# Resource Allocation and Scheduling Techniques in Cloud Computing: A Comprehensive Review

**Archana Mani**

Assistant Professor, Jagannath University

**Abstract**

Efficient resource allocation and scheduling are critical for optimizing performance, minimizing costs, and ensuring user satisfaction in cloud computing environments. This paper provides a comprehensive review of existing techniques, examining the challenges and advancements in this crucial domain. We explore the fundamental principles of resource management, including key performance indicators, service models, and the trade-offs between static and dynamic allocation. A detailed analysis of various scheduling algorithms, including heuristic, metaheuristic, machine learning-based, and hybrid approaches, is presented. Furthermore, we discuss advanced considerations such as fault tolerance, energy efficiency, security, and quality of service. Finally, we identify key research directions for future advancements, including the integration of AI and machine learning, the handling of heterogeneous resources, and the challenges posed by emerging technologies like serverless computing and edge computing. This review aims to provide a valuable resource for researchers and practitioners seeking to understand the current state-of-the-art and navigate the complexities of resource management in the evolving cloud computing landscape.

## Introduction:

### The Challenge of Efficient Resource Management in Cloud Computing

This paper provides a comprehensive review of resource allocation and scheduling techniques in cloud computing, examining the challenges and advancements in this critical area. Efficient resource management is paramount for cloud providers to maximize profitability while ensuring user satisfaction [1], [2]. The dynamic and heterogeneous nature of cloud environments, however, presents significant complexities [2], [1]. These complexities stem from the constantly fluctuating demands of numerous concurrent users, each with varying resource requirements and service level expectations. Furthermore, the underlying infrastructure itself is heterogeneous, comprising a diverse mix of hardware, software, and network components. This necessitates sophisticated algorithms and strategies to optimize resource utilization, minimize costs, and ensure the timely delivery of services. This review will analyze various approaches to resource allocation and scheduling, highlighting their strengths, weaknesses, and suitability for different scenarios, ultimately aiming to provide a clear understanding of the current state-of-the-art and future research directions in this crucial field.

## The Fundamentals of Resource Allocation and Scheduling

### A. Defining the Problem: Resource Constraints and User Demands

Cloud computing's core value proposition is on-demand access to a vast pool of resources [3]. This seemingly limitless availability, however, masks the underlying reality of finite resources. Efficiently allocating these finite resources to numerous concurrent users with diverse needs constitutes a complex optimization problem [3], [4]. Understanding this fundamental constraint is key to appreciating the challenges inherent in cloud resource management. Key performance indicators (KPIs) are used to evaluate the effectiveness of resource allocation and scheduling algorithms. Makespan, representing the total time required to complete all tasks, is a critical metric. Throughput, measuring the number of tasks completed per unit of time, reflects the overall system efficiency. Response time, the time elapsed between a request and its completion, directly impacts user experience. Finally, resource utilization, indicating the proportion of allocated resources actively in use, reflects the efficiency of resource allocation [3], [4], [5]. Optimizing these KPIs simultaneously presents a significant challenge, as improvements in one often come at the expense of others. For instance, minimizing makespan might require over-provisioning resources, impacting resource utilization and cost-effectiveness. Therefore, effective resource management necessitates a balanced approach that considers all relevant KPIs within the context of specific application requirements and service level agreements.

### B. Service Models and Resource Types: IaaS, PaaS, SaaS and Beyond

Different cloud service models (IaaS, PaaS, SaaS) [6] introduce variations in resource allocation challenges. IaaS (Infrastructure as a Service) provides users with raw computing resources, such as virtual machines (VMs), storage, and networking, offering maximum flexibility but also requiring more management effort. PaaS (Platform as a Service) provides a platform for developing and deploying applications, abstracting away some of the underlying infrastructure management. SaaS (Software as a Service) provides ready-to-use applications, requiring minimal management from the user. The resource allocation strategies employed differ significantly across these models. In IaaS, efficient VM placement and resource allocation within VMs are critical for optimization [7], [6]. PaaS requires managing resources at a higher level, focusing on application deployment and scaling. SaaS requires resource allocation at the provider level, ensuring sufficient capacity to meet user demand. Beyond these core models, newer paradigms such as serverless computing and edge computing introduce further complexities [8]. Serverless computing eliminates the need for managing servers, but efficient function allocation and scaling remain important considerations. Edge computing distributes processing closer to data sources, requiring decentralized resource management and optimized data transfer between edge nodes and the cloud. Regardless of the service model, efficient resource management requires careful consideration of various resource types, including CPU, memory, storage, and network bandwidth [7], [6]. The impact of containerization [7] is particularly significant. Containers offer lightweight virtualization, improving resource utilization compared to traditional VMs. However, efficient container orchestration and resource allocation remain crucial challenges. The choice of service model and the specific resource types involved significantly shape the design and implementation of resource allocation and scheduling algorithms.

