Volume 6, Issue 4, April 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Allocation of Work Load at Balancer Level Using Advanced Round Robin Scheduling Algorithm in a Public Cloud

A Kiran Kumar, B P Deepak Kumar, Kranthi Kiran G
Asst. Prof., Dept. of CSE, CMR Technical Campus,
Hyderabad, India

Abstract— Cloud computing is a technology where one can avail various services like storage, software usage and also some other shared equipments based on requirement of the users on rental basis. Generally cloud computing can work with a normal system and the Internet only. The number of users who uses cloud technology was increasing day by day. The increase in the number of users results in high traffic and improper load balancing. To balance the load equally to all nodes, various static and dynamic algorithms have been proposed and these algorithms consider various parameters like performance, response time, fault tolerance, high availability, cost parameters, number of services, scalability, flexibility, reduced overhead for users, etc. Each algorithm has its advantages and disadvantages. Hence there is a need to do more research work in this area i.e. load balancing. In this paper, cloud architecture for distributing the load to various nodes by the balancers using Enhanced Round Robin scheduling algorithm is proposed. The work load is distributed to the various nodes by the balancers is done using two parameters: one is node status (idle, normal and overload) and time stamp (point of time at which node status changed). Based upon these two parameters all nodes are sorted and stored in a list. Using round robin scheduling algorithm the incoming jobs are allotted to nodes in the list.

Keywords— Throttled Load Balancer (TLB), Least Connection Scheduling Algorithm (LCSA).

I. INTRODUCTION

Load balancing is a process of distributing the overall load to the all available nodes in the cloud system. The main objective of balancing the load is to utilize the resource maximum, to increase the response time and to perform efficiently. Distribution of load in the cloud depends on load balancing algorithm. While developing a load balancing algorithm, one has to consider the number of tasks submitted to cloud, the various resources required to complete the task, and load status of all available nodes etc.

The development of algorithm should consider various metrics and implementation of load balancing algorithm has many issues. Each load balancing algorithm has definite objectives. There are number of algorithms already developed for balancing, there are some problems that Some algorithms, aimed at maximization of throughput , some algorithms aimed at increasing response time and some algorithms aimed at maximum utilization of resources and each and every one has its own limitations and aimed at optimizing one or a few metric's only. But the cloud system needs an algorithm that aims at optimizing all metric and increases the system performance. In this paper, we proposed new cloud architecture, here a balancer distributes the incoming jobs to the various nodes based on the enhanced round robin scheduling algorithm.

II. DISTRIBUTION OF WORK LOAD TO VARIOUS NODES BY THE BALANCERS

Each balancer maintains the information about the status of various nodes under its control. Different statuses of nodes are idle, normal or overloaded. These statuses are determined based on values of three static parameters: CPUs, the CPU processing speed, and the memory size, and three dynamic parameters: the memory utilization ratio, the CPU utilization ratio and network bandwidth. The computations to determine node status are as given below

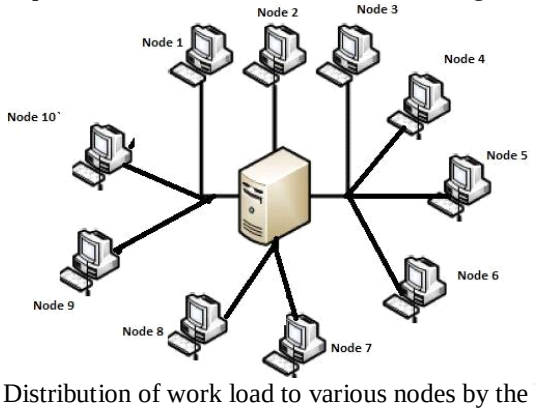

Figure 1: Distribution of work load to various nodes by the balancers

For each of these parameters a weight W_i is assigned so that the sum of the weights is equal to one.

$$\sum_{i=1}^{6} Wi$$

The load on n^{th} node is computed as the weighted sum of the parameter values of n^{th} node.

$$Node\_Load(n)= \sum_{i=1}^{6} W_i * P_i$$

Where P_i indicates i^{th} parameter value of n^{th} node. Average values of Node_load is computed as

$$Average\_Node\_load= \sum_{i=1}^{n} new\_load(i)/n$$

where n is the number of nodes

If Node_load=0, then node status is "IDLE".

