International Journal of Information Retrieval Research
Volume 12 • Issue 3

Taxonomy of Load Balancing Practices in the Cloud Computing Paradigm

Mukund Kulkarni, Dr. Babasaheb Ambedkar Technological University, Lonere, India

Prachi Deshpande, Dr. Babasaheb Ambedkar Technological University, Lonere, India*

Sanjay Nalbalwar, Dr. Babasaheb Ambedkar Technological University, Lonere, India

Anil Nandgaonkar, Dr. Babasaheb Ambedkar Technological University, Lonere, India

ABSTRACT

Rapid growth in communication technology allows users location-independent access to IT infrastructure at pay-per-use via cloud computing. This has paved a new paradigm in information processing for the consumers. Due to cloud's inherent characteristics, most service providers shift to the cloud and its data centers. To retain cloud service reliability, it's essential to carry out the minimum latency tasks and cost-effectively. Various techniques to improve performance and use of assets are focused on load control, task management, resource management, service quality, and workload management. Data load balancing helps data centers to avoid overload/underload of virtual machines, a difficulty in the world of cloud computing. This study reports a state-of-the-art analysis of current load balancing approaches, problems, and complexities to design more successful algorithms.

KEYWORDS

Cloud Computing, Load Balancing, Optimization, Resource Allocation, Task Scheduling, Virtual Machine, Workload Management

1. INTRODUCTION

Rapid development in information technology (IT) replaced traditional computing techniques with cloud computing. The Cloud allows consumers to connect many configurable computation assets (computers, memory, networks, apps) to provide 24x7 facilities to customers at pay-per-use rates (Brown, 2011, Buyya, 2010). It also enables resource allocation across the globe to perform various information centers, which allow cost-effective services to both cloud service providers (CSPs) and users (Chiregi, 2016).

As the Cloud offers various services to its users, cloud load balancing is one of the main concerns to the CSPs to avoid overloading scenarios during task estimation. Load balancing provides the capability to divide the burden evenly with the available resources. Thus, load balance aims to reduce the response time for tasks and optimize resources, which increases system performance at a lower cost. Further, load balance's objectives are to reduce energy consumption and carbon emissions, avoid bottlenecks, supply resources and meet the QoS requirements for load balance. Global research groups are interested in the design and development of the best possible methods for resource allocation.

DOI: 10.4018/IJIRR.300292

*Corresponding Author

Hence the present study and assessments focus on them. Cloud load balance is a method of distributing the load on unbundled virtual machines to improve device flow. Numerous difficulties alongside load managing include asset programming, tracking efficiency, QoS Management, power usage, and internet storage accessibility (Kaur, 2012; Malladi, 2015). This paper provides an extensive examination of the multiple kinds of cloud planning, load balancing, and task sharing methods.

Ghomi et al. performed a comprehensive evaluation of charge equilibrium in memory, dividing the study into six classifications: Hadoop MapReduce charge balance methods, Natural charge equilibrium phenomena methods, agent-based charge balancing methods, General charge balancing methods, Network-aware job planning, and count processing methods. They primarily focused on Hadoop MapReduce and Efficiency of Energy, which affect cloud count balancing. Nevertheless, their job requires task and load balancing dependent on the cluster (Ghomi, 2017).

Milani and Jafari examined and categorized multiple current load balance systems into vibrant and hybrid subdomains. The behavior, disadvantages, and difficulties of these methods were defined based on the various parameters. They have identified challenges in developing more efficient algorithms to reduce resources and energy consumption and increase the efficiency of load balancing technologies. However, they have not discussed task-based load balance, cluster-based load balance, and energy consumption problems (Milani, 2016). Singh et al. provided the comprehensive evaluation of algorithms for metaheuristic workflow planning, recognized numerous problems relating to cloud job planning, and reported a comparative assessment based on the meta-heuristic strategy for both dependency and independence (Singh, P., 2017). Singh and Chana reported six distinct views: planning workload, tracking, QoS necessity, implementation layout, auto-management of workload, and assessed automated cloud resource management (Singh and Chana, 2015). Ivanisenko and Radivilova presented a comparative assessment of the load-balancing techniques centered on metrics (response time, relocation moment, scalability, and asset usage). However, they missed out on the problems in current technology and difficulties and potential developments (Ivanisenko, 2015). Katyal and Mishra have assessed the load-balancing algorithms centered on SLA user requirements for different cloud settings. They address benefits, disadvantages, and difficulties in the primary classifications of current methods (Katyal, 2014). However, they did not assess the basis of varying load balance parameters. Hence, there is a need to define the taxonomy of load balancing techniques in cloud scenarios. The paper reports a detailed taxonomy of load balancing techniques in a cloud scenario.

