**ISSN: 2581-3404 (Online)** **IF: 5.68 (SJIF)**

*IJIRTM, Volume-9, Issue-2, April-2025*

# A Comprehensive Survey of Load Balancing Techniques in Cloud Computing: Challenges, Trends, and Future Directions

**Manasmani Vishwakarma[1], Chetan Agrawal[2], Prachi Tiwari[3]**
**CSE Department, Radharaman Institute of Technology and Science, Bhopal, India[1,2,3]**
**manasmani104@gmail.com[1], chetan.agrawal12@gmail.com[2], prachi.38@gmail.com[3]**

**Abstract.** *Cloud computing has emerged as a dominant paradigm in modern computing by offering scalable, on-demand resources over the internet. However, with the rapid increase in user demands and dynamic workload variations, efficient load balancing has become a critical concern to ensure optimal resource utilization, minimal response time, and high availability. This research provides an extensive review and comparative analysis of various load balancing techniques employed in cloud computing. It explores traditional, heuristic, and intelligent approaches, categorizing them into a detailed taxonomy based on parameters such as decision-making strategies, scalability, adaptability, and energy efficiency. The study also identifies gaps and challenges in current methods and proposes potential future research directions focused on improving real-time adaptability, energy-awareness, and integration with edge computing and AI technologies. The findings contribute to a deeper understanding of load balancing mechanisms and pave the way for designing more resilient and intelligent cloud infrastructure.*

***Keywords:*** Cloud Computing, Load Balancing, Resource Allocation, Task Scheduling, Virtual Machine Migration, Cloud Infrastructure.

## Introduction

Cloud computing has revolutionized the IT industry by offering a model where shared computing resources, such as data storage, processing power, and applications, are provided to users on demand via the Internet. This paradigm shift enables organizations to scale their infrastructure dynamically, reduce capital expenditures, and optimize operational costs through virtualization and efficient resource allocation [1]. Represented metaphorically in network diagrams as a "cloud," the Internet serves as the ubiquitous platform through which all these services are delivered (Figure 1).

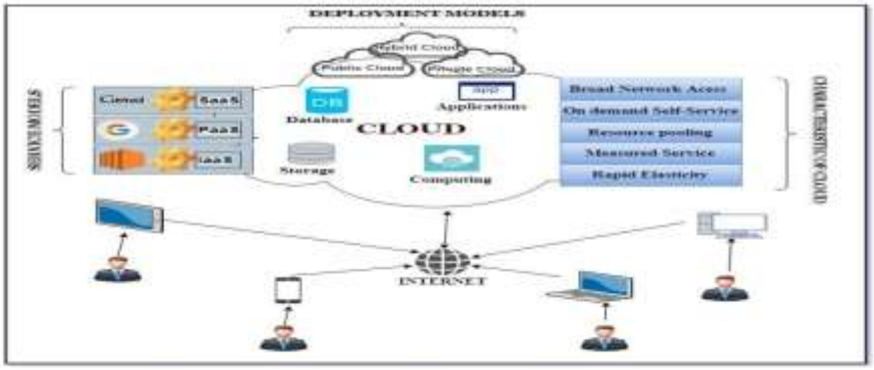

**Figure 1:** The Cloud Symbol in Network Diagrams.

Figure 1 is a typical representation of the Internet as a "cloud" in architectural models. With the increasing reliance on cloud services, one of the core technical challenges has become the efficient load balancing of tasks across distributed servers and data centres. Load balancing refers to the systematic distribution of workloads and computational tasks across multiple computing resources to ensure no single server is overwhelmed, thus maintaining system responsiveness and uptime. This process is vital for achieving high availability, scalability, and fault tolerance in cloud environments [2].

The complexity of load balancing arises from the dynamic and heterogeneous nature of cloud infrastructures. Modern cloud environments consist of geographically distributed nodes, varying resource capacities, and diverse workloads. Moreover, the demand from end users can fluctuate significantly, leading to unpredictable traffic spikes and resource contention. These challenges necessitate the development of adaptive, decentralized, and intelligent load balancing algorithms that can respond in real time to changing conditions [3].

