ORIGINAL RESEARCH ARTICLE | UHD JOURNAL OF SCIENCE AND TECHNOLOGY

Limitations of Load Balancing and Performance Analysis Processes and Algorithms in Cloud Computing

Asan Baker Kanbar[1,2]*, Kamaran Faraj[3,4]

[1]Technical College of Informatics, Sulaimani Polytechnic University , Sulaimani 46001, Kurdistan Region, Iraq, [2]Department of Computer Science,Cihan University Sulaimaniya, Sulaimaniya 46001, Kurdistan Region, Iraq, [3]Department of Computer Science, University of Sulaimani, Sulaimani, 46001, Kurdistan Region, Iraq, [4]Department of Computer Engineering, Collage of Engineering and Computer Science, Lebanse Frence University, Erbil, Iraq

ABSTRACT

In the modern IT industry, cloud computing is a cutting-edge technology. Since it faces various challenges, the most significant problem of cloud computing is load balancing, which degrades the performance of the computing resources. In earlier research studies, the management of the workload to address all resource allocation challenges that caused by the participation of a large number of users has received important attention. When several people are attempting to access a given web application at once, managing all of those users becomes exceedingly difficult. One of the elements affecting the performance stability of cloud computing is load balancing. This article evaluates and discusses load balancing, the drawbacks of the numerous methods that have been suggested to distribute load among nodes, and the variables that are taken into account when determining the best load balancing algorithm.

Index Terms: Cloud Computing, Load Balancing, Task Scheduling, Resource Allocation, Task Allocation, Performance Stability

1. INTRODUCTION

Cloud computing is a new technology for a large-scale environments. Hence, it faces many challenges and the main problem of cloud computing is load balancing which lowering the performance of the computing resources [1]. Management is the key to balancing performance and management costs along with service availability. When Cloud Data Centres (CDCs) are configured and utilized effectively, they offer huge benefits of computational power while reducing cost and saving energy. Cloud computing has three types of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Fundamental resources can be accessed through IaaS. PaaS provides the application runtime environment, besides development and deployment tools. SaaS enables the provision of software applications as a service to end users. Virtual entities are created for all hardware infrastructure elements. Virtualization is a technique that allows multiple operating systems (OSs) to coexist on a single physical machine (PM). These OSs are separated from one another and from the underlying physical infrastructure by a special middleware abstraction known as virtual machine (VM). The software that manages these multiple VMs on PM is known as the VM kernel [2]. With the help of virtualization technology, CDCs are able to share a few HPC resources and their services among many users, but virtualization

Access this article online

DOI:10.21928/uhdjst.v7n1y2023.pp71-77	**E-ISSN:** 2521-4217 **P-ISSN:** 2521-4209

Corresponding author's e-mail: Asan Baker Kanbar, Assistant Lecturer, Department of Computer Science, Cihan University Sulaimaniya, Sulaimaniya 46001, Kurdistan Region, Iraq, Asan. E-mail: asan.baker@sulicihan.edu.krd

Received: 21-11-2022 Accepted: 02-03-2023 Published: 18-03-2023

increases the complexity of resource management. Task scheduling is one of the key issues considered for efficient resource management. It aims to allocate incoming task(s) to available computing resources, and it belongs to the set of NP-class problems. Therefore, heuristic and meta-heuristic-based approaches are commonly used to generate scheduling solutions while optimizing one or more goals such as makespan, resource utilization, number of active servers, through-put, temperature effects, energy consumption, etc. Customers in the cloud can access resources at any time through the web and only pay for the services they use. With the dramatic increase in cloud users, decreasing task completion time is beneficial for improving user experience. The primary goals of task scheduling are to reduce task completion time and energy consumption while also improving resource utilization and load balancing ability [3]. Improving load balancing ability contributes to fully utilizing VMs to prevent execution efficiency from decreasing due to resource overload or waste caused by excessive idle resources. Various algorithms have been proposed to balance the load between multiple cloud resources, but there is currently no algorithm that can balance the load in the cloud without degrading performance. Load balancing is a method used to improve the performance of networking by distributing the workload among the various resources involved in computing network tasks. The load here can be processor capacity, memory, network load, etc.

