International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

A Survey on Various Load Balancing Techniques in Cloud Computing Environments

Asst. Prof. Patil Yogita[1], Pooja kulkarni[2] , Basavalingayya[3]
[1]Department of Computer Science and Engineering, Visvesvaraya university,
AIET, GULBARGA 585103, India
agyogita@gmail.com
[2] M.Tect (CSE) Student, AIET, GULBARGA 585103 , Visvesvaraya university, India
kulkarni.pooja.26@gmail.com
[3] B.E(CSE) Student, GNDEC, BIDAR 585401, Visvesvaraya university, India
basavalingit@gmail.com

Abstract: Load balancing is employed across different data centers in order to establish network availability and to increase the network capacity. In the case of cloud computing environments there are various challenges like load balancing techniques, security, fault tolerance etc. Load Balancing is the one of the most important parts of the current virtual environment. Many researchers have proposed various techniques to improve the load balancing. This paper describes a survey on load balancing schemes in cloud computing environment along with their corresponding advantages, disadvantages and performance metrics.

Keywords: **Cloud computing, Load Balancing, Response time, Performance.**

1. INTRODUCTION

In the field of information technology, cloud computing is a recent trend that moves computing and data away from desktop and portable computers into large data centers [1]. Cloud computing allows everyone to use software and computing services on-demand at anytime, anywhere and anyplace through the internet. Cloud computing mainly deals with computation, software, data access and storage services that may not require end-user knowledge of the physical location and configuration of the system that is delivering the services [2]. Load balancing is one of the central issues in cloud computing [3]. It is a mechanism that distributes the Dynamic local workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. It helps to achieve a high user satisfaction and resource utilization ratio, hence improving the overall performance and resource utility of the system.

Load balancing is the process of reassigning the total loads to the individual nodes of the collective system to make the best response time and also good utilization of the resources. Cloud computing is an internet computing in which the load balancing is the one of the challenging task. Various methods are to be used to make a better system by allocating the loads to the nodes in a balancing manner but due to network congestion, bandwidth usage etc, there were problems are occurred. These problems were solved by some of the existing techniques. A load balancing algorithm which is dynamic in nature does not consider the previous state or behaviour of the system, that is, it depends on the current behaviour of the system. There were various goals that related to the load balancing such as to improve the performance substantially, to maintain the system stability etc. Depending on the current state of the system, load balancing algorithms can be categorized into two types they are static and dynamic algorithms. In the static algorithm there was prior knowledge of the system is needed and not depend on the current system. In the case of dynamic algorithm it is based on the current system and it is better performance than the static algorithm.
The objective and motivation of this survey is to give a systematic review of various load balancing techniques in cloud computing and encourage the amateur researcher in this field, so that they can contribute in developing more efficient load balancing algo-rithm. This will benefit interested researchers to carry out further work in this research area. The rest of the paper is organized as follows: Section II focuses on the need of load balancing in cloud computing. Section III discusses about the existing load balancing techniques in cloud computing. Section IV identifies the metrics for load balancing. Section V describes the conclusion and future work.

2. NEED OF LOAD BALANCING IN CLOUD COMPUTING

Load balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio [4], making sure that no single node is overwhelmed, hence improving the overall performance of the system. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. It also helps in implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning, reducing response time etc.

Apart from the above-mentioned factors, load balancing is also required to achieve Green computing in

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

clouds which can be done with the help of the following two factors:

• ***Reducing Energy Consumption*** - Load balancing helps in avoiding overheating by balancing the workload across all the nodes of a cloud, hence reducing the amount of energy consumed.

• ***Reducing Carbon Emission*** - Energy consumption and carbon emission go hand in hand. The more the energy consumed, higher is the carbon footprint. As the energy consumption is reduced with the help of Load balancing, so is the carbon emission helping in achieving Green computing.

