Architecture For Distributing Load Dynamically In Cloud Using Server Performance Analysis Under Bursty Workloads.

Mehta Rutvik[#1], Patel Yask[#2], Trivedi Harshal[#3]

[#1#2] *Information Technology Department,*
Parul Institute of Engineering & Technology, Vadodra, Gujarat, India.

[#3] *Computer Engineering Department*
Vinus International College of Technology, Gandhinagar, Gujarat, India.

Abstract

The cloud computing systems uses distributed resources to deliver a service to end users using several technologies in combination. Over utilization of these resources is responsible for lengthy response time and under utilization of these resources is responsible for wastage of the available resources. Burstiness in user demands also degrades the performance of the cloud computing system. Major challenge for cloud computing system is to satisfy the peak user demands with the most effective utilization of available resources. Current load balancing algorithm does not consider the current resource utilization and burstiness in user demands. This paper presents a dynamic load balancing algorithm which maintains the state of all virtual machine (VM) resources, and based on CPU, memory and storage space utilization, selects the less utilized VM resource to handle the request. Based on the predicted information of burstiness , this algorithm selects the best VM resource on-the-fly to handle the request. This load balancing algorithm improves the performance by selecting the best sever under both bursty and non-bursty workloads.

***Keyword*: cloud computing, resource management, load balancing, virtual machine, bursty workloads**

I. Introduction

The computing power of any distributed system can be realized by allowing its nodes, to work cooperatively so that large loads are allocated among them in a fair and effective manner. Any strategy for load distribution among node is called load balancing. An effective load balancing policy ensures optimal use of the distributed resources where no resource is under or over utilized.

Cloud computing is an on demand service in which shared resources, information, software packages and other resources are provided according to the clients requirement at specific time. Its a term which is generally used in case of Internet. The whole Internet can be viewed as a cloud. Capital and operational costs can be cut using cloud computing[2]. cloud computing environment provides the users for accessing the shared pool of distributed resources. Cloud is a pay- go model where the consumers pay for the resources utilized instantly, which necessitates having highly available resources to service the requests on demand. Hence, the management of resources becomes a complex job from the business perspective of the cloud service provider [1].

There are many different kinds of load balancing algorithms available for cloud computing system, which can be categorized mainly into two groups:

1. Static algorithms:

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more imperfectly [4].

2. Dynamic algorithms:

In dynamic algorithms decisions on load balancing are based on current state of the system. No prior knowledge is needed for load balancing [3]. So it is better than static approach. Dynamic load balancing can be done in two ways:

- **Distributed dynamic load balancing :**

In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. A benefit, of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt, it instead would affect the system performance to some extent[3].

- **Non-distributed dynamic load balancing :**

In the non-distributed one, the dynamic load balancing algorithm is executed by a single node of the system and the task of load balancing is dependent only on that node. In this approach if the load balancing node fails, it will cause the total load balancing process to halt.

Resource allocation in cloud computing can be done at two different levels. First, when an application is uploaded to the cloud, the load balancer assigns the requested instances to physical computers, attempting to balance the computational load of multiple applications across physical computers[5]. Major problem with the current load balancing algorithm is they does not consider the current utilization of VM resources. These algorithms divide the upcoming request equally without considering the available memory and storage space and current CPU utilization of the VM resource. These applications are dependent on other applications. These applications are executed either in parallel or sequentially. Cloud users try to access the multiple instances of different applications during a short time period. This will cause a significant arrival peak. This will increase the competition between these applications to access the available resources and hence the load unbalancing for the cloud system. Current algorithms do not consider the bursty workloads and hence it will decrease the system performance.

Our proposed algorithm considers the current VM resource utilization and bursty workloads for distributing the load to each VM instances. We expect that using the proposed algorithm cloud service provider can meet the service level agreements (SLA) without purchasing additional resources. Our proposed algorithm also ensures that none of VM resources is over utilized when another one is underutilized. This will increase the system performance and provide faster response time. This will also increase the economic profit of an organization as all the resources are better utilized so there is no need for extra resources for handling the request.

II. Related Work

There are various load balancing algorithms available in the market for distributing the load for a cloud system. Static algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic [4]. Problem with this kind of algorithm is that these algorithms are not able to handle bursty work loads. Even these algorithms do not consider the current situation of each node of the system.

In another algorithm (LB3M) the strategy is to calculate the average completion time of each task for all nodes, respectively and find the task that has the maximum average completion time.

Further find the unassigned node that has the minimum completion time less than the maximum average completion time for the task selected in early stage of the algorithm. Then this task is dispatched to the selected node for computation [6]. This strategy achieves better performance than the static algorithms but it also does not consider the current resource utilization of the cloud system.

In another algorithm (ARA) the strategy is to predict the changes in user demands and shifts between the schemes that are greedy i.e. select the best server and random i.e. select the random server based on the predicted workloads. This scheme will improve the performance by making a smart site selection but the problem is that is does not consider the current utilization of available resources [5].

