LOAD BALANCING TECHNIQUES IN CLOUD: A REVIEW

Dr. G. Dalin[1*] and Mrs. V. Radhamani[2]
Assistant Professor, PG & Research Department of Computer Science, Hindustan College of Computer Science, Coimbatore, Tamilnadu, India[1*]
Ph.D. Research Scholar, Department of Computer Science, Hindustan College of Arts and Science, Coimbatore, Tamilnadu, India.[2]

ABSTRACT : Cloud computing is a modern paradigm to provide services through internet. In cloud computing system, resources are distributed all around the world for faster servicing to clients. Cloud computing has faced many challenges including load balancing, security, resource scheduling scaling, Quality of Service (QoS) management, service availability and data center energy consumption. Load balancing is one of the main challenges and concerns in cloud environment. Load balancing is the process of assigning and re-assigning the load among the available resources in order to maximize the throughput while minimizing response time and cost, improving resource utilization and performance as well as energy saving. Hence providing the efficient load balancing algorithms and mechanisms is a key to the success of cloud computing. This paper presents a survey on different load balancing techniques in cloud environment. Initially, different techniques developed by previous researchers are studied in detail. Then, the limitations in those techniques are also addressed to suggest further improvement on load balancing in cloud using advanced techniques. Comparison based on parameters is also done to prove the efficiency of the various proposed load balancing techniques. The comparison results show the best load balancing technique among them.

Keywords: Cloud computing, load balancing, resource scheduling, resource utilization.

1. INTRODUCTION

In the field of network technology, the cloud computing technology [1, 2] showing a phenomenal growth due to the explosive use of internet, advancement of communication technology and solve large scale problems. It provides both software and hardware as resources over the internet for the cloud user. Cloud Computing is an internet based computing model that shares resources (storage, applications, networks, services, and servers), information and service to various devices of the user on demand. The efficient and scalable feature of cloud computing can achieve by maintaining proper management of cloud resources. The resources in the cloud are in the virtual form which is the most important characteristics of the cloud system. The Cloud Service Provider (CSP) provides services to the users in rented basis and it is more complex one with the available virtual cloud resources. This load balancing [3] has a major impact on the system performance.

Load balancing in cloud may be among physical hosts or Virtual Machines (VMs). The load balancing techniques distributes the dynamic workload evenly among all the VMs or hosts. The load balancing in the cloud is also called as load balancing as a service (LBaas). The load balancing algorithms are classified as static and dynamic based balancing algorithm [4]. The static-based balancing algorithms are more suitable for stable environment with homogenous system. The dynamic-based balancing algorithms are more efficient and adaptable in both homogeneous and heterogeneous environment. In cloud computing technology, the allocation of different tasks to VMs is called as load. The problem of load balancing are described in different ways as

- **VM/Task Migration Management:** VM migration is the movement of VM from one PM to another PM to improve the resource utilization of the data center for which the PM is overloaded. Similarly, the migration of task's current state from one VM to another VM or VM of one host to another host is termed as task migration.
- **Task allocation:** The random distribution of a finite number of tasks into different Physical Machines (PMs) which again allocated to different VMs of respective PM. The efficiency of task allocation to the cloud determines the effectiveness of the load balancing techniques.

In this paper, various load balancing techniques in cloud are analyzed based on their merits and demerits and compared each techniques in terms of total VM cost, average response time, resource utilization, waiting time, running time, response time, memory usage, CPU usage, makespan, energy consumption, fault tolerant level, performance degradation and communication cost reduction.

2. SURVEY ON LOAD BALANCING TECHNIQUES

State-Based Load Balancing (SBLB) algorithm [5] was proposed to balance load among Virtual Machines (VMs) in cloud. In addition to this, three different cloud brokering algorithm called as Cost Aware (CA), Load Aware (LA), Load Aware Over Cost (LAOC) were proposed. Once the broker selected a data center for service deployment, the next VM load balancer distributed the loads among VMs based on the VMs performance manner. Based on states of VMs, SBLB algorithm retained two different tables. It checked whether each VM in cloud reached a usage threshold. If it so, then that VM was placed in the busy state otherwise it is flagged as in the available state. The data center controller passed the user requests to the load balancer and it returned the available VMs from the state table based on the user requests. Simultaneously the table was updated after the allocation of requests to VMs. If the data center controller does not find any available VMs, then data center controller waited for resource availability. The load balancer reallocates the VMs for another task when the processing is finished in a specific VM.

