

# LOAD RE-BALANCING APPROACH FOR DISTRIBUTED LOAD IN CLOUD ENVIRONMENT

Manmohan Sharma<sup>1</sup>, Anil Kumar<sup>2</sup>, Sourabh Singh Verma<sup>3</sup>

Abstract-IT industry is growing each day and so is the need for computing and storage resources. Cloud computing is one of the solution that helps the customers(including organization and individuals) to get resources as per his/her needs and pay only for the time they are using those resources. Due to the huge demand of cloud computing, efficient load balancing becomes critical to ensure that computational tasks are evenly distributed across servers to avoid issues related to latency. In this paper we discuss about load balancing and rebalancing approach to remove some of the bottleneck related to cloud usage.

Keywords: Virtual Machines (VMs), processes, load balancing.

## 1. INTRODUCTION

Cloud computing is an emerging technology in the market. It provides online resources for computation and also for storage. Only the customer has an internet facility to access these resources. In Cloud computing cloud provider outsourced all the resources to their client. There are number of issues related to cloud computing adoption. One of the problems is load balancing in cloud computing [4]. Load balancing helps to distribute all loads coming from different users between all the nodes. It also ensures that every computing resource is distributed efficiently and fairly.

## 2. LOAD BALANCING

Load balancing is dividing the amount of work that a computer has to do between two or more computers so that more work gets done in the same amount of time and, in general, all users get served faster. For ex A web application or websites can be accessed by a plenty of users at any point of time. It is very difficult for the applications to manage all these users at one time and provide response frequently [1][2]. For a website owner it creates a feeling that his potential customer will go down because of not getting the response in a stipulated time. Here we have this website be hosted on cloud computing environment. This type of problem is very common in distributed environment [3][5][6]. To overcome this type of problem we have the concept of load balancing.

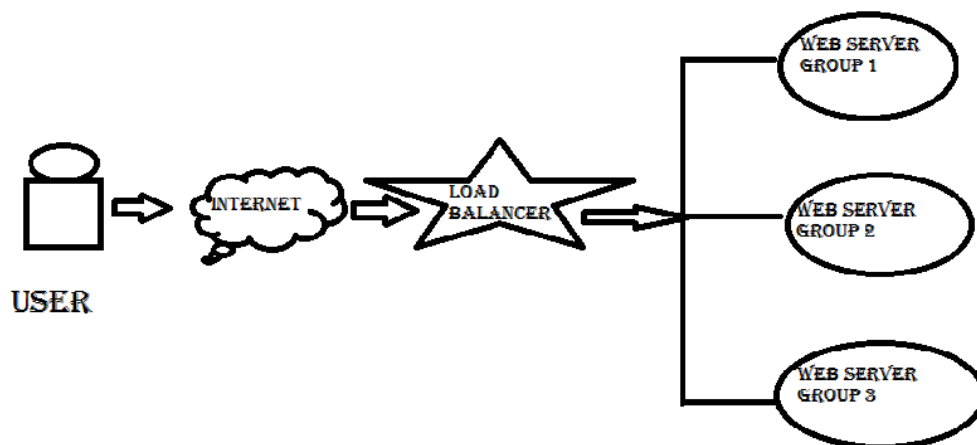


Fig.1: Load balancing in cloud

<sup>1</sup> Mody University Of Science and Technology,Laxmangarh, Rajasthan, India

<sup>2</sup> Mody University Of Science and Technology,Laxmangarh, Rajasthan, India

<sup>3</sup> Mody University Of Science and Technology,Laxmangarh, Rajasthan, India

The main objective of using load balancing is:

1. To maintain system durability.
2. To improve system performance.
3. To protect against system failure.

For balancing the load we have number of load balancing algorithms [4][7]. We categorize these algorithms basically in two types:

Static load balancing: Static load balancing is used for homogeneous resources. In this the load balancer has the initial information about the servers and virtual machines and load is distributed on the basis of initial information.

Dynamic Load balancing: Dynamic load balancing is used for heterogeneous resources. Here in this system the load balancer maintain all the information’s about the servers /VMs state and load on each server and VMs. This information related to VMs is updated periodically. It is more stable as compared to static because the load balancer has the current information about the VM (or resources).

