A Review of Scheduling and Resource Allocation Algorithms with a Load Balancing Approach in Cloud Computing

Yaser Ramzanpoor Foomeshi[1]
1- Department of Computer Engineering, Qaemshahr Branch,Islamic Azad University, Qaemshahr, Iran.
Email: yaser.ramzanpoor@iau.ac.ir

Received: 8 March 2023 Revised: 22 April 2023 Accepted: 20 May 2023

ABSTRACT:
Cloud computing is a distributed environment for providing services over the Internet. Load balancing of computing resources has emerged as a crucial element in this industry as a result of the expanding use of cloud computing and the expectations of customers to receive more services and better outcomes. The workload and system behavior of cloud computing are quite dynamic. And this can cause the resources in the data center to be overloaded. Ultimately, a load imbalance in some data center resources could result in increased energy use, decreased performance, and resource waste. Response time, expense, throughput, performance, and resource usage are among the quality of service indicators that load balancing can enhance. In this article, we analyze and evaluate scheduling and resource allocation methods with a view to load balancing, review the most recent approaches, and give a classification of these algorithms. Also, several significant problems and difficulties with cloud load balancing will be discussed in an upcoming study to create new algorithms.

KEYWORDS: Cloud Computing, Scheduling, Resource Allocation, Load Balancing.

1. INTRODUCTION

After several decades of evolution in computing capabilities, cloud computing has emerged as a result of the quick development of information technology and its wide range of applications. There were several difficulties and drawbacks with earlier computing technologies. By making the new technology more advanced, expandable, and interoperable with other technologies, the next technology aims to address or prevent those drawbacks. Cloud computing is related to many technologies, such as the Internet of Things [1, 2], Electronic Health [3], and vehicular ad hoc networks [2]. This paradigm hosts these services in sizable data centers to give customers access to more effective use of varied resources, software platforms, etc. [4]. Several models are used by cloud computing providers to offer their services. Three popular approaches include software, platforms, and infrastructure as a service [5]. Cloud computing faces many challenges in various fields, such as scalability, security, service quality management, resource scheduling, data center energy consumption, service availability, and appropriate load balancing [6, 7]. Load balancing is one of the main challenges and concerns in cloud environments [8]. The goal of load balancing is to maximize operational throughput while reducing cost and reaction time, enhancing performance, and conserving energy [9, 10]. Load allocation and re-allocation amongst available resources are done in this process.

Therefore, the load must be distributed across the resources in a cloud-based architecture such that each resource performs roughly the same amount of tasks at any point in time. Providing some solutions to balance the requests to deliver the application solution faster is a very important requirement.

Load balancing is done automatically in the infrastructure of all cloud service vendors. This allows customers to increase the number of processors or memory for their resources to match increased demand. These services are optional and depend on customers' business needs. Hence, the effectiveness of load balancing algorithms and techniques is essential for cloud computing success. Although cloud computing still faces many problems, load balancing is considered as one of the main challenges.

In this article, we reviewed the history of the study and examined recent advancements to aid future research in the design of load balancing algorithms. This article's goal is to examine the available techniques, outline their features, and explain both their benefits and drawbacks. The following are the article's primary goals:

- Study of current load balancing techniques.

Paper type: Research paper
DOI: 10.30486/MJTD.2023.1981169.1028
How to cite this paper: Y. Ramzanpoor Foomeshi, "**A Review of Scheduling and Resource Allocation Algorithms with a Load Balancing Approach in Cloud Computing**", *Majlesi Journal of Telecommunication Devices*, Vol. 12, No. 2, pp.95-103, 2023.

- Classifying the various load balancing techniques.
- Examining the benefits and drawbacks of each class's load balancing algorithms.
- Explaining the key points that can lead to the improvement of load balancing algorithms in future research.

The rest of the article is organized as follows. In section 2, the challenges of cloud-based load balancing algorithms are presented. The classification, criteria and policies of load balancing algorithms are explained in Section 3. Section 4 provides an overview of existing methods and some useful statistics. Open issues are described in Section 5. Finally, in section 6, conclusions are presented.

