Virtual Machine Load Balancing Model Framework for Cloud Computing

E. Suganthi
PG & Research Department of Computer Science, Government College for Women (A) (Affiliated to Bharathidasan University, Trichy), Kumbakonam, India
esuganthics@gmail.com (corresponding author)

F. Kurus Malai Selvi
PG & Research Department of Computer Science, Government College for Women (A) (Affiliated to Bharathidasan University, Trichy), Kumbakonam, India
f.kurusmalaiselvi@gcwk.ac.in

Received: 13 February 2025 | Revised: 26 February 2025 | Accepted: 7 March 2025

 DOI: https://doi.org/10.48084/etasr.10556

ABSTRACT

Cloud Computing (CC) is a comprehensive paradigm that enables individuals and businesses to acquire necessary services on demand. CC provides numerous services, including archiving, distribution platforms, and easy access to online services. Implementing CC necessitates overcoming various difficulties, such as resource identification, protection, scheduling, and Load Balancing (LB). This study examines LB, which distributes workloads across cloud systems to ensure fair resource allocation and prevent Virtual Machines (VMs) from becoming over- or under-loaded. An effective LB solution is essential to maximize VM resource utilization while ensuring high user satisfaction. This study develops the VM LB model framework for CC, which includes a state and random model, a Weight Factor (WF) and priority-based model, and a two-stage optimal model. These models efficiently allocate the VM to the Physical Machine (PM) using Cloudsim. The PlanetLab workload evaluates the performance of the models in terms of Energy Consumption (EC) and Service Level Agreement Violation (SLAV). The experimental results indicate that the proposed model improves Service Level Agreement (SLA) compliance and energy efficiency.

Keywords-cloud computing; load balancing; virtual machine; state; priority; optimization

I. INTRODUCTION

CC is an internet-based technology that has evolved significantly to provide online access to computing resources and services to customers with diverse needs. The CC paradigm encompasses the infrastructure and software data centers used to distribute programs over the internet [1]. Users typically expect a certain Quality of Service (QoS), which is achieved by implementing an emerging virtualization technology, VMs, which is a CC realization that provides computing resources in a virtualized format [2]. Cloud service providers store data across multiple Cloud Data Centers (CDCs) and the tasks are assigned to separate servers by VMs, each of which handles specific responsibilities. Distributed computing ensures efficient task allocation across VMs. Certain VMs may be overloaded, whereas others may require increased utilization when tasks are assigned [3].

LB is a critical and intricate area of research in CC, as it involves the distribution of workloads among VMs in CDCs [4]. LB is essential to maximize resource utilization and QoS in diverse CC environments. Load balancers are necessary to fairly and efficiently allocate resources to workloads to ensure customer satisfaction and reduce costs. However, current LB systems face significant problems that require immediate attention. This has led researchers to propose enhanced LB strategies to address these challenges [5]. In CC, workload distribution is a crucial element of resource allocation. To optimize resource utilization, cloud systems incorporate a number of LB mechanisms [6]. However, most traditional LB techniques suffer from significant computational cost, EC, limited scalability, time constraints, and other challenges.

This paper proposes a framework for VM LB in CC that includes state-based and random-based models, WF and priority-based models, and a two-stage optimal model. The state-based and random-based model first evaluates the current state of the PM, i.e., overloaded, underloaded, or normal, based on the host utilization, after employing a randomized strategy to select the optimal PM for VM allocation. In the WF and priority-based paradigm, the PM current state is determined by the resources (CPU, RAM, and bandwidth) using a PM load identification method based on the resource WF. The priority-based VM allocation model selects the appropriate PM for the VM. In the two-stage optimization model, in the first stage, a VM is selected based on minimum utilization and migration duration. In the second stage, a multi-objective optimization method known as Modified Fish Swarm Optimization (MFSO) is used to allocate VMs. The main contributions of the present study are:

- The state and random model assesses the PM status and uses a randomized strategy for optimal VM allocation.
- The WF and priority-based model allocates VMs based on resource utilization and priority.
- The two-stage optimal model minimizes utilization and migration time and optimizes VM allocation using MFSO.