**3. LOAD REBALANCING**

The load rebalancing task is used to eliminate the load on central node or the system. Using load rebalancing algorithm the load of VMs is balanced by transferring the loads on different VMs [3][9].Load Rebalancing is also one of the techniques to increase the performance of the cloud service provider.

Suppose we have 5VMs (VM1,VM2,VM3,VM4,VM5) with same CPU capacity.Initially we have 3 Users(U1,U2,U3) having a number of processes or files as shown in table1.Here in this rebalancing algorithm we are using term process in place of file system.

S.No	User	Processes
1	U1	P1,P2,P3
2	U2	P1,P2,P3,P4,P5
3	U3	P1,P2,P3,P4,P5
4	U4	P1,P2

Table 1

VM1	VM2	VM3	VM4	VM5
U1-P1	U1-P2	U1-P3		
U2-P1	U2-P2	U2-P3	U2-P4	U2-P5
U3-P1	U3-P2	U3-P3	U3-P4	U3-P5
U4-P1	U4-P2			
4(processes)	4(processes)	3(processes)	2(processes)	2(processes)

Table 2

VM1	VM2	VM3	VM4	VM5
U1-P1	U1-P2	U1-P3	U4-P1	U4-P2
U2-P1	U2-P2	U2-P3	U2-P4	U2-P5
U3-P1	U3-P2	U3-P3	U3-P4	U3-P5
3(processes)	3(processes)	3(processes)	3(processes)	3(processes)

Table 3

Each process of table is requiring approximately equal processing capacity. Now without load rebalancing approach we allocate processes randomly on each VM as shown in table 2.

From table 2 we have clearly seen that VM1, VM2 are maximum loaded VM3 is moderately loaded and VM4 and VM5 are partially loaded.

From the table 2 we conclude that VM1 and VM2 will take maximum time to respond because of heavy load where as VM4 and VM 5 will take minimum time to respond because of lightly loaded. So we need to rebalance the above scenario. Table 3 will show the results after rebalancing.

All the above tables will show the hypothetical way to rebalance the load on VMs.Now each VM have an equal load (3 processes each).The response time is now going to be improved for VM 1and VM2.All the above results we can easily demonstrate with the help of cloud analyst tool also.

Load Rebalancing consists of 3 steps.

Process Allocation: Here we say that every Users have a number of processes (like p1,p2,p3,p4,...pn) requiring equal compute power. Each process can be added, deleted and appended to VMs based on their state. It helps to preserver the data and reduces the data loss.

Lookup table: To structure the VMs over the network a lookup table is maintained. Each user process assign with unique identifier, each user process having rapid key lookup in Lookup table. Lookup table guarantees that if VMs joins or add then it allocates the processes which are stored in successor and if any VMs leaves or delete then allocated process are migrated to its successor. Network specifies the location of process that can be integrated with existing distributed user system, due to the transparency of Lookuptable. In this system master system manages mapping of processes to storage VMs and namespace of user system. Figure2 shows the structure of Lookup Table, and Lookup is visible to all VMs.