2. THE CHALLENGES

To ensure the reliability of the services provided by Cloud computing, load balancing is a significant task that requires unique focus. Moreover, several other problems like virtual machine migration, VM safety, customer compliance with QoS, and asset use involve the same direction as the other aspects that must be accounted for during the information processing system's design. Therefore, the challenges in load balancing technique can be listed as:

- **Geographical Distributed Nodes:** Generally, cloud information centers are dispersed geographically for computing reasons. Therefore, it should be considered to design load-balancing systems for distant servers.
- **Single Point of Failure:** If the central unit collapses, the general computing environment will be affected. Therefore, some distributed algorithms have to be developed in which a given node does not regulate the entire scheme.
- **VM Migration:** Virtualization enables multiple VMs to be created on one physical machine. These VMs have distinct settings and are autonomous in design. Some VMs have a VM-load-balancing strategy to move to a remote place if a physical device is overloaded.
- **Heterogeneous Nodes:** In cloud computing, users' demands alter dynamically, requiring them to be executed on heterogeneous nodes to use resources and minimize response time efficiently.

- **Storage Management:** The issue of high hardware costs for storage has been solved in cloud storage. The Cloud permits clients to save information without any access issues heterogeneously (T.Wu, 2012). However, an effective load balancing method is required to consider the allocation of implementation and associated information based on a temporary replication scheme.
- **Load-Balancer Scalability:** To cope with the dynamic environment of the Cloud, the load balancer mechanism shall be highly scalable.
- **Algorithm Complexity:** Algorithms should be easy to execute in cloud computing. A complicated algorithm reduces the cloud service reliability and cloud application output.

3. THE LOAD BALANCING IN CLOUD

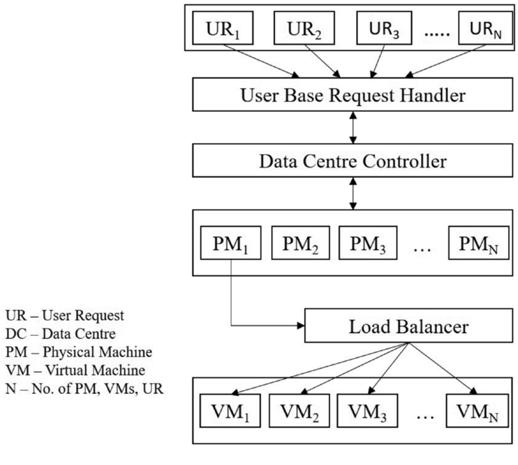
Cloud on request includes a shared resource base (e.g., Cloud, space, and network) requiring a high client task management and maintenance level (Zeng, 2015; Zhang, 2010). Therefore, an excellent load balancing system is necessary to assign assignments to VMs based on their QoS demands to handle customer demands for accessible resources (Bhardwaj, 2018). Figure 1 shows the load balancer model and workflow. In the Cloud, customer demands involve the dynamic environment to perform the functions vary considerably. If the Cloud detects any user requests, a cloud service broker can detect resource accessibility and consult with additional brokers on efficiency and resource expense. After analyzing available resources, the broker transfers the request from a user to the selected data center, where they are accepted by the Data Center (DCN).

During this stage, the load balancer receives VMs from the state table and changes the table after assignment. The DCN sets the tasks in the queue and waits for the availability of resources. A VM manager is also available in the data center, which manages all VMs on physical machines. The load balancer is accountable for assigning an appropriate VM for assignments where the job is a critical problem in the Cloud. The load balancer also ensures that VMs are not overloaded. Thus, a load balancer improves the use and accessibility of resources and minimizes the response time for assignments.

3.1. Classification Strategies Based on the State of the System

Load balancing methodologies are mainly classified depending on resource initiation and deployment. Fig. 1 depicts typical load balancing methodologies. Based on resource initiation, the load balancing is classified as:

Figure 1. Load balancing model



- **Sender Initiated:** If the node is overloaded, other nodes are searched for which the workload is easily accessed. If nodes become congested, the sender will begin the method of finding the underloaded nodes.
- **Receiver Initiated:** This method searches for highly loaded nodes to communicate the receptors' workload or weakly charged nodes.
- **Symmetric:** This method combines sender-initiated and receiver-initiated method process techniques.

Based on the deployment of resources, the load balancing strategies can be classified as:

1. **Static:** Static load balancing methods do not depend on the present state of the system. This method is easy to implement but usually unable to identify the connected computers, resulting in uneven resource allocation (Liang, 2017). Thus it is inappropriate for dynamically changing distributed systems.
2. **Dynamic:** In this method, the assignments can be transferred from an overloaded to an underloaded device. Dynamic load handling methods are versatile, leading to device efficiency improvements. It tracks a load of nodes continually. It shares load and status details between servers at a specified time window to calculate their workload and redistribute the working force between servers. An agent-based method, Honey Bee Behaviour and Throttled Load Balancing are popular techniques for dynamic load balancing (Singh, A., 2017; Babu, D., 2013; Shridhar, 2013).
3. **Distributed:** Here, all nodes are involved in the distribution of load. To manage the duties effectively, all nodes retain data for communication. If all of the nodes within the scheme operate together to attain a common objective or decision making, it's called cooperative; otherwise, it is non-cooperative.
4. **Non-distributed:** A separate node chooses load distribution in non-distributed methods (Das,2003; Ahmad,1991). In the case of behavior, non-distributed methods can be distributed or semi-coupled. A single node conducts the load management processes and is accountable for the load balance in distributed methods.

