© 2017 IJSRSET | Volume 3 | Issue 6 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Review paper on Various Dynamic Load Balancing Algorithms in Cloud computing environment

Smaranika Parida
PG student, Department of Information Technology, L.D. College of Engineering, Ahemedabad, Gujurat

ABSTRACT

Cloud computing is an emerging approach enabling ever-present network access, computing resources, deploying, organizing, and accessing enormous distributed computing applications over the network. In cloud computing, Load balancing is one of the main challenges which are required to distribute the workload equally across all the nodes. Load balancing uses services offered by many computer network service provider corporations. Load balancing can be different types like network load, storage capacity, and memory capacity and CPU load. Load balancing 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, and decreases costs associated with document management systems and maximizes the availability of resources. This paper describes a survey of different dynamic load balancing algorithms in the cloud environment with their comparisons on the bases of different load balancing metrics.

Keywords : Cloud Computing, Load balancing, static load balancing, dynamic load balancing algorithm, load balancing metrics

I. INTRODUCTION

In Cloud computing environment the utilization of computing resources may be available through the service by cloud consumers to service providers over the internet. Due to popularity of Cloud computing environment, the cloud computing users are increasing day by day and that has become one of the important challenges for the cloud providers in terms of load balancing.[1]

The US National Institute of Standards and Technology

(NIST) [2] characterizes cloud computing as ". . . a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

This definition describes Cloud Computing using [2], [3]:

Service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Deployment models: Private Clouds, Community Clouds, Public Clouds, and Hybrid Clouds.

Any cloud computing system consists of three major components which are:

Client: The end users, which interact with the clouds to manage information related to the cloud. Clients can be Mobile client, Thin client and Thick client.

Datacenter: Datacenter is the collection of servers hosting different application, exist at a far away from the clients.

Distributed Servers: Distributed servers are the part of a cloud which actively checks services of their hosts and available throughout the internet hosting different applications.

This paper mainly focuses on dynamic load balancing algorithms.

The rest of the paper is organized as follows:

Characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.

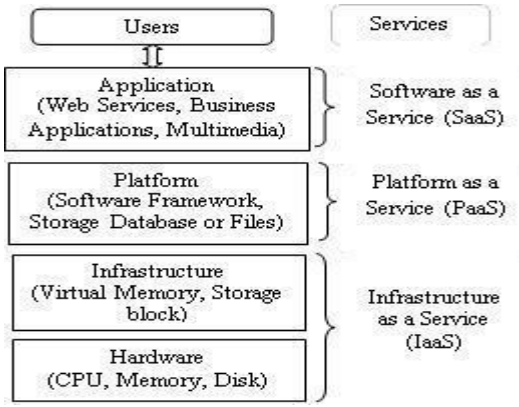

Figure 1 : Cloud Computing Architecture

Section 2 contains an introduction of load balancing. Section 3 describes the existing load balancing algorithms. Comparison of all the algorithms are analyzed in section 4 using different load balancing metrics and final conclusion of the work is given in section 5.

II. LOAD BALANCING

Load balancing is a technique that distributes the workload evenly across all the nodes. Load balancing is used for achieving better 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]. Figure 2 presents a framework under which different load balancing algorithms work in a cloud environment.