Load balancing optimizes resource usage, reduces response time, and avoids system overload by distributing the load across several components. Many researchers are working on the problem of load balancing, and as a result of their research, many algorithms are proposed every day. In this paper, we overview some of the optimistic algorithms that have shown some improvement in load balancing and increased the level of performance. Besides we will also show the limitations of these algorithms.

2. LOAD BLANCING IN CLOUD COMPUTING

Load balancing is performed for resource allocation and managing load in each data center, as illustrated in Fig. 1. Load balancing in a cloud computing environment has a significant impact on performance; good load balancing can make cloud computing more efficient and improve user satisfaction. Load balancing is a relatively new technology that allows networks and resources to deliver maximum throughput with a minimum response time.

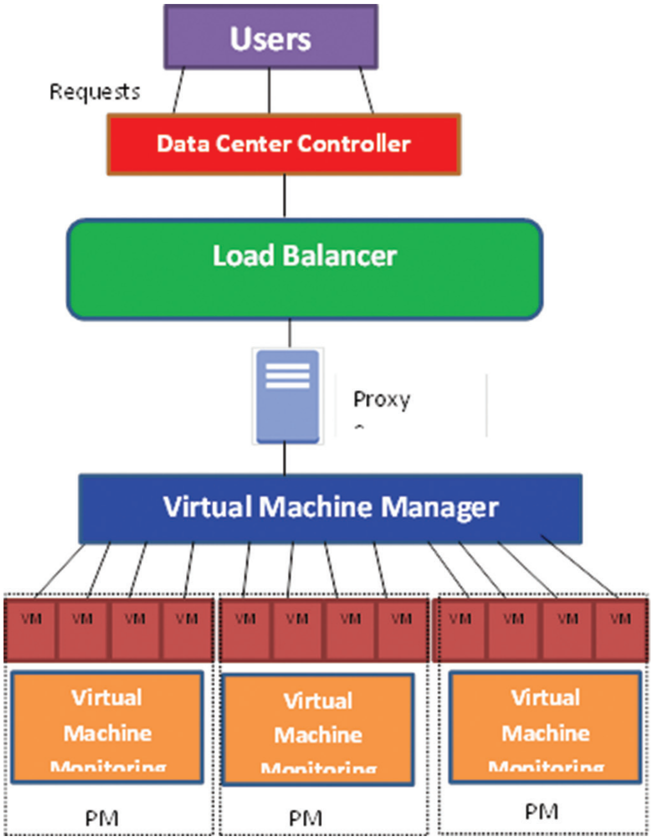

Fig. 1. Model of load balancing.

Good load balancing helps to optimize the use of available resources, thus minimizing resource consumption. By sharing traffic between servers, you can send and receive data without experiencing significant delays. Different types of algorithms can be used to help reduce traffic load between available servers. A basic example of load balancing in everyday life can be related to websites. Without load balancing, users may experience delays, timeouts, and the system may become less responsive. By dividing the traffic among servers, data can be sent and received without significant delay.

Load balancing is done using a load balancer (Fig. 2), where each incoming request is redirected and transparent to the requesting client.

Based on specified parameters such as availability and current load, the load balancer uses various scheduling algorithms to find which server should handle the request and sends the request to the selected server. To make a final decision, the load balancer obtains information about the candidate server's state and current workload to validate its ability to respond to this request [4].