3.2. Load Balancing Metrics

A load balancer is mandatory to spread the computation load over accessible resources to increase resource use and enhance efficiency. Following metrics are considered for load balancing techniques:

1. **Performance:** After implementing the method, the system's efficiency must be validated compared to other current load handling methods.
2. **Response Time:** It is the total time required for the execution of a system request.
3. **Throughput:** The total number of tasks or operations completed on a system over a while. The greater the output, the more efficient the system.
4. **Scalability:** The system can achieve a uniform load balancing when the number of nodes required increases.
5. **Fault Tolerance:** The ability to evenly accomplish the load balancing method when any connection or node is broken down.
6. **Migration Time:** The migration time is used to determine the total time needed for a request/task sent from an overloaded machine to the under the loaded machine. The less time you migrate, the faster the cloud system performs.
7. **Resource Use:** This is analyzed to guarantee the correct use of all resources in the system. Greater resource use reduces general and energy costs and the decrease in the cloud scheme carbon emissions.

3.3. Load-Balancing Policies

1. **Selection Policy:** This strategy sets out all the tasks to be passed from one node to another. It opts for the overhead-based tasks needed to migrate, the number of non-local system calls, and the time to accomplish the task.
2. **Location Policy:** This policy defines the underloaded or open computing nodes and transfers processing tasks to them. It decides the target node according to available techniques. In the process of negotiations, nodes discuss load balance with each one.
3. **Transfer Policy:** This policy identifies the conditions under which tasks must be transferred from a local node to a regional/remote node.
4. **Information Policy:** This is another dynamic load-sharing strategy that maintains all details in the system used to create selections by other measures. It determines the time to collect data.

Both the transfer strategy and the location policy gather the information required to make a choice.

3.4. Existing Methodologies

The load balancing methods are primarily classified into static and dynamic systems depending on the system's state. The current techniques for load balancing are as:

1. **Static Load-Balancing Technique:** Static load balance strategies need not know the system's present state; only machine resources such as run time, space, storage capabilities, and node handling capability are kept in progress. Therefore, static load balance does not enable resource distribution at runtime. These methods can be easily implemented and applied but are helpful for smaller systems or smaller networks. Since the scheme's present state is not considered, these methods are not beneficial for continuous calculation applications.
2. **Dynamic Load-Balancing Technique:** Static techniques are inappropriate for distributed computer systems, which are dynamically changing. Therefore, a flexible approach is required for cloud load-balancing. A comparative analysis of different dynamic load balance techniques depending on the load balance criteria is reported in this section:
    a. General Load Balancing Techniques
    b. Natural Phenomena Centred Load Balancing Techniques
    c. Hybrid Load Balancing Technique
    d. Agent-Based Load balancing
    e. Task-based Lad Balancing
    f. Cluster-Based Load Balancing

Tables 1 to 7 summarises the above methodologies with a detailed analysis of each method. Fig. 2 depicts the classification of load balancing techniques.

4. THE RESEARCH TRENDS

There are many questions and concerns to be examined and addressed in the future in cloud load equilibrium. We found some of the future areas from the literature review in which the Cloud needs to focus. Several potential approaches need to be addressed to manage cloud performance: service reliability (QoS), the extent of service agreement, services availability, and load balancing, among other matters. Cloud service providers need to maintain QoS and SLA with several resources. SLAs are developed and implemented according to QoS guidelines, and if an SLA violation happens, a service provider must pay a fee.

Table 1. General load balancing methodologies

Contributions	The System	Methodology	Concept	Advantages	Disadvantages
(Wang and Chen,2015)	Dynamic	Speedy storage workload and resource management framework	Live VM Migration	Low task execution	A single task, homogeneous Virtual machine
(Chien and Son,2016)	Dynamic	Use of the approximate time of work	Load balancing VMs using the time to end service	Reduced response time, time to process	Present processing power is hard to calculate; more energy is used
(Chana and Bala,2016)	Dynamic	The predictive solution to load balance for machine learning	Use machine learning to distinguish packed and underloaded node	High use of resources, lower overhead migration, lower migration numbers	Not tested on a real cloud
(Kulkarni and Ghoneem, 2017)	Dynamic	The improved load-balancing technique for Active VM	Use of the reservation table to distribute demands equally.	Minimum time to response for assignments, load balance between Virtual Machines, elasticity improvement	Assigns tasks to the single data center equally
(Liang and Chen, 2017)	Dynamic	A unique approach to load balance to reduce the load on servers	Static load balancing dynamical annexed method.	Enhanced use of resources, make span, and Quality of Service	Improved response time with minimum criteria

Table 2. Natural phenomena–motivated load-balancing technique

Contributions	The System	The Approach	Notion	Advantages	Disadvantages
(Babu and Krishna, 2013)	Dynamic	The behavior of the Honey bee inspires load balancing.	Use of the behavior of bees.	Low time to get through and answer.	Does not operate for dependent activities.
(Falco and Laskowski, 2015)	Dynamic	Extreme load balancing optimization	Execution of concurrent job in a dynamic environment	Lower time to execute, fewer task transfers, and increased use of resources.	Does not support multi-objective optimization and graph optimization.
(Babu, R., Samuel, P, 2016)	Dynamic	Improved load balancing in the bee colony	Use of honey bee technology to reduce utilization of resources and time to response	Low time to respond, better use of resources, lower relocation of tasks	Poor scalability, complexities
(Devi, C., Uthariaraj, R, 2016)	Dynamic	Weighted round-robin approach.	Reduce response time of activities when taking into account time to execute.	Low time to respond.	Uniform execution of environment.