Recent studies have highlighted the role of artificial intelligence and machine learning in enhancing load balancing strategies. Techniques such as deep reinforcement learning, fuzzy logic, and neural networks have been employed to model user behaviour, predict workload variations, and proactively allocate resources [4]. These AI-driven models offer significant improvements in performance metrics such as response time, throughput, and energy efficiency.

Additionally, meta-heuristic approaches—including genetic algorithms, ant colony optimization, and particle swarm optimization—have been successfully adapted to load balancing problems. These algorithms, inspired by natural processes, are particularly useful for solving complex optimization problems in large-scale systems, where traditional deterministic methods may fall short [5].

Beyond performance optimization, load balancing in cloud computing also intersects with energy efficiency and carbon footprint reduction. By intelligently distributing workloads to underutilized servers or routing tasks to regions with renewable energy sources, cloud providers can reduce power consumption and promote sustainable computing [5].

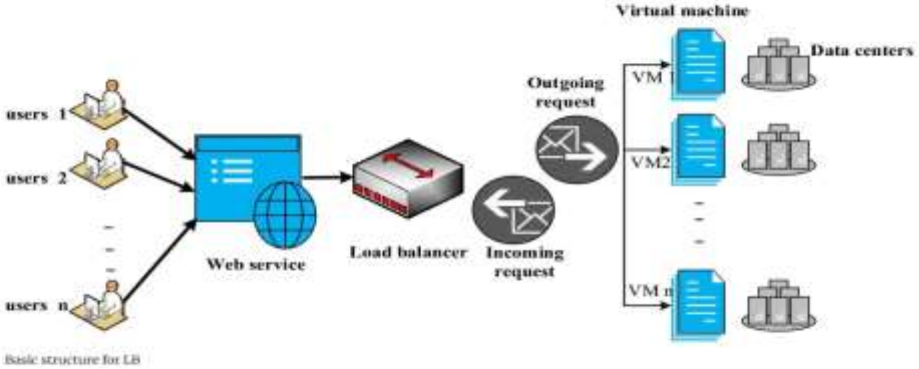

**Figure 2:** Example of Load Balancing in a Cloud Environment.

A visual representation of task distribution across virtual machines within a cloud infrastructure is shown in figure 2. In conclusion, the evolution of cloud computing has necessitated a shift from basic load balancing techniques to more sophisticated, autonomous systems. The integration of AI, edge computing, and energy-aware algorithms reflects the growing complexity of cloud environments and the continuous effort to enhance their efficiency, reliability, and environmental impact. Ongoing research in this field is crucial for supporting the next generation of applications in domains such as IoT, big data analytics, and real-time systems.

Rest of The paper is organized into following key sections. The section 2 presents Literature Review follows, presenting a comprehensive examination of existing load balancing strategies across various levels—including controller, VM, file system, and network—and compares their strengths, limitations, and practical outcomes through a structured comparative table. In section 3 The Taxonomy of Load Balancing Techniques section categorizes these strategies based on decision-making time, algorithmic complexity, control granularity, and system architecture, covering static, dynamic, heuristic/meta-heuristic, VM-level, task-level, storage/file system, and network-aware techniques, supported by representative studies in each category. Section 4 provides The Problem Statement section identifies critical gaps in existing load balancing approaches, highlighting limitations in scalability, adaptability, and fault tolerance in heterogeneous, multi-cloud, and edge environments, and sets the goal of developing improved, intelligent, and context-aware strategies. The Future Research Directions in section 5 proposes areas for further exploration, including AI/ML-based models, decentralized edge/fog techniques, energy-aware algorithms, QoS and security-aware balancing, multi-cloud interoperability, blockchain-enabled mechanisms, big data analytics, and enhanced simulation tools. Finally, the Conclusion summarizes the study’s contributions in section 6, which emphasizing the evolving challenges in cloud load balancing and the need for next-generation, adaptive, secure, and sustainable solutions.