A novel approach called as Dynamic Load Balancing with effective Bin Packing and VM configuration (DLBPR) [6] was proposed in cloud. The main intension of this approach was to process the jobs within their deadline and balance the load among the resources. Initially in the DLBPR approach, the jobs were classified using the deadline based job scheduler and stored in a different job queue based on the expected processing speed of the job. The VMs were dynamically clustered and then jobs were mapped into a suitable VM existing on the cluster. After the reconfiguration, the VMs were dynamically regrouped based on the processing speed of the VMs. This approach consists of three tiers are web tier, schedule tier and resource allocation tier. When a user requests were submitted to the tier, it was

forwarded to the scheduler tier. The deadline based scheduler classified and prioritized the incoming jobs. These jobs were processed effectively by VMs in the allocation tier.

A new method [7] was presented to migrate VMs between cluster nodes using TOPSIS algorithm for load balancing in cloud. A fuzzy decision making software tool was applied in the process of selecting the most overloaded server. The load in the most overloaded server was moved to the least overloaded servers while attempting to minimize data copying incurred during migration. The decision algorithm sorted the nodes and labeled those nodes with a number between 0 and 1. If there is node violates the threshold values then that node tried to mitigate its load by migration the most appropriate VM running on that machine to the least loaded node which was determined in the previous step. Then at the next level consider the most critical node and apply the fuzzy decision algorithm for the second time. At the end of this level, the best VM candidate for migration is selected.

A Genetic Algorithm (GA) based load balancing strategy [8] was proposed for load balancing in cloud. GA algorithm tried to balance the load of the cloud infrastructure while trying to reduce completion time of a given task. GA is a soft computing approach and it is composed of three operations are population generation, crossover and mutation. It created a population of possible solutions for the load balancing problem and lets them evolve over multiple generations to determine better and better solutions. After the population generation, best fitter pair of individuals was selected for crossover process. It generated a new pair of individuals. Then selected a mutation probability and based on mutation value the bits of chromosomes were toggled from 0 to 1 or 1 to 0. This strategy tried to eliminate the challenge of the inappropriate distribution of the execution time which was used to create the traffic on the server.

A soft computing approach [9] was proposed for load balancing in cloud computing using stochastic hill climbing. A local optimization approach called stochastic hill climbing was used for allocation of incoming jobs to the servers or VMs. The stochastic hill climbing was simply a loop where over utilized VMs moved in the direction of increasing value which is uphill. The moving process stopped when it reached a peak where no neighbor has a higher value. The randomly chosen variant among uphill moves and its probability will vary with its steepness of the same move. Thus it maps assignments to a set of assignments by making minor changes to the original assignment. Each element of the set was evaluated according to some criteria designed to move closer to a valid assignment to improve the evaluation score of the state. The best element of the set is made the next assignment. This basic operation is repeated until either a solution is found or a stopping criteria was reached. So it has two main components a candidate generator which maps one solution candidate to a set of possible successors, and an evaluation criteria which ranks each valid solution (or invalid full assignments), such that improving the evaluation leads to better (or closer to valid) solutions.

Honey Bee Behavior inspired Load Balancing (HBB-LB) algorithm [10] was proposed to balance loads across VMs for maximizing the throughput. Inspired from the searching and collecting food behavior of honeybees, the proposed HBB-LB algorithm considered the removal of tasks from overloaded nodes. Once a task was submitted to a VM, the number of priority tasks and load of that VM was updated. Then HBB-LB algorithm informed other tasks to help then in selecting a VM. In this algorithm tasks are signified as the honeybees and the VMs are signified as the food sources. This algorithm takes considered the tasks priorities which minimized the waiting time of tasks in a queue.

A collaborative agent based problem solving technique [11] was proposed for load balancing in cloud. This technique was capable of balancing workloads across commodity, heterogeneous servers by making use of VM live migration. The agents were provided with migration heuristics to determine which VMs should be migrated and their destination hosts, migration policies to decide when VMs should be migrated, VM acceptance policies to find out which VMs should be hosted and front-end load balancing heuristics. This technique considered both server heterogeneity and VM resource usage heterogeneity simultaneously to dynamically balance the loads in cloud. It was monitor and balance different workload types in a distributed manner.

A multi agent based load balancing (MA) algorithm [12] was proposed in Infrastructure as a Service (IaaS) cloud environment. In order to achieve well dynamic load balancing across virtual machines, the MA algorithm shifted the load in the IaaS architecture and it also maximizes the utilization of resources. This algorithm performed both sender initiated and receiver initiated approach to reduce the waiting time of the tasks and guarantee the Service Level Agreement (SLA). It was comprised of three agents are VM Migration (VMM) agent, Datacenter Monitor (DM) and Negotiator Ant (NA). The loads are monitored by VMM by collecting the bandwidth, CPU and memory utilization of the individual VM hosted by different types of tasks. The VMM's information was monitored by DM agent through information policy. It categorized the VMs based on their characteristics. DCM agents initiated NA agents. They move to other datacenters and communicate with the DCM agent of those datacenters to acquire the status of VMs there, searching for the desired configuration.