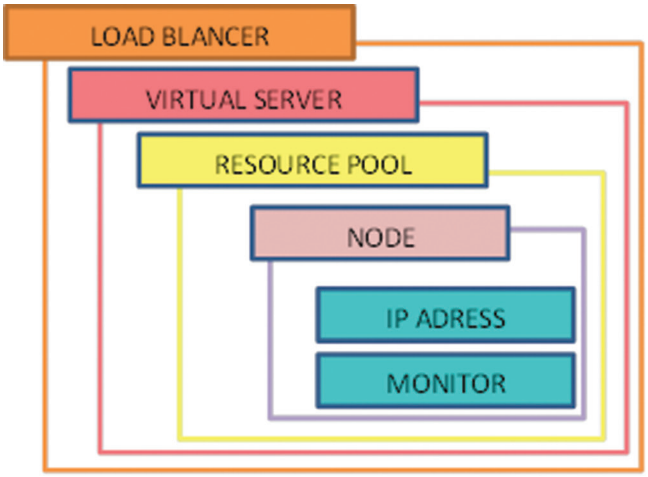

Fig. 2. Load Balancer.

3. CHALLENGES IN CLOUD COMPUTING LOAD BALANCING

Before we could review the current load balancing approaches for cloud computing, we need to identify the main issues and challenges involved and that could affect how the algorithm would perform. Here, we discuss the challenges to be addressed when attempting to propose an optimal solution to the issue of load balancing in Cloud Computing. These challenges are summarized in the following points.

3.1. Cloud Node Distribution

Many algorithms have been proposed for load balancing in cloud computing; among them, some algorithms can provide efficient results in small networks or networks with nodes close to each other. Such algorithms are not suitable for large networks because they cannot produce the same efficient results when applied to larger networks. The development of a system to regulate load balancing while being able to tolerating significant delays across all the geographical distributed nodes is necessary [5]. However, it is difficult to design a load balancing algorithm suitable for spatially distributed nodes. Some load-balancing techniques are designed for a smaller area where they do not consider the factors such as network delay, communication delay, distance between the distributed computing nodes, distance between user and resources, and so on. Nodes located at very distant locations are a challenge, as these algorithms are not suitable for this environment. Thus, designing load-balancing algorithms for distantly located nodes should be taken into account [6]. It is used in large-scale applications such as Twitter and Facebook. The DS of the processors in the cloud computing environment is very useful for maintaining system efficiency and handling fault tolerance well. The geographical distribution has a significant impact on the overall performance of any real-time cloud environment.

3.2. Storage/Replication

A full replication algorithm does not take efficient storage utilization into account. This is because all replication nodes store the same data. Full replication algorithms impose higher costs because more storage capacity is required. However, partial replication algorithms may store partial datasets in each node (with some degree of overlap) depending on each node's capabilities (such as power and processing capacity) [7]. This can lead to better usability, but it increases the complexity of load balancing algorithms as they try to account for the availability of parts of the dataset on different cloud nodes.

3.3. Migration Time

Cloud computing follows a service-on-demand model, which means when there is a demand for a resource, the service will be provided to the required client. Therefore, while providing services based on the needs of our customers, we sometimes have to migrate resources from remote locations due to the unavailability of nearby locations. In such cases, the time of migration of the resources from far locations will be more which will affect system performance. When developing algorithms, it is important to note that resource migration time is an important factor affecting system performance.

3.4. Point of Failure

Controlling the load balancing and collecting data about the various nodes must be designed in a way that avoids having a single point of failure in the algorithm. If the algorithm's patterns are properly created, they can also help provide effective and efficient techniques to address load balancing problems. Using a single controller to balance the load is a major difficulty because, failure might have severe consequences and lead to overloading and under-loading issues. This difficulty must be addressed in the design of any load balancing algorithm [8]. Distributed load balancing algorithms seem to offer a better approach, but they are much more complex and require more coordination and control to work properly.

3.5. System Performance

This does not mean that if the complexity of the algorithm is high, then the system performance will be very high. Any time a load balancing algorithm must be simple to implement and easy to operate. If the complexity of the algorithm is high, then the implementation cost will also be higher and even after implementing the system, performance will be decreased due to the increased delays in the functionality of the algorithm.

3.6. Algorithm Complexity

In terms of implementation and operation, the load balancing algorithm is preferably not that complicated. Higher implementation complexity will lead to more complex procedures, which can lead to negative performance issues Furthermore, when the algorithms require more information and higher communication for monitoring and control, delays would cause more problems and reduce efficiency. Therefore, to reduce overhead on cloud computing services, load-balancing algorithms should be as simple and effective as possible [9].