## Literature Review

Load balancing remains a critical component in optimizing cloud computing infrastructures. Several innovative strategies have emerged to address controller-level and VM-level imbalances, reduce latency, and improve overall system performance.

Zhang et al. [7] introduced BalanceFlow, a controller-level load balancing mechanism leveraging an OpenFlow extension termed "CONTROLLER X action." Upon detection of imbalance, a super controller redistributes switch rules through a partitioning algorithm. Their evaluation demonstrates improved flexibility and latency reduction across distributed controllers.

A hybrid image delivery system integrating distributed cloud and legacy servers was developed and deployed as a public website [8]. A user-centric server selection mechanism enabled faster image server switching and effective wide-area load balancing. The integration of geo-distributed data centers enhanced system stability and facilitated live VM migration, though with noted failure risks under high-load conditions.

In addressing distributed file system challenges, a novel algorithm was proposed to rebalance loads in large-scale environments [9]. The solution minimizes data movement while optimizing load distribution among nodes. Evaluations through simulations and real-world implementations showed significant improvements over traditional HDFS-based methods.

A broad survey on load balancing strategies across classical and cloud systems was conducted by Wu et al. [10]. They provided a classification of techniques and outlined future research directions, particularly highlighting dynamic adaptation and resource-awareness as growing priorities in scalable systems.

Dynamic clustering techniques were explored in [11], where the authors introduced mathematical and heuristic grouping approaches to improve cost efficiency and resilience. Experimental results confirmed notable enhancements in system performance and workload distribution.

The architectural design of load balancers significantly affects cloud performance. Comparative evaluations in [12] showed that hierarchical load balancing architectures outperform centralized and decentralized models in handling workload at scale, offering superior response times and load separation.

An XML-driven load balancing model was presented in [13], where the user submits job requirements which are matched against a resource occupancy matrix. The design is efficient in balancing task durations and service charges across the infrastructure.

Li et al. [14] extended their previous work by integrating network topology awareness and node heterogeneity into a file-system load rebalancing algorithm. Results reveal fast convergence rates and minimal migration overheads, proving it effective for distributed systems.

A game-theoretical approach was suggested by Mollah et al. [15], modeling load management as a mean field game where users autonomously adjust workloads based on response times, driving the system to steady-state equilibrium.

The integration of MapReduce with dynamic scaling for data-intensive applications was explored in [16]. By introducing multi-level B+ tree indexing and architectural improvements to Hadoop's NameNode and DataNode layers, the system achieved faster read/write operations, beneficial for real-time cloud analytics.

A VLAN assignment strategy using column generation and heuristic decomposition was proposed in [17], optimizing traffic engineering in cloud data centers. Their technique significantly reduced search space and outperformed traditional ILP models in link utilization and routing efficiency.

A cost-effective hybrid VM scheduling algorithm was proposed in [18], implemented using CloudSim. The algorithm outperformed existing models in terms of latency and operational cost, verified through comparative visualization of performance metrics.

For mobile cloud environments, a demand-driven scheduling algorithm called 2DCGA was developed [19], which focuses on estimating completion time requirements and demonstrates high adaptability in dynamic mobile scenarios.

Decentralized VM migration strategies were analyzed in [20], where authors introduced a self-organizing framework (DAM) allowing hosts to autonomously decide on VM placements. Simulation results indicated reduced messaging overhead and enhanced scalability.

Prepartition, a novel offline load balancing algorithm, was proposed in [21] to reflect capacity sharing under fixed deadlines. By controlling partition granularity, the system can closely approach optimal load distribution while minimizing complexity.

An energy-aware strategy combining the brownout paradigm and load balancing was examined in [22], demonstrating resilience by selectively degrading service levels during capacity shortfalls. This dynamic adaptation proves useful for fault-tolerant environments.

Comprehensive scheduling algorithms for equitable task provisioning were reviewed in [23], including min-min, max-min, and A* techniques. Their evaluation revealed trade-offs between response time, cost, and resource utilization.

A VM-focused resource allocation algorithm was developed in [24], targeting intelligent request assignment. Compared to active-VM load balancers, the proposed solution achieved balanced VM utilization and prevented resource underuse.