2. CHALLENGES OF LOAD BALANCING

The primary goal of load balancing is to evenly distribute the workload among the nodes so that no node is over- or underloaded. We first determine the primary problems and difficulties influencing the algorithm's performance before reviewing the current load balancing techniques. In this section, we discuss the difficulties that must be overcome in order to offer the best solution to the load balancing issue in cloud computing. The following points provide a summary of these difficulties:

2.1. Cloud Nodes Distribution Space

Certain load balancing techniques are made for lower scales, where things like network delay, communication delay, distance between distributed computing nodes, user-to-resource distance, etc. are not taken into account. With this class of algorithms, nodes in remote locations present a challenge because they are not suited for this environment. Consequently, it is important to take high latency into account while designing load balancing algorithms for nodes that are far away.

2.2. Heterogeneous Nodes

In the process of developing load balancing algorithms for the cloud environment, the initial assumption of the researchers was the homogeneity of the cloud nodes. But in cloud computing, the user's needs change dynamically, which requires their implementation on heterogeneous nodes to effectively use resources and minimize response time. Thus, it presents a challenge for researchers to develop effective load-balancing techniques for heterogeneous environments.

2.3. Management of Storage Resources and Data Replications

Users can store diverse data in the cloud without experiencing any access issues [11]. As the amount of cloud storage increases, redundant data storage is necessary for fast access and data consistency. Because replication points are required to store duplicate data, full data replication techniques are not very effective. However, partial replication methods can keep portions of the dataset on each node (with a given level of overlap) depending on each node's characteristics, such as processing power and capacity. But these techniques impose higher costs due to the need for more storage space [12]. It may not be necessary to have much redundancy in the data replication process, but there may be a problem with the data set's availability, which makes load balancing techniques more complex. As a result, an effective load balancing solution based on a partial replication system should be created to take into account the distribution of applications and related data.

2.4. Algorithm Complexity

Algorithms for cloud computing should be straightforward and uncomplicated to use. Increased implementation complexity creates a convoluted procedure that may result in issues and poor performance. Also, the delay brought on by complexity increases issues and lowers efficiency, as algorithms need more data and more communication for monitoring and control. Thus, the most straightforward load-balancing algorithms should be created.

2.5. Point of Failure

Some algorithms (centralized algorithms) can provide efficient and effective methods to solve load balancing in a particular pattern. However, these algorithms have a controller for the whole system. In such cases, if the central controller fails, the entire system will fail. A load balancing algorithm should be designed to overcome this challenge. The control of load balancing and data collection about different nodes should be designed to avoid having a single point of failure in the algorithm. As a result, it is necessary to create some distributed algorithms that do not depend entirely on a single node to function. However, in order for them to work correctly, they are much more complex and call for more control and coordination.

2.6. Scalability

The demand-based service provision feature in the cloud allows users to quickly increase or decrease computing resources whenever they want, according to their needs. To properly accommodate these changes, a decent load balancing algorithm should take into account the frequent changes in demands for processing power, storage, system structure, etc. [13].

2.7. Migration

Many virtual hosts can be built on a single physical

machine thanks to virtualization. These virtual computers have various settings and are independent in nature. If a physical host becomes overloaded or its load size is not proportional and is less than a threshold value, some virtual hosts should be migrated to other physical hosts using a load balancing approach. The process of cloud load balancing involves a crucial phase called migration, and the latter would be ineffective without the former. Virtual host migration and task migration are the two types of cloud migration. Virtual host migration, which can be divided into live and non-live migration, is the transfer of a virtual host from one physical host to another in order to solve the overload issue. Similarly, in task migration, moving tasks between virtual hosts is of two types. In the first type, the load balancing process is performed between virtual hosts located on each physical host, and in the second type, the load balancing on virtual hosts in different physical hosts is done. Several migration approaches have been presented in the research background. Effective load balancing results from an efficient migration mechanism. According to research studies, the migration process has evolved from virtual host migration to task migration, and the task migration method takes more time and costs more than virtual host migration [14, 15, 16, and 17].

2.8. Resource Allocation

An allocation policy is used to allocate resources. There are many scheduling and allocation policies in the research background. Along with speeding up execution using scheduling algorithms, allocation policy is also needed to properly manage resources and improve resource performance. The effectiveness of the scheduling algorithm and the allocation strategy affect how effective the load balancing algorithm is [18, 19, and 20].

