International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

A Hybrid Task Scheduling Algorithm for Improving Resource Utilization in Cloud Environments

1st K. Mani deepika 2nd Mummidi Sharanya 3rd K.Sarath
Dept. Computer Application, Aditya University, Surampalem, India.
deepikakmani@gmail.com saranyamummidi@gmail.com saratkondepudi6@gmail.com
4th Chintakula Sai Surya 5th K Gowtham Reddy
Dept. Computer Application, Aditya University, Surampalem, India
suryavicky.143225@gmail.com gowthamreddyse@gmail.com

***Abstract*—Cloud computing has emerged as a fundamental technology that enables scalable and on-demand access to computing resources such as storage, processing power, and networking. However, efficient task scheduling remains one of the major challenges in cloud environments due to the heterogeneous nature of resources and dynamic workload conditions. Inefficient scheduling can lead to poor resource utilization, increased execution time, and higher operational costs. To address these issues, this study proposes a hybrid task scheduling algorithm designed to improve resource utilization and optimize task execution in cloud computing environments. The proposed approach integrates heuristic scheduling techniques with machine learning-based optimization to dynamically allocate tasks to available virtual machines based on resource availability, workload characteristics, and execution priorities.**

The algorithm aims to minimize makespan, improve throughput, and balance workload across computing nodes. Experimental evaluation is conducted using a simulated cloud environment with multiple datasets representing heterogeneous workloads. The results demonstrate that the proposed hybrid scheduling model significantly improves resource utilization and reduces task completion time compared to traditional scheduling methods such as First Come First Serve (FCFS), Round Robin, and Min-Min algorithms. The proposed framework also shows improved load balancing and scalability under varying workload conditions.

These findings suggest that hybrid scheduling techniques can effectively enhance the efficiency and performance of cloud computing infrastructures. Future research may explore integration with reinforcement learning models and real-time workload prediction mechanisms to further optimize scheduling decisions.
Keywords:Cloud Computing, Task Scheduling, Resource Utilization, Hybrid Scheduling Algorithm, Virtual Machines, Load Balancing, Distributed Computing

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Cloud computing has revolutionized modern computing infrastructures by providing scalable, flexible, and cost-effective computing services over the internet. Organizations increasingly rely on cloud platforms to store data, deploy applications, and perform large-scale computations. The fundamental concept behind cloud computing is the availability of shared computing resources that can be dynamically allocated to users based on demand. These resources include virtual machines, storage units, and networking components that collectively support the execution of computational tasks [1] [9]. Despite the advantages offered by cloud environments, efficient task scheduling remains a critical challenge. Task scheduling refers to the process of assigning tasks to available computing resources in a way that optimizes system performance and resource utilization. In large-scale cloud systems, thousands of tasks may arrive simultaneously, each with different computational requirements and execution priorities. Without effective scheduling strategies, resources may remain underutilized while some nodes become overloaded, resulting in increased response time and reduced system performance.

Traditional scheduling algorithms such as First Come First Serve, Round Robin, and Priority Scheduling were initially designed for centralized computing systems. These algorithms often fail to handle the dynamic and distributed nature of cloud environments effectively. As cloud infrastructures continue to grow in scale and complexity, more advanced scheduling strategies are required to manage resource allocation efficiently [2].

Recent research has explored the use of hybrid scheduling approaches that combine heuristic methods with intelligent optimization techniques. Hybrid algorithms can leverage the strengths of multiple scheduling strategies to achieve better performance outcomes. For example, heuristic algorithms can quickly generate initial scheduling decisions, while machine learning models can refine these decisions based on historical workload patterns and resource availability.

Resource utilization is another crucial aspect of cloud computing efficiency. Poor resource utilization leads to increased operational costs for cloud service providers and reduced performance for end users. Efficient scheduling algorithms must ensure that computing resources are evenly distributed among tasks while minimizing idle time across virtual machines [6] [8].

This research proposes a hybrid task scheduling algorithm designed to improve resource utilization and task execution efficiency in cloud environments. The proposed algorithm combines heuristic scheduling techniques with adaptive resource allocation mechanisms to dynamically assign tasks to available virtual machines. By considering factors such as task length, resource capacity, and system workload, the proposed framework aims to achieve better load balancing and reduced execution time.

Identify applicable funding agency here. If none, delete this.

The remainder of this paper is organized as follows. Section 2 reviews existing research on cloud task scheduling techniques. Section 3 presents the proposed hybrid scheduling methodology. Section 4 describes the experimental setup and implementation details. Section 5 discusses the experimental results and performance evaluation. Finally, Section 6 concludes the study and outlines potential directions for future research.