Automatic availability of resources minimizes communication between clients and service providers. Load balancing for the appropriate use of the available services is required to maintain the SLA and QoS. A load balancer helps to maintain minimal asset costs at a high performance. Various load-equilibrating strategies, such as efficiency, response time, deployment time, project

Table 3. Hybrid load-balance methodologies

Contributions	The System	The Approach	Notion	Advantages	Disadvantages
(Wang, Z., Chen, H, 2015)	Dynamic	Adaptive scheduling technique for parallel tasks	Task transfer with minimum cost strategy to increase resource usage	Optimized resource usage with minimum task migration and improved QoS	Operational cost, Energy efficiency are not considered
(Norouzi, M., Sharifi, M., 2014)	Dynamic	The mixture of load balancing and Resource identification	Scalability improvement by coordination of resource allocation and load balancing	Highly scalable, Independent of neighboring nodes for operation	High resource discovery time
(Cho, K., Tsai, P., 2015)	Dynamic	ACO combined with PSO	Improved usage of resources by clubbing ACO and PSO	Optimal resource use with the lowest latency	High computation cost, homogeneous server support
(Liu, Y., Zhang, C.,2015)	Dynamic	Task scheduling with hybrid load balancing	Master-Slave approach for load balancing	Independent and dependent task scheduling, lower response time	High transmission and scheduling time
(Naha, R., Othman, M., 2016)	Dynamic	Combination of broker and load-balancing techniques	Combining load-balancing and broker techniques to reduce response time	Low processing time and quick response time	Low performance, high execution time
(Liang, S., Chen, Y.,2017)	Dynamic	Cloud load-balancing (CLB) technique	Load balancer architecture to monitor server response failure	Highly scalable	High response time

Table 4. Agent-based load balancing technique

Contributions	The System	Methodologies	Notion	Advantages	Disadvantages
(Chen, C., Zhu,X.,2013)	Dynamic	New emerging cloud task assignment	Emerging cloud tasks by competition and the concept of dynamic improvement	Efficient resource allocation	Increased time for processing, increased time for transmission
(Tasquier, L., 2015)	Dynamic	Load balance on a multiagent basis	Use of various agents in a multi-cloud environment to provide resources and monitor	Offers high elasticity and uses multi-cloud resources	QoS is not taken into account
(Garcia, O., Nafarrate, A.,2015)	Dynamic	Load balancing based on agent	Agent-based Virtual Machine migration	Heterogeneous server and Virtual Machine	Weak scalability, substantial transfer overhead
(Keshvadi, S., Faghih, B.,2016)	Dynamic	Load balancing architecture based on multiagent.	Maximization of multiagent resource use	Low time to respond, enhanced resource use	DcM agents have to destroy parents ' messages without timers to autodestruct themselves.

Table 5. Task-Oriented load-balancing technique

Contributions	The System	Approach	Notion	Advantages	Disadvantages
(Fahimeh, R., Jie, L., 2014)	Dynamic	Task dependent load-balancing approach by maximizing the particle swarm	Live Virtual Machine migration	Low task execution	Homogeneous VM, independent task
(Zhilong, W., Sheng, X., 2016)	Dynamic	A hybrid algorithm for task planning of genetic ant colony	Genetic quest for available resources. ant colony selections for optimal execution	Improved balance of the load, improved efficiency, reduced time for execution	Untested in existing cloud environments
(Haiying, S., Lei, Y., 2016)	Dynamic	Considering topology of the network and time for schedule	Implement MapReduce task planning to minimize time and cost of data transmission	Better use of the cluster, the shortened job for completion time	No allocation for bandwidth, no design is tested under various network conditions
(Samir, E., Shahenda, S.,2017)	Dynamic	Dynamic task algorithm with a hybrid round-robin strategy	Place and perform short and long work independently in different ready queues to reduce famine.	Waiting time and starvation time for tasks reduced, time for response and turnaround improved	Task quantum is less effective
(Yu, X., Zhi, X.,2017)	Dynamic	Multiple schedulers cost-effective for concurrent tasks	Multiple schedules for running simultaneous tasks and weighted machine assignment	Enhanced use of resources, reduced task weighting, and time for execution	Assessing the parameter is not optimal

migration time, and resource usage, have been developed. However, no methodology has considered all criteria of load balancing, which enhances total data center efficiency. In the cloud load balancing, the following are significant open issues and challenges:

1. Proper resource distribution is required to improve the efficiency of the Cloud.
2. All operational (SLAs) and non-functional (QoS) need to be respected.
3. Manage a large number of user requests instantly and manage SLA to carry out the tasks currently performed.
4. To maintain application costs with various service providers and heterogeneous environments in the case of implementation.
5. Service providers build many data centers. The selection of the exemplary service as per their criteria is a significant challenge for clients.
6. For increasing cloud providers, a scenario can emerge that needs a task to be moved to another cloud provider and present a challenge because of different information and service regulations.
7. Data lockup can also be a concern when the application has to be moved to another cloud provider needing other rules to address certain problems.
8. Boost cloud platform performance through component evaluation and a control point-based methodology.