II. RELATED WORK

In this section, several works related to LB in cloud environments are reviewed based on diverse criteria. The Dynamic Virtual Machine Consolidation (DVMC) model-based LB solution proposed in [7] employs the Pearson correlation algorithm to minimize EC. Authors in [8] present a DVMC technique that balances EC and QoS to facilitate the efficient consolidation of virtual resources. This technique reduces VM migrations and EC rates while maintaining a high QoS. Authors in [9] introduce a greedy methodology for the VM placement strategy. By limiting the number of active PMs and reducing the total EC, the energy efficiency of VMs is emphasized. In addition, reducing the total resource waste involves minimizing the resource allocation and consumption for a PM-deployed VM. Authors in [10] propose a greedy randomized VM placement strategy in a large-scale CDC characterized by diverse and multidimensional resources. To improve energy efficiency and resource utilization, their strategy assigns VMs to energy-efficient PMs. At the same time, it reduces the overall resource consumption and energy utilization. Authors in [11] provide an innovative Machine Learning (ML) approach to integrate a swarm intelligence model with an ML classifier. Authors in [12] present a methodology for VM migration and server consolidation. Authors in [13] propose a utilization-aware VM placement. Authors in [14] advocate the integration of pattern analysis models with time series forecasting methods. Authors in [15] introduce an enhanced genetic-based VM consolidation technique that prioritizes real-time VM replacement. This method employs a Genetic Algorithm (GA) to assign VMs to appropriate PMs. Authors in [16] introduce the Modified Feeding Birds Algorithm (MFBA) model. The technique employs adaptive position update rules and models to reduce VM migration. Authors in [17] propose a VM placement strategy for CDCs utilizing the Harris Hawk Optimization (HHO) model. This model aims to determine the optimal assignment of VMs to suitable hosts considering minimum load and power consumption factors. Authors in [18] present a meta-heuristic optimization method, the Multi-Objective Mayfly VM Placement (MOM-VMP) algorithm. This approach employs a comprehensive CDC with diverse and multidimensional resources. An integrated approach is adopted through a multi-objective dynamic VMP technique.

The existing LB models in cloud environments face limitations, such as increased EC, unnecessary VM migrations, performance degradation, and computational complexity [19]. Methods, such as DVMC, greedy strategies, and Gas, do not efficiently reduce SLAVs or optimize resource allocation. To address these issues, novel approaches are needed. The current study proposes a framework that aims to mitigate EC and SLAVs while improving overall performance.

III. METHODOLOGY

This study introduces a framework for VM LB in cloud environments using three VM allocation algorithms: State-based Random VM_LB (SRVM_LB), WF and Priority-based VM_LB (WFPVM_LB), and Two-Stage Optimal VM_LB (TSOVM_LB). The framework flow is depicted in Figure 1.

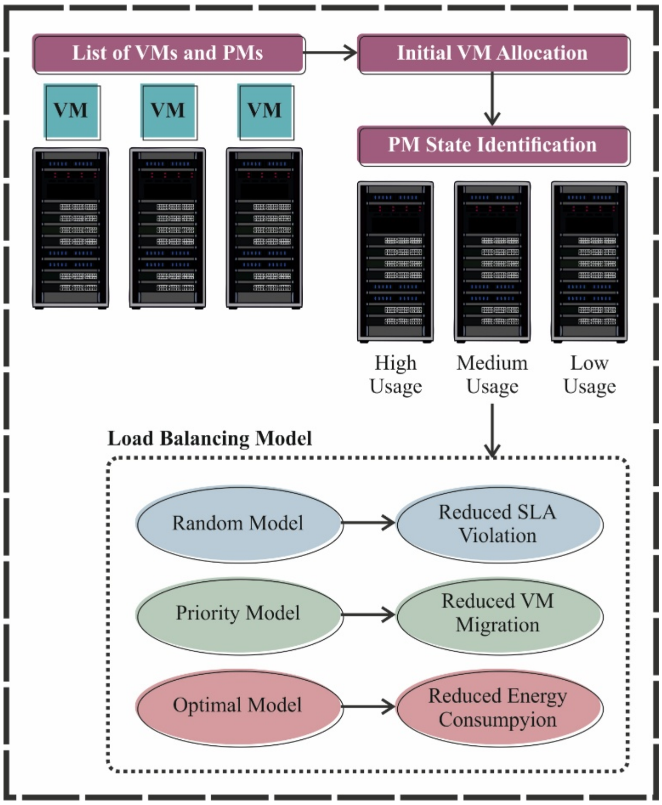

Fig. 1. Framework flow.

