International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-4, Issue-7)

Research Article

July 2015

Various Load Balancing Algorithms in Cloud Environment

Sushil Chandra Dimri
Department of Computer Science
Graphic Era University, Dehradun, India

Abstract—

Now a days cloud computing is an innovative and emerging standard for implementing, organizing, and accessing mass distributed computing applications over the network. In cloud computing, Load balancing is one of the crucial challenges which is required to distribute the workload equally across all the nodes. Load is an amount of work that a computation system performs. It can be classified as network load, storage capacity, memory capacity and CPU load. It helps to achieve a high user satisfaction and resource utilization ratio by confirming an efficient and fair allocation of every computing resource. Proper load balancing support in implementing failover, enabling scalability, over-provisioning, minimizing resource consumption and avoiding bottlenecks, etc. This paper describes different dynamic load balancing algorithms in the cloud environment.

Keywords— Load Balancing, Cloud Computing, Dynamic Algorithm.

I. INTRODUCTION

"A cloud computing is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing platforms on demand, which could be accessed in a simple and pervasive way"[1].
The US National Institute of Standards and Technology (NIST) has published a working definition [2] as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". This definition describes Cloud Computing using [2], [3]:

- **Three service models:** Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).
- **Four deployment models:** Private Clouds, Community Clouds, Public Clouds, and Hybrid Clouds.
- **Five characteristics:** on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.

II. LOAD BALANCING

In cloud environment, Load balancing is a technique that distributes the excess dynamic local workload evenly across all the nodes. Load balancing is used for achieving a better service provisioning, resource utilization and improving the overall performance of the system. For the proper load distribution a load balancer is used which received tasks from different location and then distributed to the data center. A load balancer is a device that acts as a reverse proxy and distributes network or application load across a number of servers [4],[5].
Load balancing is a technique of distributing the total load to the individual nodes of the collective system to the facilitate networks and resources to improve the response time of the job with maximum throughput in the system [6]. The important things which said about load balancing are estimation of load, load comparison, different system stability, system performance, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones to consider while developing such algorithm [7]. In the area of cloud computing, the main objective of load balancing techniques is to improve performance of computing in the cloud, backup plan in case of system failure, maintain stability and scalability for accommodating an increase in large scale computing, reduces associated costs and response time for working

2.1 Components of Dynamic Load Balancing Algorithms

A load balancing algorithm has five major components [9]

- **Transfer Policy:** In this policy it is responsible to determine when a task should be transferred from one node to the other node.
- **Selection Policy:** In this policy it focuses on choosing the processor for load transfer so that the overall response time and throughput may be improved.
- **Location Policy:** In this policy it determines the availability of essential resources for providing services and makes a selection based on location of resources.
- **Information Policy:** In this policy it acquires workload related information about the system such as nature of workload and the average load on each node. It is also responsible for exchanging the information from one node to another, along with method of exchange and the amount of the information to be exchanged. For exchanging load information of a node, three methods can be used which are Broadcast approach, Global System Load, Polling approach.

- **Load Estimation Policy:** In this policy it determines the total workload of a node in a system.

There are various issues while dealing with load balancing in a cloud computing environment. Each load balancing algorithm must be such as to achieve the desired goal. Some algorithms aim to achieving higher throughput, minimum response time, and maximum resource utilization.

III. EXISTING LOAD BALANCING ALGORITHMSIN CLOUD COMPUTING

In cloud computing environment, there are various Load Balancing Algorithms which are closely analyzed and compared on the bases of some predefined metrics, including *throughput, response time, overhead, performance, fault tolerance, migration time, resource utilization, and scalability*. Some of the commonly known *Load Balancing Algorithms* are;

In this algorithm [10]: the load on a server is represented as a virtual graph having connectivity with each node. Each server is symbolized as a node in the graph, with each in degree directed to the free resources of the server. Whenever a node executes a job, it deletes an incoming edge, which indicates a reduction in the availability of free resource. After completion of a job, the node adds on an incoming edge, indicating an increase in the availability of free resource. Random sampling is used for the increment and decrement processes. The last node in the walk is selected for allocation of load; instead any other node based on certain criteria could also be preferred. A node on receiving a job, will execute it only if its current walk length is equal to or greater than the threshold value. Else, the walk length of the job under consideration is incremented and another neighbour node is selected randomly. Again a new directed graph is formed and load balancing is achieved in a fully decentralized manner, thus making it suitable for large network systems like cloud.

