Controller notifies the Throttled Vm Load Balancer of the new allocation.
   d. Throttled Vm Load Balancer updates the allocation table accordingly.

   If not found:

   e. The Throttled Vm Load Balancer returns null.
   f. The Data Center Controller queues the request until a VM is available.
5. When the VM finishes processing the request, and the Data Center Controller receives the response cloudlet, it notifies the Throttled Vm Load Balancer of the VM de-allocation.
6. The Throttled Vm Load Balancer updates the allocation table by decreasing the allocation count for that VM by 1.
7. The Data Center Controller checks if there are any waiting requests in the queue. If there are, it continues from step 3.

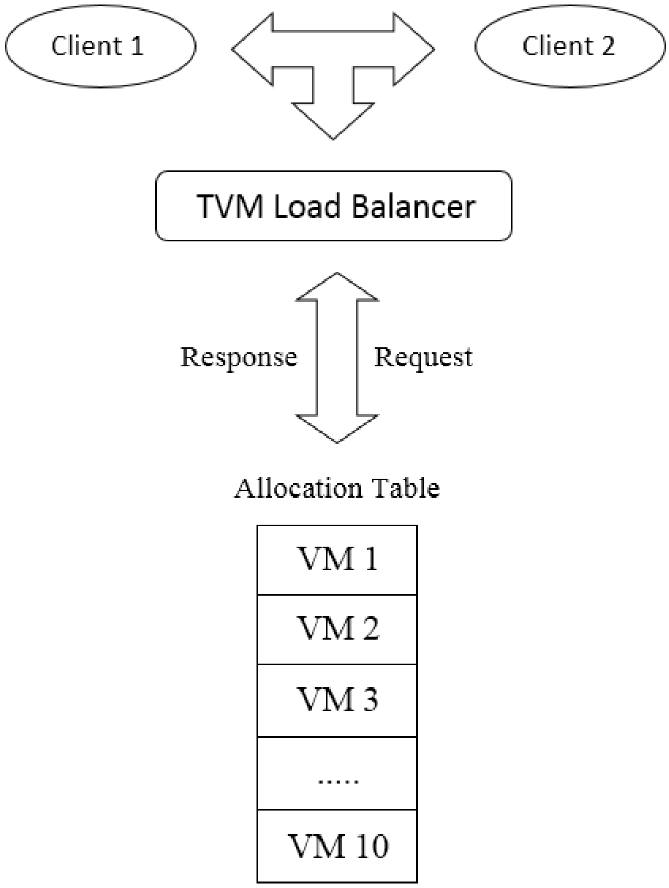

Fig.5. Throttled Load Balancing (TLB) Algorithm

5. RESULTS AND DISCUSSION

After analyzing the graphs plotted in figure 1 and figure 2 for response time and request severing time it is found. But when compared with request severing time that is request processing time form different regions it is found that Closest Data Center, Optimize Response Time work effectively in compare to Reconfigure Dynamically with Load balancing. Closest Data Center and Optimize Response Time work efficiently with low processing time.

6. CONCLUSION

The paper studied the basic of different polices implemented in cloud computing environment and suggested the methods for improving the implementation of broker polices for data center, which further extends in the to produce much more higher outputs for cloud computing environment. The results calculated shows the comparative study of different broker policies which can be further implemented in the real world clod computing environment.

REFERENCES

[1] Z. Lu, S. Takashige, Y. Sugita, T. Morimura, and Y. Kudo, “An analysis and comparison of cloud data center energy-efficient resource management technology,” International Journal of Services Computing (IJSC).

[2] B. Jennings and R. Stadler, “Resource management in clouds: Survey and research challenges,” Journal of Network and Systems Management, vol. 23, no. 3, pp. 567 – 619, 2015.

[3] J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon, and R. Buyya, “Cloudsimsdn: Modeling and simulation of software-defined cloud data centers,” in 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, May 2015, pp. 475–484.

[4] P.-J. Maenhaut, H. Moens, B. Volckaert, V. Ongenae, and F. D. Turck, “Design of a hierarchical software-defined storage system for dataintensive multi-tenant cloud applications,” in 2015 11th International Conference on Network and Service Management (CNSM), November 2015, pp. 22–28.

[5] H. Moens, B. Hanssens, B. Dhoedt, and F. D. Turck, “Hierarchical network-aware placement of service oriented applications in clouds,” in 2014 IEEE Network Operations and Management Symposium (NOMS), May 2014, pp. 1–8.

[6] P.-J. Maenhaut, H. Moens, B. Volckaert, V. Ongenae, and F. D. Turck, “A simulation tool for evaluating the constraint-based allocation of storage resources for multi-tenant cloud applications,” in 2016 IEEE/IFIP Network Operations and Management Symposium (NOMS 2016), Istanbul, Turkey, April 2016, pp. 1017–1018.

[7] W. Forrest, "How to cut data centre carbon emissions?" Website, December 2008. [Online]. Available: http://www.computerweekly.comlArticles/ 2008/12/05/2337 48/how -to-cut-data -centre-carbon -emissions.htm.