3. MODEL, CLASSIFICATION, CRITERIA AND LOAD BALANCING POLICIES

Using physical resources by distributing them among virtual hosts is made possible by virtualization technology. A virtual host is a computer software implementation on which operating systems and programs can run. Virtual machines process user requests. The requests sent by users, who hail from all over the world, are sent at random. Virtual hosts must be assigned to requests in order to process them. Therefore, task allocation is an important issue in cloud computing. The quality of service will decline if the load balancing of the virtual hosts is improper, resulting in some machines being overloaded while others are idle or underutilized.

A lot of control and administration over user tasks and resources are needed since the cloud offers on-demand access to a pool of shared resources (such as servers, storage, and networks) [21]. An effective load balancer is required to distribute tasks to virtual hosts in accordance with their QoS requirements in order to handle user demands for available resources [22]. The user requests for the cloud are always changing, necessitating a dynamic environment to carry out activities. When a user submits a request to the cloud, a cloud service broker determines whether resources are available and confers with other brokers about resource performance and cost.

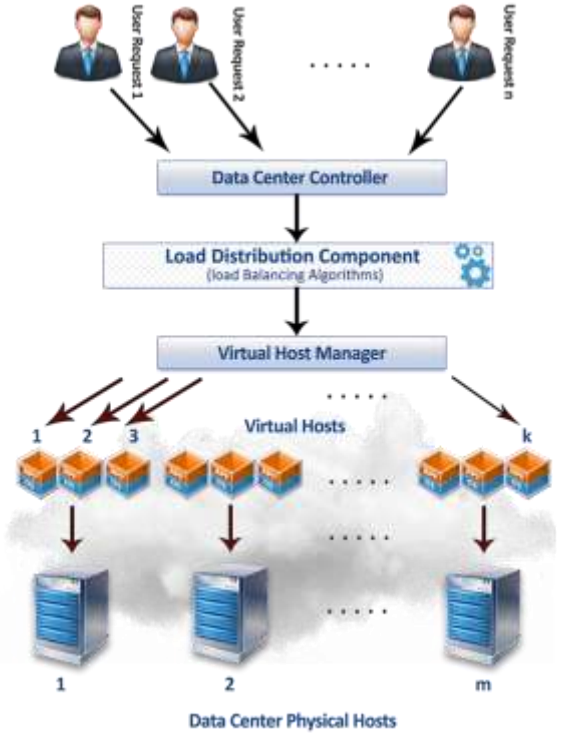

Fig. 1. Load distribution model.

Following a resource analysis, the broker sends the user request to the chosen data center, where the data center controller accepts it for further processing. To distribute tasks among virtual hosts, a load balancer on the server receives requests from a data center and forwards them to the server.

The load balancing model and workflow is shown in Figure 1. In this paradigm, the load-distributing component allocates user requests among virtual hosts by executing load-balancing algorithms. The load-distributing component determines which virtual host should be assigned to the following request. Task management falls within the purview of the data center controller. The load distributing component receives the tasks and executes the load balancing algorithm to assign the tasks to the appropriate virtual host based on the host's current load condition. A virtual host manager in the data center oversees all virtual hosts running on

physical servers. The load-distributing component considers both task scheduling and resource assignment. A competent load balancer reduces task response times while improving resource availability and utilization.

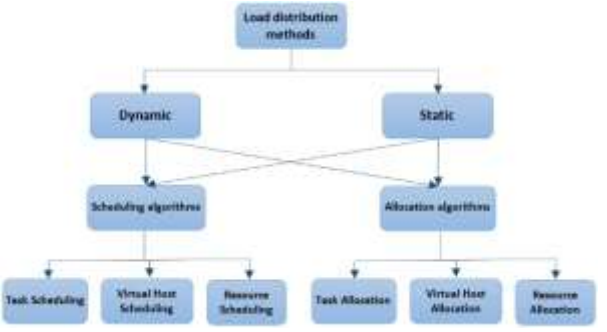

Fig. 2. Classifying load distributing techniques, based on the state of the system.