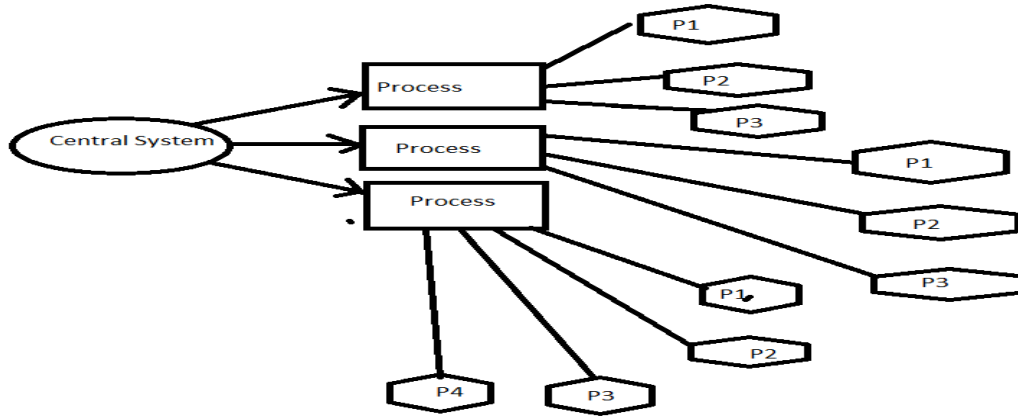


Fig.2:Process Allocation

Load Rebalancing: First we need to find whether a particular VM is under-loaded (lightly loaded) or overloaded (heavily loaded). A VM is under loaded if, number of processes < Minimum threshold value of VM ( $\Delta T_{min}$ ) and a node is overloaded if, Number of processes > Maximum threshold value of VM ( $\Delta T_{max}$ ). All heavily loaded VMs shared its load with lightly loaded VMs. If node “i” is the lightly loaded then it transfer its load to its successor “i+1” and then join instantly as successor to heavy node. The lightly loaded node request for process from heavily loaded VMs and it traverse through physical network link.

**4. SIMULATION**

We had used the cloudsim and cloud analyst for the simulation purpose. Cloud sim is generally used for simulation of load balancing problems. We are using simulation tools and virtual machines environment because installation and preparation of distributed environment is costing a lot for testing purpose.

In cloud analyst we first configure the simulation environment. In that we first design the data centers having number of VMs with their capacity. We also design the user base with their request size (that is to be measured in terms of file size or processes size).

After that we selected load balancing policy as round robin. Once everything is set we run the simulation and get the results shown in fig 3



Fig.3: Simulation result

From fig 3 we concluded that rebalancing processes will divide the workload equally on each VM or Data Centre. The size of the file is of approximately 380KB. We have also seen that it takes around 48782 milliseconds for load balancing and 6441 milliseconds for rebalancing. From fig 3 we also concluded that rebalancing time is far more less than balancing.

## 5. CONCLUSION AND FUTURE SCOPE

Load balancing is one of the important challenges in front of cloud service providers. If there is a proper load balancing algorithm in place response time to customer is going to be reduced. Balancing the load by using rebalancing algorithms is also very effective for distributed systems. In this paper we have shown rebalancing approach using simulated environment. There is also to much scope of improvement in rebalancing in terms of exact load rebalancing with less time of load migration.

## 6. REFERENCES:

- [1] H.-C. Hsiao, H. Liao, S.-S. Chen, and K.-C. Huang, "Load Balance with Imperfect Information in Structured Peer-to-Peer Systems," *IEEE Trans. Parallel Distributed Systems*, vol. 22, no. 4, pp. 634-649, Apr. 2011.
- [2] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load Balancing in Structured P2P Systems," *Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS '02)*, pp. 68-79, Feb. 2003.
- [3] Hung-Chang Hsiao, Hsueh-Yi Chung, Haiying Shen, Yu-Chang Chao, "Load Rebalancing for Distributed File Systems in Clouds", *IEEE transactions on parallel and distributed systems*, vol. 24, no. 5, may 2013.
- [4] SmitaSalunkhe, S. S. Sannakki, "Load Balancing in Clouds," *International Journal of Emerging Trends in Electrical and Electronics*, vol. 11(2), pp. 126-131, June 2015.
- [5] Y. Zhu and Y. Hu, "Efficient, Proximity-Aware Load Balancing for DHT-Based P2P Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 16(4), pp. 349-361, Apr. 2005.
- [6] K. S. Sundaram, "Load Balancing for Distributed File System," *International Journal of Advanced Computational Engineering and Networking*, ISSN Vol. 1(1), pp. 25-30, Mar. 2013
- [7] Suriya Begum and Dr. Prashanth, "Investigational Study of 7 Effective Schemes of Load Balancing in Cloud Computing," *International Journal of Computer Science Issues*, Vol. 10(1), pp. 276-287, Nov. 2013.
- [8] B. Godfrey, K. Lakshminarayanan, R. Karp and I. Stoica S. Surana, "Load Balancing in Dynamic Structured P2P Systems," *Performance Evaluation*, vol. 63(6), pp. 217-240, 2006 Mar.
- [9] S.Penmatsa and T.Chronopoulos, Game-theoretic static load balancing for distributed systems, *Journal of Parallel and Distributed Computing*, vol.71, no.4, pp.537-555, Apr. 2011.