3. EXISTING LOAD BALANCING TECHNIQUES IN CLOUD COMPUTING

Following load balancing techniques are currently prevalent in clouds

- **VectorDot-** A. Singh et al. [5] proposed a novel load balancing algorithm called VectorDot. It handles the hierarchical com-plexity of the data-center and multidimensionality of resource loads across servers, network switches, and storage in an agile data center that has integrated server and storage virtualization technologies. VectorDot uses dot product to distin-guish nodes based on the item requirements and helps in removing overloads on servers, switches and storage nodes.
- **CARTON-** R. Stanojevic et al. [6] proposed a mechanism CARTON for cloud control that unifies the use of LB and DRL. LB (Load Balancing) is used to equally distribute the jobs to different servers so that the associated costs can be mini-mized and DRL (Distributed Rate Limiting) is used to make sure that the resources are distributed in a way to keep a fair resource allocation. DRL also adapts to server capacities for the dynamic workloads so that performance levels at all serv-ers are equal. With very low computation and communication overhead, this algorithm is simple and easy to implement.
- **Compare and Balance-** Y. Zhao et al. [7] addressed the prob-lem of intra-cloud load balancing amongst physical hosts by adaptive live migration of virtual machines. A load balancing model is designed and implemented to reduce virtual ma-chines' migration time by shared storage, to balance load amongst servers according to their processor or IO usage, etc. and to keep virtual machines' zero-downtime in the process. A distributed load balancing algorithm COMPARE AND BAL-ANCE is also proposed that is based on sampling and reaches equilibrium very fast. This algorithm assures that the migration of VMs is always from high-cost physical hosts to low-cost host but assumes that each physical host has enough memory which is a weak assumption.
- **Event-driven-** V. Nae et al. [8] presented an event-driven load balancing algorithm for real-time Massively Multiplayer Online Games (MMOG). This algorithm after receiving capacity events as input, analyzes its components in context of the resources and the global state of the game session, thereby generating the game session load balancing actions. It is ca-pable of scaling up and down a game session on multiple resources according to the variable user load but has occa-sional QoS breaches.
- **Scheduling strategy on LB of VM resources -** J. Hu et al. [9] proposed a scheduling strategy on load balancing of VM resources that uses historical data and current state of the system. This strategy achieves the best load balancing and reduced dynamic migration by using a genetic algorithm. It helps in resolving the issue of load imbalance and high cost of migration thus achieving better resource utilization.
- **CLBVM-** A. Bhadani et al. [10] proposed a Central Load Bal-ancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud computing environment. This policy improves the overall performance of the system but does not consider the systems that are fault-tolerant.
- **LBVS-** H. Liu et al. [11] proposed a load balancing virtual stor-age strategy (LBVS) that provides a large scale net data stor-age model and Storage as a Service model based on Cloud Storage. Storage virtualization is achieved using an architec-ture that is three-layered and load balancing is achieved using two load balancing modules. It helps in improving the efficien-cy of concurrent access by using replica balancing further reducing the response time and enhancing the capacity of disaster recovery. This strategy also helps in improving the use rate of storage resource, flexibility and robustness of the system.
- **Task Scheduling based on LB-** Y. Fang et al. [12] discussed a two-level task scheduling mechanism based on load balanc-ing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first map-ping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud com-puting environment.
- **Honeybee Foraging Behavior-** M. Randles et al. [13] investi-gated a decentralized honeybee-based load balancing tech-nique that is a nature-inspired algorithm for self-organization. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased sys-tem diversity but throughput is not increased with an increase in system

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

size. It is best suited for the conditions where the diverse population of service types is required.