A. *State-based Random Virtual Machine Load Balancing*

This approach determines the PM state using lower and upper thresholds derived from PM utilization. A random-based algorithm is employed for VM allocation, and the PM's current resource utilization is assessed to select the appropriate PM for the VM allocation. This model is chosen for its simplicity and efficiency in handling dynamic resource allocation efficiently.

The PM load state in a data center is associated with the system's EC metric. High utilization affects response times and QoS, whereas low utilization increases EC. Identifying PM states is key to efficient resource allocation, and this methodology/.The proposed methodology determines three PM states: high, medium, and low utilization. In addition, VM allocation is crucial for mitigating EC and SLAVs, with this methodology utilizing a randomization technique for effective

VM allocation. A detailed explanation of the algorithm can be found in [20].

B. Weight Factor and Priority-based Virtual Machine Load Balancing

This approach identifies the PM state using the WF of resources (CPU, RAM, and bandwidth). The priority-based technique is used to allocate VMs, ensuring effective resource utilization and minimizing PM congestion. The PM has three operational states: over, under, and normal. The current state of the PM is determined by two thresholds, T_{up} and T_{low}. When T_{up} is low, resource waste occurs; when T_{up} is high, there is continuous congestion and significant SLAVs. In this approach, T_{low} and T_{up} are calculated dynamically based on the current resource utilization. A comprehensive explanation of the algorithm is provided in [21].

C. Two-Stage Optimal Virtual Machine Load Balancing

This approach uses a threshold-based model to determine the current utilization of PMs, ensuring an accurate evaluation of the PM states for enhanced resource allocation. PMs are classified into three states: typically loaded, underloaded, and overloaded. The minimum utilization and migration time of the VM is used to identify suitable VMs for host migration. MFSO evenly distributes migrated VMs and achieves optimal service performance for VM allocation.

IV. SIMULATION RESULTS

The proposed methodology was evaluated using the CloudSim 4.0 simulator, a framework designed to simulate CC environments that can emulate most resources and attributes of cloud systems. In this experiment, two distinct types of hosts and four varieties of VMs were employed to model a CDC. Table I presents the details of their configurations.

TABLE I. PM AND VM CONFIGURATION

	Type		MIPS	Cores	RAM (MB)
PM	HP ProLiant ML110	G4 - Xeon 3040	1860	2	4096
		G5 - Xeon 3075	2660	2	4096
VM	High		2500	1	870
	Medium		2000	1	1740
	Tiny		1000	1	1740
	Micro		500	1	613

Workload data from the PlanetLab project [22] were used to evaluate the performance of the algorithm in a real-world context. PlanetLab is a global initiative that collects CPU utilization data from VMs at over 500 sites. The data, collected from March 3 to April 20, 2011, include various VM counts and distinct resource utilization attributes, including average CPU utilization and standard deviation. Each VM contains 288 CPU utilization records captured every 5 m. The workload dataset is described in Table II [23].

The following metrics evaluate algorithm performance: EC, the number of migrations required to fulfill the workload, SLAV, and SLAV Time per Active Host (SLATAH). EC is the aggregate number of energy processing devices used to achieve a workload. Minimizing energy usage is crucial to efficient allocation because EC varies with CPU utilization and host types. The total number of migrations during each phase reflects network congestion caused by various policies. Minimizing the frequency of live VM migrations is crucial because they can result in a suboptimal user experience in real-world scenarios. The impact of migration on host performance is evaluated by Performance Degradation due to Migration (PDM), which illustrates how VM migrations have impacted QoS. PDM is computed by:

$$\mathrm{PDM} = \frac{1}{N}\sum_{j=1}^{N}\frac{Cd_j}{Cr_j} \tag{1}$$

where Cr_j represents the total resources requested by the VM, Cd_j represents the anticipated performance degradation resulting from migrations, and N represents the total number of VMs.