## Static vs. Dynamic Resource Allocation: A Comparative Analysis

### A. Static Allocation: Simplicity vs. Inefficiency

Static resource allocation involves pre-allocating resources to users or applications based on predicted needs [7]. This approach offers a degree of simplicity and predictability, making it easier to manage and plan resource usage. However, its inherent inflexibility often leads to suboptimal resource utilization. Over-provisioning, where more resources are allocated than actually needed, results in wasted resources and increased costs. Conversely, under-provisioning, where insufficient resources are allocated, can lead to performance bottlenecks and SLA violations [7], [9]. Static allocation is generally suitable for predictable workloads with consistent resource demands. However, the dynamic nature of most cloud environments makes static allocation less appropriate for many scenarios. The simplicity of static allocation often comes at the cost of significant inefficiency in resource utilization, especially in environments with fluctuating workloads. The inability to adapt to changing demands makes static allocation a less attractive option in many modern cloud computing contexts [7], [9].

### B. Dynamic Allocation: Adaptability and Complexity

Dynamic resource allocation, in contrast, adjusts resource allocation in real-time based on current demand and availability [7]. This adaptive approach offers significantly improved resource utilization and efficiency compared to static allocation. By continuously monitoring resource consumption and user requests, dynamic allocation ensures resources are allocated only when and where needed. This avoids the waste associated with over-provisioning in static allocation. Furthermore, dynamic allocation can adapt to sudden spikes in demand, preventing performance degradation and SLA violations. However, this adaptability comes at the cost of increased complexity. Designing and implementing efficient dynamic allocation algorithms requires sophisticated techniques to handle continuous monitoring, resource allocation decisions, and potential resource contention. The management overhead associated with dynamic allocation can be substantial, particularly in large-scale cloud environments [7], [9]. The choice between static and dynamic allocation involves a trade-off between simplicity and efficiency. Static allocation is simpler to manage but less efficient, while dynamic allocation is more complex but significantly more efficient in resource utilization. The optimal choice depends on the specific characteristics of the workload, the desired level of efficiency, and the available management resources.

## Resource Scheduling Algorithms:

This section categorizes and analyzes various resource scheduling algorithms, grouping them by their underlying approaches. The selection of an appropriate algorithm depends heavily on the specific characteristics of the cloud environment, the nature of the workloads, and the desired performance metrics.

### A. Heuristic Algorithms: Rule-Based Approaches

Heuristic algorithms utilize rules of thumb or approximations to find good solutions within reasonable time constraints [10]. These algorithms are often simpler to implement than more sophisticated optimization techniques, making them attractive for environments with limited computational resources or stringent real-time requirements. Examples include First-Come, First-Served (FCFS), which allocates resources in the order of arrival; Shortest Job First (SJF), which prioritizes shorter tasks; and Longest Expected Processing Time (LEPT), which prioritizes longer tasks to minimize preemption [10]. While these algorithms can provide acceptable performance in certain scenarios, they often fail to achieve optimal resource utilization and may suffer from starvation or unfairness, where certain tasks are consistently delayed or neglected [10], [5]. The simplicity of heuristic algorithms is often counterbalanced by their inability to guarantee optimal solutions or adapt to dynamic changes in the workload. Their effectiveness is highly dependent on the specific characteristics of the workload and the underlying resource environment.