According to[11],[12]: Active Clustering is considered as a self aggregation algorithm, works on the principle of grouping the similar nodes and work together on these available groups. A set of processes is iteratively executed by each node on the network. Initially any node can become an initiator and selects another node from its neighbours to be the matchmaker node satisfying the criteria of being a different type than the former one. The matchmaker node then forms a connection between neighbors of it which are similar to the initiator. The matchmaker node then removes the connection between itself and the initiator.

As per the view of [13]: This algorithm is derived from the behaviour of honey bees for finding and reaping food. In order to check for fluctuation in demand of services, servers are grouped under virtual servers (VS), having its own virtual service queues. Each server processing a request from its queue calculates a profit or reward on basis of CPU utilization, which is corresponds to the quality that the bees show in their waggle dance and advertise on the advert board. Each of the servers takes the role of either a forager or a scout. A server serving a request, calculates its profit and compare it with the colony profit, if profit was high, then the server stays at the current virtual server and if it was low, then the server returns to the forager or scout behavior, thus balancing the load with the server.

According to [14] Join Idle Queue load balancing algorithm is applied for dynamically scalable web services. This technique involves a dispatcher to whom processors informs at the time of their idleness, without interfering with job arrivals. Thus removing the load balancing work from the critical path of request processing, system load is reduced; no communication overhead at job arrivals and no increment in actual response time.

In the given algorithm [15] and new [16] had main objective to minimize execution time of each task, also avoid unnecessary replication of task on the node thereby minimizing overall completion time. Opportunistic Load balancing algorithm when combined with LBMM (OLB + LBMM) [15] keeps every node in working state to achieve load balance. Similar to LBMM, LB3M [16] also calculate average completion time for each task for all nodes. Then mark the task with maximum average completion time. After that it dispatches the task of marked node to the unassigned node with minimum completion task, thus balancing the workload evenly among all nodes.

As per the [17],[18] algorithm is mainly proposed for load balancing of nodes. This approach aims efficient distribution of workload among the nodes. The ant will start to move towards the source of the food from the head node when the request is initialized. Ant records their data for future decision making and it keeps records for every node and it makes a visit to the record. Every ant is build with their own individual result set and further built for giving the complete solution. It makes to update continuously with a single result set rather than own result set is updating. This ant works in searching of new sources food with the use of existing food sources to shift the food back to the nest. This mainly aims that efficient distribution of the load among the nodes. It does not encounter the dead end of the movement to the node for building an optimum solution set. In ACO [18] two types of pheromones are used *Foraging Pheromone* (FP) used to explore overloaded node by forward movement of ants while *Trailing Pheromone* (TP) used to discover its path back to the under loaded node.In order to limit the number of ants in the network, they would commit suicide once it finds the target node.

IV. LOAD BALANCING METRICS

After studying the dynamic load balancing algorithms, we have compared all the algorithms on the bases of some predefined metrics. These metrics are as follows:

Throughput: Throughput is used to calculate the number of jobs whose execution has been completed. It should be high to improve the performance of the system.

Overhead: It determines the amount of overhead involved while implementing a load balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.

Fault Tolerance: Fault tolerance system is a system in which the processing does not get affected because of the failure of any particular processing device in the system. The load balancing should be fault tolerant.

Migration time: Migration is the time of movement of job of the master system to the slave system and vice versa in case of results. Migration time is the overhead, which cannot be removed but should be minimized.

Response Time: It is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.

Resource Utilization: It is used to check the utilization of resources. It should be optimized for an efficient load balancing.

Scalability: It is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.

Performance: It is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

V. CONCLUSION

In the cloud computing environment, load balancing is one of the major issues that is highly needed to distribute local workload to all the nodes in the cloud to improve the performance and maximize resource utilization. This paper explains cloud computing, load balancing, types of load balancing algorithms, components of dynamic load balancing algorithms and load balancing metrics. This paper primarily focuses on dynamic load balancing algorithm in cloud environment. For this, various existing dynamic load balancing algorithms are surveyed. By comparing the algorithms on different metrics we tried to find the scope for improving throughput, fault tolerance, performance, resource utilization and minimizing response time, migration time, overhead in the load balancing algorithm. Future work is related to designing a new dynamic load balancing algorithm with fault tolerance for better resource utilization, minimum response time and fast throughput of the cloud computing environment.