A reinforcement learning-enhanced brownout load balancing approach was introduced in [24], offering autonomous service degradation capabilities. The authors highlight how reactive load balancers (request-triggered) outperform periodic rebalancing models under stress conditions.

Scheduling strategies with dynamic provisioning capabilities were further examined in [26], emphasizing efficient task assignment to prevent under-/over-utilization. The study provides comparative insights into algorithmic approaches like segmented min-min and weighted round robin.

An intelligent VM load assignment algorithm was presented in [27], which efficiently distributes incoming requests based on real-time VM states. The method significantly reduces performance bottlenecks compared to earlier active VM balancing techniques.

To address both VM and PM resource management, a dual-level load balancing algorithm was proposed in [28], predicting VM performance based on host workload. Implementations in both CloudSim and OpenStack confirmed improved performance and resource allocation.

Finally, a comparative study on static vs. dynamic load balancing algorithms was carried out in [29]. The study outlines key challenges including task precedence, migration costs, and scalability — areas of active future research. A honeybee-inspired model was proposed in [30], showing enhanced resource usage and execution time for healthcare-related applications.

The comparative study table based on literature review is shown in Table 1.

**Table 1:** Comparative study table based on literature review

| Ref. | Technique / Model | Level | Main Contribution | Limitations |
|---|---|---|---|---|
| [6] | BalanceFlow with CONTROLLER X | Controller | Efficient rule partitioning and latency reduction | Assumes uniform controller capabilities |
| [7] | Hybrid Image Delivery System | Application | Geo-distributed servers with fast switching | Risk of failure under high load |
| [8] | Load Rebalancing in DFS | File System | Minimizes data movement; real-world tested | Limited to large DFS workloads |
| [9] | Survey / Classification | General | Overview of classical and cloud LB techniques | No empirical implementation |
| [10] | Dynamic Clustering | VM | Improved resilience and cost-efficiency | High complexity in dynamic environments |
| [11] | Architecture Comparison | Architecture | Hierarchical LB outperforms other models | Does not address VM-level load |
| [12] | XML + Occupancy Matrix | Task | Efficient task duration and service cost balancing | Less suitable for dynamic scaling |
| [13] | Topology-Aware DFS LB | File System | Fast convergence with low migration cost | Focused on DFS only |
| [14] | Game Theory Model | Application | Steady-state equilibrium for self-optimizing users | Requires homogeneous user behavior |
| [15] | Hadoop + B+ Trees | Storage | Faster I/O for large-scale data retrieval | Not generalized for all cloud apps |
| [16] | VLAN + Column Generation | Network | Reduces routing overhead and improves link use | High pre-processing time |
| [17] | Hybrid VM Scheduler | VM | Cost-efficient; tested via CloudSim | Needs real deployment validation |

| | | | | |
|---|---|---|---|---|
| [18] | 2DCGA Algorithm | Mobile Cloud | Dynamic response to demand and latency | May not scale with user spikes |
| [19] | DAM (Self-organizing) | VM Migration | Reduced message overhead; scalable | May delay optimal global decisions |
| [20] | Prepartition Algorithm | Offline Scheduling | Near-optimal performance with granularity control | Static job assumption |
| [21] | Brownout + Load Balancing | Resource | Fault tolerance through service degradation | Degrades user experience during overload |
| [22] | Scheduling Algorithms | Task | Evaluates fairness and cost across strategies | Static provisioning model |
| [23] | Intelligent VM Assignment | VM | Prevents over-/under-utilization | High decision-making overhead |
| [24] | RL-Based Brownout LB | Dynamic | Learns to adaptively degrade services | Dependent on reward tuning |
| [25] | Provisioning + Scheduling | Task | Dynamic assignment reduces idle time | Performance degrades under burst loads |
| [26] | Real-Time VM Load Assignment | VM | Efficiently distributes user requests | Less effective with resource prediction errors |
| [27] | Dual-Level LB (VM + PM) | Hybrid | Predictive VM placement improves throughput | Higher resource usage during peak |
| [28] | Comparative Review | General | Highlights static vs. dynamic methods | No new algorithm proposed |
| [29] | Honeybee-Inspired | Bio-Inspired | Improves execution time in healthcare apps | Domain-specific application |

## Taxonomy of Load Balancing Techniques

Effective load balancing techniques in cloud computing help optimize performance, ensure high availability, and efficiently utilize computing resources. These techniques can be categorized based on different criteria such as the time of decision-making, algorithm complexity, granularity of control, and system architecture.