3.1. Classification of Load Balancing Methods

The initial load distribution process and system state are the main factors used to categorize load distributing methods, followed by feature. According to the virtual host's current state, cloud allocation and scheduling algorithms can either be static or dynamic, as shown in Figure 2. Static load distribution techniques adhere to a predetermined set of guidelines that are independent of the system's current condition. Static algorithms are rigid and need previous knowledge of resources like communication time, node memory and storage, node computing power, etc. Although this approach is clear and easy to use, it typically fails to recognize related actors, which leads to an inequitable distribution of resources [23]. While using the static technique, decisions are made without taking into account the system's present state. As a result, it is inappropriate for distributed systems that change dynamically and only works well when there is less load variation at the nodes. Static load distributing methods produce two types of results: optimal and near to optimal.

Dynamic techniques base their decisions on the system's current state and take it into account. The ability to move tasks from a host that is under a lot of pressure to one that is under a lot less load is the main benefit of these technologies. System performance is enhanced by the adaptability of dynamic load-distribution techniques. The processing nodes' load is continuously monitored via a dynamic approach. It calculates the workload of each node and redistributes the workload across the nodes at regular intervals based on the load and status information exchanged between the nodes. A node that is overloaded distributes the load to a node that is not as overloaded. Distributed and non-distributed categories are used to categorize dynamic load distribution techniques. Allocation and scheduling algorithms are essential to resource management and cloud performance monitoring, both of which have a positive impact on the user experience. The following three categories are used to group scheduling algorithms:

- Task Scheduling: It is a technique for distributing user tasks to computing resources so they can be completed.
- Resource scheduling: It is the process of planning, managing, and monitoring computing resources to execute tasks.
- Virtual Host Scheduling: On a physical host, it refers to the process of creating, deleting, and managing virtual hosts. This process is separate from managing virtual hosts during the migration process across hosts.

Likewise, allocation algorithms are divided into the following three categories:

- Task Allocation: The process of allocating a task to a resource that will carry it out.
- Resource Allocation: To finish a task, allocate a resource to it. Allocating tasks and allocating resources are mutually exclusive.
- Virtual Host Allocation: Assigning a virtual host to a user or a group of users.

3.2. Load Distribution Criteria

In this section, we examine load balancing criteria in cloud computing. To better utilize resources and improve performance, a load balancing component is necessary to distribute the computing load among available resources. Various load balancing methods and different criteria for applying these methods have been proposed by different researchers for greater user satisfaction and optimal utilization of resources. Researchers must make sure that all parameters are satisfied to their highest potential in order to improve the system's overall performance. These criteria are discussed in some sources such as [23, 24, 25, 26, 27, and 28], which are summarized as follows:

- Throughput: The amount of tasks or processes accomplished per unit of time in a system. Performance of the system improves with throughput.
- Make span: The maximum completion time, or the period during which resources are assigned to a user, is determined using this measure.
- Scalability: The capacity of an algorithm to balance the system's load consistently in light of the growing number of nodes.
- Fault Tolerance: the algorithm's potential to undertake load balancing in the event that some nodes or links fail.

- Migration Time: The time required to move a request or task from a node that is overloaded to a node that is underloaded. The cloud system performs better the faster the migration is completed.
- Degree of Imbalance: This measure describes the imbalance or changes between virtual machines.
- Efficiency: the effectiveness of the system after implementing a load balancing algorithm compared to other available load balancing methods.
- Response Time: The time taken to process the system's request in its entirety.
- Energy Consumption: the amount of energy consumed by all nodes. Load balancing helps to reduce energy consumption in all nodes.
- Resource Utilization: Evaluated to make sure that the cloud system's resources are all being used effectively. More efficient resource utilization will lower overall costs, as well as the cloud system's energy use and carbon emission rates.

3.3. Load Distribution Guidelines

The categories of static and dynamic load balancing methods were covered in the preceding sections. To complete tasks, dynamic algorithms use the following policies and the system state [26, 29, and 30].