TABLE II. PLANETLAB WORKLOAD DATASET [23]

Workload	Date	# VM	Mean (%)	Std. dev. (%)
1	03-03-2011	1052	12.31	17.09
2	06-03-2011	898	11.44	16.83
3	09-03-2011	1061	10.70	15.57
4	22-03-2011	1516	9.26	12.78
5	25-03-2011	1078	10.56	14.14
6	03-04-2011	1463	12.39	16.55
7	09-04-2011	1358	11.12	15.09
8	11-04-2011	1233	11.56	15.07
9	12-04-2011	1054	11.54	15.15
10	20-04-2011	1033	10.43	15.21

The congestion condition for the entire data center is represented by SLATAH, which is the average overload duration as a percentage of the total time of all active hosts. If the VM requires more resources than the host can provide, the host with insufficient resources is considered overloaded:

$$\mathrm{SLATAH} = \frac{1}{M}\sum_{i=1}^{M}\frac{To_i}{Tr_i} \tag{2}$$

where M represents the total number of PMs, To_i represents the overload duration, and Tr_i represents the host's running time. The SLAV rate refers to the proportion of time that violations occur during workload execution and it reflects the QoS maintained by the consolidation policy. Policies that consolidate workloads while minimizing SLAVs are favored. SLAV affects the equally essential and autonomous elements of SLATAH and PDM. The combination of SLATAH and PDM, referred to as SLAV, is calculated using:

$$\mathrm{SLAV} = \mathrm{SLATAH} \times \mathrm{PDM} \tag{3}$$

EC and SLAV are limited to presenting the performance of numerous algorithms from two perspectives; they cannot comprehensively evaluate the performance of multiple methods. The EC and SLAV, an objective evaluation of the advantages and disadvantages of different algorithms, is quantified by the Effective Service Violation (ESV), described as:

$$\mathrm{ESV} = \mathrm{EC} \times \mathrm{SLAV} \tag{4}$$

An increase in either metric increases the ESV score, as the ESV is a function of these two variables. A lower ESV score indicates a more significant trade-off between EC and the SLAV.

Table III shows the EC comparison for the three approaches. The average EC of SRVM_LB, WFPVM_LB, and TSOVM_LB is 66.164, 50.196, and 48.608, respectively. The TSOVM_LB approach reduces the EC compared to other LB approaches.

TABLE III. EC COMPARISON

Workload	SRVM_LB	WFPVM_LB	TSOVM_LB
03-03-2011	63.71	48.04	46.68
06-03-2011	56.51	43.49	42.42
09-03-2011	62.9	47.47	46.02
22-03-2011	76.68	58.29	56.24
25-03-2011	64.3	48.86	47.20
03-04-2011	72.8	54.8	52.72
09-04-2011	70.07	53.22	51.45
11-04-2011	71.55	54.16	52.24
12-04-2011	62.66	47.03	46.08
20-04-2011	60.46	46.6	45.03
Average	66.164	50.196	48.608

Table IV depicts the SLAV comparison for the three approaches. The average SLAV of SRVM_LB, WFPVM_LB, and TSOVM_LB is 0.000161, 0.000376, and 0.000165, respectively. The SRVM_LB approach reduces the SLAV compared to other LB approaches.

TABLE IV. SLAV COMPARISON

Workload	SRVM_LB	WFPVM_LB	TSOVM_LB
03-03-2011	0.0002	0.00021	0.0002
06-03-2011	0.0002	0.00028	0.00025
09-03-2011	0.00017	0.0021	0.0002
22-03-2011	0.00013	0.00013	0.00012
25-03-2011	0.00015	0.00017	0.00014
03-04-2011	0.00014	0.00015	0.00014
09-04-2011	0.00014	0.00016	0.00013
11-04-2011	0.00012	0.00014	0.00011
12-04-2011	0.00017	0.0002	0.00017
20-04-2011	0.00019	0.00022	0.00019
Average	0.000161	0.000376	0.000165

Figure 2 compares the EC for different workloads. The TSOVM_LB approach efficiently minimizes the number of active hosts and appropriately distributes resources among hosts, thereby reducing the host switching frequency. Consequently, the TSOVM_LB has some advantages in reducing EC. Figure 3 compares SLAV for different workloads. The SRVM_LB approach can effectively prevent excessive consolidation and reduce the likelihood of resource shortages. Table V illustrates the ESV comparison. The average ESV of SRVM_LB, WFPVM_LB, and TSOVM_LB is 0.010508, 0.000922, and 0.000786, respectively. The TSOVM_LB approach reduces the ESV compared to other LB approaches. Table VI portrays the comparison of the number of VMs migrated. The average VM migration of SRVM_LB, WFPVM_LB, and TSOVM_LB is 1158, 924, and 927, respectively. The WFPVM_LB approach reduces the number of migrated VMs compared to other LB approaches.