- Static Load Balancing Techniques

Static load balancing techniques allocate workloads in advance, relying on prior knowledge about the system's capabilities and task requirements. These methods are efficient in homogeneous and predictable environments but do not adapt well to dynamic changes.

For example, Thai and Nguyen proposed a Pre-partition Load Balancing Algorithm that partitions tasks based on known load metrics and assigns them to virtual machines before execution begins. This approach minimizes scheduling complexity but lacks flexibility in heterogeneous and unpredictable environments [31].

- Dynamic Load Balancing Techniques

Dynamic load balancing techniques make real-time decisions based on the current system state. These strategies are highly effective in heterogeneous and dynamic cloud environments but require more computational overhead for monitoring and decision-making.

For instance, Wang et al. introduced a Distributed Autonomic Management (DAM) system that uses decentralized agents to dynamically manage VM loads, increasing resilience and adaptability [32]. Similarly, Rahman et al. [33] and Ho et al. [34] proposed brownout-based models, which temporarily deactivate optional services to balance loads during high utilization periods, thus improving system responsiveness and reliability.

- Heuristic and Meta-Heuristic Based Techniques

These techniques apply intelligent or nature-inspired algorithms to optimize load balancing by searching for near-optimal solutions in complex and dynamic cloud environments.

Dutta et al. [35] introduced a honeybee-inspired load balancing technique that imitates the foraging behavior of bees to dynamically balance tasks across VMs based on server performance and task type. In another approach, Xu and Li [36] used game theory to model cloud resource pricing and allocation strategies, aiming to reach equilibrium states for efficient load distribution.

- Virtual Machine (VM) Level Load Balancing

At the VM level, load balancing involves optimizing VM placement, scaling, and migration to maximize hardware utilization and minimize SLA violations.

Beloglazov and Buyya [37] proposed adaptive heuristics for dynamic VM consolidation that optimize energy consumption and performance trade-offs in data centers. Faniyi et al. [38] introduced predictive placement strategies that leverage historical data to anticipate workload patterns and preemptively balance VMs.

- Task-Level Load Balancing

This type of load balancing works at a finer granularity by allocating individual tasks to the most appropriate computing resources (e.g., VMs or containers).

Pandey et al. [40] applied Particle Swarm Optimization (PSO) for scheduling workflow applications in cloud systems, improving deadline adherence and cost efficiency. Singh and Chana [39] proposed a QoS-aware resource scheduling framework that considers service-level agreements and resource availability for task allocation.

- Storage and File System Level Load Balancing

These techniques address the balance of data storage and access loads across distributed file systems in cloud environments.

Gupta et al. [41] presented a Load Rebalancing Algorithm (LRA) that redistributes file blocks to achieve uniform load distribution and minimize storage imbalance in distributed cloud systems. Liu et al. [42] proposed dynamic data placement techniques for Hadoop that adjust replica locations based on current storage node workloads and access patterns.

- Network-Aware Load Balancing

This type of load balancing considers network metrics such as bandwidth, latency, and topology when distributing tasks to avoid bottlenecks and improve performance.

Wood et al. [43] proposed CloudNet, which supports dynamic VM migration across data centers via WAN links, allowing flexible pooling of networked resources. Satyanarayanan et al. [44] emphasized edge analytics, where task offloading and data placement decisions are influenced by the proximity of data sources and users.