### B. Metaheuristic Algorithms: Optimization Techniques

Metaheuristic algorithms employ iterative processes to explore the solution space, aiming to find near-optimal solutions [11], [12]. These algorithms are generally more computationally intensive than heuristic algorithms but are capable of achieving significantly better resource utilization and performance. Examples include genetic algorithms [13], [12], which mimic biological evolution to find optimal solutions; particle swarm optimization [14], [15], which simulates the social behavior of bird flocks; and ant colony optimization [3], [14], which models the foraging behavior of ants. Many other nature-inspired algorithms have also been applied to cloud resource scheduling, such as the Honey Badger Algorithm [12] and the Locust-Inspired Algorithm [16]. The strengths of metaheuristic algorithms lie in their ability to explore a wide range of solutions and escape local optima, leading to improved resource allocation and scheduling efficiency [11], [12], [13]. However, their computational cost can be a limiting factor, particularly in real-time environments. The choice of a specific metaheuristic algorithm often involves a trade-off between solution quality and computational complexity.

### C. Machine Learning-Based Scheduling: Data-Driven Approaches

Machine learning techniques leverage historical data to learn patterns and predict future resource demands [17]. This data-driven approach allows for more adaptive and proactive resource allocation and scheduling. By analyzing past workload patterns, machine learning models can predict future resource needs and proactively allocate resources to prevent bottlenecks. Reinforcement learning [17], [18], where an agent learns to make optimal decisions through trial and error, is particularly well-suited for dynamic resource allocation. Deep learning [19] techniques can capture complex relationships within the data, potentially leading to even more accurate predictions and improved resource allocation decisions. The integration of machine learning into resource scheduling allows for a more intelligent and responsive system capable of adapting to changing conditions and optimizing resource utilization in real-time [8]. However, the effectiveness of machine learning-based scheduling depends heavily on the quality and quantity of available data. The development and training of accurate models can be computationally expensive and require significant expertise.

### D. Hybrid Approaches: Combining the Best of Different Worlds

Hybrid approaches combine multiple algorithms or techniques to leverage their complementary strengths [20], [21]. This approach often leads to improved performance compared to using a single algorithm alone. For example, combining a heuristic algorithm for initial resource allocation with a metaheuristic algorithm for fine-tuning can result in a system that is both efficient and effective. Similarly, integrating machine learning with optimization techniques can improve the accuracy of predictions and the efficiency of resource allocation [22]. The design of a hybrid approach requires careful consideration of the strengths and weaknesses of each component algorithm and how they can be effectively integrated to achieve the desired overall performance. The complexity of hybrid approaches can be greater than that of single-algorithm approaches, but the potential for improved performance often justifies this increased complexity [20], [21], [15].

## Advanced Considerations in Cloud Resource Management

### A. Fault Tolerance and Resilience: Handling Resource Failures

The dynamic nature of cloud environments makes resource failures inevitable [23]. Building fault-tolerant resource allocation and scheduling systems is crucial for ensuring the reliability and availability of cloud services. Techniques such as replication, where tasks are duplicated across multiple resources, provide redundancy and protect against individual resource failures. Migration, the ability to move tasks from a failed resource to a healthy one, allows for continued operation even in the face of failures. Checkpointing, saving the state of a task at regular intervals, enables recovery from failures with minimal data loss [23], [24], [25]. However, these fault-tolerance techniques introduce overheads in terms of increased resource consumption and management complexity. The choice of fault-tolerance strategy involves a trade-off between the level of resilience and the associated performance overhead [23], [24]. The optimal strategy depends on the criticality of the application and the acceptable level of performance degradation during failures.

### B. Energy Efficiency: Minimizing Power Consumption

Cloud data centers consume vast amounts of energy [3], making energy efficiency a critical concern for both economic and environmental reasons. Energy-aware resource allocation and scheduling techniques aim to minimize power consumption while maintaining acceptable performance. Server consolidation, running multiple virtual machines on a single physical server, reduces the number of active servers and lowers overall power consumption. Power capping, limiting the maximum power consumption of individual servers, prevents excessive energy usage. Dynamic voltage scaling, adjusting the voltage supplied to processors based on workload demand, further reduces power consumption [26], [27], [21]. These techniques require careful balancing to avoid performance degradation while achieving significant energy savings. The development of energy-efficient algorithms and hardware is crucial for achieving sustainable cloud computing.

### C. Security and Privacy: Protecting Cloud Resources

Security and privacy are paramount in cloud computing [19]. Resource allocation and scheduling must incorporate security considerations to prevent unauthorized access, data breaches, and other security threats. Secure resource access control mechanisms ensure that only authorized users can access specific resources. Data encryption protects sensitive data both in transit and at rest. Intrusion detection systems monitor resource usage for suspicious activity, alerting administrators to potential threats [19], [8]. Security mechanisms should be integrated into resource allocation and scheduling algorithms to ensure a secure and reliable cloud environment. This requires a holistic approach that addresses security at all levels, from the underlying infrastructure to the applications running on the cloud.