3.7. Energy Management

A load balancing algorithm should be designed in such a way that the operational cost and energy consumption of the algorithm must be low. Increasing energy consumption is one of the biggest issues facing cloud computing today. Even though using energy efficient hardware architectures which slows down the processor speed and turn off machines that are not under use the energy management is becoming difficult. Hence, to achieve better results in energy management, the load balancing algorithm should be designed according to Energy Aware Job Scheduling methodology [10].

3.8. Security

Security is one of the problems that cloud computing has as its top priority. The cloud is always vulnerable in one way or the other way to security attacks like DDOS attacks, etc. While balancing the load there are many operations that take place like VM migration, etc. at that time there is a high probability of security attacks. Hence, an efficient load balancing algorithm must be strong enough to reduce security attacks but should not be vulnerable.

4. RELATED WORKS AND LIMITATIONS OF USED ALGORITHMS AND PROCESSES

The author [11] proposed a hybrid optimization algorithm for load balancing. This is firefly optimization and enhanced multi-criteria based on the Particle Swarm Optimization (PSO) algorithm called (FIMPSO). To initialize the population in PSO, the Firefly algorithm is used, since it gives the optimal solution. Only two parameters are considered here, such as task arrival time and task execution time. The results are executed, taking into account parameters such as run time, resource consumption, reliability, makespan, and throughput. Limitations: Hybrid algorithms require high latency to run. In particular, PSO falls into a local optimum problem when processing a large number of requests, and the convergence speed is low. Overloading occurs here because more iteration is needed to achieve the optimal solution. In the paper [12], propose the use of three-layer cooperative fog to reduce bandwidth cost and delay in cloud computing environments, this article discusses the composite objective function of bandwidth cost reduction and load balancing, where we consider both link bandwidth and server CPU processing levels. Assign weights to every objective of the composite objective function to determine priority. The minimum bandwidth cost has a higher priority and runs first on Layer1 fog. However, the load balancer gets the priority it used to reduce latency. The MILP (Mixed-Integer Linear Programming) algorithm is used to minimize the composite objective function. Two types of resources are used, one is a network resource (bandwidth) and the other is a server resource (CPU processing layer). Limitations: This work is not suitable for real-time applications, because it takes a high execution time for selecting the bandwidth and CPU. It only focuses on reducing bandwidth costs and load balancing, so it takes a long time to find the optimal solution. Priority is based on the minimum bandwidth utilization in a large scale environments, many regions are used the minimum bandwidth utilization so congestion is occurring; it takes much time to execute the task, which also reduces the QoS values. Author [13] Task offloading and resource allocation were proposed for IoT fog cloud architecture based on energy and time efficiency. The ETCORA algorithm is used to improve energy efficiency and request completion time. It performs two tasks. One is computational offload selection and the other is transmitting power allocation. Three layers are presented in this work. The first tier contains some IoT devices. The second tier is the fog tier, which consists of fog servers and controllers located in different geographic locations. The third tier is the cloud tier, which consists of cloud servers. However, the entire task is outsourced within the fog layer, so the fog layer is also overloaded. In many regions, a request is sent to the users at a certain time, the fog layer cannot control the load balancing. All users in the region access the cloud server, which triggers load balancing. The author [14] proposed using probabilistic load balancing to avoid congestion due to VM migration and also to minimize congestion across migrations. For VM migration, this paper takes into account the distance between the source PM and the destination PM. The architecture features a VM migration controller, stochastic demand forecasting, hotspot detection, and VMs, PMs. Load balancing is addressed by profiling resource demand, hotspot demand, and hotspot migration. Resource demand profiling tracked the following: VM resource utilization on CPU, memory, network bandwidth, and disk I/O. It is used to update the periodic information to the balancer. For discovering the hotspot they periodically change the resource allocation status from the VMs and PMs' Resource demands. The hotspot migration process uses the hotspot migration algorithm.