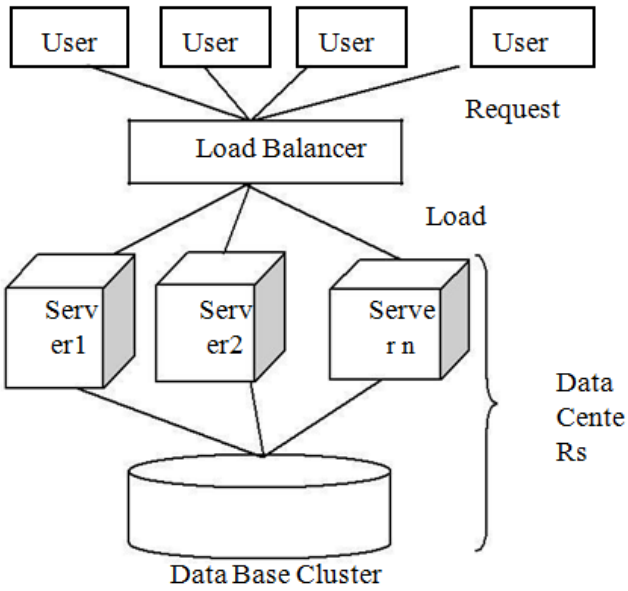

Figure 1 : Framework for working of Dynamic Load Balancing

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 [6]. 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 in the cloud and also maximizes the availability of resources [7].

Next subsection explains types of Load balancing techniques and main components of a dynamic load balancing algorithm.

2.1 Static load balancing algorithm

Static load balancing algorithm uses priori knowledge of the applications and statistical information about the system and distributes the load evenly between servers.

2.2 Dynamic load balancing algorithms

Dynamic load balancing algorithms are those algorithms which search for the lightly loaded server in the network and then trasfers appropriate load on it. In this, work load is distributed among the processors at runtime. The

algorithms in this category are considered complex, but have better fault tolerance and overall performance.

III. LOADBALANCING ALGORITHMS

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;

Biased Random Sampling: Biased Random Sampling algorithm [10] is a distributed and scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. In this algorithm 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 [34]. 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[33].

Active Clustering: Active Clustering algorithm [11][12] works on the principle of grouping the similar nodes and work together on the 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 [34]. The matchmaker node then forms a connection between neighbours of it which are similar to the initiator. The matchmaker node then removes the connection between itself and the initiator [33].

Honey Bee Foraging: Honey Bee Foraging algorithm [13] 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, 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 behaviour, thus balancing the load with the server[33].

Load Balance Min-Min (LBMM): LBMM scheduling algorithm [15] and new optimized *Load Balancing Max-Min-Max (LB3M)* [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[33].

Ant Colony Optimization (ACO): ACO algorithm [17] is mainly proposed for load balancing of nodes and aims efficient distribution of workload among the nodes[34]. 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 [33].

Exponential Smooth Forecast based on Weighted Least Connection (ESWLC): ESWLC algorithm [20] is improved form of Weighted Least-Connection (WLC) along with its features; it also takes into account time series and trials. WLC counts the connections of each server and reports the appropriate server based on the multiplication of a server weight and its count of connections. ESWLC algorithm concludes assigning a certain task to a node only after getting to know about the node capabilities. ESWLC builds the decision based on the experience of the node's CPU power, memory, number of connections and the amount of disk space currently being used. ESWLC then predicts which node is to be selected based on exponential smoothing[33].

Honey Bee Behavior inspired Load Balancing (HBB-LB): According to [21][22] ,HBB-LB is a technique, which helps to achieve even load balancing across virtual machine to maximize throughput. It considers the priority task waiting in queue for execution in virtual machines. After that, the work load on VM calculated decides whether the system is overloaded, under load or balanced and based on these VMs are grouped [34]. According to the load on VM the task is scheduled on VMs, which is removed earlier. To find the correct low loaded VM for the current task, tasks which are removed earlier from over loaded VM are helpful. Forager bee is used as a Scout bee in the next steps [33].

Equally Spread Current Execution (ESCE): According to [24], ESCE is a dynamic load balancing algorithm, which handles the process with priority. It determines the priority by checking the size of the process. This algorithm distributes the load randomly by first checking the size of the process and then transferring the load to a virtual machine, which is lightly loaded. The load balancer spreads the load on different nodes, and hence, it is known as spread spectrum technique.