Table 6. Cluster-Based load-balancing technique

Contributions	The System	Approach	Notion	Advantages	Disadvantages
(Eman, D.,Shyan, Y.,2015)	Dynamic	A limited global overlay network for optimizing adaptive load balancing	Greedy built controls in distributed mega data centers	Effective use of energy, enhanced efficiency of the data center	High power consumption
(Byungseok, K., Hyunseung, C.,2016)	Dynamic	Job dispatch methodology based on clusters	Improvement of massive inter-cloud engagement for dynamic and real-time digital media streaming load balancing	Better response time	Loss of packet due to congestion
(Jia, Z., Kun, Y., Xiaohui,2016)	Dynamic	Bays and clusters load balancing (LB-BC)	The possibility of external hosts afterward is associated with the principle of clustering	Lower time to react	Good just for LAN, not appropriate for real-time systems
(Yiming, H., Anthony, C., 2017)	Dynamic	Decentralized self-planning systems to maximize load balancing.	Improved load balancing and scalability by distributed self-scheduling method	Increased scalability, better overall efficiency, decreased overhead for communication	simultaneous running is not allowed

9. When demand for cloud infrastructure and data centers rises, power consumption often exceeds; a significant issue is to reduce power requirements.
10. A large amount of data, which needs high-quality storage and stellar methodology for easy extraction and evaluation, is produced daily from various sources such as banking, social sites, and the e-commerce market.
11. Resource and system management is a challenging task in the heterogeneous cloud environment.
12. A very well-designed system for allocating resources is a significant problem for service providers to maximize assets' efficient use.

The present era of information processing has rapidly changed and moving ahead towards 5G. Further, IoT and big data have revolutionized human life with their mesmerizing capacities. These applications require considerable processing power with minimum latency. Cloud computing is the best alternative available so far as compared to the traditional computing alternatives. However, the underlying mechanism's load must be distributed judiciously to avoid the processing bottleneck to support the real-time applications. Therefore, load balancing techniques will play a pivotal role.

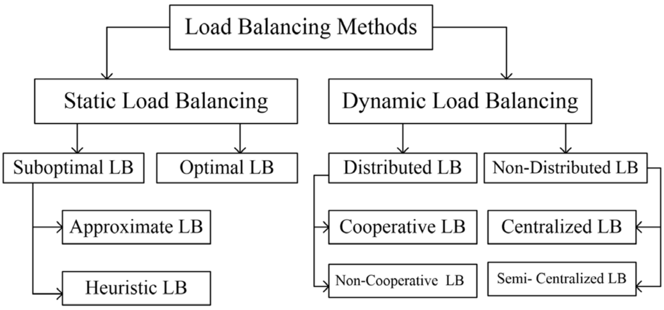
5. CONCLUSION

The load balancing of Virtual Machine tasks is a significant challenge in cloud computing, which has received considerable research exposure. The paper reports an in-depth analysis of various load-balancing methods with strategies in each class benefits, drawbacks, principles, and obstacles—the approach to maximize the system's performance in the future needs to be strengthened. In contrast, load-balancing methods prioritize Sustainable technology, fuel consumption, and workload control to increase system performance and power use. Furthermore, the latest load balancing strategies for simulators based on various load balancing parameters from multiple categories will also need

Table 7. The state-of-the-art Cloud load-balancing

Reference	Time to Response	Time for Migration	Throughput	Make-span	Resource Usage	Scalability	Fault Tolerance	Cost
(Wei, W., Yue, C.,2013)	✓	✓		✓	✓			✓
(Yingchi, M., Daoning, R.,2013)	✓		✓		✓			
(Sue, C., Ming, L.,2014)	✓							
(Garcia, O., Nafarrate, A., 2015)	✓		✓					
(Elina, P., Cristian, M.,2015)	✓	✓	✓		✓			
(Florin, P., Ciprian, D., 2015)	✓				✓			✓
(Moona, Y., Seyed, M.,2015)		✓	✓		✓			
(Shridhar, D., Ram, R.,2013)	✓			✓	✓	✓		
(Babu, R., Samuel, P.,2015)	✓							✓
(Keshvadi, S., Faghih, B.,2016)	✓	✓	✓		✓			
(Zhilong, W., Sheng, X., 2016)	✓			✓				
(Somayeh, K., Nasrolah, M., 2016)				✓	✓			
(Soumi, G., Chandan, B.,2016)	✓			✓				
(Bok, K., Jaemin, H.,2016)				✓				
(Priyanka, S., Palak, B.,2016)	✓	✓		✓	✓	✓		
(Jixiang, Y., Ling, L., 2016)	✓			✓			✓	
Mohsen, S. et al.,2016		✓	✓	✓	✓			
(Ahmad, I., Ghafoor, A., 1991)	✓				✓			
(Faouzia, Z., Abdellah, I., 2016)	✓				✓			
(Samir, E., Shahenda, S., 2017)	✓		✓		✓			
(Ghoneem, M., Kulkarni, L., 2017)				✓		✓		
(Bebal and Dejey,2017)				✓	✓	✓		✓
(Maria, R., Buyya, R.,2017)	✓				✓			
(Kaur, R., Luthra, P., 2012)				✓	✓			

Figure 2. Classification of load-balancing strategies



to be tested to predict these methods' feasibility before they can be implemented in the current cloud environment.