Author [15] proposed a static load balancing algorithm totally based on discrete PSO for distributed simulations in cloud computing. For static load balancing, adaptive pbest discrete PSO (APDPSO) is used. PSO updates particle velocity and position vectors. The distance metric is used to update the velocity and position vectors from the pbest and gbest values. Non-Dominated Genetic Sorting Algorithm II (NSGA II) is one of the evolutionary algorithms that preserves the optimal solution. For each iteration, NSGA II considers three important processes selection, mutation and crossover. However, PSO suffers from local optima and poor convergence when handling a large number of requests, resulting in increased latency. In paper [16], the author proposed multi-goal task scheduling based on SLA and processing time, which is suitable for cloud environments. This article proposes two scheduling algorithms called the Threshold Based Task Scheduling (TBTS) algorithm and the Service Level Agreement Load Balancing (SLA-LB) algorithm. TBTS scheduled a task for a batch TNTS threshold (expected time of completion) generated from ETC. SLA-LB is based on an online model that dynamically schedules a task based deadline and budget criteria. SLA-LB is used to find the required system to reduce the makespan and increasing the cloud usage. This paper discuses following performance metrics such as makespan, penalty, achieve cost, and VM utilization. The results are shows that the proposed method is superior when compared to existing algorithm in terms of both scalability and VMs. However, the value of threshold is based on the completion time, if assuming that completion time is increased, the threshold value will be burst. It reduces the SLA and QoS values. Author [17] proposed a multi-agent system for dynamic consolidation of VMs with optimized energy efficiency in cloud computing. This proposed system eliminates the centralized failure, so that, the decentralized server presented with Gossip Control (GC) with a multi-agent framework the GC has two protocols: Gossip and Contract Network Protocol. With the assist of GC developed DVMS (Dynamic VM Consolidation) compared two sercon strategies centralized strategy and an ECO Cloud distributed Strategy. Sercon is used to minimize server count and VM migration. During integration, Eco Cloud considers two processes: First is the migration procedure and the second is the allocation procedure GC-based strategy works best for SLA violations and power consumption. In paper [18] using cloud theory for wind power uncertainty, the author proposed a multi-objective feeder reconfiguration problem. Proposed used the cloud theory properties of qualitative–quantitative bidirectional transmission to solve the problem of multi-objective feeder reconfiguration with the backward and forward cloud generator algorithm, Proposed system used a fuzzy decision-making algorithm to get the best solution. Authors in [19] proposed an approach to perform cloud resource provisioning and scheduling based on metaheuristic algorithms. To design the supporting model for the autonomic resource that schedules the applications effectively, the binary PSO (BPSO) algorithm is used. This work consists of three consecutive phases as user-level phase, the cloud provisioning and scheduling phase, and the infrastructure level phase. Finally, experimental evaluation was performed by modifying the BPSO algorithm's transfer function to achieve high exploration and exploitation. Authors in [20] introduced intelligent scheduling and provisioning of resource methods for cloud computing environments. This work overcomes the existing problems of poor quality of service, increased execution time, and high costs during service. Existing problems are addressed by an intelligent optimization framework that schedules jobs for users using spider monkey optimization. This will result in faster execution time, lower cost, and better QoS for the user. Here, the job is scheduled by the Spider-Monkey optimization algorithm. However, Job sensitivity (i.e., risk or non-risk) is not considered, resulting in poor QoE. Author [21] proposed to consider a multi-objective optimization for energy-efficient allocation of virtual clusters in CDCs. This research paper describes four optimization goals related to VC and data centers: Availability, power consumption, average resource utilization, and resource load balancing. The architecture contains three-layers which are the core layer, aggregation layer, and edge layer. In the core, layer contains a core switch, which is classified into many aggregated switches. The aggregation layer contains an edge switch and PMs. The edge switches are connected to the PMs. The PMs are connected to the VMs cluster. If the edge switch is failed, it will not access the PMs and VMs cluster.