- **Biased Random Sampling-** M. Randles et al. [13] investigated 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. The performance of the system is improved with high and similar population of resources thus resulting in an in-creased throughput by effectively utilizing the increased sys-tem resources. It is degraded with an increase in population diversity.
- **Active Clustering-** M. Randles et al. [13] investigated a self-aggregation load balancing technique that is a self-aggregation algorithm to optimize job assignments by con-necting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby in-creasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.
- **ACCLB-** Z. Zhang et al. [14] proposed a load balancing mech-anism based on ant colony and complex network theory (ACCLB) in an open cloud computing federation. It uses small-world and scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excel-lent in fault tolerance and has good scalability hence helps in improving the performance of the system

- **(OLB + LBMM)-** S.-C. Wang et al. [15] proposed a two-phase scheduling algorithm that combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms to utilize better executing efficiency and maintain the load balancing of the system. OLB scheduling algorithm, keeps every node in working state to achieve the goal of load balance and LBMM scheduling algorithm is utilized to mini-mize the execution time of each task on the node thereby minimizing the overall completion time. This combined ap-proach hence helps in an efficient utilization of resources and enhances the work efficiency.
- **Decentralized content aware-** H. Mehta et al. [16] proposed a new content aware load balancing policy named as work-load and client aware policy (WCAP). It uses a parameter named as USP to specify the unique and special property of the requests as well as computing nodes. USP helps the scheduler to decide the best suitable node for processing the requests. This strategy is implemented in a decentralized manner with low overhead. By using the content information to narrow down the search, it improves the searching perfor-mance overall performance of the system. It also helps in reducing the idle time of the computing nodes hence improv-ing their utilization.
- **Server-based LB for Internet distributed services-** A. M. Nakai et al. [17] proposed a new server-based load balancing policy for web servers which are distributed all over the world. It helps in reducing the service response times by using a protocol that limits the redirection of requests to the closest remote servers without overloading them. A middleware is described to implement this protocol. It also uses a heuristic to help web servers to endure overloads.
- **Join-Idle-Queue-** Y. Lua et al. [18] proposed a Join-Idle-Queue load balancing algorithm for dynamically scalable web services. This algorithm provides large-scale load balancing with distributed dispatchers by, first load balancing idle pro-cessors across dispatchers for the availability of idle proces-sors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. By remov-ing the load balancing work from the critical path of request processing, it effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time.
- **Lock-free multiprocessing solution for LB-** X. Liu et al. [19] proposed a lock-free multiprocessing load balancing solution that avoids the use of shared memory in contrast to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. It is achieved by modifying Linux kernel. This solution helps in improving the overall performance of load balancer in a multi-core environ-ment by running multiple load-balancing processes in one load balancer.

- Table I- shows the analysis of existing load balancing techniques in cloud computing. This review identifies the techniques, their environ-ment and description according to what has been stated in each of the selected papers.

Table 1-Analysis Of Existing Load Balancing Techniques

Techniques	Findings
T1[5]	1.Handles hierarchial and multi dimenshional constraint
T2 [6]	1. Simple 2. Easy to implement 3. Very low computation and communication overheaD
T3[7]	1.Balances load amongst servers 2.Reaches equilibrium fast
T4 [8]	1. Capable of scaling up and down a game session on multiple resources according to the variable user load 2. Occasional QoS breaches
T5[9]	1. Solves the problems of load imbalance and high migration cost
T6[10]	1. Balances the load evenly to improve overall performance 2. Does not consider fault tolerance

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

T7[11]	1. Enhances flexibility and robustness 2. Provides large scale net data storage and storage as a service
T8[12]	1. Improves task response time 2. Improves resource utilization
T9[13]	1.Performs well as system diversity increases
T10[13]	1.Performs better with high and similar population of resources
T11 [13]	1. Performs better with high resources 2. Utilizes the increased system resources to increase throughput
T12[14]	1. Adaptive to dynamic environments 2. Excellent in fault tolerance 3. Good scalability
T13[15]	1. Efficient utilization of resources
T14[16]	1.Improves the searching performance hence increasing overall perfor-mance
T15[17]	1. Reduces service response times by redirecting requests to the closest server without overloading them
T16[18]	1. Effectively reduces the system load 2. Incurs no communication overhead at job arrivals
T17[19]	1.Improves overall performance of load balancer