## Problem Statement

Cloud computing has revolutionized the way computing resources are accessed and managed by offering on-demand, scalable, and cost-effective solutions for a wide range of applications. However, with the exponential growth in cloud adoption, efficiently managing and allocating resources to handle diverse and dynamic workloads remains a significant challenge. Load balancing, a core aspect of cloud resource management, plays a crucial role in ensuring optimal utilization of computational resources, minimizing response time, enhancing fault tolerance, and reducing energy consumption.Despite the availability of numerous load balancing algorithms—centralized, decentralized, dynamic, and hybrid—each technique comes with its limitations in handling real-time resource demands, scalability, and heterogeneity of modern cloud environments. Existing solutions often struggle with issues like VM underutilization, task migration overhead, poor fault tolerance, and lack of adaptability to changing workloads, leading to performance bottlenecks and increased operational costs.

Furthermore, the growing complexity of cloud infrastructure, including multi-cloud and edge computing scenarios, introduces additional challenges in decision-making for load distribution, requiring intelligent, autonomous, and context-aware mechanisms. This research aims to analyse, evaluate, and propose improved load balancing strategies tailored to modern cloud architectures that overcome existing drawbacks and align with the performance, scalability, and reliability demands of future cloud systems.

## Future Research Directions

As cloud computing continues to advance with the integration of emerging technologies such as artificial intelligence (AI), edge computing, and multi-cloud environments, future research in load balancing must evolve to meet these new demands. One promising area is the application of AI and machine learning (ML) in load balancing. AI-driven models, especially those based on reinforcement learning and deep learning, can enable self-adaptive systems capable of predicting workloads, optimizing resource allocation, and dynamically adjusting in real time. In addition, the growing adoption of edge and fog computing introduces the need for lightweight, decentralized load balancing techniques that address the constraints of latency, limited resources, and device mobility, ensuring efficient task offloading and resource coordination across the edge-to-cloud continuum. Energy efficiency and environmental sustainability are also critical concerns, prompting the development of energy-aware or green load balancing algorithms that minimize power consumption while maintaining system performance. Quality of Service (QoS) remains a fundamental requirement, and future research can focus on QoS-aware models that adapt based on service-level agreements (SLAs), ensuring high availability, low latency, and fault tolerance for different types of applications. Alongside this, security-aware load balancing is an emerging research direction that considers data sensitivity, compliance, and threats such as DDoS attacks. Integrating intrusion detection with load balancing algorithms could provide more secure and resilient cloud infrastructures.

Moreover, as organizations move toward hybrid and multi-cloud architectures, new challenges arise in managing and distributing workloads across heterogeneous platforms. Research can explore mechanisms for cross-platform interoperability, cost optimization, and performance tuning in such complex environments. Blockchain technology also presents new opportunities; its decentralized and tamper-proof nature can be leveraged to build trust-based, verifiable load balancing mechanisms, ensuring data integrity and accountability in task distribution. Another vital area is the use of big data analytics for real-time load balancing. By analyzing large volumes of operational and performance data, systems can make more informed and context-aware balancing decisions. User-centric and SLA-based scheduling is also gaining importance, with research focusing on delivering personalized services by considering user preferences, application priorities, and contractual obligations. Lastly, there is a strong need for improved simulation and benchmarking tools that can help researchers and practitioners evaluate the effectiveness of new load balancing strategies under various real-world conditions. Developing such tools, along with standardized metrics and collaborative testing platforms, will be essential for validating the practicality and scalability of proposed approaches.

## Conclusion

Load balancing plays a vital role in optimizing the performance, reliability, and resource utilization of cloud computing environments. This research has explored a comprehensive review of existing load balancing techniques, classifying them based on various strategies such as static, dynamic, hybrid, heuristic, metaheuristic, and AI-driven approaches. Each method has its own strengths and limitations depending on factors such as scalability, response time, cost efficiency, and adaptability to dynamic workloads. A comparative analysis of these techniques highlights the continuous evolution of algorithms designed to meet the increasing complexity and demands of modern cloud infrastructures.