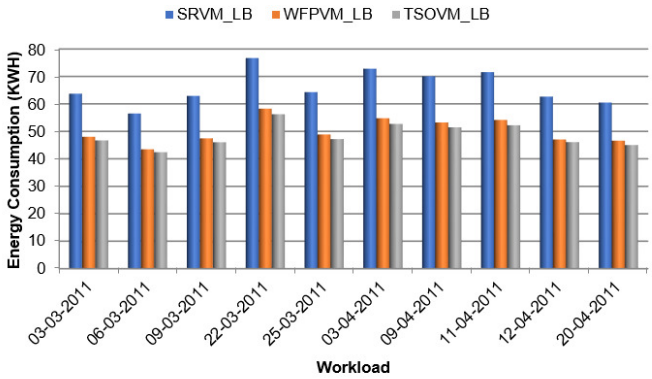

Fig. 2. EC for different workloads.

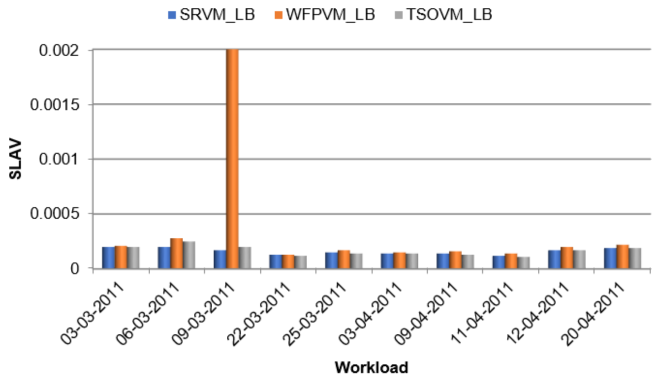

Fig. 3. SLAV for different workloads.

TABLE V. ESV COMPARISON

Workload	SRVM_LB	WFPVM_LB	TSOVM_LB
03-03-2011	0.012742	0.00101	0.00093
06-03-2011	0.011302	0.00122	0.00106
09-03-2011	0.010693	0.001	0.00092
22-03-2011	0.0099684	0.00076	0.00067
25-03-2011	0.009645	0.00083	0.00066
03-04-2011	0.010192	0.00082	0.00074
09-04-2011	0.0098098	0.00085	0.00067
11-04-2011	0.008586	0.00076	0.00057
12-04-2011	0.0106522	0.00094	0.00078
20-04-2011	0.0114874	0.00103	0.00086
Average	0.010508	0.000922	0.000786

TABLE VI. NUMBER OF MIGRATED VMS

Workload	SRVM_LB	WFPVM_LB	TSOVM_LB
03-03-2011	1109	893	889
06-03-2011	976	790	792
09-03-2011	1080	868	875
22-03-2011	1379	1069	1099
25-03-2011	1107	878	886
03-04-2011	1312	1048	1043
09-04-2011	1271	1016	1014
11-04-2011	1192	943	941
12-04-2011	1071	858	857
20-04-2011	1083	877	870
Average	1158	924	927

Figure 4 compares the ESVs for different workloads. The ESV metric indicates both the total amount of EC and the QoS provided. The TSOVM_LB has the lowest ESV compared to other approaches. Figure 5 compares the number of VMs migrated for different workloads. The WFPVM_LB strategy reduces the probability of host overloading and decreases migration rate.

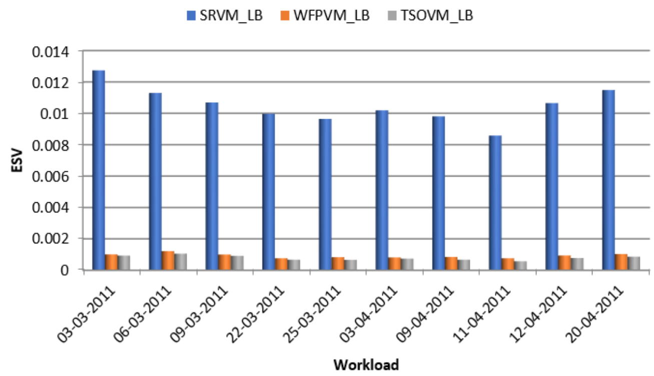

Fig. 4. ESV for different workloads.