If Node_load≤ Average_node_load, then node status is "NORMAL".

If Node_load>Average_node_load, then node status is "OVERLOADED".

Here, each node provides its status like idle, normal, and overloaded along with a time_stamp to the balancer. A time_stamp of a node indicates the point of time at which the node status has changed. The balancer uses this information and maintains list of nodes in the order of status as idle, and normal. Within value of each status, nodes are maintained in the increasing order of their time_stamps. It means, first all nodes whose status are idle are added to the list in the increasing order of time_stamp, then all nodes whose status are normal are added in the increasing order of their time_stamp. This list is refreshed for every fixed period say T. When jobs are sent to the balancer, it assigns the jobs to the nodes in the list in a round robin fashion.

Algorithm:

Step 1: Create a list of nodes whose statuses are idle and normal.

Step 2: Sort this list with the priority of idle status as first and normal status as second.

Step 3: Again sort this list in the increasing order of time stamp of idleness within idle state, and in the increasing order of time stamp of normalness within normal state.

Step 4: When job requests come, the balancer allots to a node in the sorted list using a round robin fashion.

Step 5: The status of the nodes in the list is refreshed at every T time.

When a job arrives for resource allocation, it will be allocated first to the appropriate cloud cluster and then to an appropriate balancer in the cluster, finally to a suitable node. This leads to inconsistent statuses of balancers and nodes during the refresh period. It means the system status i.e balancer status and the node status changes by the arrival of new job, but this information is not updated in the lists maintained by the balancer and the main controller. This results in erroneous node selection. To overcome this problem two lists are maintained at main controller and two lists are maintained at balancers.

One of the lists is flagged as "Read" and another list flagged as "Write". The list flagged as "Read" is used for load balancing. The list flagged as "Write" is used to store the latest information about the status during the refresh period. Once the data in the list is refreshed its flagged is changed from "Write" to read. And this list is used for load balancing. Other list flag is changed from read to write. Thus the flags of the list are alternated to solve this inconsistency.

III. PERFORMANCE ANALYSIS

Enhanced Round Robin Algorithm is simulated on Cloudsim and its performance is analyzed with different algorithms and analysis results are presented below. The other two algorithms used for comparison are Throttled Load Balancer (TLB) and Least Connection Scheduling Algorithm (LCSA). The processing time of different algorithms are given below. Here analysis is done by using three different environments for Data centers DC1, DC2, and DC3. The three different configurations are presented in the below table.

Table 1: Configurations of different data centers

Data Center	Virtual machines	memory	Band width
DC1	15	1 GB	1000
DC2	15	2GB	1000
DC3	15	4GB	1000

The processing time of different algorithms along with the different data centers are given in the below table.

Table 2: Processing time of different algorithms in different data centers

Algorithms	DC1	DC2	DC3
	Avg Processing Time(ms)	Avg Processing Time(ms)	Avg Processing Time(ms)
TLB	0.383	0.472	0.552
LCSA	0.369	0.449	0.537
ERRA	0.355	0.426	0.517

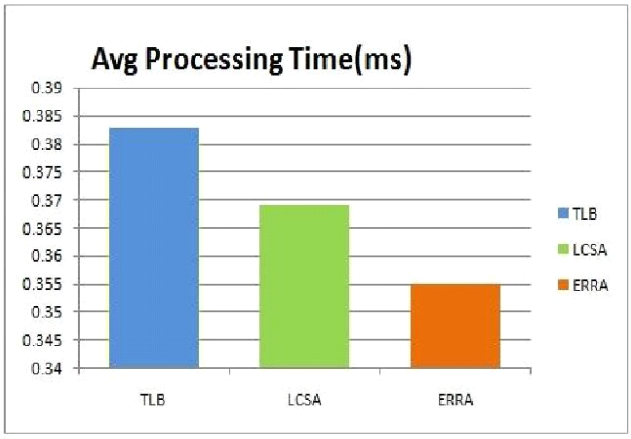

Figure:2 Processing time of different algorithms in a data center

IV. CONCLUSION

In this paper, cloud architecture is proposed at balancer level for load balancing. When a job comes for resource allocation, balancers allocates an appropriate nodes based on the node status and its time stamp of change of its status. The results of this algorithm are compared with Throttled Load Balancer and Least Connection Scheduling Algorithm and results showed that proposed algorithm performed better than other two algorithms.

