Dynamic Resource Allocation Using Load Balancing

Sanchaya S
Department of Computer Science and Engineering,
Jain Global Campus, Jain University,
Jakkasandra Post, Kanakpura Taluk,
Ramanagar District-562112

Madhu B.R
Assistant Professor
Department of Computer Science and Engineering,
Jain Global Campus, Jain University,
Jakkasandra Post, Kanakapura Taluk,
Ramanagara District-562112

Abstract: **In cloud computing environment, resource allocation under bursty workloads can be achieved by enhancing the load balancer. Load balancer is a method to distribute workloads across multiple computers, central processing units, disk drives or many other resources. Resource utilization and maximizing the throughput can be achieved by minimizing the response time and avoid overloading. In the existing load balancing system the algorithm does not consider burstiness as well as current resource utilization in user demands. Hence the existing algorithm which is static in nature can be thought of made dynamic. This improves the system performance and provides faster response time.**

Keywords: Cloud computing, Load Balancing, Burstiness.

1. INTRODUCTION

Cloud computing is an on demand service since it offers dynamic flexible resource allocation for reliable and guaranteed services in pay as you use manner to the public. Multiple cloud users can request number of cloud services simultaneously when necessary. So there must be a provision that all resources are made available based upon the request made by the user in efficient manner to satisfy their needs.

In cloud platforms allocation of resources takes place at two stages. The first stage involves the application to be uploaded to the cloud, the load balancer starts assigning the requested instances to the physical computers in order to balance the load across many physical computers. The second stage involves the application to receive multiple incoming requests, here each requests should be specifically assigned to the application instance in order to balance the computational load across many instances of the same application. Amazon Elastic Compute Cloud uses elastic load balancing to control and handle the incoming requests.

The presence of burstiness in the user workloads usually causes the degradation of the application performance. In order to satisfy the peak user demands, load balancer usually does not consider the case of bursty arrivals and hence results in performance degradation. Burstiness also causes load unbalancing in clouds and as a results degrades the system overall performance.

As a result, finding out the burstiness and providing high quality of service along with system availability is important and challenging as well. When the resources are over-utilized, it results in increased response time. Similarly when the resources are under-utilized, it results in wastage of resources.

Load Balancing is necessary for efficient operation in distributed environment. Cloud computing is a well known platform for providing storage of data in an inexpensive manner that is available over the internet and hence load balancing has necessarily become one of the important and interesting topics in the research fields.

When the number of requests is being generated simultaneously, balancing the load is necessary for achieving better user satisfaction as well as to utilize the resources based on the availability. There many algorithms those are available for providing efficient mechanism and to enhance the cloud performance. Hence provides satisfying and efficient services to the user.

2. RELATED WORK

Burstiness has been known as an important characteristic of traffic in communication networks and has fueled much research over the past two decades. Recently the presence of burstiness has also been identified in a variety of settings, including enterprise systems grid storage systems and file systems. The impact of burstiness has been examined and reported in [5].

Tai Jianzhe et.al [5] implemented a new smart load balancer by adjusting the tradeoffs between randomness and greediness in the site selection process.

Shreyas Mulay., et al [10] implemented the cluster sorting of servers for load balancing. Hence with help of cluster sorting technique requests are handled easily handled by server clusters.

Rashmi K. S., et al [2] A load balancing algorithm has been proposed to avoid deadlocks among the Virtual Machines (VMs) while processing the requests received from the users by VM migration.

Ram Prasad P., et al [3] implemented the idea regarding "Load balancing in cloud computing system" which includes distributed servers along with high fault tolerance, availability scalability and so on.

Mishra, Ratan et.al [4] An ant colony optimization has been proposed to initiate the service load distribution under

cloud computing architecture. The pheromone update mechanism has been proved as a efficient and effective tool to balance the load. This modification supports to minimize the make span of the cloud computing based services and portability of servicing the request also has been converged using the ant colony optimization technique. This technique does not consider the fault tolerance issues.

3. PROPOSED SYSTEM

The proposed load balancing system is a combination of ARA online algorithm and Enhanced Equally distributed algorithm. This proposed load balancing system has an advantage over the previous algorithms in terms of its response time, efficiency and throughput. The design of the proposed work is shown in Figure1.