II. Background Study

A. Related Work

Task scheduling in cloud computing has been extensively studied due to its significant impact on system performance and resource utilization. Researchers have proposed numerous algorithms aimed at improving scheduling efficiency and reducing task completion time in distributed environments. These algorithms can generally be categorized into heuristic-based, meta-heuristic-based, and machine learning-based approaches.

Heuristic scheduling algorithms are among the earliest methods used for task allocation in distributed computing systems. Algorithms such as First Come First Serve (FCFS), Round Robin, and Min-Min are commonly implemented due to their simplicity and low computational overhead. FCFS schedules tasks in the order they arrive, while Round Robin distributes tasks among available resources in a cyclic manner. The Min-Min algorithm assigns tasks with the minimum completion time to the fastest available resource. Although these algorithms are easy to implement, they often fail to achieve optimal resource utilization in large-scale cloud environments [8].

To overcome the limitations of heuristic methods, researchers have introduced meta-heuristic algorithms inspired by natural processes. Techniques such as Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization have been widely applied to cloud scheduling problems. These algorithms aim to find near-optimal solutions by exploring a large search space of possible task allocations. For instance, genetic algorithms use evolutionary processes such as selection, crossover, and mutation to optimize scheduling decisions. Similarly, particle swarm optimization simulates collective behavior among particles to identify optimal resource allocations. In recent years, machine learning techniques have been integrated into task scheduling frameworks to improve decision-making capabilities. Machine learning models can analyze historical workload patterns and predict future resource demands. These predictions allow scheduling systems to dynamically adjust resource allocation strategies in real time. Deep learning models have also been explored for workload classification and performance prediction in cloud systems [7].

Hybrid scheduling approaches have gained considerable attention due to their ability to combine the strengths of multiple optimization techniques. Hybrid algorithms typically integrate heuristic scheduling methods with machine learning or meta-heuristic optimization techniques. By leveraging complementary strategies, hybrid algorithms can achieve improved scheduling performance compared to traditional methods.

Several studies have demonstrated the effectiveness of hybrid scheduling models in cloud computing environments. These models often incorporate load balancing mechanisms to distribute tasks evenly across available computing nodes. Load balancing ensures that no single resource becomes overloaded while others remain idle. As a result, overall system performance and resource utilization can be significantly improved. Despite the progress made in this area, challenges still remain in designing scalable and efficient scheduling algorithms capable of handling heterogeneous workloads and dynamic resource conditions. Therefore, the development of hybrid scheduling strategies continues to be an active area of research in cloud computing.

III. Proposed Methodology

The proposed system introduces a hybrid task scheduling framework designed to improve resource utilization in cloud computing environments. The framework integrates heuristic scheduling techniques with dynamic resource monitoring mechanisms to optimize task allocation across virtual machines. The primary objective of the proposed model is to reduce task execution time while maintaining balanced resource utilization across the cloud infrastructure [5].

The scheduling process begins with task submission by users through a cloud interface. Each task is characterized by parameters such as computational requirements, execution priority, and estimated processing time. These tasks are then forwarded to a scheduling manager responsible for analyzing task attributes and determining appropriate resource allocations.

The proposed hybrid scheduling algorithm operates in two main stages. In the first stage, a heuristic scheduling mechanism generates an initial mapping of tasks to available virtual machines based on resource capacity and estimated execution time. This stage aims to quickly distribute tasks across computing nodes while minimizing initial scheduling overhead.

In the second stage, the algorithm applies a dynamic optimization strategy that evaluates resource utilization across the cloud environment. If certain nodes become overloaded while others remain underutilized, the system redistributes tasks to achieve better load balancing. This adaptive scheduling process ensures efficient resource allocation and prevents performance degradation [3] [4].

The system continuously monitors resource usage metrics such as CPU utilization, memory consumption, and network bandwidth. These metrics are used to update scheduling decisions in real time, allowing the system to adapt to changing workload conditions.

