Load Balancing Technique in Cloud Computing : A Review

Narendra Chamoli, Himanshu Suyal, Amit Panwar
M.Tech
Department of Computer Science & Engg.
GBPEC, Pauri Garhwal

Ravinder Chauhan
Assistant Executive, Data Processing
Indira Gandhi National Open University, Regional Centre, Dehradun

ABSTRACT
Cloud Computing (CC) term came into existence using existing technologies like Parallel computing, Grid computing, Distributing computing, Peer to Peer technology and Virtualization, etc. Load Balancing is a technique of sharing cloudlet of overloaded nodes to slug nodes. In CC, due to its elastic characteristic, load balancing (LB) is a critical issue as data processing occurs centrally in network using Virtual Machines (VM). LB helps in minimizing over consumption of resources, fault tolerance, scalability, increase throughput, response time, etc. The paper summarizes various recent techniques introduced for load balancing in Cloud. The metrics of analyses are objectives, achievements, challenges of discussed load balancing techniques and their comparison.

Keywords
Cloud Computing (CC), Virtual Machine (VM), Virtualization, Load Balancing (LB), Elasticity

1. INTRODUCTION
Cloud Computing (CC) technology came up with the idea of resource optimization of global network. It is known that central information processing is fast and more efficient which is done by large farms of computing and storage systems accessible via the Internet. When computing is done by distant data centers rather than local it is called as network-centric computing and networkcentric content. With this idea and with advancement in Internet technologies two new computing models are widely

accepted, Grid computing and Cloud or Utility computing, in which CC is latest [1]. CC is a new paradigm of internet technology for the provisioning of computing infrastructure [2]. It refers to on demand applications and hardware delivered as a service through virtualization of hardware and systems software in the datacenters [3]. The hardware and systems software in the datacenters is referred to as a Cloud. It led to the utility computing as hardware and software are concentrated in large data centers and the users can pay as they consume computing storage and communication resources [4] [5]. It on its base use Internet technologies to offer elastic services. The term "Elastic Computing" means ability to change the number of resources used with the dynamic changes in workload in real time as per requirement [6]. These attractive features of Cloud are making huge industry moment toward it and making it as the next dominant computing paradigm. As CC is on its initial phase, it is suffering from various issues like security, virtualization, capacity allocation, load balancing, energy optimization and providing Quality of Service (QoS) guaranty [1]. Cloud inherits some of these challenges from parallel and distributed computing but it faces major challenges of its own. This paper mainly focuses on Load Balancing (LB), which is the major problem in CC. The motivation of this work is to help those researchers who are new to this problem area of Cloud. This article will help researchers to understand why LB is required in Cloud and take them to the end of work happened for this problem. This paper is organized as follows. Section II presents the need of load balancing in cloud. Section III identifies the metrics in the existing load balancing techniques. Section IV discusses various techniques of load balancing algorithms. Section V carries out the summary based on identified metrics and finally Section VI concludes the article

2. WHY LOAD BALANCING?
Load Balancing is the major concern in any network or system because it affects three main aspects of the system, i.e., performance, functionality and so the cost in cloud [7]. It is a resource management technique to utilize all resources at minimum. It is a technique to evenly distribute the workload among slug nodes in the network. For instance, we have four identical servers; A, B, C and D whose relative loads are 80%, 60%, 40% and 20%, respectively, of their capacity; as a result of a perfect load balancing each would have 50% of load each. LB middleware is used extensively to improve scalability and overall system throughput in distributed systems [8]. Why LB is a major concern in cloud when there are many scheduling techniques already exist? Answer is because of its elasticity. The resource provisioning is often provided by the independent companies. So, these companies can increase and decrease their provided resources number according to their need or competition strategy. As a result it is a work of load balancer to make a decision as to which server component gives maximum profit among the listing of available server components after receiving a particular request [9].

3. LOAD BALANCING METRICS FOR CLOUD
The existing load balancing techniques in clouds, consider various parameters "metrics" viz. response time, scalability, throughput, resource utilization, fault tolerance, migration time, associated overhead, energy consumption and carbon emission, etc [10].