### D. Quality of Service (QoS): Meeting User Expectations

Cloud providers often have service-level agreements (SLAs) with users [1], specifying performance guarantees such as response time, throughput, and availability. Resource allocation and scheduling play a critical role in meeting these QoS requirements. Algorithms should be designed to prioritize tasks based on their QoS requirements, ensuring that critical applications receive the necessary resources to meet their SLAs [28], [1], [29]. Monitoring and management tools are essential for tracking QoS metrics and identifying potential issues. Proactive resource management, anticipating and addressing potential bottlenecks before they impact QoS, is crucial for maintaining user satisfaction and meeting SLA commitments. A robust QoS management system is essential for ensuring the success and reliability of cloud computing services.

## Tools and Simulators for Evaluating Resource Allocation and Scheduling Algorithms

Evaluating the performance of resource allocation and scheduling algorithms requires specialized tools and simulators. CloudSim [30], [12] is a widely used open-source simulator that provides a platform for modeling and evaluating various cloud resource management strategies. It allows researchers to simulate different cloud environments, workloads, and algorithms, enabling comparative analysis and performance evaluation [30], [12]. iCanCloud [31] is another popular simulator that offers flexibility and scalability, allowing for the simulation of large-scale cloud environments. These simulators provide a cost-effective and efficient way to test and compare different algorithms before deploying them in real-world environments [30], [31]. The appropriate choice of simulator depends on the specific research questions and the complexity of the simulated environment. The use of these tools is crucial for rigorous validation and comparison of different resource management approaches.

**Future Research Directions**

Despite significant progress, several open research challenges remain in cloud resource allocation and scheduling. The development of more sophisticated predictive models is crucial for accurately forecasting resource demands and proactively managing resource allocation [32]. Improved handling of heterogeneous resources, addressing the diverse capabilities and characteristics of different resources within the cloud environment, is also a key area for future research [33]. The integration of emerging technologies such as serverless computing and edge computing presents new opportunities and challenges for resource management [8]. Serverless computing requires efficient function scaling and allocation, while edge computing necessitates decentralized resource management and optimized data transfer. The challenges posed by quantum computing [34], a rapidly developing field with the potential to revolutionize computing, require the development of new resource allocation and scheduling strategies tailored to the unique characteristics of quantum computers. The exploration of novel algorithms, the integration of AI and machine learning, and the development of robust and scalable tools and simulators will be crucial for addressing these challenges and driving further advancements in cloud resource management.

**Conclusion: Towards Optimal Resource Management in the Cloud**

This paper has provided a comprehensive overview of resource allocation and scheduling techniques in cloud computing. While significant progress has been made in developing efficient and effective algorithms, several open challenges remain. Future research should focus on developing more adaptive, efficient, and secure resource management strategies to meet the ever-increasing demands of cloud computing. The integration of AI and machine learning techniques holds significant promise for achieving optimal resource utilization and maximizing the efficiency of cloud infrastructures [35]. The ongoing evolution of cloud computing, with the emergence of new service models and technologies, will continue to drive innovation and research in this critical area. Addressing the challenges of heterogeneity, fault tolerance, energy efficiency, security, and QoS will be key to realizing the full potential of cloud computing and ensuring its continued growth and success.

**REFERENCE**

1. Singh, Harvinder, Bhasin, Anshu, Kaveri, Parag Ravikant, and Chavan, Vinay. 2020. "Cloud Resource Management: Comparative Analysis and Research Issues". None. https://doi.org/None
2. Manzoor, Muhammad Faraz, Abid, Adnan, Farooq, Muhammad Shoaib, Nawaz, Naeem A., and Farooq, Uzma. 2020. "Resource Allocation Techniques in Cloud Computing: A Review and Future Directions". Kaunas University of Technology. https://doi.org/10.5755/j01.eie.26.6.25865
3. Gogula, Et Al. Sreenivasulu. 2023. "A Study Resource Optimization Techniques Based Job Scheduling in Cloud Computing". International Journal on Recent and Innovation Trends in Computing and Communication. https://doi.org/10.17762/ijritcc.v11i10.8746
4. Lakshmi, J., Reddy, V., and Naresh, T.. 2020. "Resource optimization and task scheduling in cloud computing". None. https://doi.org/None
5. Patel, Swachil J. and Bhoi, Upendra R.. 2013. "Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review". None. https://doi.org/None