In-depth research in cloud load balancing methodologies will help humanity irrespective of their socio-economical strata. The present study will assist scientists, academicians, and researchers in identifying research problems in load balance and summarize the available methods for load balance.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

Ahmad, I., & Ghafoor, A. (1991). Semi-distributed load balancing for massively parallel multicomputer systems. *IEEE Transactions on Software Engineering*, *17*(10), 987–1004. doi:10.1109/32.99188

Babu, D., & Krishna, V. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, *13*(5), 2292–2303. doi:10.1016/j.asoc.2013.01.025

Babu, R., & Samuel, P. (2016). Enhanced bee colony algorithm for efficient load balancing and scheduling in Cloud. *Innovation in Bio-Inspired Computing and Application*, *4*, 135–142.

Bala, A., & Chana, I. (2016). Prediction-based proactive load balancing approach through VM migration. *Engineering with Computers*, *32*(4), 1–12. doi:10.1007/s00366-016-0434-5

Bhardwaj, A., & Rama Krishna, C. (2018). Efficient multistage bandwidth allocation technique for virtual machine migration in cloud computing. *Journal of Intelligent & Fuzzy Systems*, *36*(5), 1–14. doi:10.3233/JIFS-169819

Bibal, B. (2017). Performance Improvement of MapReduce for heterogeneous clusters based on efficient locality and replica aware scheduling (ELRAS) strategy. *Wireless Personal Communications*, *95*(3), 2709–2733. doi:10.1007/s11277-017-3953-5

Bok, K., Jaemin, H., Jongtae, L., Yeonwoo, K., & Jaesoo, Y. (2016). An efficient MapReduce scheduling scheme for processing large multimedia data. *Multimedia Tools and Applications*, *76*(16), 17273–17296. doi:10.1007/s11042-016-4026-6

Brown, E. (2011). *Final Version of NIST Cloud Computing Definition.* https:// www. Bluepiit.com/blog/different-types-of-cloud-computing-service-models/

Buyya, R., Broberg, J., & Goscinski, A. (2010). *Cloud Computing: Principles and Paradigms* (Vol. 87). John Wiley & Sons.

Byungseok, K., & Hyunseung, C. (2016). A cluster-based decentralized job dispatching for the large-scale Cloud. *EURASIP Journal on Wireless Communications and Networking*, *25*, 1–8.

Chen, C., Zhu, X., Bao, W., & Sim, K. (2013). An agent-based emergent task allocation algorithm in clouds. *Proceedings of the 10th IEEE International Conference on High-Performance Computing and Communications*, 1490–1497. doi:10.1109/HPCC.and.EUC.2013.210

Chien, N., Son, N., & Loc, H. (2016). Load balancing algorithm based on estimating finish time of services in cloud computing. *Proceedings of the 18th IEEE International Conference on Advanced Communication Technology (ICACT)*, 228–233.

Chiregi, M., & Navimipour, N. (2016). A new method for trust and reputation evaluation in the cloud environments using the recommendations of opinion leaders entities and removing the effect of troll entities. *Computers in Human Behavior*, *60*, 280–292. doi:10.1016/j.chb.2016.02.029

Cho, K., Tsai, P., Tsai, C., & Yang, C. (2015). A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. *Neural Computing & Applications*, *26*(6), 1297–1309. doi:10.1007/s00521-014-1804-9

Das, S., Viswanathan, H., & Rittenhouse, G. (2003). Dynamic load balancing through coordinated scheduling in packet data systems. *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications*, 1, 786–796.

Devi, C., & Uthariaraj, R. (2016). Load balancing in cloud computing environment using improved weighted round-robin algorithm for non pre-emptive dependent tasks. *TheScientificWorldJournal*, *2016*, 1–14. doi:10.1155/2016/3896065 PMID:26955656

Elina, P., Cristian, M., & Carlos, G. (2015). Balancing throughput and response time in online scientific Clouds via ant colony optimization. *Advances in Engineering Software*, *84*, 31–47. doi:10.1016/j.advengsoft.2015.01.005

Eman, D., & Shyan, Y. (2015). A small world-based overlay network for improving dynamic load-balancing. *Journal of Systems and Software*, *107*, 187–203. doi:10.1016/j.jss.2015.06.001

Fahimeh, R., Jie, L., & Farookh, H. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. *International Journal of Parallel Programming*, *42*(5), 739–754. doi:10.1007/s10766-013-0275-4

Falco, I., Laskowski, E., Olejnik, R., & Tudruj, M. (2015). External optimization applied to load balancing in execution of distributed programs. *Applied Soft Computing*, *30*, 501–513. doi:10.1016/j.asoc.2015.01.048

Faouzia, Z., Abdellah, I., & Hajar, R. (2016). Resource allocation with efficient load balancing in cloud environment. *Proceedings of International Conference on Big Data and Advanced Wireless Technologies (BDAW), 46*, 1-7.