Those are discussed below: a) Response Time: is the amount of time taken by computing system to give first response to the given task. It should be low. b) Throughput: It is the number of jobs completed in a given time period. It should be higher for better performance. It should be high. c) Resource Utilization: This means how many resources a system is using to complete given amount of load in optimized manner. All resources should be used to fulfill the request in context of minimizing response time and increasing the throughput of CC environment. d) Scalability: means the technique is able to

manage load in a changing number of nodes environment. e) Fault Tolerance: is the ability of load balancing technique to balance the load uniformly of a failure node to other nodes. In CC environment this property is very important because CC came as a business model. f) Associated Overhead: determines the amount of overhead involved in calculating the movement of tasks, inter-processor and inter-process communication. It should be minimized so that a LB technique can work fast. g) Migration Time: is the time to migrate a cloudlet from one node to another. It should be minimized for better performance. h) Energy Consumption: is the amount of energy (i.e. electricity) consumed by resources for execution. It must be kept low from cost perspective as well as natures'. i) Carbon Emission: means the amount of carbon generated from the system. It is directly proportional to the energy consumption. The more energy consume the more carbon will be emitted.

4. MAJOR LOAD BALANCING TECHNIQUES AVAILABLE IN LITERATURE

4.1 Prediction Base

In [11] authors have proposed and implemented a method of load balancing for Virtual Machine Cluster Based on Cloud Service. This technique focuses on to remove the unnecessary migration of Virtual Machines (VMs) which would trigger on a small transient spike of load. They proposed a load balancing method and a migration policy for virtual machine cluster to predict which VM from a cluster will migrate to where. In their work they proposed six steps for load balancing. First, get the Load status of all the nodes. Second, evaluate the status of nodes by setting threshold value and if resource utilization of VMi is below the threshold value then it will be considered as light-load llow and if resource utilization of VMi is above the threshold then it will be considered as heavy-load lhigh. Third, predict the future of load flow of next period from the previous load trends. Fourth, estimation of Benefit which means the cost of migration and the cost it will give after migration. If the estimation of benefit is less then cost without migration, it consider the migration is beneficial to the system, else it will not. Fifth, selection of receiver nodes is done by the information collected in first step. Load will migrate toward the VMs which have lowest llow. Sixth, migrate the selected VMs to the selected receiver nodes.

This technique is tested on different VMs which use same hardware and achieved well-balanced distribution of workload. This work is manly based on future prediction of workload which is calculated from past experience so it will always suffer when the peak time or seasonal time come to end.

4.2 Ant Colony Optimization

In [12] authors have proposed a method of load balancing using ant colony formation. This method used the behavior of ant for food searching, collecting and modification of route when encounter an obstacle. The main objective of this work is to synchronize the movement of ants for optimum load balancing. According to this algorithm, first, a node is chosen in a particular region in cloud which can be referred as head node. The head node is chosen such that it must have maximum number of neighbors in that region, as it will help ants to traverse all nodes in the cloud. The movement of ants will originate from head node and they will traverse in such a way that they will always know about the location of underloaded or overloaded nodes in the cloud. These ants along with the traversal also update a pheromone table, which will keep tab on the resources utilization of each node. The movement of ants according to this algorithm is of two types:

- **Forward Movement**: The ants continuously move forward till they find both overloaded and underloaded nodes**.**
- **Backward Movement:** When an ant finds an overloaded node after visiting an underloaded node then it will move backward to find if it is still underloaded or not. If it is still underloaded then it will redistribute work evenly and vice-versa. For these two types of movement the ant use two types of pheromones**:**
- **Foraging Pheromone (FP):** This Pheromone lay down by an ant after encountering the underloaded node for searching of overloaded node. That means when an ant reach upto an underloaded node it will search of an overloaded node by foraging pheromone
- **Trailing Pheromone (TP):** This Pheromone used by an ant after encountering the overloaded node to find its path back to the underloaded node. The ants first originate from head node and by default follow the Foraging pheromone for finding overloaded nodes and simultaneously update the FP trails. After coming on overloaded nodes they follow Trailing pheromone to redistribute the cloudlet and simultaneously update the TP trails. The main challenge in this algorithm is to limit the time for ants generation and the counter of nodes it will traverse because unmanaged creation of ants it can overload the network or make algorithm ineffective.