6. Shekokare, Rajashri. S., Kha, Rais Abdul Hamid, Baldhare, Pawan, and Muqeem, Mohammad.. 2024. "Convinient Load Balancing by Dynamic Memory Allocation for Cloud Computing Model in Virtual Machines". None. https://doi.org/10.38124/ijisrt/ijisrt24may963
7. Vhatkar, K. and Bhole, G.. 2022. "A comprehensive survey on container resource allocation approaches in cloud computing: State-of-the-art and research challenges". International Conference on Wirtschaftsinformatik. https://doi.org/10.3233/web-210474
8. Kokila, R. and Rammohan, D. S. R.. 2023. "An Advanced Cloud Data Streaming Framework for Optimized Container Resource Allocation, Job Scheduling, And Security Enhancement". None. https://doi.org/10.52783/tjjpt.v44.i3.577
9. Shukur, Hanan M., Zeebaree, Subhi R. M., Zebari, Rizgar R., Zeebaree, Diyar Qader, Ahmed, Omar M., and Salih, Azar Abid. 2020. "Cloud Computing Virtualization of Resources Allocation for Distributed Systems". None. https://doi.org/10.38094/jastt1331
10. Gawali, Mahendra Bhatu and Shinde, Subhash K.. 2018. "Task scheduling and resource allocation in cloud computing using a heuristic approach". Springer Nature. https://doi.org/10.1186/s13677-018-0105-8
11. Garg, M., Kaur, A., and Dhiman, Gaurav. NaN. "A Novel Resource Allocation and Scheduling Based on Priority Using Metaheuristic for Cloud Computing Environment". None. https://doi.org/10.4018/978-1-7998-5040-3.ch008
12. Rajagopal, R., Arunarani, AR., Arivarasi, A., Ingle, Anup, T, Ravichandran, and Prakash, R. V.. 2023. "Enhanced Honey Badger Algorithm for Resource Allocation and Task Scheduling in Cloud Environment". None. https://doi.org/10.1109/ICOSEC58147.2023.10275908
13. Li, Xunzhang, Wei, Shiwei, Ke, Jie, and Zhou, Huiyi. 2022. "A new multi-subpopulation co-evolutionary genetic algorithm for cloud resource scheduling". International Conference on Computational Intelligence and Security. https://doi.org/10.1109/CIS58238.2022.00078
14. Saleh, Heba, Nashaat, Heba, Saber, Walaa, and Harb, Hany. 2018. "IPSO Task Scheduling Algorithm for Large Scale Data in Cloud Computing Environment". Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/access.2018.2890067
15. Kamalinia, Amin and Ghaffari, Ali. NaN. "Hybrid Task Scheduling Method for Cloud Computing by Genetic and PSO Algorithms". None. https://doi.org/None
16. Saraswathy, S., Malathi, J., Subiksha, N., Lakshminarayanan, S., Kumar, K., and Livinesh, L.. 2024. "A Hybrid Strategy Integrating Ant Colony Optimization and Locust-Inspired Algorithm (HACO-LA) to Boost Efficiency and Performance in cloud Resource Management". None. https://doi.org/10.1109/PARC59193.2024.10486486
17. Li, Yansong. 2022. "Research on cloud computing resource scheduling based on machine learning". None. https://doi.org/10.1109/MLISE57402.2022.00090
18. Mubin, Md Mehefujur Rahman, Ullah, Sajjad, Purification, James Anthony, and Islam, Md. Motaharul. 2023. "Energy Aware Scheduling and Resource Allocation for Virtual Machine". None. https://doi.org/10.1109/STI59863.2023.10464530
19. Bal, Prasanta Kumar, Mohapatra, Sudhir Kumar, Das, Tapan Kumar, Srinivasan, Kathiravan, and Hu, YuhChung. 2022. "A Joint Resource Allocation, Security with Efficient Task Scheduling in Cloud Computing Using Hybrid Machine Learning Techniques". Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/s22031242