Energy aware hybrid fruitfly optimization technique [13] was proposed for load balancing in cloud. This technique used hybrid fruitfly optimization with stimulated annealing to attain the best optimum solution. The main intention of this technique was to define multi objective function for optimal use of resources by reducing the three dimensional aspects such as cost, makespan and energy optimization. The hybrid fruitful technique is comprised of two stages. In the first stage of hybrid fruitfly optimization algorithm, each swarm of flies moved in different directions to follow a uniform distribution. In the second stage of hybrid fruitfly optimization, integrated simulated annealing to update the current locations and solutions to force the hurdle of fruitfly optimization algorithm out of premature convergence, due to its exploration and exploitation ability.

Guaranteeing Fault-Tolerant requirement Load Balancing Scheme (GFTLBS) [14] was proposed based on VM migration. This scheme was migrated the VMs to balance the load without violating the fault tolerant requirement of all services. With GFTLBS, by moving storage content, memory content, network connections of VM, CPU state, VMs can be migrated from the host with the heaviest load to the lightest one while not violating the fault tolerant requirements of all services. Moreover, based on migration of VM, availability, hardware utilization, scalability, power savings and availability was increased without disrupting the customer applications running in all VMs.

Resource Intensity Aware Load balancing (RIAL) [15] method was proposed in cloud for load balancing. For each Physical Machine (PM) RIAL assigned different weights to different resources based on their intensities. The weights were then used in choosing VMs to migrate and finding destination PMs in each load balancing operation. Hence, an overloaded PM migrated out its VMs with high consumption on high intensity resources and low consumption on low intensity resources. Hence, it is relieving its load while fully utilizing its resources. In addition to, an extended version of RIAL was proposed with three additional algorithms. First algorithm determined the optimal weight. The second one is more strict migration triggering algorithm which avoids unnecessary migration. Third algorithm selected the destination PMs in a decentralized manner.

3. RESULTS AND DISCUSSIONS

This section illustrates an overview of merits and demerits of different load balancing techniques in clouds whose processes are discussed in above section. Through the literature survey on load balancing techniques in cloud, the following limitations are observed. In CA, LA, LAOC, SBLB algorithm based load balancing technique, CA needs more processing time. DLBPR based load balancing technique has a limitation of high space consumption. The major drawback of TOPSIS is when the number of VMs in cloud increases, the complexity of TOPSIS is also increased. The natural phenomena based load balancing techniques such as GA, Stochastic Hill Climbing, HBB-LB and fruitfly based load balancing techniques has different limitations such as lack of scalability, high computational complexity, starvation for lower priority load and threshold value influence the workload assignment respectively.

The agent based load balancing technique such as collaborative agent based problem solving technique and multi agent based load balancing algorithm are used for load balancing. But, in collaborative agent based problem solving technique VM migration policy is failed when either the CPU based and memory based migration threshold is violated. In multi agent based load balancing algorithm, datacenter management ants do not have a timer for self destroying and wait for message from parent. The GFTLBS based load balanced technique is less efficient technique. The RIAL based load balancing technique needs improvement in terms of effectiveness and efficiency. From the following Table 1, the most challenging issues in load balancing in cloud are observed and an ideal solution is identified to overcome those issues in cloud environment.

Table.1 Comparison of Different load balancing techniques in cloud

Ref. no.	Methods	Merits	Demerits	Performance Metrics
[5]	CA, LA, LAOC, SBLB algorithm	SBLB improves average response time	CA requires more processing time	**Total VM cost (US$)** **Cloud Scenario SC3:** CA- SBLB=1070 LA-SBLB=1150 LAOC-SBLB=1090 **Average Response Time (ms)** **Cloud Scenario SC3:** CA- SBLB=150 LA-SBLB=240 LAOC-SBLB=90
[6]	DLBPR	Increases the throughput, Increases the resource utilization	Space consumption is high	**Resource utilization (%)** **System load 0.95:** DLBPR= 80% **Waiting Time (ms)** **No. of cloudlets 50:** DLBPR= 580
[7]	TOPSIS	Minimize the migration time	Complexity of TOPSIS increased when the number of VMs increases	**Running Time (s)** **No. of VMs 1000:** TOPSIS= 120
[8]	Genetic Algorithm	Reducing job time span	Lack of scalability	**Response Time (ms)** CC1 cloud configuration GA (25 VMs) =329.01
[9]	Stochastic Hill Climbing	Effective optimization tool	High computational complexity	**Response Time (ms)** **Cloud Configuration CC1:** Stochastic Hill Climbing= 328.02
[10]	HBB-LB	Maximizing the throughput	Lack of scalability, starvation for lower priority load	**Response Time (s)** Number of tasks= 40 HBB-LB= 4 Makespan=29
[11]	Collaborative agent based problem solving technique	Balance loads in a distributed manner	VM migration policy is failed when either the CPU based and memory based migration threshold is violated	**Memory usage (%)** Collaborative agent based problem solving technique = 20.2% **CPU usage (%)** Collaborative agent based problem solving technique = 10%
[12]	Multi agent based load balancing algorithm	Maximizing resource utilization, Reduce migration cost, Avoid or reduce dynamic migration	Datacenter management ants do not have a timer for self destroying and wait for	**Response Time (s)** **Number of tasks 100:** MA =21 **Makespan** MA = 39