4.3 Throttled Algorithm

In [13] authors have proposed Modified Throttled Algorithm (MTA) for load balancing in cloud. This work mainly focused on distribution of cloudlets in the cloud. In this algorithm first maintains a list of available VMs with their status (Busy/Available). When a Data Center Controller (DCC) receives a new request, it queries the MTA load balancer for allocation. MTA load balancer checks its list for the available VMs and returns the first available VM id to DCC and updates the corresponding VM's status to Busy. If it does not found any available VM then it returns -1. To resolve the case of unavailability DCC notifies the balancer when any VM gets available. This algorithm works like Round Robin in case of allocation of VM from the list. This algorithm differs from Throttled algorithm in case of new allocation, where searching for allocation is always starts from first index but in MTA allocation is done in round robin fashion. This algorithm distributes workload evenly to the VMs but this algorithm does not consider any queue at the VM so it stores all the newly arrived cloudlets at DCC. Research works proposed in [14] [15] [16] [17] [18] [19] also follows the same work.

4.4 User-Priority Guided Min-Min Algorithm

In [20] authors have proposed a version of Min-Min algorithm [21] for decreasing the make span and balance the load by considering the user priority as parameter. As Min-Min algorithm fails to utilize resource properly which leads to load imbalance, so, they proposed Load Balance Improved Min-Min Scheduling Algorithm(LBIMM), to improve the balancing of load by utilizing every resource and for reducing overall completion time they extend LBIMM to User-Priority Awared Load Balance Improved Min-Min Scheduling Algorithm (PA-LBIMM). According to LBIMM algorithm, first starts the basic Min-Min algorithm. At the second step it

picks up the smallest size task from the heaviest loaded resource and calculates its completion time on other load free resources. If the completion time of that task is less than the calculated makespan of Min-Min algorithm on any of the resources then the task will be reassigned to that corresponding resource and ready time of both the resources are updated. This process continues until the load from heaviest loaded resource need not to reassign. Thus the loaded resources will get free and idle resources will utilize as well as will reduce the overall completion time. PA-LBIMM is just an extend version of LBIMM with an additional parameter User Priority. This algorithm first divides the task into two groups; first group has tasks for high priority users and second group of normal priority users' tasks. Second, it applies basic Min-Min algorithm to estimate make span of both groups. Third it applies LBIMM to both groups as describe above then the execution will go through according to the final schedule. Thus PA-LBIMM more focuses on reduction of completion time of high priority users with reduction of overall completion time.

4.5 Cloud Partitioning Method

In [22] authors' objective is to simplify the load balancing problem in big and complex clouds by dividing them. In this model suggests a method to balance the load in public cloud by partitioning a cloud among various partition. According to this, a cloud partition is a part of public cloud that is divided based on the geographic location. Every partition has its own load balancer to balance the load in that particular region of the cloud and all load balancers are connected to the central main controller which assigns cloudlets to suitable cluster of cloud. The process of this model is as follows: First, cloudlet comes to main controller which checks for suitable partition of cloud by its status, i.e., idle or have average load at that time. Then it sends the cloudlet to that partition. Here the status of the cloud partition calculated by calculating the status of every node in that partition. This work is done by the load balancer which keeps a Load Status Table in which information of each node is kept. When a cloudlet comes to the partition, it checks is Load Status Table and then according to the scheduling algorithm sends the cloudlet to the appropriate node. Here it keeps two table of Load Status to get updated status. Second, if any partition is idle then it gets the cloudlets first. Now this partition use Round Robin based on the load degree evaluation scheduling algorithm, in which all nodes will be arranged according to their load degrees. If cloudlets submitted to the partition which has average load status, then it will use strategy based on Game Theory [23]. The main challenges of this model are that it does not clarify the method of division of Cloud, does not clarify how to say any partition idle or overloaded. It has been not compared with other load balancing strategies. F. Honey bee inspired: In [24] authors have proposed an approach for load balancing using foraging behavior of honey bees [25]. The objective of this approach is to balance the load by searching the best node for a cloudlet according to its priority. In this approach VMs are divided into three groups; that are overloaded VMs, underloaded VMs and balanced VMs; by checking every VM load status and then also generate standard deviation to find group is balanced or not. Tasks from overloaded group are treated as honey bees and VMs of underloaded VMs group are destination. Tasks which are removed from overloaded VMs are act like Scout bees and according to their priorities they will search the VMs in underloaded VMs group and act like Forager bees. Here priority is also consider in searching of suitable VM which means task will search for the VM which has least number or priority task so that it will get fast response. This algorithm is better than FIFO (First In First Out), WRR (Weighted Round Robin) and DLB (Dynamic Load Balancing) for Grid in terms of make span, response time, number of task migrations. This algorithm only took priority as a QoS parameter in load balancing. Table I gives the brief summary of all the discussed techniques.