The proposed hybrid scheduling model is designed to improve three key performance metrics: makespan, resource utilization, and system throughput. By combining heuristic and adaptive scheduling strategies, the framework aims to

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

achieve higher efficiency compared to conventional scheduling algorithms

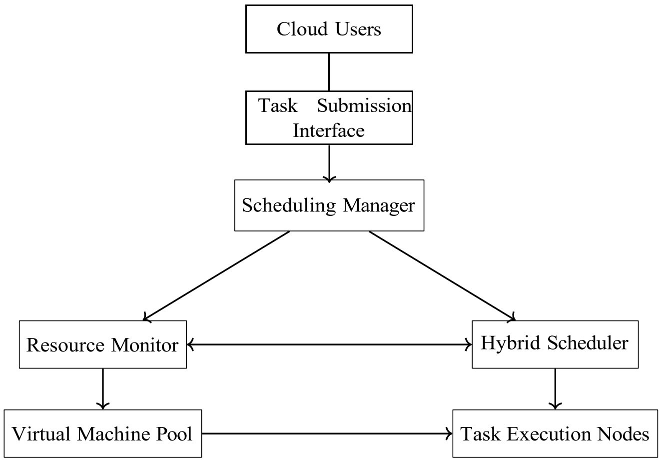

Fig. 1. Architecture of the Hybrid Cloud Task Scheduling Framework

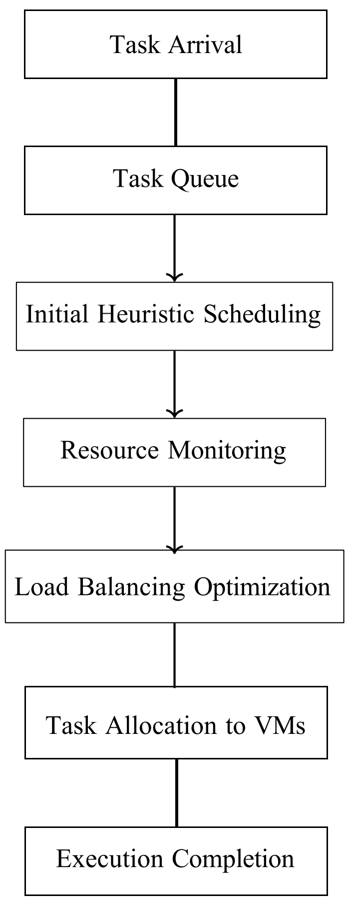

Fig. 2. Workflow of the Hybrid Task Scheduling Algorithm

IV. Experimental Setup and Implementation

The experimental evaluation of the proposed hybrid task scheduling algorithm was conducted in a simulated cloud computing environment to analyze its effectiveness in improving

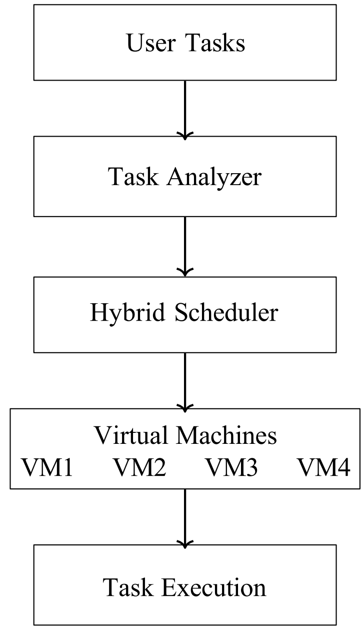

Fig. 3. System Model for Cloud Task Scheduling

resource utilization and task execution performance. The simulation environment was designed to emulate a real-world cloud infrastructure consisting of multiple virtual machines operating under heterogeneous resource configurations. The experiments were carried out using a cloud simulation framework that supports task scheduling and resource allocation experiments. The simulated environment consisted of multiple physical hosts, each capable of running several virtual machines with varying CPU capacities, memory sizes, and processing speeds. A large set of computational tasks was generated to simulate realistic workload conditions in a cloud environment. These tasks varied in execution length, resource requirements, and priority levels. The proposed hybrid scheduling algorithm was implemented within the scheduler module of the simulation framework, enabling dynamic task allocation and resource monitoring during runtime.

The scheduling algorithm evaluates each incoming task by analyzing its computational complexity and estimated execution time. Based on this analysis, the scheduler initially assigns tasks using a heuristic strategy that identifies suitable virtual machines with available processing capacity. Once tasks are distributed, the system continuously monitors the resource utilization levels of each virtual machine. If certain nodes experience excessive workload while others remain underutilized, the hybrid scheduler dynamically redistributes tasks to maintain balanced system performance.

Several performance metrics were used to evaluate the effectiveness of the proposed scheduling approach. These metrics include makespan, resource utilization, throughput, and average response time. Makespan refers to the total time required to complete all submitted tasks, while resource utilization indicates the percentage of computing resources actively used during execution. Throughput represents the number of tasks

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

completed per unit time, and response time measures the delay experienced by tasks before execution begins.