- Selection Policy: Deciding which tasks ought to be moved from one node to another. Tasks are picked depending on the amount of overhead necessary for the migration, the number of non-local calls, and the time required to execute.
- Location Policy: This policy transfers tasks for processing to nodes that have a low or free load that are specified. After checking the availability of services required for task migration, the destination node is selected based on approaches such as exploration, negotiation, or randomly. Also, a node in the exploration technique examines other nodes in the system before choosing a destination. The negotiation strategy involves nodes negotiating load balancing with one another.
- Transfer Policy: This policy specifies the conditions for transferring tasks from a local node to other nodes. In accordance with this guideline, either all current tasks or the most recent task received should be transferred. In the first method, depending on the workload of each node, a rule is used to determine whether a task should be transferred or locally rescheduled for processing. In the second approach, among all input tasks, the last migration task is given.
- Information policy: In this policy, which is related to dynamic load balancing, all resource information is kept in the system for subsequent decisions. This policy determines when information is collected. Different methods for collecting information from nodes include agent-based methods, broadcast methods, and centralized sampling. Different information policies are: demand-based, periodic and status-based.

Following are the relationships between several policies: First, the transfer policy processes tasks that are entered into a system. Following processing, a decision is made regarding the transfer of tasks to a remote node for load balancing. The task is processed locally if it is not eligible for transfer. The location policy is turned on for tasks that need to be shifted in order to locate a node that can handle them. The task is queued for local processing if a remote node is not available for execution. The information policy offers data to the transfer and location policies to aid in decision-making.

4. REVIEW OF LOAD DISTRIBUTION METHODS

In this section, we examine and categorize the literature pertaining to current approaches to load balancing in cloud systems using various standards. To classify them for this reason, we looked into a number of journals and conference pieces. The classification procedure employs a top-down strategy. The algorithm's current state and the load-balancing function are two additional classification factors. Load balancing algorithms can be static, dynamic, or hybrid, depending on the system state. Algorithms for load balancing are categorized as scheduling and allocation algorithms based on the attributes they employ. In Table 1, we have organized the available techniques in accordance with Figure 2.

Table 1. Cloud load balancing methods in research background.

Ref.	Algorithm used	Algorithm state	Property	Load balancing type	Method	Advantages	Disadvantages
[31]	Artificial bee colony + reinforcement learning	Hybrid	Task scheduling	Virtual host load balancing	Meta-heuristic	Optimal use of resources High throughput of the virtual machine Reduce the Makespan	Failure to guarantee optimal performance regarding the priority implementation of large tasks Inefficiency in all datasets
[32]	Non-Classical	Dynamic	Task scheduling	Virtual host load balancing	Non-classical,	Reduce execution time Reduce the Makespan Optimal use of resources	Improved a limited number of service level agreement parameters

					deterministic		
[33]	Non-Classical	Dynamic	Task scheduling	Virtual host load balancing	Non-classical, deterministic	Reduce response time Increase the utilization of hosts Reducing the overall cost of migration Reduce the Makespan	High communication cost Not suitable for dependent tasks
[34]	Wall algorithm + QOHC	Hybrid	Task scheduling	Virtual host load balancing	Optimization	Reduce response time Reducing the number of migrations Increase the error accuracy rate	Improving a limited number of service quality parameters
[35]	Non-Classical	Dynamic	Task scheduling	Task load balancing	Non-classical, deterministic	Better overall time Better resource utilization Less waiting time Less execution time	The deadline for the task is not considered Less error tolerance Less energy efficient
[36]	Classical and linear programming	Dynamic	Task scheduling	Task load balancing	Optimization (based on linear programming)	Better overall time Better resource utilization	Decreasing the quality of service
[37]	Genetics and Min-Min	Hybrid	Task scheduling	Task load balancing	Heuristic (evolutionary)	Better scalability Less response time Lower execution cost	Low utilization of resources Lower degree of balance
[38]	Lamarck evolution + BFO	Hybrid	Resource Scheduling	CPU load balancing	Optimization	Low virtual machine failure Improved execution time Less transfer time	Low scalability and throughput Less resource utilization
[39]	Genetics	Dynamic	Virtual machine scheduling	Virtual host load balancing	Meta-heuristic	Less response time Less task completion time	Low throughput Low scalability Low degree of balance Low resource utilization
[40]	Honey Bee	Dynamic	Task scheduling	Task load balancing	Optimization	Low response time Low runtime Low execution cost	Low throughput and scalability Low degree of balance Low resource utilization
[41]	Bat	Dynamic	Task and resource scheduling	Task and resource load balancing	Optimization	Less execution time Low execution cost	High Completion time Low throughput Low resource utilization Less energy efficient
[42]	Active monitoring	Dynamic	Virtual machine and task scheduling	Virtual host and task load balancing	Heuristic	High scalability Low response time High resource utilization	Low throughput Low fault tolerance High completion time
[43]	Active monitoring	Dynamic	Resource Scheduling	Resource load balancing	Heuristic	Low overhead Low completion time High resource utilization	Low throughput Less energy efficient