Table 1. Summary

LB Techniques	Objectives	Challenges
Prediction Base	Minimize the unnecessary migration of cloudlets based on past experience	May also imbalance the cloud because it predict based on past experience
Ant Colony Optimization	Uniform distribution of cloudlet around the slug nodes using ants' movement.	Limit the time of ant generation to reduce the network load
Throttled Algorithm	Balance the load at the scheduling time.	Does not provide solution to the dynamic change in the network. All cloudlets stores at Data Centre.
User Priority Guided Min-Min Algorithm	Decrease the Make span. Give fast response to higher priority cloudlets.	Low priority cloudlets may starve.
Cloud Partitioning Method	Simplify the load balancing process in small region by virtually dividing the Cloud.	Does not give proper method to divide the Cloud. Also does not clarify how to say any partition is idle or overloaded.
Honey Bee Inspired Method	Searching appropriate slug node for particular priority cloudlet.	Only took priority as QoS parameter.

Table 2. Analysis

LB Tech-niques	Response Time	Throughput	Resource Utilization	Scalability	Fault Tolerance	Associated Over-head	Migration Time	Energy Consu-mption	Carbon Emiss-ion
A	LOW	LOW	GOOD	GOOD	-	AVVERAGE	AVERAGE	HIGH	HIGH
B	HIGH	HIGH	GOOD	GOOD	GOOD	HIGH	HIGH	HIGH	HIGH
C	LOW	LOW	GOOD	GOOD	-	LOW	LOW	HIGH	HIGH
D	LOW	LOW	GOOD	GOOD	-	AVERAGE	AVERAGE	HIGH	HIGH
E	-	-	-	GOOD	-	LOW	LOW	HIGH	HIGH
F	HIGH	HIGH	GOOD	GOOD	-	AVERAGE	HIGH	HIGH	HIGH

5. REFERENCES

[1] Dan C Marinescu, Cloud Computing: Theory and Practice.: Newnes, 2013.

[2] Luis Rodero-Merino, Juan Caceres and Maik Lindner Luis M. Vaquero, "A Break in the Clouds: Towards a Cloud Definition," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50-55, 2008.

[3] Pradeep, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, and Kenneth Salem Padala, "Adaptive control of virtualized resources in utility computing environments," ACM SIGOPS Operating Systems Review,ACM, vol. 41, no. 3, pp. 289-302, 2007.

[4] Armando Fox, Rean Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica. Michael Armbrust, "Above the clouds: A Berkeley view of cloud computing," vol. 58, no. 4, pp. 50-58, 2010.

[5] Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic Buyya, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation computer systems, Elsevier, vol. 25, no. 6, pp. 599-616, June 2009.

[6] Nikolas Roman, Samuel Kounev, and Ralf Reussner Herbst, "Elasticity in Cloud Computing: What It Is, and What It Is Not," Proceedings of the 10th International Conference on Autonomic Computing (ICAC), May 2013.

[7] Eunmi Choi, and Ian Lumb Bhaskar Prasad Rimal, "A taxonomy and survey of cloud computing systems," In INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on, IEEE, pp. 44-51, 2009.