			message from parent	
[13]	Hybrid Fruitfly Optimization	Improves convergence rate, improves optimization accuracy	Threshold value influence the workload assignment	**Makespan (s)** Hybrid Fruitfly Optimization= 26 **Energy consumption (kWh) No. of tasks 400:** Hybrid Fruitfly Optimization= 3.1
[14]	GFTLBS	High scalability	Less efficient	**Fault tolerant level No. of service 12:** GFTLBS= 3
[15]	RIAL	Fast and constant convergence with fewer migrations	Still needs an improvement in terms of effectiveness and efficiency of load balancing	**Performance Degradation ($\times 10^3$) No. of VMs 2500:** RIAL=0.18 **Communication cost reduction Time 8 hours:** RIAL=80

4. CONCLUSION

In this paper, a detailed survey on load balancing techniques in cloud was presented. It is obvious all researchers have tried in different techniques to balance loads across VMs or hosts to improve the system performance of cloud. The discussed load balancing techniques provides the recent developments in the load balancing in cloud are analyzed by describing the novel ideas incorporated in them. The analysis of these techniques provides better understanding of the steps involved in each process thus increasing the scope for finding the efficient techniques to achieve better performance. Based on the analysis, an RIAL based load balancing technique has better performance than the other load balancing techniques. This survey helps in deriving the motivation for our future researches as well.

References

[1] Kherani, F. F., & Vania, J. (2014). Load balancing in Cloud Computing. *International Journal of Engineering Development and Research, 2(1),*907-912.

[2] Moghaddam, F. F., Ahmadi, M., Sarvari, S., Eslami, M., & Golkar, A. (2015, May). Cloud computing challenges and opportunities: A survey. In *Telematics and Future Generation Networks (TAFGEN), 2015 1st International Conference on* (pp. 34-38). IEEE.

[3] Mehmood, M., Sattar, K., Khan, A. H., & Afzal, M. (2015). Load balancing approach in cloud computing. *Journal of Information Technology & Software Engineering*, *5*(03), 1-5.

[4] Karimi, A., Zarafshan, F., Jantan, A., Ramli, A. R., & Saripan, M. (2009). A new fuzzy approach for dynamic load balancing algorithm. *arXiv preprint arXiv:0910.0317*.

[5] Naha, R. K., & Othman, M. (2016). Cost-aware service brokering and performance sentient load balancing algorithms in the cloud. *Journal of Network and Computer Applications*, *75*, 47-57.

[6] Komarasamy, D., & Muthuswamy, V. (2016). A novel approach for dynamic load balancing with effective bin packing and VM reconfiguration in cloud. *Indian Journal of Science and Technology*, *9*(11).

[7] Tarighi, M., Motamedi, S. A., & Sharifian, S. (2010). A new model for virtual machine migration in virtualized cluster server based on fuzzy decision making. *arXiv preprint arXiv:1002.3329*.

[8] Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A genetic algorithm (ga) based load balancing strategy for cloud computing. *Procedia Technology*, *10*, 340-347.

[9] Mondal, B., Dasgupta, K., & Dutta, P. (2012). Load balancing in cloud computing using stochastic hill climbing-a soft computing approach. *Procedia Technology*, *4*, 783-789.

[10] Krishna, P. V. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, *13*(5), 2292-2303.

[11] Gutierrez-Garcia, J. O., & Ramirez-Nafarrate, A. (2015). Agent-based load balancing in cloud data centers. *Cluster Computing*, *18*(3), 1041-1062.

[12] Keshvadi, S., & Faghih, B. (2016). A multi-agent based load balancing system in IaaS cloud environment. *International Robotics & Automation Journal*, *1*(1), 3-8.

[13] Lawanyashri, M., Balusamy, B., & Subha, S. (2017). Energy-aware hybrid fruitfly optimization for load balancing in cloud environments for EHR applications. *Informatics in Medicine Unlocked*, *8*, 42-50.

[14] Yao, L., Wu, G., Ren, J., Zhu, Y., & Li, Y. (2013). Guaranteeing fault-tolerant requirement load balancing scheme based on VM migration. *The Computer Journal*, *57*(2), 225-232.

[15] Shen, H. (2017). RIAL: Resource intensity aware load balancing in clouds. *IEEE Transactions on Cloud Computing*, *PP(99)*, 1-14.