In this method, the total available servers are initially grouped into a set of 3 servers each known as clusters (Data Center). This is because, when the servers are grouped into clusters, the sorting of clusters will be easier and quicker compared to sequential server sorting. Secondly, clusters acts as backups for each other i.e., if one of the cluster is over loaded, the request will be handled by other clusters until that cluster gets back to normal state. So, this increases the efficiency and response time of the system.

Logic of operation–This system involves two stage sorting i.e. one at the cluster level and other at the server level. Each one of them is associated with a variable called Cluster counter variable (CCV) and Server counter variable (SCV) respectively. These variables will be updated automatically as the cluster and server status changes. Thus, load balancer will sort the cluster and server in descending order of their values.

Cluster counter variable defines the maximum number of request that the cluster can handle. E.g. If CCV is 300, then the 3 servers in that cluster can handle 300 requests simultaneously (loads may or may not be equally distributed within the cluster).Similarly, Server counter variable defines the number of requests that each servers can handle. E.g. If SCV is 100; it means that the server can handle 100 requests simultaneously.

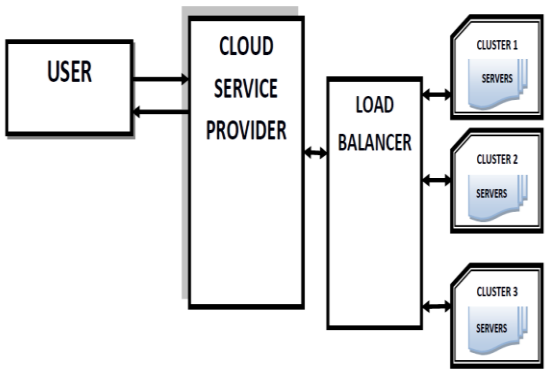

Figure1: Load Balancing in Cloud Computing

Method of operation – Initially, when the user requests arrive from the client, the load balancer (LB) counts the number of incoming requests. Later the LB sends a query to the clusters to know its status. Once the CCV is received, LB arranges the cluster in the descending order of their CCV values. That means, the cluster with maximum request handling capability will be at the Priority 1 level and the cluster with least request handling capability will be at the priority K level, where K refers to the total no. of clusters in the LB system.

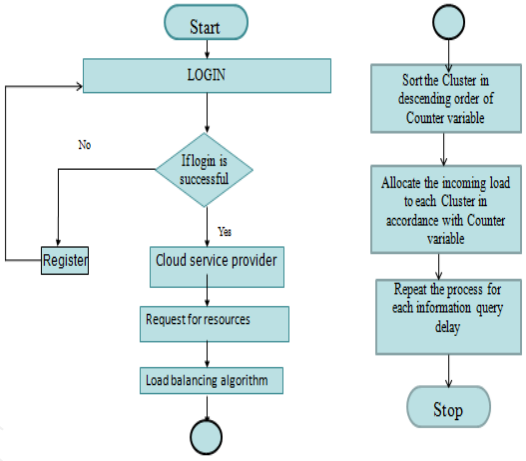

Figure 2: Proposed Flow Chart

The Figure 2 shows the flow of the project. During this time, the LB also receives the server counter variable, from each cluster and sorts the servers in descending order of their SCV value. That means, inside each cluster, the server with maximum request handling capability will be placed on the top and one with least request handling capability will be placed at the bottom. Later the prediction algorithm is executed to know the type of incoming requests. If the request is a bursty type, then the LB will automatically switch to Random mode that is the requests will be randomly allocated to the sorted servers without observing the kind of requests. This improves the response time but the performance output may not be up to the expectation.

If the incoming request is weak bursty or idle type, then LB switches to greedy mode, that is, servers will be allocated which best suits the requests. This improves the performance output and there will be no delay in response as the requests are few in number.

Depending on the no. of incoming requests, the no. of clusters will be varied. E.g. suppose each cluster can handle 500 requests simultaneously and there are 10 clusters in a cloud under one LB, if the request size is 4000, then it could be handled by 8 clusters. So, the remaining 2 clusters will be kept in passive mode, so that any other LBs in the cloud can utilize that clusters for their services. This increases the efficiency of the server utilization and also the throughput in cloud computing. Once again if the request increases, then the clusters return to the parent LB and provide service for them.