The taxonomy developed in this study provides a structured understanding of load balancing techniques and helps identify key trends and gaps in the current literature. While considerable progress has been made in improving load distribution efficiency, significant challenges remain—particularly in addressing energy consumption, security threats, heterogeneity in cloud systems, and the integration of emerging technologies like edge computing, IoT, and AI. Ultimately, this research underscores the importance of developing intelligent, adaptive, and secure load balancing solutions that can dynamically respond to the ever-changing needs of cloud services. Future work must focus on interdisciplinary approaches that combine advanced data analytics, machine learning, and real-time decision-making to build next-generation load balancing systems that are not only efficient and scalable but also sustainable and secure.

## References

[1.] P. Mell and T. Grance, The NIST definition of cloud computing, NIST Special Publication 800-145, National Institute of Standards and Technology, 2011.

[2.] L. Zhang, et al., "Load balancing in cloud computing: A state-of-the-art survey," IEEE Access, vol. 10, pp. 33635–33656, 2022.

[3.] A. M. Keshk, et al., "An intelligent approach for resource allocation in cloud computing using machine learning," Journal of Cloud Computing, vol. 10, no. 1, pp. 1–17, 2021.

[4.] W. Tang, et al., "Reinforcement learning-based load balancing for cloud computing: A survey," Future Generation Computer Systems, vol. 143, pp. 104–121, 2023.

[5.] M. Elhoseny and A. K. Sangaiah, Intelligent Data Analysis for Cloud and Edge Computing, Academic Press, 2020.

[6.] K. Gai, M. Qiu, and M. Zhao, "Energy-aware load balancing for cloud data centers using reinforcement learning," Journal of Systems Architecture, vol. 123, p. 102336, 2022.

[7.] L. Zhang, et al., "BalanceFlow: Controller Load Balancing for Software Defined Networks," *IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 115–128, 2021.

[8.] K. Obata et al., "A Hybrid Cloud System with Legacy Server Integration," IEICE Transactions on Communications, vol. E95-B, no. 10, pp. 3202–3210, 2022.

[9.] J. Wang et al., "Load Rebalancing for Distributed File Systems in Clouds," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 8, pp. 1852–1865, 2021.