Florin, P., Ciprian, D., Valentin, C., Nik, B., Fatos, X., & Leonard, B. (2015). Deadline scheduling for aperiodic tasks in inter-Cloud environments: A new approach to resource management. *The Journal of Supercomputing*, *71*(5), 1754–1765. doi:10.1007/s11227-014-1285-8

Garcia, O., & Nafarrate, A. (2015). Agent-based load balancing in cloud data centers. *Cluster Computing*, *18*(3), 1041–1062. doi:10.1007/s10586-015-0460-x

Ghomi, E., Rahmani, A., & Qader, N. (2017). Load-balancing algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, *88*, 50–71. doi:10.1016/j.jnca.2017.04.007

Ghoneem, M., & Kulkarni, L. (2017). An adaptive MapReduce scheduler for scalable heterogeneous systems. *Proceedings of the International Conference on Data Engineering and Communication Technology*, 603–611. doi:10.1007/978-981-10-1678-3_57

Haiying, S., Lei, Y., Liuhua, C., & Zhuozhao, L. (2016). Goodbye to fixed bandwidth reservation: Job scheduling with elastic bandwidth reservation in clouds. *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 1–8.

Ivanisenko, I., & Radivilova, T. (2015). Survey of major load balancing algorithms in distributed system. *Information Technologies in Innovation Business Conference (ITIB),* 89–92. doi:10.1109/ITIB.2015.7355061

Jia, Z., Kun, Y., Xiaohui, W., Yan, D., Liang, H., & Gaochao, X. (2016). A heuristic clustering-based task deployment approach for load balancing using Bayes theorem in cloud environment. *IEEE Transactions on Parallel and Distributed Systems*, *27*(2), 305–316. doi:10.1109/TPDS.2015.2402655

Jixiang, Y., Ling, L., & Haibin, L. (2016). A hierarchical load balancing strategy considering communication delay overhead for large distributed computing systems. *Mathematical Problems in Engineering*, *2016*, 1–9.

Katyal, M., & Mishra, A. (2014). A comparative study of load balancing algorithms in cloud computing environment. *International Journal of Distributed and Cloud Computing*, *1*(2), 5–14.

Kaur, R., & Luthra, P. (2012). Load balancing in cloud computing. *Proceedings of the International Conference on Recent Trends in Information, Telecommunication and Computing (ITC'12)*, 374–381.

Keshvadi, S., & Faghih, B. (2016). A multiagent-based load balancing system in an IaaS cloud environment. *International Robotics & Automation Journal*, *1*(1), 1–6. doi:10.15406/iratj.2016.01.00002

Liang, S., Chen, Y., & Kuo, S. (2017). CLB: A novel load balancing architecture and algorithm for cloud services. *Computers & Electrical Engineering*, *58*, 154–160. doi:10.1016/j.compeleceng.2016.01.029

Liu, Y., Zhang, C., & Niu, J. (2015). DeMS: A hybrid scheme of task scheduling and load balancing in computing clusters. *Journal of Network and Computer Applications*, *83*, 213–220. doi:10.1016/j.jnca.2015.04.017

Malladi, R. (2015). An approach to load balancing in cloud computing. *International Journal of Innovative Research in Science, Engineering and Technology*, *4*(5), 2319–8753.

Maria, R., & Buyya, R. (2017). Scheduling dynamic workloads in multi-tenant scientific workflow as a service platforms. *Future Generation Computer Systems*, *79*(2), 739–750.

Milani, A., & Navimipour, N. (2016). Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends. *Journal of Network and Computer Applications*, *71*, 86–98. doi:10.1016/j.jnca.2016.06.003

Mohsen, S. (2016). Stochastic-based robust dynamic resource allocation for independent tasks in a heterogeneous computing system. *Journal of Parallel and Distributed Computing*, *97*, 96–111. doi:10.1016/j.jpdc.2016.06.008

Moona, Y., Seyed, M., Ghafari, Y., & Ahmad, P. (2015). Proposing a load-balancing method based on cuckoo optimization algorithm for energy management in cloud computing infrastructures. *Proceedings of the 6th International Conference on Modelling, Simulation, and Applied Optimization (ICMSAO)*, 1–5.

Naha, R., & 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. doi:10.1016/j.jnca.2016.08.018

Norouzi, M., & Sharifi, M. (2014). A model for communication between resource discovery and load balancing units in computing environments. *The Journal of Supercomputing*, *68*(3), 1538–1555. doi:10.1007/s11227-014-1124-y

Priyanka, S., Palak, B., & Saurabh, G. (2016). Assorted load-balancing algorithms in cloud computing: A survey. *International Journal of Computers and Applications*, *143*(7), 1–8.

Samir, E., Shahenda, S., & Manar, J. (2017). A novel hybrid of shortest job first and round-robin with dynamic variable quantum time task scheduling technique. *Journal of Cloud Computing*, *6*(1), 1–12.

Shridhar, D., & Ram, R. (2013). Load balancing in cloud computing using modified throttled algorithm. *Proceedings of the IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 1–5.