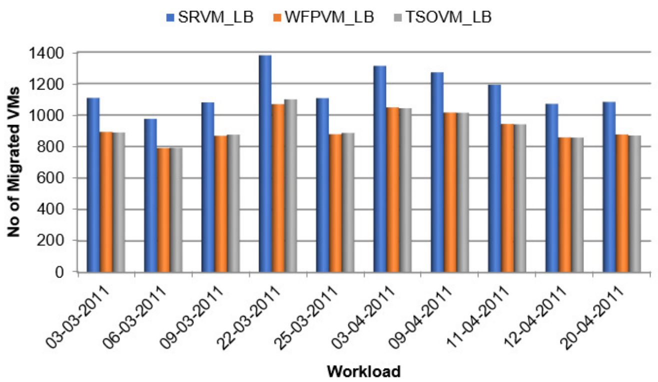

Fig. 5. No of migrated VMs for different workloads.

Based on the results for all workloads, it can be concluded that SRVM_LB reduces the SLAV, WFPVM_LB reduces the number of migrated VMs, and TSOVM_LB reduces EC and ESV.

V. CONCLUSION

Load Balancing (LB) approaches are critical in Cloud Computing (CC) to optimize resource consumption and improve load distribution. Various strategies and methodologies have been proposed to address the challenges associated with LB. This study presents a highly efficient load balancing framework model for CC, which includes state-based and random-based models, Weight Factor (WF) and priority-based models, and a two-stage optimal model. The state-based and random-based model first evaluates the current state of the Physical Machine (PM), i.e., overloaded, underloaded, or normal, according to the host utilization, and then uses a randomized approach to identify the most appropriate PM for Virtual Machines (VMs) allocation. The WF and priority-based model evaluates the current state of the PM in terms of resources (CPU, RAM, and bandwidth) using a PM load identification technique based on resource WFs. The priority-based VM allocation model determines the appropriate PM for the VM allocation. In the two-stage optimal model, the first stage involves selecting a VM based on minimum utilization and migration duration. In the second stage, a multi-objective optimization technique known as Modified Fish Swarm Optimization (MFSO) is deployed for VM allocation. The research was conducted using extensive real-world data derived from PlanetLab. The results demonstrate that the state-based and random-based approach reduces Service Level Agreement Violations (SLAVs), the WF and priority-based approach reduces the number of migrated VMs, and the two-stage optimal approach reduces Energy Consumption (EC) and Effective Service Violation (ESV). The existing LB models face problems, such as high EC and frequent VM migrations, therefore, future research should focus on optimizing energy efficiency, mitigating migration overhead, and improving performance at scale.

REFERENCES

[1] H. M. Shukur, S. R. M. Zeebaree, R. R. Zebari, D. Q. Zeebaree, O. M. Ahmed, and A. A. Salih, "Cloud Computing Virtualization of Resources Allocation for Distributed Systems," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 98–105, Jun. 2020, https://doi.org/10.38094/jastt1331.

[2] 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, https://doi.org/10.1109/ACCESS.2021.3065308.

[3] M. Jesi, A. Appathurai, M. Narayanaperumal, and A. Kumar, "Load Balancing in Cloud Computing via Mayfly Optimization Algorithm," *Revue Roumaine des Sciences Techniques — Série Électrotechnique et Énergétique*, vol. 69, no. 1, pp. 79–84, Apr. 2024, https://doi.org/10.59277/RRST-EE.2024.1.14.

[4] S. Swarnakar, S. Bhattacharya, and C. Banerjee, "A Bio-Inspired and Heuristic-Based Hybrid Algorithm for Effective Performance With Load Balancing in Cloud Environment," *International Journal of Cloud Applications and Computing*, vol. 11, no. 4, pp. 59–79, Oct. 2021, https://doi.org/10.4018/IJCAC.2021100104.

[5] H. Singh, S. Tyagi, P. Kumar, S. S. Gill, and R. Buyya, "Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions," *Simulation Modelling Practice and Theory*, vol. 111, Sep. 2021, Art. no. 102353, https://doi.org/10.1016/j.simpat.2021.102353.

[6] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 3910–3933, Jul. 2022, https://doi.org/10.1016/j.jksuci.2021.02.007.

[7] J. P. B. Mapetu, L. Kong, and Z. Chen, "A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing," *The Journal of Supercomputing*, vol. 77, no. 6, pp. 5840–5881, Jun. 2021, https://doi.org/10.1007/s11227-020-03494-6.