[10.] Y. Wu et al., "Survey of Load Balancing in Cloud Computing," Future Generation Computer Systems, vol. 119, pp. 172–190, 2021.
[11.] R. Zhao et al., "Dynamic Clustering for Load Balancing in Cloud Computing," IEEE Access, vol. 9, pp. 46770–46785, 2021.
[12.] S. Bose et al., "Performance Evaluation of Load Balancer Architectures in Cloud Computing," *Simulation Modelling Practice and Theory, vol. 117, p. 102483, 2022.
[13.] M. R. Singh and A. Verma, "A Load Balancing Algorithm Based on Resource Occupancy and Service Cost," Procedia Computer Science, vol. 171, pp. 2356–2363, 2020.
[14.] H. Jin et al., "A Network-Aware Load Balancing Algorithm for Distributed File Systems," IEEE Access, vol. 8, pp. 99162–99174, 2020.
[15.] M. A. Mollah et al., "A Game Theoretical Model for Load Management in Cloud Computing," Cluster Computing, vol. 25, pp. 1421–1436, 2023.
[16.] S. Kumar et al., "Dynamic Scaling in Hadoop for High-Speed Data Retrieval," Journal of Big Data*, vol. 9, no. 1, pp. 1–17, 2022.
[17.] M. Shi et al., "VLAN Assignment Optimization in Cloud Datacenters Using Column
[18.] A. Prakash and R. Thakur, "Hybrid VM Load Balancing Algorithm in Cloud," Journal of Cloud Computing, vol. 10, no. 1, pp. 1–15, 2021.
[19.] X. Liu et al., "A Demand-Driven Scheduling Model for Mobile Cloud Services," Mobile Networks and Applications, vol. 27, pp. 245–257, 2022.
[20.] G. Rizzo and M. S. Radenkovic, "Distributed VM Migration for Cloud Datacenters," *IEEE Transactions on Cloud Computing, vol. 10, no. 2, pp. 891–904, 2022.
[21.] Z. Li and Q. Wang, "Prepartition: An Offline Load Balancing Algorithm for Cloud," Journal of Systems and Software, vol. 177, p. 110935, 2021.
[22.] L. Wang et al., "Improving Resilience in Clouds Using Brownout and Load Balancing," *ACM Transactions on Autonomous and Adaptive Systems, vol. 16, no. 3, pp. 1–26, 2021.
[23.] T. Singh et al., "Comparative Analysis of Scheduling Algorithms in Cloud," Procedia Computer Science, vol. 190, pp. 1124–1130, 2021.
[24.] N. Patel et al., "An Intelligent VM Assignment Algorithm for Load Balancing," International Journal of Grid and High Performance Computing, vol. 13, no. 4, pp. 45–60, 2021.
[25.] Y. Sun and T. Wang, "Brownout-Based Load Balancing with Self-Adaptive Clouds," Journal of Parallel and Distributed Computing, vol. 165, pp. 56–70, 2022.
[26.] R. K. Singh, "Survey on Scheduling and Provisioning in Cloud," *International Journal of Cloud Applications and Computing, vol. 12, no. 1, pp. 34–52, 2022.
[27.] H. Mehta and S. Patel, "A VM-Aware Load Balancing Technique for Cloud Systems," *International Journal of Computers and Applications, vol. 44, no. 4, pp. 301–310, 2022.
[28.] W. Wei et al., "A Greedy-Based VM Load Balancing Algorithm for Cloud," *IEEE Transactions on Services Computing, vol. 15, no. 2, pp. 690–701, 2022.
[29.] D. Sharma, "Load Balancing Algorithms: A Comparative Review," Journal of Cloud.
[30.] B. Gupta et al., "Honeybee-Inspired Load Balancing Algorithm for Cloud-Based Health Services," IEEE Access, vol. 9, pp. 83753–83765, 2021.
[31.] M. T. Thai and T. D. Nguyen, "A prepartition load balancing algorithm for cloud computing," Int. J. Cloud Comput., vol. 4, no. 3, pp. 247–261, 2015.
[32.] X. Wang et al., "A distributed autonomic management system for load balancing in cloud data centers," IEEE Trans. Cloud Comput., vol. 10, no. 3, pp. 621–635, 2022.

[33.] M. R. Rahman et al., "Brownout aware load balancing for cloud applications," in Proc. IEEE IC2E, 2019, pp. 156–161.
[34.] T. V. D. Ho et al., "A reinforcement learning-based brownout approach for adaptive cloud load balancing," Future Gener. Comput. Syst., vol. 114, pp. 192–208, 2021.
[35.] S. Dutta et al., "A honeybee inspired load balancing technique for cloud computing," in Proc. Int. Conf. Adv. Comput. Commun. Inform., 2018, pp. 1573–1578.
[36.] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization," IEEE Trans. Cloud Comput., vol. 1, no. 2, pp. 158–171, 2013.
[37.] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurr. Comput. Pract. Exp., vol. 24, no. 13, pp. 1397–1420, 2012.
[38.] F. Faniyi et al., "Predictive virtual machine placement and load balancing in cloud computing environments," J. Cloud Comput., vol. 10, no. 1, pp. 1–18, 2021.
[39.] S. Singh and I. Chana, "QRSF: QoS-aware resource scheduling framework in cloud computing," J. Supercomput., vol. 71, pp. 241–292, 2015.
[40.] S. Pandey et al., "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in Proc. IEEE AINA, 2010, pp. 400–407.
[41.] G. K. Gupta et al., "Load rebalancing for distributed file systems in clouds," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 12, pp. 2425–2434, 2013.
[42.] J. Liu et al., "Dynamic data placement strategy for Hadoop with load balance in cloud computing," J. Supercomput., vol. 74, pp. 5901–5922, 2018.
[43.] T. Wood et al., "CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines," IEEE Netw., vol. 25, no. 5, pp. 56–65, 2011.
[44.] M. Satyanarayanan et al., "Edge analytics in the Internet of Things," IEEE Pervasive Comput., vol. 14, no. 2, pp. 24–31, 2015.