20. Paulraj, D., Sethukarasi, T., Subramani, Neelakandan, Prakash, M., and Baburaj, E.. 2023. "An Efficient Hybrid Job Scheduling Optimization (EHJSO) approach to enhance resource search using Cuckoo and Grey Wolf Job Optimization for cloud environment". PLoS ONE. https://doi.org/10.1371/journal.pone.0282600
21. Kumar, M. S., Hussain, D. M., Rohini, M., and Manoj, S. O.. 2024. "Advanced Hybrid Optimization Algorithms for Energy Efficient Cloud Resource Allocation". Radio Science. https://doi.org/10.1029/2024rs008012
22. Gongada, Dr. Taviti Naidu, Desale, Prof. Girish Bhagwant, Ghodake, S., Sridharan, Dr. K., Rao, Dr.Vuda Sreenivasa, and El-Ebiary, D. Y. A.. NaN. "Optimizing Resource Allocation in Cloud Environments using Fruit Fly Optimization and Convolutional Neural Networks". International Journal of Advanced Computer Science and Applications. https://doi.org/10.14569/ijacsa.2024.01505119
23. Chawla, Sonu and Kaur, Amandeep. 2024. "Fault-Tolerant Heuristic Task Scheduling Algorithm for Efficient Resource Utilization in Cloud Computing". None. https://doi.org/10.1109/AUTOCOM60220.2024.10486076
24. Sathiyamoorthi, V., Keerthika, P., Suresh, P., Zhang, Zuopeng, Rao, Adiraju Prasanth, and Logeswaran, K.. 2021. "Adaptive Fault Tolerant Resource Allocation Scheme for Cloud Computing Environments". IGI Global. https://doi.org/10.4018/joeuc.20210901.oa7
25. Soltanshahi, Minoo. 2016. "Improving the palbimm scheduling algorithm for fault tolerance in cloud computing". None. https://doi.org/None
26. Qiu, Meikang, Chen, Zhi, Ming, Zhong, Qin, Xiao, and Niu, Jianwei. 2014. "Energy-Aware Data Allocation With Hybrid Memory for Mobile Cloud Systems". Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/jsyst.2014.2345733
27. Sonkar, S. K. and Kharat, M.. NaN. "A review on resource allocation and VM scheduling techniques and a model for efficient resource management in cloud computing environment". None. https://doi.org/10.1109/ICTBIG.2016.7892646
28. Murali, Juliet A.. NaN. "Ecient Resource Allocation in Cloud Computing Using Hungarian Optimization in Aws". None. https://doi.org/None
29. Katti, Anvesha. 2024. "An Analysis of Low-Overhead and High Efficiency Task Scheduling in the Cloud and Fog Environments". None. https://doi.org/10.1109/AUTOCOM60220.2024.10486130
30. Tani, Hicham Gibet and Amrani, Chaker El. 2016. "Cloud Computing CPU Allocation and Scheduling Algorithms using CloudSim Simulator". Institute of Advanced Engineering and Science (IAES). https://doi.org/10.11591/ijece.v6i4.pp1866-1879
31. Durga, A. and Madhumathi, R.. NaN. "Priority Based Fairshare Scheduling Algorithm in Cloud Computing Environment". None. https://doi.org/None
32. Sohani, Mayank and Jain, S. C.. 2021. "A Predictive Priority-Based Dynamic Resource Provisioning Scheme With Load Balancing in Heterogeneous Cloud Computing". Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/access.2021.3074833
33. Zhang, AnNing, Chu, ShuChuan, Song, Pei-Cheng, Wang, Hui, and Pan, JengShyang. 2022. "Task Scheduling in Cloud Computing Environment Using Advanced Phasmatodea Population Evolution Algorithms". Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/electronics11091451

34. Lu, Binhan, Chen, Zhaoyun, and Wu, Yuchun. 2024. "QSRA: A QPU Scheduling and Resource Allocation Approach for Cloud-Based Quantum Computing". None. https://doi.org/None
35. Karamthulla, Musarath Jahan, Narkarunai, Jesu, Malaiyappan, Arasu, and Tillu, Ravish. 2023. "Optimizing Resource Allocation in Cloud Infrastructure through AI Automation: A Comparative Study". Online (Weston, Conn.). https://doi.org/10.60087/jklst.vol2.n2.p326