5. DISCUSSION

The existing works were addressed the issues of task scheduling and load balancing in IoT-fog-cloud environment. Many of the

TABLE 1: Summary of Related Works and Limitations of Used Algorithms and processes

References	Task classification	Task scheduling	Load balancing	Task allocation	Algorithm/process used	limitations
[11]	x	x	✓	x	Firefly improved multi-objective particle swarm optimization (FIMPSO).	• More number of iterations and low convergence rate Performs
[12]	x	x	✓	x	When traffic exceeds a region's capacity, the fog layer performs load balancing.	• Long execution time is observed • Task requirements were not considered resulting in SLA violation.
[13]	x	x	✓	✓	ETCORA algorithm for energy efficient offloading	• Constraints exist when it comes to increasing the scalability of tasks in a particular region.
[14]	x	x	✓	x	Resource requirements for each task are profiled and offloaded.	• Throughput is affected because of frequent migrations • High migration time because of long congestion in link.
[15]	x	x	✓	x	Load balancing based on Adaptive Pbest Discrete PSO (APDPSO) to reduce communication costs	• Blind spot problem • Throughput is affected because of frequent migrations
[16]	x	✓	✓	x	TBTS and SLA-LB to dynamically perform scheduling and load balancing.	• Ineffective determination of threshold value increases complexity
[17]	x	x	✓	x	Gossip Control based Dynamic Virtual machine Consolidation for effective load balancing	• Lack of security of data during migration. • High migration time because of long congestion in link.
[18]	x	✓	x	✓	BPSO based resource provisioning and scheduling	• High privacy leakage during data migration
[19]	x	x	✓	x	Cloud theory based optimization of load in order improve QoS	• Increased latency and end to end delay because centralized processing in cloud layer
[20]	x	✓	x	✓	ARPS framework based scheduling and provisioning of resources	• High congestion occurs due data overloading
[21]	x	x	x	✓	Virtual clustering based multi objective task allocation is carried out using IBBBO.	• long execution time • Slow convergence of IBBBO algorithm

research aims to reduced makespan, energy consumption, and latency during task scheduling, allocation, and VM migration for load balancing. However, the existing works consider limited features for tasks scheduling and allocation which leads to poor scheduling and QoS. In addition, some of the works selects target VM for migration by considering only load which was not enough for optimal VM migration. Due to lack of significant features also increases frequent migration, which increase high overload and latency during load balancing. The existing works used optimization algorithm with slow convergence such as PSO, genetic algorithm, etc. for task scheduling and VM migration which leads to high latency and overload during load balancing in IoT-fog-cloud environment; hence, we need to addressed these issues for providing efficient task scheduling and load balancing results.

6. CONCLUSION

Cloud computing is growing rapidly and users are demanding more and more services, that's why cloud computing load balancing has become such a thoughtful impetus and important research area. Load on the cloud is growing extremely with the expansion of new applications [22] to overcome the load because of huge requests and increase the quality of service many load balancing techniques are used. In this paper, we surveyed many cloud load balancing techniques and focusing to the limitations of each to help the researchers to propose new methods to overcome the limitations to solve the problem of load balancing in cloud computing environment. Table 1 shows the summary of related works the used load balancing algorithms and processes limitations.

REFERENCES

[1] A. B. Kanbar and K Faraj. “Regional aware dynamic task scheduling and resource virtualization for load balancing in IoT-Fog multi-cloud environment”. *Future Generation Computer Systems*, 137C, pp. 70-86, 2022.

[2] H. Nashaat, N. Ashry and R. Rizk. “Smart elastic scheduling algorithm for virtual machine migration in cloud computing”. The Journal of Supercomputing, vol. 75, pp. 3842-3865, 2019.

[3] H. Gao, H. Miao, L. Liu, J. Kai and K. Zhao. “Automated quantitative verification for service-based system design: A visualization transform tool perspective”. *International Journal of Software Engineering and Knowledge Engineering*, vol. 28, no. 10, pp. 1369-1397, 2018.

[4] S. V. Pius and T. S. Shilpa. “Survey on load balancing in cloud computing”. In: *International Conference on Computing, Communication and Energy Systems*, 2014.