III. Problems Unrevealed

The current algorithms for distributing the load for cloud computing does not consider the performance of VM instance. These algorithms equally distribute the load to each instances. Problem with these algorithms is that the resources are under or over utilized. These algorithms also do not consider the bursty workloads. If the user demands will change gradually, it will decrease the performance of the cloud system. This is the most complex problem now a days in cloud computing as the number of users that uses cloud services, increases day by day. Load balancer must consider the performance of each instances and bursty work loads for efficient utilization of resources. These parameters are not considered together by the current algorithms and are major concern of cloud service provider.

IV. Proposed Architecture

Cloud controller server:

The Cloud Controller Server (CLS) is the front end to the entire cloud infrastructure. CLS provides web service interface to the client tools on one side and interacts with the rest of the components of the eucalyptus infrastructure on the other side. CLS also provides a web interface to users for managing certain aspects of the cloud infrastructure.

Node controller server:

A node controller server (NCS) is a virtual extension (VT) server. Node controller server runs on each node and controls the life cycle of instances running on the node. The NCS interacts with the OS running on the node on one side and the cloud controller on the other side.

The figure given below shows the proposed model of cloud for load balancing.

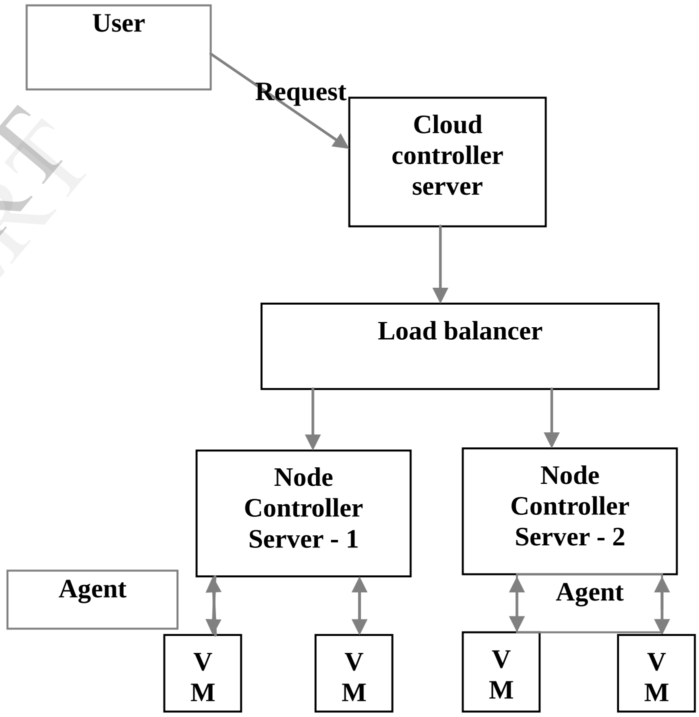

Fig 1: Proposed architecture for distributing the load dynamically in cloud.

Agents:

Agents are the services which keeps the record of CPU usage, memory and storage space usage for each virtual instances. It also keeps the record of current number of connection to a virtual instance.

Virtual machines (VMs):

VMs are one kind of instances of the cloud. Separate instances are created for every user on demand of services. All the services are provided to users through VM instances. All instances are running on NCS.

V. Approach

In the above architecture shown in figure the cloud user would be accessing his services from the cloud controller server. If user makes a request for a cloud service, the request will first go to the cloud controller server. This request will be transferred to the load balancer. A monitoring agent would be continuously monitoring the CPU usage, memory and storage space usage and expected load and current load data for each virtual instances. All the data are transferred to the load balancer by monitoring agent. Based on the data of each virtual instances the request is transferred to the appropriate node controller server where virtual machines are running and from where different instances are provided to different users. Finally the request is transferred to the virtual instance that is selected by the load balancer.

VI. Conclusion

As cloud computing is a new area for research and development, developing a dynamic load balancing algorithm is a major challenge for cloud service provider. This algorithm will ensure the optimum utilization of cloud resources. This algorithm will provide faster response time and it will improve the system performance in the case of changing user demands. This will help the cloud service provider to meet the service level agreements. This algorithm will cut the economic cost for an organization because less resources will be required than static algorithms to handle the user requests.

References

[1] Rashmi, K. S. (2012). Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud, (June).

[2] Padhy, R. P., & Rao, P. G. P. (2011). Load Balancing in Cloud.

[3] Mishra, R., & Jaiswal, A. (2012). Ant colony Optimization : A Solution of Load balancing in Cloud, 3(2), 33–50.

[4] Chaczko, Z., Mahadevan, V., Aslanzadeh, S., & Mcdermid, C. (2011). Availability and Load Balancing in Cloud Computing,

[5] Tai, J., Zhang, J., Li, J., Meleis, W., & Mi, N. (2011). ArA: Adaptive resource allocation for cloud computing environments under bursty workloads. 30th IEEE International Performance Computing and Communications Conference, 1–8

[6] Hung, C., Wang, H., & Hu, Y. (n.d.). Efficient Load Balancing Algorithm for Cloud Computing Network Case study

[7] Bakkali, H. E. L. (2012 IEEE). Load Balancing Cloud Computing : State of Art

[8] http://www.akashsharma.me/private-cloud-setup-using-eucalyptus-and-xen/

[9] http://cloudcomputing.sys-con.com/node/2261725