IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-12 E-ISSN: 2347-2693

Distribution of Work Load at Main Controller Level Using Enhanced Round Robin Scheduling Algorithm in A Public Cloud

G.Thejesvi [1*] and T.Anuradha [2]

[1*,2] Dept. of Computer Science,
Dravidian University, Kuppam, India,

www.ijcseonline.org

Received: 27/Nov/2015 Revised:09/Dec/2015 Accepted:21/Dec/2015 Published: 31/Dec/21015

***Abstract*—** The enormous development in the computer and communication technology led to use various web based software applications using with the Internet. Cloud computing is an emerging technology where millions of clients and individuals use various cloud services like storage, software's and infrastructure on rental basis. Tremendous increase in the number of users has led to some issues and problems. One of the main issues is balancing the work load and increasing the performance of the system. An efficient and dynamic algorithm called "Enhanced Round Robin scheduling algorithm" has been proposed in this paper for balancing the load. The work distribution to the various balancers by the main controller is done using two parameters: one is balancer status (idle, normal and overload) and the other is percentage of overall balancer status (idleness and normalness). Based upon these parameters, the balancers are sorted and stored in a list. The work is distributed to various balancers in a round robin fashion.

***Keywords*—**Main controller,Balancer, Nodes, Enhanced Round Robin Sceduling, Idle, Normal and Overload

I. INTRODUCTION

The unparallel success of the Internet in the last few years had made the computer resources available ubiquitously. This has led to the development of a new computing concept called cloud computing. Cloud computing provides flexible infrastructures, softwares and platforms as services on demand on rental basis. Even though it provides excellent facilities to large business enterprises, small companies and startups etc, by sharing computing resources over the internet, there are many challenges. One of the privileged challenges is load balancing. Load balancing is the process of distributing the dynamic work load across the different nodes of cloud for efficient utilization of resources on high or low load. Already many algorithms are proposed for balancing of load in the cloud environment. These algorithms are either static or dynamic. A static algorithm does not consider the present state or behavior of a node while distributing the load and uses only prior knowledge regarding system resources and tasks. Since the cloud environment is dynamic, these types of algorithms cannot perform proper load balancing. Dynamic algorithms consider the current status of the nodes, for distributing the load. They perform better than static algorithms but they are complex and difficult to implement. Some of the load balancing algorithms are honey bee foraging algorithm, Biased random sampling algorithm, Active clustering algorithm, Two phase scheduling, Min-min, Min-max, Opportunistic load balancing and Round robin etc.

In this chapter, a dynamic round robin algorithm is proposed. In a static Round robin load balancing algorithm, a node is selected randomly and jobs are allocated to all other nodes in a round robin fashion. The tasks are assigned to these nodes in a circular order without considering any priority in assigning a task. Here, since the running time of any process is not known in advance, the work load is not distributed in a uniform manner. Some nodes are heavily loaded and some nodes are less loaded. In order to overcome this drawback, a dynamic load balancing algorithm called "Enhanced Round Robin algorithm" is proposed here. Here, Round Robin algorithm is chosen for enhancement because of its simplicity and easiness in implementation.

The proposed algorithm considers two parameters as maintaining the status of all balancers in the particular region. Generally the status of the balancers will be idle, normal or overloaded and the second parameter is percentage of its load status. The percentage of load status of a balancer is the number of nodes having their majority status can be considered. Based on these two parameters all the balancers are sorted and stored in the list. Finally using Round Robin fashion the main controller assigns the incoming job to the balancers in the list.

II. PROPOSED CLOUD ARCHITECTURE USING ENHANCED ROUND ROBIN SCHEDULING ALGORITHM

A new enhanced round robin algorithm is proposed for public cloud. The public cloud contains many nodes which are placed at different geographical locations. To manage

these nodes of large public cloud in a better way, the concept of cloud portioning is used. Cloud portioning is nothing but a grouping of nodes in a subarea of cloud i.e. groups of nodes are formed on geographical area. The proposed algorithm is based on cloud portioning concept. The overall architecture of the proposed system is shown in the figure 1.

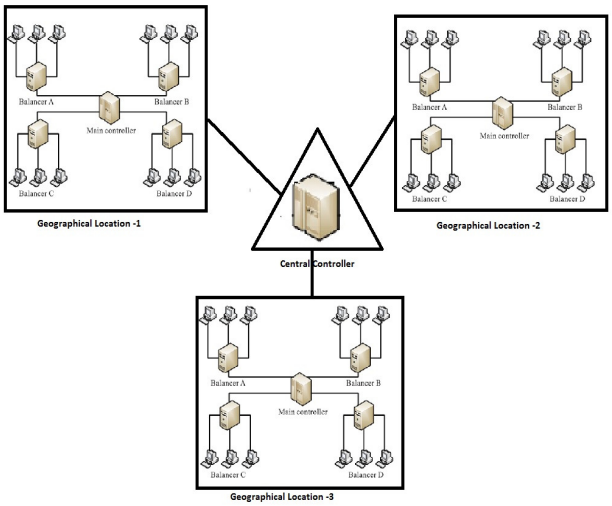

Figure-1: Proposed Cloud Architecture

In this architecture, central controller selects an appropriate cluster for load distribution. Each cluster has a main controller that selects appropriate balancer for job allocation from group of balancers under its control. When a job arrives, the central controller selects an appropriate cluster for resource allocation to this job and this job is sent to the main controller of the selected cluster. The main controller is responsible for selecting an appropriate balancer within the cluster. The balancer selects a node to be allocated to the job from a set of nodes under its control. The decision to select a node to attend a job is done at two levels, first at main controller and then at balancer level.
The proposed Enhanced Round Robin algorithm is executed at main controllers and at balancers. It is assumed that the clusters are formed by the cloud provider, and an appropriate strategy is applied to send a job to the clusters' main controller.