4. EXPERIMENTAL ANALYSIS AND RESULTS

The registered client requests for the resources available in the cloud. The requests are redirected to the servers based on the availability. The cluster with more number of servers available for providing services is given the highest priority.

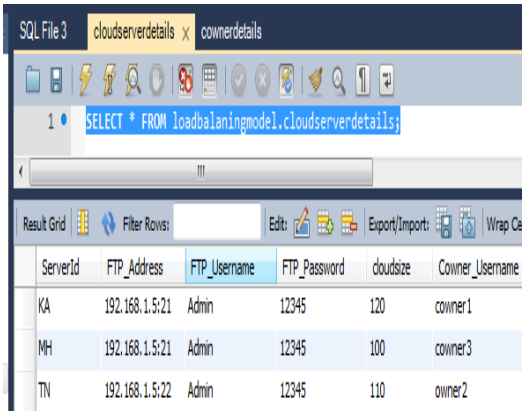

Figure 3: Servers KA, MH, TN are available with cloud size 120 MB, 100MB, 110MB respectively.

In Figure 3, the server details along with the cloud size is shown. Since KA is the server with highest priority based upon the cloud size, when a request is made by the client the server KA is used for allocation.

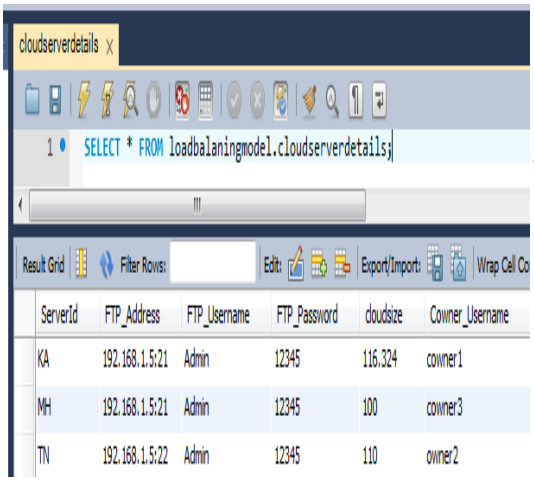

Figure 4: Updated server details after resource allocation

In Figure 4, the size of the cloud is being reduced after the allocation of the resource is made by the server with the highest priority (i.e. KA).

5. CONCLUSION AND FUTURE WORK

In this paper the description is about allocating the resources under bursty workloads. The proposed work focuses on allocation of resources to the servers based on server sorting. The resources are efficiently utilized by assigning the counter variables based upon the availability of the server. Hence efficiency of the system can be achieved. The future work involves the concept of cluster sorting based on the availability of servers.

REFERENCES

[1] Naimesh D. Naik and Ashilkumar R. Patel "Load Balancing Under Bursty Environment For Cloud Computing."International Journal Engineering Research and Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 6, June – 2013.

[2] Rashmi K. S,Suma V.,Vaidehi M., "Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud" in Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies June 2012.

[3] Padhy Ram Prasad, & P. Gautam Prasad Rao. "Load Balancing in Cloud Computing System" at Department of Computer Science and Engineering National Institute of Technology, Rourkela-769 008, Orissa, India May, 2011.

[4] Mishra Ratan and JaiswalAnant. "ANT Colony Optimization :A solution of load balancing in cloud in International Journal of Web & Semantic Technology IJWET Vol.3, No.2, April 2012.

[5] Tai Jianzhe, Zhang Juemin, Li Jun, Meleis Waleedand MiNingfang "ArA: Adaptive resource allocation for cloud computing environments under bursty workloads".In the 30th IEEE International Performance Computing and Communications Conference.

[6] Shreyas Mulay and Sanjay Jain "Enhanced equally distrusted load balancing for cloud computing Volume 2,Issue no 6,June 2013.

[7]http://www.computer.org/csdl/proceedings/pccc/2011/0010/00/06108060-abs.html

[8]http://www.ijert.org/view.php?id=1621&title=architecture-for-distributing-load dynamically-in-cloud-using-server-performance-analysis-under-bursty-workloads

[9]http://www.collaborative.com/uploads/Drive%20OnDemand%20Performance%20Testing%20with%20Cloud%20Computing%20and%20Proactively%20Meet%20Your%20Market%20Needs.pdf

[10] http://warse.org/pdfs/ijmcis01112012.pdf