REFERENCES

[1] Z. Chaczko, V. Mahadevan, Sh. Aslanzadeh and Ch. Mcdermid, " Availability and Load Balancing in Cloud Computing, " 2011 International Conference on Computer and Software Modeling, IACSIT Press, Singapore, vol.14, 134-140.

[2] Jasp reetKaur et al. "Comparison of load balancing algorithms in a Cloud", International Journal of En gineerin g Research and Applications (IJERA), May-Jun 2012.

[3] T. Zuhori, T. Shamrin, R. Tanbin and F. Mahmud, "An Efficient Load Balancing Approach in Cloud Environment by using Round Robin Algorithm," International Journal of Artificial Intelligence and Mechatronics, vol. 1, 2013

[4] Gaochao Xu, Junjie Pang, and Xiaodong Fu, "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud", IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013/

[5] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, N. Nitin, and R. Rastogi, Load balancing of nodes in cloud using ant colony optimization, in Proc. 14th International Conference on Computer Modelling and Simulation (UKSim), Cambridgeshire, United Kingdom, Mar. 2012, pp. 28-30.

[6] K. Nuaimi, N. Mohamed, M. Nuaimi and J. Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenes and Algorithmsg," 2012 IEEE Second Symposium on Network Cloud Computing and Applications, 2012.

[7] T. Kokilavani and G. Amalarethinam, "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing" International Journal of Computer Applications, vol. 20, No. 2, April 2011, pp. 43-49.

[8] Rajesh George rajan,V. jeyakrishna "A survey on Load balancing in cloud computing environment", International journal of advanced research in computer engineering, December 2013 .

[9] Ajaltouni, E. E., Boukerche, A., & Zhang, M. (2008). An Efficient Dynamic Load Balancing Scheme for Distributed Simulations on a Grid Infrastructure. 2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, 61-68. Ieee. doi:10.1109/DS-RT.2008.38

[10] Wang, S-C., K-Q. Yan, W-P. Liao and S-S. Wang, "Towards a load balancing in a three-level cloud computing network," in proc. 3rd International Conference on. Computer Science and Information Technology (ICCSIT), IEEE, Vol. 1, pp: 108-113, July 2010.

[11] Radojevic, B. and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." In proc. 34th International Convention on MIPRO, IEEE, 2011.

[12] K. S. Rashmi, V. Suma and M. Vaidehi, "Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud," Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications (ACCTHPCA), June 2012, pp. 31–35.

AUTHOR'S PROFILE

Adepu Kiran Kumar presently working as a Assistant Professor in CMR Technical campus, Hyderabad, Telangana, INDIA. He received his M.Tech (CSE) degree from Medak Engineering College in the year 2014. He received B.Tech. (CSE) degree from Jayamukhi Institute of Technological Sciences , in the year 2005. His fields of interests are Cloud Computing, Web Technologies, computer Networks, etc.

B. P. Deepak Kumar presently working as a Assistant Professor in CMR Technical campus, Hyderabad, Telangana, INDIA. He received his M.Tech (CSE) degree from Princeton college of Engineering and Technology in the year 2014 .He received B.Tech(CSE) degree from Institute of Aeronautical Engg, in the year 2008. His fields of interests are Cloud Computing, Web Technology, Computer Networks, etc.

Kranthi Kiran G presently working as a Assistant Professor in CMR Technical campus, Hyderabad, Telangana, INDIA. He received his M.Tech (CSE) degree from Sphoorthy Engineering College in the year 2012.He received B.E. (CSE) degree from M.V.S.R Engineering College, in the year 2009. His fields of interests are Cloud Computing, Machine learning, Network Security, etc.