5. OPEN ISSUES IN THE FIELD OF LOAD BALANCING

There are still a lot of problems and difficulties with load balancing that will need to be discussed in the future. We have identified some directions for further study on cloud augmentation from the literature review. Issues including satisfaction with service, agreements on service levels, resource provisioning, load distribution, etc., must be taken into account in order to maintain cloud performance. To uphold service level agreements and service quality standards with cloud service providers, numerous resources are needed. Service level agreements are designed and implemented based on service quality rules, and if there is any violation of the service level agreement, the service provider must pay a fine. The amount of time users and service providers interact is reduced through automatic resource provisioning. The proper use of allotted resources necessitates load distribution in order to uphold service level agreements and quality of service. A load distribution algorithm aids in maximizing throughput while utilizing the fewest resources possible. There are many load distribution techniques that take into account different variables like performance, response time, execution time, task migration time, and resource usage. To increase the overall performance of data centers, no solution has taken all load balancing factors into account. The following are significant problems and difficulties with cloud load balancing:

- In a heterogeneous cloud environment, managing resources and apps is a highly challenging process.
- A well-designed approach to resource allocation is an important issue for service providers to continuously improve resource utilization.
- Component-level testing and a checkpoint-based strategy can increase the robustness of a distributed environment.
- Data lock-in can become a problem when there is a need to move workloads to another cloud provider, so it is necessary to provide policies to solve such problems.

6. CONCLUSION

Researchers have focused on load balancing as a significant issue in cloud computing. This article provides an overview of recent research on load balancing concerns and obstacles. Based on this study, a thorough analysis of numerous load distribution techniques has been carried out, taking various parameters into account. These load distribution techniques have been divided into two groups: state-based techniques and attribute-based techniques. We have covered the benefits, drawbacks, ideas, and difficulties of each category of these techniques. There are several load distribution methods that support the majority of the load balancing requirements, offer the best resource use, and speed up response. To raise the system's performance in the future, we must, however, improve the approaches we currently use. Load balancing techniques emphasize green computing, energy efficiency, and workload management in addition to bettering system performance and resource utilization, which calls for the creation of new algorithms. This paper offers an overview of the various load balancing techniques now in use, which will be helpful for researchers in identifying research topics in the field.

REFERENCES

[1] A. Botta, W. De Donato, V. Persico and A. Pescap, "**Integration of Cloud computing and Internet of Things: A survey**," *Future Generation Computer Systems* , vol. 56, pp. 684-700, 2016.

[2] S. Distefano, G. Merlino and A. Puliafito, "**A utility paradigm for IoT:The sensing clou**d," *Pervasive and Mobile Computing,* vol. 20, pp. 127-144, Jul. 2015.

[3] O. Diallo, J. J. P. C. Rodrigues, M. Sene and J. Niu, "**Real-time query processing optimization for cloud-based wireless body area networks**," *Information Sciences,* vol. 284, pp. 84-94, 2014.

[4] T. DeStefano, R. Kneller and J. Timmis, "**Cloud Computing and Firm Growth**," 2020. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3618829. [Accessed 15 July 2021].

[5] Y. Chen, X. Li and F. Chen, "**Overview and analysis of cloud computing research and application**," in *International Conference on E-Business and E-Government (ICEE)*, Shanghai, China, 6–8 May 2011.