To provide a comprehensive evaluation, the proposed algorithm was compared with several traditional scheduling algorithms commonly used in cloud computing systems. These include First Come First Serve, Round Robin, and the Min-Min scheduling algorithm. Each algorithm was tested under identical workload conditions to ensure a fair comparison of performance outcomes [10].

The experiments were conducted under varying workload scenarios to analyze the scalability of the proposed scheduling model. The number of tasks was gradually increased from 500 to 5000 tasks to observe the system's behavior under heavy workload conditions. Additionally, the number of virtual machines available in the cloud infrastructure was adjusted during experiments to evaluate the algorithm's adaptability to dynamic resource availability.

The implementation results demonstrate that the proposed hybrid scheduling framework can effectively distribute workloads across available resources while minimizing task execution delays. The algorithm also exhibits improved scalability compared to traditional scheduling methods when handling large-scale task submissions.

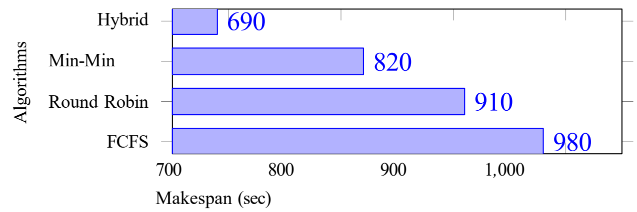

Fig. 4. Makespan Comparison

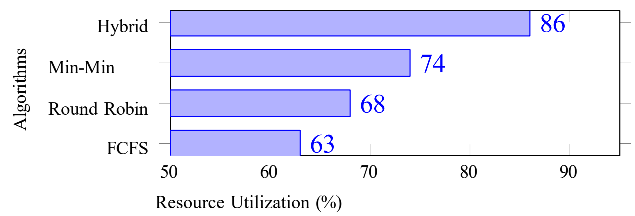

Fig. 5. Resource Utilization

TABLE I
SIMULATION ENVIRONMENT CONFIGURATION

Parameter	Value
Data Centers	1
Hosts	10
Virtual Machines	40
VM CPU Capacity	1000–3000 MIPS
RAM	2–8 GB
Tasks	500–5000
Algorithms	FCFS, RR, Min-Min, Hybrid

TABLE II
PERFORMANCE COMPARISON OF SCHEDULING ALGORITHMS

Algorithm	Makespan	Utilization	Throughput	Response
FCFS	980	63	5.1	2.9
Round Robin	910	68	5.5	2.5
Min-Min	820	74	6.1	2.1
Hybrid	690	86	7.4	1.6

TABLE III
RESOURCE UTILIZATION UNDER DIFFERENT WORKLOADS

Tasks	FCFS	RR	Min-Min	Hybrid
500	61	65	70	82
1000	62	67	72	84
2000	63	68	73	85
5000	64	69	74	86

V. RESULTS AND DISCUSSION

The experimental results clearly demonstrate the effectiveness of the proposed hybrid task scheduling algorithm in improving system performance and resource utilization in cloud computing environments. Compared to traditional scheduling algorithms, the hybrid model achieves significantly lower makespan and higher throughput, indicating that tasks are completed faster and system resources are used more efficiently [4] [10].

One of the key observations from the experimental evaluation is the improvement in resource utilization achieved by the hybrid scheduling algorithm. Traditional algorithms such as FCFS and Round Robin often result in uneven workload distribution across computing nodes. Some virtual machines may become overloaded while others remain idle, leading to inefficient resource usage. The proposed hybrid approach addresses this issue by continuously monitoring resource utilization levels and dynamically redistributing tasks when necessary.

The results also show a noticeable reduction in task response time when using the hybrid scheduler. Response time is a critical performance metric in cloud computing environments because it directly affects the user experience. The proposed scheduling framework reduces the waiting time experienced by tasks before execution begins, thereby improving overall system responsiveness.

Another important finding from the experiments is the scalability of the hybrid scheduling model. As the number of tasks increases, traditional scheduling algorithms tend to experience performance degradation due to inefficient task distribution. In contrast, the proposed hybrid algorithm maintains stable performance even under high workload conditions. This indicates that the system can effectively manage large-scale task submissions without significant delays or resource bottlenecks.