[8] H. Li, T. Li, and Z. Shuhua, "Energy-performance optimisation for the dynamic consolidation of virtual machines in cloud computing," *International Journal of Services Operations and Informatics*, vol. 9, no. 1, pp. 62–82, Jan. 2018, https://doi.org/10.1504/IJSOI.2018.088517.

[9] H. Wang and H. Tianfield, "Energy-Aware Dynamic Virtual Machine Consolidation for Cloud Datacenters," *IEEE Access*, vol. 6, pp. 15259–15273, 2018, https://doi.org/10.1109/ACCESS.2018.2813541.

[10] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "GRVMP: A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2571–2582, Jun. 2021, https://doi.org/10.1109/JSYST.2020.3002721.

[11] S. Talwani *et al.*, "Machine-Learning-Based Approach for Virtual Machine Allocation and Migration," *Electronics*, vol. 11, no. 19, Oct. 2022, Art. no. 3249, https://doi.org/10.3390/electronics11193249.

[12] R. Sharma and B. Sinha, "Load Balancing and Server Consolidation for Energy Management in Cloud Data Center," *ITM Web of Conferences*,

vol. 54, 2023, Art. no. 01017, https://doi.org/10.1051/itmconf/20235401017.

[13] D. Dabhi and D. Thakor, "Utilisation-aware VM placement policy for workload consolidation in cloud data centres," *International Journal of Communication Networks and Distributed Systems*, vol. 28, no. 6, pp. 704–726, Jan. 2022, https://doi.org/10.1504/ijcnds.2022.126224.

[14] S. Vila, F. Guirado, and J. L. Lérida, "Cloud computing virtual machine consolidation based on stock trading forecast techniques," *Future Generation Computer Systems*, vol. 145, pp. 321–336, Aug. 2023, https://doi.org/10.1016/j.future.2023.03.018.

[15] M. Radi, A. A. Alwan, and Y. Gulzar, "Genetic-Based Virtual Machines Consolidation Strategy With Efficient Energy Consumption in Cloud Environment," *IEEE Access*, vol. 11, pp. 48022–48032, 2023, https://doi.org/10.1109/ACCESS.2023.3276292.

[16] D. Alsadie and M. Alsulami, "Efficient Resource Management in Cloud Environments: A Modified Feeding Birds Algorithm for VM Consolidation," *Mathematics*, vol. 12, no. 12, Jun. 2024, Art. no. 1845, https://doi.org/10.3390/math12121845.

[17] H. S. Madhusudhan, S. Kumar T., P. Gupta, and G. McArdle, "A Harris Hawk Optimisation system for energy and resource efficient virtual machine placement in cloud data centers," *Plos One*, vol. 18, no. 8, Aug. 2023, Art. no. e0289156, https://doi.org/10.1371/journal.pone.0289156.

[18] S. Durairaj and R. Sridhar, "MOM-VMP: multi-objective mayfly optimization algorithm for VM placement supported by principal component analysis (PCA) in cloud data center," *Cluster Computing*, vol. 27, no. 2, pp. 1733–1751, Jun. 2023, https://doi.org/10.1007/s10586-023-04040-8.

[19] P. K. Mishra and A. K. Chaturvedi, "An Improved Laxity based Cost Efficient Task Scheduling Approach for Cloud-Fog Environment," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19037–19044, Feb. 2025, https://doi.org/10.48084/etasr.8595.

[20] E. Suganthi and F. Kurus Malai Selvi, "Cloud Computing by Implementing State and Random-Based Virtual Machine Load Balancing Model," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 92–101, Jun. 2024, https://doi.org/10.22266/ijies2024.0630.08.

[21] E. Suganthi and F. Kurus Malai Selvi, "Weight factor and priority-based virtual machine load balancing model for cloud computing," *International Journal of Information Technology*, vol. 16, no. 8, pp. 5271–5276, Dec. 2024, https://doi.org/10.1007/s41870-024-02119-y.

[22] Z. Á. Mann and M. Szabó, "Which is the best algorithm for virtual machine placement optimization?," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, Apr. 2017, Art. no. e4083, https://doi.org/10.1002/cpe.4083.

[23] Z. Ma, D. Ma, M. Lv, and Y. Liu, "Virtual Machine Migration Techniques for Optimizing Energy Consumption in Cloud Data Centers," *IEEE Access*, vol. 11, pp. 86739–86753, 2023, https://doi.org/10.1109/ACCESS.2023.3305268.