ACKNOWLEDGEMENTS

I am very thankful to the management of Graphic Era University for always being very supportive for me and also providing such a commendable research platform for all of us.

REFERENCES

[1] L. Wang, G. Laszewski, “Scientific cloud computing: Early definition and experience”, in Proceedings of 10th IEEE International Conference on High Performance Computing and Communications Dalian, China, 2008, pp. 825–830.

[2] Mell, Peter, and Tim Grance. "Draft NIST working definition of cloud computing."Referenced on June. 3rd 15 2009.

[3] Rimal, Bhaskar Prasad, Eunmi Choi, and Ian Lumb. "A taxonomy and survey of cloud computing systems." INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on. IEEE, 2009.

[4] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, “A break in the clouds: towards a cloud definition,” SIGCOMM ACM Computer Communication Review,vol. 39, pp. 50–55, December 2008.

[5] Rahman, Mazedur, Samira Iqbal, and Jerry Gao. "Load Balancer as a Service in Cloud Computing." In Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on, pp. 204-211. IEEE, 2014.

[6] R. Shimonski. “Windows 2000 & Windows Server 2003 Clustering and Load Balancing”, Emeryville. McGraw-Hill Professional Publishing, CA, USA (2003), p 2, 2003.

[7] Ali M. Alakeel, “A Guide to Dynamic Load Balancing in Distributed Computer Systems”, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[8] M.Armbrust, A.Fox, R. Griffit,et al., “A view of cloud computing”, Communications of the ACM, vol. 53, no.4, pp. 50–58, 2010.

[9] M. Amar, K. Anurag, K. Rakesh, K. Rupesh, Y. Prashant (2011). SLA Driven Load Balancing For Web Applications in Cloud Computing Environment, Information and Knowledge Management, 1(1), pp. 5-13, 2011.

[10] O. Abu- Rahmeh, P. Johnson and A. Taleb-Bendiab, “A Dynamic Biased Random Sampling Scheme for Scalable and Reliable Grid Networks”, INFOCOMP - Journal of Computer Science, ISSN 1807-4545, 2008, VOL.7, N.4, December, 2008, pp. 01-10.

[11] F. Saffre, R. Tateson, J. Halloy, M. Shackleton and J.L. Deneubourg, “Aggregation Dynamics in Overlay Networks and Their Implications for Self-Organized Distributed Applications.” The Computer Journal, March 31st, 2008.

[12] Dhurandher, Sanjay K., Mohammad S. Obaidat, Isaac Woungang, Pragya Agarwal, Abhishek Gupta, and Prateek Gupta. "A cluster-based load balancing algorithm in cloud computing." In Communications (ICC), 2014 IEEE International Conference on, pp. 2921-2925. IEEE, 2014.

[13] Randles, M., D. Lamb and A. Taleb-Bendiab, “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing,” in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, Australia, April 2010.

[14] Yi Lua, QiaominXiea, Gabriel Kliotb, Alan Gellerb, James R. Larusb, Albert Greenbergc, “ Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services” Volume 68 Issue 11, November, 2011, pp:1056-1071, Elsevier Science Publishers, 2011.

[15] S. Wang, K. Yan, W. Liao, and S. Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, September 2010, pages 108-113.

[16] Che-Lun Hung, Hsiao-hsi Wang and Yu-Chen Hu "Efficient Load Balancing Algorithm for Cloud Computing Network", International Conference on Information Science and Technology (IST 2012), April 28-30, pp; 251-253.

[17] Nishant, K. P. Sharma, V. Krishna, C. Gupta, KP. 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), IEEE, pp: 3-8, March 2012.

[18] Dam, Santanu, Gopa Mandal, KousikDasgupta, and Paramartha Dutta. "An Ant Colony Based Load Balancing Strategy in Cloud Computing." In Advanced Computing, Networking and Informatics-Volume 2, pp. 403-413. Springer International Publishing, 2014.