The improved performance of the hybrid scheduler can be attributed to its two-stage scheduling strategy. The initial heuristic scheduling phase ensures rapid task allocation with minimal overhead, while the dynamic optimization phase enhances resource distribution based on real-time monitoring data. This combination allows the system to achieve both efficiency and adaptability in task scheduling decisions.

Overall, the results confirm that hybrid scheduling ap-

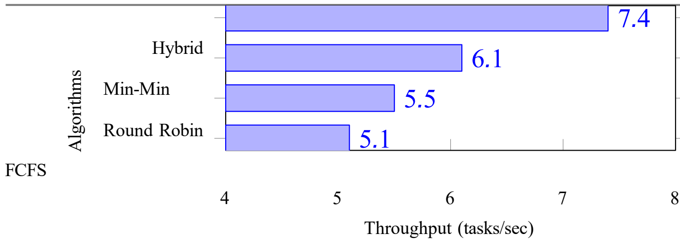

Fig. 6. Throughput Comparison

proaches can significantly enhance cloud computing performance by optimizing resource utilization and reducing task completion time. The proposed model demonstrates strong potential for deployment in real-world cloud infrastructures where efficient workload management is essential for maintaining service quality.

VI. Conclusion and Future Work

This study presented a hybrid task scheduling algorithm designed to improve resource utilization and task execution performance in cloud computing environments. The proposed framework integrates heuristic scheduling techniques with dynamic resource monitoring mechanisms to achieve balanced workload distribution across virtual machines. By considering factors such as task execution time, resource capacity, and system workload, the hybrid scheduler can allocate tasks more efficiently than traditional scheduling algorithms.

The experimental evaluation conducted using a simulated cloud environment demonstrates that the proposed algorithm significantly reduces makespan and response time while improving system throughput and resource utilization. Compared to commonly used scheduling algorithms such as FCFS, Round Robin, and Min-Min, the hybrid model consistently delivers superior performance under varying workload conditions. These results highlight the effectiveness of combining heuristic and adaptive scheduling strategies in modern cloud infrastructures.

Another important contribution of this research is the demonstration of the scalability of the proposed scheduling framework. As cloud computing systems continue to expand in size and complexity, efficient resource management becomes increasingly critical. The hybrid scheduling algorithm proposed in this study shows strong potential for handling large-scale workloads without compromising system performance.

Despite these promising results, several opportunities remain for further improvement and extension of the proposed framework. Future research may explore the integration of machine learning techniques to predict workload patterns and optimize scheduling decisions in real time. Reinforcement learning models could also be applied to enable the scheduler to learn from past performance outcomes and continuously improve its decision-making capabilities.

Additionally, future studies may consider incorporating energy-aware scheduling mechanisms to reduce power consumption in cloud data centers. Energy efficiency has become an important consideration for large-scale cloud infrastructures due to increasing operational costs and environmental concerns.

In conclusion, the proposed hybrid task scheduling algorithm provides an effective solution for improving resource utilization and system performance in cloud computing environments. The findings of this study contribute to the ongoing development of intelligent scheduling techniques capable of supporting the growing demands of modern cloud-based applications.

VII. Reference

References

[1] M. Armbrust *et al.*, "Above the Clouds: A Berkeley View of Cloud Computing," University of California, Berkeley Technical Report, 2009.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[3] S. Singh and I. Chana, "QRSF: QoS-aware Resource Scheduling Framework in Cloud Computing," *Journal of Supercomputing*, vol. 71, no. 1, pp. 241–292, 2015.

[4] H. Topcuoglu, S. Hariri, and M. Wu, "Performance-effective and Low-complexity Task Scheduling for Heterogeneous Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.

[5] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Special Publication 800-145, 2011.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[7] X. Liu, Z. Ni, J. Yang, and Q. Ding, "A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing," *Future Generation Computer Systems*, vol. 89, pp. 408–417, 2018.

[8] K. Etminani and M. Naghibzadeh, "A Min-Min Max-Min Selective Algorithm for Grid Task Scheduling," in *Proc. IEEE International Conference on Internet, Multimedia Services Architecture and Applications*, pp. 1–7, 2007.

[9] L. F. Bittencourt and E. R. Madeira, "HCOC: A Cost Optimization Algorithm for Workflow Scheduling in Hybrid Clouds," *Journal of Internet Services and Applications*, vol. 2, no. 3, pp. 207–227, 2011.

[10] A. Verma, P. Ahuja, and A. Neogi, "Power-aware Dynamic Placement of HPC Applications," in *Proc. 22nd Annual International Conference on Supercomputing*, pp. 175–184, 2008.