4. METRICS FOR LOAD BALANCING

Various Metrics considered in existing load balancing techniques in cloud computing are discussed below:

- **Throughput** is used to calculate the number of tasks whose execution has been completed. It should be high t improve the performance Of the system.
- **Overhead Associated** determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, interprocessor and inter-process communication.It should be minimized.
- **Fault tolerance** is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure.
- **Migration time** is the time to migrate the jobs or resources from one node to another. It should be minimized.
- **Response Time** is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This should be minimized.
- **Resource utilization** is used to check the utilization of resources. It should be optimized for an efficient load balancing.
- **Scalability** is the ability of an algorithm to perform load bal-ancing for a system with any finite number of nodes. This metric should be improved.
- **Performance** 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.

5. CONCLUSION AND FUTURE WORK

The load balancing of the current system is one of the greatest issues. Various techniques and algorithms are used to solve the problem. In this paper we survey various existing load balancing methods in different environments. A large number of parameters and different types of soft computing techniques can be included in the future for the better utilization and needs of the user. The various load balancing techniques are also being compared here.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the Clouds: A Berkeley View of Cloud Computing”, EECS Department, University of California, Berkeley, Technical Report No., UCB/EECS-2009-28, pages 1-23, February 2009.
[2] R. W. Lucky, “Cloud computing”, IEEE Journal ofSpectrum, Vol. 46, No. 5, May 2009, pages 27-45.
[3] B. P. Rima, E. Choi, and I. Lumb, “A Taxonomy and Survey of Cloud Computing Systems”, Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pages 44-51.
[4] Z. Zhang, and X. Zhang, “A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation”, Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pages 240- 243.
[5] Singh A., Korupolu M. and Mohapatra D. (2008) ACM/IEEE conference on Supercomputing.
[6] Stanojevic R. and Shorten R. (2009) *IEEE ICC*, 1-6.
[7] Zhao Y. and Huang W. (2009) 5*th International Joint Confer-ence on INC, IMS and IDC*, 170-175.
[8] Nae V., Prodan R. and Fahringer T. (2010) 11*th IEEE/ACM* International Conference on Grid Computing (Grid), 9-17.
[9] Hu J., Gu J., Sun G. and Zhao T. (2010) 3*rd* International Symposium on Parallel Architectures, Algorithms and Pro-gramming, 89-96.

[10] Bhadani A. and Chaudhary S. (2010) *3rd AnnualACM Banga-lore Conference.*
[11] Liu H., Liu S., Meng X., Yang C. and Zhang Y. (2010) *Interna-tional Conference on Service Sciences (ICSS)*, 257-262.
[12] Fang Y., Wang F. and Ge J. (2010) *Lecture Notes in Comput-er Science*, 6318, 271-277.
[13] Randles M., Lamb D. and Taleb-Bendiab A. (2010) 24*th Inter-national Conference on Advanced Information Networking and Applications Workshops*, 551-556.
[14] Zhang Z. and Zhang X. (2010) 2*nd International Conference on Industrial Mechatronics and Automation*, 240-243.
[15] Wang S., Yan K., Liao W. and Wang S. (2010) 3*rd Internation-al Conference on Computer Science and Information Technol-ogy*, 108-113.
[16] Mehta H., Kanungo P. and Chandwani M. (2011) *International Conference Workshop on Emerging Trends in Technology*, 370-375.
[17] Nakai A.M., Madeira E. and Buzato L.E. (2011) 5*th Latin-American Symposium on Dependable Computing*, 156-165.
[18] Lua Y., Xiea Q., Kliotb G., Gellerb A., Larusb J. R. and Green-ber A. (2011) *Int. Journal on Performance evaluation.*
[19] Liu Xi., Pan Lei., Wang Chong-Jun. and Xie Jun-Yuan. (2011) 3*rd International Workshop on Intelligent Systems and Appli-cations*, 1-4.