Throttled Load Balancer (TLB): Throttled load balancer is a dynamic load balancing algorithm [24] in which the client first requests the load balancer to find a suitable virtual machine to perform the required operation. In Cloud computing, there may be multiple instances of virtual machine. These virtual machines can be grouped based on the type of requests they can handle. Whenever a client sends a request, the load balancer will first look for that group, which can handle this request and allocate the process to the lightly loaded instance of that group.

Genetic Algorithm (GA): According to [25], Genetic Algorithm has been used as a soft computing approach, which uses the mechanism of natural selection strategy. A simple Genetic Algorithm is composed of three operations: genetic operation, selection, and replacement operation. The advantage of this technique is that it can handle a vast search space applicable to complex objective function and can avoid being trapped in locally optimal solution. A generation is a collection of artificial creatures (strings). In every new generation, a set of strings is created using information from the previous ones. Occasionally, a new part is effort for good measure. According to [26] Genetic Algorithms are randomized, but they are not simple random walks. They adept exploit historical information to speculate on new search points with expected improvement. The effectiveness of the GA depends in appropriate mix of exploration and exploitation.

Stochastic Hill Climbing: According to [29], Stochastic Hill

Climbing is a soft computing based load balancing approach which is used for allocation of incoming jobs to the servers or virtual machines (VMs). There are two main families of procedures for solving an optimization problem. Complete methods which guarantee either to find a valid assignment of values to variables or prove that no such assignment exists. These methods frequently exhibit good performance, and guarantee a correct and optimal answer for all inputs. Unfortunately, they require exponential time in the worst case, which is not acceptable in the cloud computing domain. The other incomplete methods may not guarantee correct answers for all inputs. Rather, these methods find satisfying assignments for solvable problems with high probability.

***Compare and Balance*:** According to [30][31], Compare and Balance algorithm uses the concept of compare and balance to reach equilibrium condition and manage unbalanced system's load on the basis of probability (number of virtual machines running on the current host and whole cloud system). The current node selects randomly a node and compares the load with itself.

IV. LOAD BALANCING METRICS AND COMPARISON OF THE ENTIRE ALGORITHMS

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 [34]:

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. Overhead 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.

Table 1 shows the comparisons of the dynamic load balancing algorithm, which were discussed in section 3.

Table 1 : Comparisons of existing Load Balancing techniques

Parameters (→) Algorithms (↓)	*Throughput*	*Overhead*	*FaultTolerance*	*MigrationTime*	*ResponseTime*	*ResourceUtilization*	*Scalability*	*Performance*
Biased	N	Y	N	N	N	Y	N	Y
Random Sampling								
Active	N	Y	N	Y	N	Y	N	N
Clustering								
Honey bee Foraging	N	N	N	N	N	Y	N	N
BMM	N	N	N	N	N	Y	N	Y
ACO	Y	N	N	N	N	Y	Y	Y
ACCLB	N	N	Y	N	N	Y	Y	Y
ESWLC	Y	N	Y	N	N	Y	N	Y
HBB-LB	Y	N	N	N	N	N	Y	Y
PALB	Y	Y	Y	Y	Y	Y	N	N
ESCE	Y	N	N	N	Y	Y	N	Y
TLB	N	N	Y	Y	Y	Y	Y	Y
Genetic Algorithm	N	N	N	N	N	Y	N	Y
Stochastic Hill Climbing	Y	N	N	N	Y	Y	N	Y
Compare & Balance	N	Y	N	Y	N	Y	N	N

V. CONCLUSION

Load balancing is one of the main issues in cloud computing, which is essential to allocate dynamic workload between the node among a list of nodes. The main aim is to improve overall performance and to maximize resource utilization .In this paper , cloud computing, load balancing, types of load balancing

algorithms, components and load balancing metrics are explained. This paper mainly focuses on dynamic algorithm of load balancing in cloud environment. So , a list of existing dynamic load balancing algorithms is surveyed and a comparing of these algorithms on different metrics tried to find out. 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.

VI. REFERENCES

[1]. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems, 25:599_616, 2009.