[8] Ossama Othman and Douglas C. Schmidt, "Issues in the design of adaptive middleware load balancing," In ACM SIGPLAN Notices, vol. 36, no. 8, pp. 205-213, 2001.

[9] Michel K. Bowman-Amuah, "Load balancer in environment services patterns.," U.S. Patent 6,578,068 B1, June 2003.

[10] Ekram M. Rewehel and Mostafa-Sami M. Mostafa, "A Survey on Load Balancing Techniques in Cloud Computing," In International Journal of Engineering Research and Technology (IJERT), vol. 3, no. 2, pp. 178-184, Feburary 2014.

[11] Rui, Wei Le, and Xuejie Zhang Wang, "Design and Implementation of an Efficient Load-Balancing Method for Virtual Machine Cluster Based on Cloud Service," Wireless, Mobile & Multimedia Networks (ICWMMN 2011), 4th IET International Conference, IEEE, pp. 321-324, 2011.

[12] Kumar, Pratik Sharma, Vishal Krishna, Chhavi Gupta, Kuwar Pratap Singh, N. Nitin, and Ravi Rastogi Nishant, "Load balancing of nodes in cloud using ant colony optimization," Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference, IEEE, pp. 3-8, 2012.

[13] S.G. Domanal and G.R.M. Reddy, "Load Balancing in Cloud Computingusing Modified Throttled Algorithm," Cloud Computing in Emerging Markets (CCEM), 2013 IEEE International Conference on, pp. 1-5, 2013.

[14] Ms. G. Vidya Mr. M. Ajit, "VM Level Load Balancing in Cloud Environment," Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on, IEEE, pp. 1-5, July 2013.

[15] Mahitha.O and Suma. V, "Deadlock Avoidance through Efficient Load Balancing to Control Disaster in Cloud Environment," Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on, IEEE, pp. 1-6, July 2013.

[16] G. Ram Mahana Reddy G.Damanal Shridhar, "Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines," Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on, IEEE, pp. 14, January 2014.

[17] Jasmin James and Bhupendra Verma, "EFFICIENT VM LOAD BALANCING ALGORITHM FOR A CLOUD COMPUTING ENVIRONMENT," International Journal on Computer Science & Engineering (IJCSE), vol. 4, no. 9, 2012.

[18] Tejinder Sharma and Dr.Vijay Kumar Banga, "Proposed Efficient and Enhanced Algorithm in Cloud Computing," International Journal of Engineering Research & Technology (IJERT), vol. 2, no. 2, Feburary 2013.

[19] Kousik Dasgupta, and Paramartha Dutta Brototi Mondal, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach, Elsevier," Procedia Technology 4, pp. 783789, 2012.

[20] Huankai, Frank Wang, Na Helian, and Gbola Akanmu Chen, "User-Priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing," Parallel Computing Technologies (PARCOMPTECH), IEEE 2013 National Conference on, pp. 1-8, 2013.

[21] Shu-Ching, Kuo-Qin Yan, Wen-Pin Liao, and ShunSheng Wang Wang, "Towards a load balancing in a three-level cloud computing network," In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol. 1, pp. 108-113, 2010.

[22] Gaochao, Junjie Pang, and Xiaodong Fu Xu, "A load balancing model based on cloud partitioning for the public cloud," Tsinghua Science and Technology, vol. 18, no. 1, pp. 34-39, 2013.

[23] Satish, and Anthony T. Chronopoulos Penmatsa, "Game-theoretic static load balancing for distributed systems, Elsevier," Journal of Parallel and Distributed Computing, vol. 71, no. 4, pp. 537-555, 2011.

[24] P. Venkata Krishna Dhinesh Babu L.D., "Honey bee behaviour inspired load balancing of tasks in cloud computing environments, Elsevier," Applied Soft Computing, Elsevier, vol. 13, no. 5, pp. 2292-2303, February 2013.

[25] Salim Bitam, "Bees Life Algorithm for job scheduling in cloud computing," International Conference on computing and Information Technology (IC2IT), pp. 186-191, 201