[5] P. Jain and S. Choudhary. “A review of load balancing and its challenges in cloud computing”. *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 9275-9281, 2017.

[6] P. Kumar and R. Kumar. “Issues and challenges of load balancing techniques in cloud Computing: A survey”. *ACM Computing Surveys*, vol. 51, no. 6, pp. 1-35, 2019.

[7] M. Alam and Z. A. Khan. “Issues and challenges of load balancing algorithm in cloud computing environment”. *Indian Journal of Science and Technology*, vol. 10, no. 25, pp. 1-12, 2017.

[8] H. Kaur and K. Kaur. “*Load Balancing and its Challenges in Cloud Computing: A Review*”. In: M. S. Kaiser, J. Xie and V. S. Rathore, editors. Information and Communication Technology for Competitive Strategies (ICTCS 2020). Lecture Notes in Networks and Systems, vol. 190. Springer, Singapore, 2021.

[9] G. K. Sriram. “Challenges of cloud compute load balancing algorithms”. *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 1, p. 6, 2022.

[10] H. Chen, F. Wang, N. Helian and G. Akanmu. “*User-priority Guided Min-min Scheduling Algorithm for Load Balancing in Cloud Computing*”. In: Proceeding National Conference on Parallel Computing Technologies (*PARCOMPTECH*), IEEE. pp. 1-8, 2013.

[11] A. F. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L Lydia and K. Shankar. “Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments”. *Journal of Parallel and Distributed Computing*, vol. 142, pp. 36-45, 2020.

[12] M. M. Maswood, M. R. Rahman, A. G. Alharbi and D. Medhi. “A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment”. *IEEE Access*, vol. 8, pp. 113737-113750, 2020.

[13] H. Sun, H. Yu, G. Fan and L. Chen. “Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture”. *Peer-to-Peer Networking and Applications*, vol. 13, no. 2, pp. 548-563, 2020.

[14] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang and Y. Pan. “Stochastic load balancing for virtual resource management in datacenters”. *IEEE Transactions on Cloud Computing*, vol. 8, pp. 459-472, 2020.

[15] Z. Miao, P. Yong, Y. Mei, Y. Quanjun and X. Xu. “A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment”. *Future Generation Computer Systems*, vol. 115, no. 3, pp. 497-516, 2021.

[16] D. Singh, P. S. Saikrishna, R. Pasumarthy and D. Krishnamurthy. “Decentralized LPV-MPC controller with heuristic load balancing for a private cloud hosted application”. *Control Engineering Practice*, vol. 100, no. 4, p. 104438, 2020.

[17] N. M. Donnell, E. Howley and J. Duggan. “Dynamic virtual machine consolidation using a multi-agent system to optimise energy efficiency in cloud computing”. *Future Generation Computer Systems*, vol. 108, pp. 288-301, 2020.

[18] F. Hosseini, A. Safari and M. Farrokhifar. “Cloud theory-based multi-objective feeder reconfiguration problem considering wind power uncertainty”. *Renewable Energy*, vol. 161, pp. 1130-1139, 2020.

[19] M. Kumar, S. C. Sharma, S. S. Goel, S. K. Mishra and A. Husain. “Autonomic cloud resource provisioning and scheduling using meta-heuristic algorithm”. *Neural Computing and Applications*, vol. 32, pp. 18285-18303, 2020.

[20] M. Kumar, A. Kishor, J. Abawajy, P. Agarwal, A. Singh and A. Y. Zomaya. “ARPS: An autonomic resource provisioning and scheduling framework for cloud platforms”. *IEEE Transactions on Sustainable Computing*, vol. 7, no. 2, pp. 386-399, 2021.

[21] X. Liu, B. Cheng and S. Wang. “Availability-aware and energy-efficient virtual cluster allocation based on multi-objective optimization in cloud datacenters”. *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 972-985, 2020.

[22] A. B. Kanbar and K. Faraj. “Modern load balancing techniques and their effects on cloud computing”. *Journal of Hunan University* (*Natural Sciences*), vol. 49, no.7, pp. 37-43, 2022.