III. Distribution of work load to various balancers by the main controller

Each main controller maintains the information about the status of various balancers under its control. Different statuses of balancers are idle, normal or overloaded. These statuses are determined based on percentage of nodes that are idle, normal or overloaded. For a particular balancer if the percentage of idle nodes is more than the percentage of normal nodes as well as percentage of overloaded , then the status of the balancer is determined as "IDLE". Similarly the "NORMAL" status or "OVERLOAD" statuses of balancers are determined. Here, each balancer provides its status like idle, normal, or overloaded along with the percentages of its status. The main controller uses this information and maintains list of balancers in the order of status: idle followed by normal. Within value of each status, balancers are maintained in the decreasing order of their status percentages. It means, first all balancers whose status are idle are added to the list in the decreasing order of percentages of their idleness, then all balancers whose status are normal are added in the decreasing order of percentages of their normalness. This list is refreshed for every fixed period say T. When jobs are sent to the main controller, it makes use of sorted list of balancers to allocate them to these jobs in a round robin fashion.

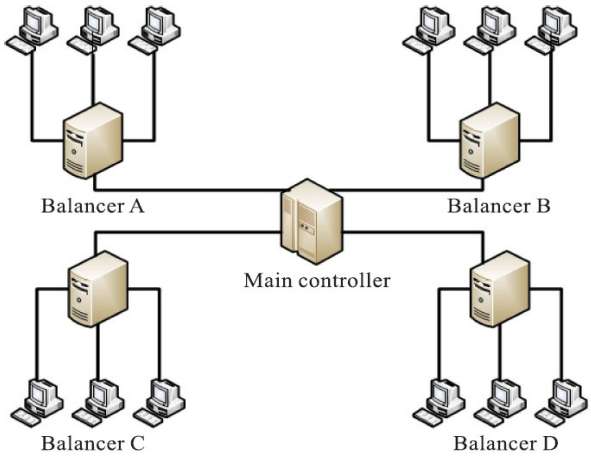

Figure-2: Distribution of work load from main controller to various balancers

Algorithm:

Step 1: Create a list of balancers 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 decreasing order of percentage of idleness within idle state, and in the decreasing order of percentage of normalness within normal state.

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

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

IV. 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 response time of different algorithms are given below.

Algorithms	Average Time(ms)	Minimum Time(ms)	Maximum Time(ms)
TLB	303.22	278.06	364.20
LCSA	303.19	278.14	364.15
ERRA	303.08	278.11	364.13

Table-1: Response time of different dynamic load balancing algorithms.

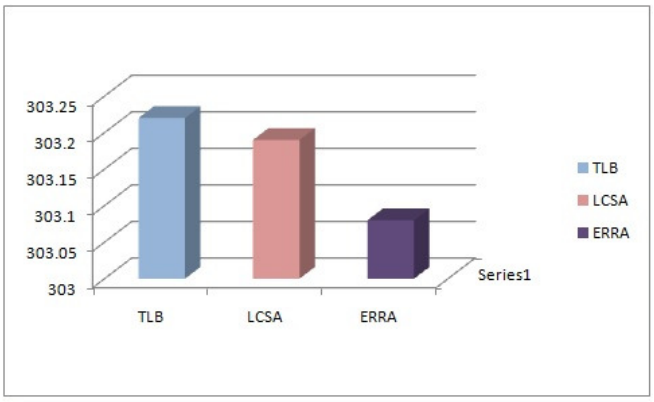

Figure-3: Response time of different dynamic load balancing algorithms

ACKNOWLEDGMENT

I thank My Guide Dr. T. Anuradha for guiding me towards my research. I also thank the Faculty Members of Department of Computer Science for providing the proper guidance.

REFERENCES

[1] B. Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale. com/info center/whitepapers/Load-Balancing-in-the-Cloud.pdf, 2012

[2] 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.

[3] 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.

[4] 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.

[5] 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, pp. 1-4.

[6] 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.

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

[8] 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

[9] 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.

[10] 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.

[11] 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.

[12] 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, pp. 134-140.

[13] S.Penmatsa and A. T. Chronopoulos , Game-theoretic static load balancing for distributed systems, Journal of parallel and Distributing Computing, vol. 71, no.4, pp. 537-555, Apr.2011.

[14] Ruhi Guptha, Rekha Bhatia "An Enhanced and secure approach of load balancing in cloud computing" International Journal of computer Science and Engineering(IJCSE), vol.02, Issue.08, pp.112-116, Aug-2014.

AUTHORS PROFILE

G.Thejesvi Received his MCA from JNTU, Anantapur in 2011 and Done M.Phil from Dravidian Universirty in 2015. His Research Areas are 'Cloud Computing' and 'Big Data'.

Dr. T.Anuradha is an Associate professor working in Dept of Computer Science, Dravidian University,Kuppam. She Received her P.hd from Padmavathi University, Tirupathi. Her Research areas are Data Mining, Cloud Computing and Networking.