Singh, A., Juneja, D., & Malhotra, M. (2015). Autonomous agent-based load balancing algorithm in cloud computing. *Procedia Computer Science*, *45*, 832–841. doi:10.1016/j.procs.2015.03.168

Singh, P., Dutta, M., & Aggarwal, N. (2017). A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowledge and Information Systems*, *52*(1), 1–51. doi:10.1007/s10115-017-1044-2

Singh, S., & Chana, I. (2015). QoS-aware autonomic resource management in cloud computing: A systematic review. *Computer Survey*, *48*(3), 1–42. doi:10.1145/2843889

Somayeh, K., Nasrolah, M., & Mehdi, K. (2016). Ant colony based constrained workflow scheduling for heterogeneous computing systems. *Cluster Computing*, *19*(3), 1053–1070. doi:10.1007/s10586-016-0575-8

Soumi, G., & Chandan, B. (2016). Priority based modified throttled algorithm in cloud computing. *Proceedings of International Conference on Inventive Computation Technologies (ICICT)*, 3, 1–6.

Sue, C., Ming, L., & Yi, C. (2014). A load-balanced MapReduce algorithm for blocking-based entity- resolution with multiple keys. *Proceedings of the 12th Australasian Symposium on Parallel and Distributed Computing*, 152, 3–9.

Tasquier, L. (2015). Agent-based load-balancer for multi-cloud environments. *Journal of Cloud Computing Research*, *1*(1), 35–49.

Wang, Z., Chen, H., Liu, D., & Ban, Y. (2015). Workload balancing and adaptive resource management for the swift storage system on Cloud. *Future Generation Computer Systems*, *51*, 120–131. doi:10.1016/j.future.2014.11.006

Wei, W., Yue, C., Win, T., & Yi, L. (2013). Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments. *The Journal of Supercomputing*, *66*(2), 783–811. doi:10.1007/s11227-013-0890-2

Wu, T., Lee, W., Lin, Y., Lin, Y., Chan, H. L., & Huang, J. (2012). Dynamic load balancing mechanism based on cloud storage. *Computing, Communications and Applications Conference*, 102-106. doi:10.1109/ComComAp.2012.6154011

Yiming, H., & Anthony, C. (2017). Scalable loop self-scheduling schemes for large-scale clusters and cloud systems. *International Journal of Parallel Programming*, *45*(3), 595–611. doi:10.1007/s10766-016-0434-5

Yingchi, M., Daoning, R., & Xi, C. (2013). Adaptive load balancing algorithm based on prediction model in cloud computing. *Proceedings of the 2nd International Conference on Innovative Computing and Cloud Computing (ICCC)*, 165–170.

Yu, X., Zhi, X., & Jing, Y. (2017). A load balance-oriented cost-efficient scheduling method for parallel tasks. *Journal of Network and Computer Applications*, *81*, 37–46. doi:10.1016/j.jnca.2016.12.032

Zeng, L., Veeravalli, B., & Zomaya, A. (2015). An integrated task computation and data management scheduling strategy for workflow applications in cloud environments. *Journal of Network and Computer Applications*, *50*, 39–48. doi:10.1016/j.jnca.2015.01.001

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18. doi:10.1007/s13174-010-0007-6

Zhilong, W., Sheng, X., Shubin, C., Zhijiao, X., & Zhong, M. (2016). A genetic-ant-colony hybrid algorithm for task scheduling in cloud system. *Proceedings of the International Conference on Smart Computing and Communication*, 183–193.

Mukund Kulkarni is working as a Senior Computer Programmer in the Institute of Petrochemical Engineering, Lonere-India. He had completed his B.E.(ECE) & M.E.(CSE) with Distinction from AIME and University of Pune, respectively. He is presently pursuing his Doctoral degree(Ph.D.) in the area of Cloud Computing and its Load Balancing strategies from Dr. Babasaheb Ambedkar Technological University, Loner-India. His research area includes-Cloud Based System Design and Analysis, Big Data Analytics, and Sensor Network Design.

Prachi Deshpande earned her Ph.D. from the Indian Institute of Technology Roorkee in 2016 in Cloud Computing Security Issues as a major. She is a recipient of the MHRD Research Fellowship during 2011-2016. She has authored/edited 03 books and several papers are published in International Journals and Conferences of International Repute. She has 15 years of academic experience. Her topic of interest are Cognitive Science and Big Data, Next generation fusion technologies and IoT Applications.

Sanjay L. Nalbalwar is working as a Professor and Head of the Department of Electronics and Telecommunication, Dr. B.A.T.U. Lonere-Raigad. He received his Ph.D. in Signal Processing from the Indian Institute of Technology, Delhi. His research interests include Signal Matched Filter Banks, Design, Characterisation, and Process Modelling. He has organized many international conferences in his field and is an active member of many professional bodies, e.g. the IETE (M123221), CSI (LM54758), ISTE (LM17072), IE (AM0820973), ISCEE (LM212), and IEEE(M80415950). He has published many papers in high-impact journals.

Anil Nandgaonkar is working as a Professor in the Department of Electronics and Telecommunication, Dr. B.A.T.U. Lonere-Raigad. He received his Ph.D. in Microwave Communication from Dr. Babasaheb Ambedkar Technological University, Lonere-India in the Year 2011. His research interests include Antenna Design, Wireless and MIMO Communications, and allied Networks. He has organized many international conferences in his field and is an active member of many professional bodies, e.g. the IETE, CSI, ISTE, and IEEE. He has published many papers in high-impact journals.