[6] K. Rajwinder and P. Luthra, "**Load balancing in cloud computing**," in *International Conference on Recent Trends in Information, Telecommunication and Computing(ITC)*, 2012.

[7] R. R. Malladi, "**An approach to load balancing in cloud computing**," *international journal of research in engineering and technology,* vol. 4, no. 5, pp. 3769-3777, 2015.

[8] Y. Jadeja and K. Modi, "**Cloud Computing - Concepts, Architecture and Challenges**," in *International Conference on Computing, Electronics and Electrical Technologies[ICCEET]*, 2012.

[9] S. Goyal and M. .. K. Verma, "**Load balancing techniques in cloud computing environment: a review**," *international journal of advanced research in computer science and software engineering,* vol. 6, no. 4, 2016.

[10] P. Singh, P. Baaga and S. Gupta, "**Assorted load-balancing algorithms in cloud computing: a survey**," *International Journal of Computer Applications,* vol. 143, no. 7, 2016.

[11] Y. W. Tin, T. L. Wei, S. L. Yu, S. L. Yih, L. C. Hung and S. H. Jhih, "**Dynamic load balancing mechanism based on cloud storage**," in *IEEE International Conference on Computing, Communications and Applications*, 2012.

[12] I. Foster, Y. Zhao, I. Raicu and S. Lu, "**Cloud Computing and Grid Computing 360-degree compared**," in *Grid Computing Environments Workshop*, 2008.

[13] R. Soumya and D. S. Ajanta, "**Execution analysis of load balancing algorithms in cloud computing environment**," *International Journal on Cloud Computing: Services and Architecture,* vol. 2, no. 5, pp. 1-13, 2012.

[14] M. Noshy, A. Ibrahim and H. A. Ali, "**Optimization of live virtual machine migration in cloud computing: a survey and future directions**," *Journal of Network and Computer Applications,* pp. 1-10, 2018.

[15] L. Gkatzikis and I. Koutsopoulos, "**Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems**," *IEEE Wireless Communications,* vol. 20, no. 3, pp. 24-32, 2013.

[16] P. Jamshidi, A. Ahmad and C. Pahl, "**Cloud migration research: a systematic review**," *IEEE Transactions on Cloud Computing,* vol. 1, no. 2, pp. 142-157, 2013.

[17] E. Shamsinezhad, A. Shahbahrami, A. Hedayati, A. K. Zadeh and H. Banirostam, "**Presentation methods for task migration in cloud computing by combination of Yu router and post-copy**," *International Journal of Computer Science Issues,* vol. 10, no. 4, 2013.

[18] S. K. Mishra, D. Puthal, B. Sahoo, S. K. Jena and M. S. Obaidat, "**An adaptive task allocation technique for**

green cloud computing," *Journal of Supercomputing*, pp. 1-16, 2017.

[19] A. H. Ibrahim, H. E. D. M. Faheem, Y. B. Mahdy and A. R. Hedar, "**Resource allocation algorithm for GPUs in a private cloud**," *International Journal of Cloud Computing*, vol. 5, no. 1, pp. 45-56, 2016.

[20] M. Jebalia, L. A. Ben, M. Hamdi and S. Tabbane, "**An overview on coalitional game-theoretic approaches for resource allocation in cloud computing architectures**," *International Journal of Cloud Computing*, vol. 4, no. 1, p. 63–77, 2015.

[21] F. Z. Ling, V. Bharadwaj and Y. Z. Albert, "**An integrated task computation and data management scheduling strategy for workflow applications in cloud environments**," *Journal of Network and Computer Applications*, vol. 50, pp. 39-48, 2015.

[22] B. Aditya and R. K. Challa, "**Efficient multistage bandwidth allocation technique for virtual machine migration in cloud computing**," *Journal of Intelligent & Fuzzy Systems*, vol. 36, pp. 1-14, 2018.

[23] L. C. Shang, Y. C. Yun and H. K. Suang, "**CLB: A novel load balancing architecture and algorithm for cloud services**," *Computers and Electrical Engineering*, vol. 58, p. 154–160, 2017.

[24] A. S. Milani and N. J. Navimipour, "**Load balancing mechanisms and techniques in the cloud environments: systematic literature review and future trends**," *International Journal of Cloud Computing*, vol. 71, pp. 86-98, 2016.