[2]. P. Mell and T. Grance, The NIST Definition of Cloud Computing, National Institute of Standards and technology, Information Technology Laboratory, Technical Report Version 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.

[5]. Lindner, "A break in the clouds: towards a cloud definition," SIGCOMM ACM Computer

[6]. Communication Review,vol. 39, pp. 50–55, December 2008.

[7]. 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.

[8]. Ali M. Alakeel, "A Guide to Dynamic Load Balancing in Distributed Computer Systems", IJCSNS International

[9]. Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[10]. 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.

[11]. 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.

[12]. 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.

[13]. 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.

[14]. 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.

[15]. 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.

[16]. Yi Lua, Qiaomin Xiea, 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.

[17]. 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.

[18]. 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.

[19]. 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.

[20]. Dam, Santanu, Gopa Mandal, Kousik Dasgupta, 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.

[21]. Zhang, Z. and X. Zhang, "A load balancing mechanism based on Ant Colony and Complex Network Theory in Open Cloud Computing federation." In proc. 2nd International Conference on. Industrial Mechatronics and Automation (ICIMA), IEEE, Vol. 2, pp:240-243, May 2010.

[22]. Ren, X., R. Lin and H. Zou, "A dynamic load balancing strategy for cloud computing platform based

on exponential smoothing forecast" in proc. International Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE, pp: 220-224, September 2011.

[23]. Dhinesh B. L.D , P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", in proc. Applied Soft Computing, volume 13, Issue 5, May 2013.

[24]. Ganesh, Amal, M. Sandhya, and Sharmila Shankar. "A study on fault tolerance methods in Cloud Computing." In Advance Computing Conference (IACC), 2014 IEEE International, pp. 844-849. IEEE, 2014.

[25]. Galloway, Jeffrey M., Karl L. Smith, and Susan S. Vrbsky. "Power aware load balancing for cloud computing." Proceedings of the World Congress on Engineering and Computer Science. Vol. 1. 2011.

[26]. Domanal, Shridhar G., and G. Ram Mohana Reddy. "Load Balancing in Cloud Computingusing Modified Throttled Algorithm." Cloud Computing in Emerging Markets (CCEM), 2013 IEEE International Conference on. IEEE, 2013.

[27]. Ye, Zhen, Xiaofang Zhou, and Athman Bouguettaya. "Genetic algorithm based QoS-aware service compositions in cloud computing." Database systems for advanced applications. Springer Berlin Heidelberg, 2011

[28]. Dam, Scintami, et al. "Genetic algorithm and gravitational emulation based hybrid load balancing strategy in cloud computing." Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on. IEEE, 2015.

[29]. Pandey, Suraj, Linlin Wu, Siddeswara Mayura Guru, and Rajkumar Buyya. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, pp. 400-407. IEEE, 2010.

[30]. Gwalior, India. "An Analysis of Swarm Intelligence based Load Balancing Algorithms in a Cloud Computing Environment." (2015).

[31]. Mondal, Brototi, Kousik Dasgupta, and Paramartha Dutta. "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach."Procedia Technology 4 (2012): 783-789.

[32]. Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE

[33]. International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea, August 2009, pages 170-175.

[34]. Kansal, Nidhi Jain, and Inderveer Chana. "Cloud load balancing techniques: A step towards green computing." IJCSI International Journal of Computer Science Issues 9.1 (2012): 238-246.

[35]. A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers", Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.

[36]. Sushil Kumar, Deepak Singh Rana and Sushil Chandra Dimri, "Fault Tolerance and Load Balancing algorithm in Cloud Computing: A survey", International Journal of Advanced Research in Computer and Communication Engineering, July 2015.

[37]. Dharmesh Kashyap, Jaydeep Viradiya, "A Survey of Various Load Balancing Algorithms In Cloud Computing", International Journal of Scientific & Technology Research, Vol. 3, Issue 11, November 2014.