[25] M. M. Abdullahi, N. M. A. Asri and S. M. Abdulhamid, "**Symbiotic organism search optimization based task scheduling in cloud computing environment**," *Future Generation Computer Systems*, vol. 56, pp. 640-650, 2015.

[26] Y. D. Eman and M. Y. Shyan, "**A small world based overlay network for improving dynamic loadbalancing**," *Journal of Systems & Software*, vol. 107, p. 187–203, 2015.

[27] F. Ramezani, L. Jie and K. H. Farookh, "**Task-based system load balancing in cloud computing using particle swarm optimization**," *International Journal of Parallel Programming*, vol. 42, no. 5, p. 739–754, 2014.

[28] M. Abdolhamid, M. Shafi'i and M. B. Bashir, "**Scheduling techniques in on-demand grid as a service cloud: a review**," *Journal of Theoretical and Applied Information Technology*, vol. 63, no. 1, pp. 10-19, 2014.

[29] T. K. Ravi and K. R. Vuyyuru, "**Performance analysis of load balancing techniques in cloud computing environment**," in *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT'15)*, 2015.

[30] S. Kumar and D. H. Rana, "**Various dynamic load-balancing algorithms in cloud environment: a survey**," *International Journal of Computer Applications*, vol. 129, no. 6, 2015.

[31] B. Kruekaew and W. Kimpan, "**Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning**," *IEEE Access*, vol. 10, pp. 17803-17818, 2022.

[32] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah and M. A. Alzain, "**A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications**," *IEEE Access*, vol. 9, pp. 41731-41744, 2021.

[33] A. Semmoud, M. Hakem, B. Benmammar and J. Charr, "**Load balancing in cloud computing environments based on adaptive starvation threshold**," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 11, pp. 1-14, 2020.

[34] S. Anuradha and P. Kanmani, "**fault tolerant load balancing with quadruple osmotic hybrid classifier and whale optimization for cloud computing**," *Scalable Computing: Practice and Experience*, vol. 23, no. 4, p. 321–338, 2022.

[35] M. Kumar and S. C. Sharma, "**Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing**," in *Computer Science 115*, 2017.

[36] M. Adhikari and T. Amgoth, "**Heuristic-based load-balancing algorithm for IaaS cloud**," *Future Generation Computer Systems*, vol. 81, pp. 156-165, 2018.

[37] S. S. Rajput and V. S. Kushwah, "**A genetic based improved load balanced min-min task scheduling algorithm for load balancing in cloud computing**," in *8th international conference on Computational Intelligence and Communication Networks (CICN)*, 2016.

[38] L. Tang, Z. Li, P. Ren, J. Pan, Z. Lu, J. Su and Z. Meng, "**Online and offline based load balance algorithm in cloud computing**," *Knowledge-Based Systems*, vol. 138, pp. 91-104, 2017.

[39] M. Vanitha and P. Marikkannu, "**Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines**," *Computers and Electrical Engineering*, vol. 57, p. 199–208, 2017.

[40] S. K. Vasudevan, S. Anandaram, A. J. Menon and A. Aravinth, "**A novel improved honey bee based load balancing technique in cloud computing environment**," *Asian Journal of Information Technology*, vol. 15, no. 9, p. 1425–1430, 2016.

[41] S. Sharma, A. K. Luhach and S. S. Abdhullah, "**An optimal load balancing technique for cloud computing environment using bat algorithm**," *indian journal of science and technology*, vol. 9, no. 28, 2016.

[42] A. M. Tripathi and S. Singh, "**PMAMA: priority-based modified active monitoring load balancing algorithm in cloud computing**," *Journal of Advanced Research in Dynamical and Control Systems*, pp. 809-823, 2018.

[43] A. N. Singh and S. Prakash, "**WAMLB: weighted active monitoring load balancing in cloud computing**," in *Aggarwal, V., Bhatnagar, V., Mishra, D. (eds) Big Data Analytics. Advances in Intelligent Systems and Computing, vol 654. Springer*, Singapore, pp 677–685, 2018.