International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

HYBRID GENETIC ALGORITHM-BASED QOS-AWARE TASK SCHEDULING IN CLOUD ENVIRONMENTS

Dr. M. V. NARAYANA [1], ARIGE PAVAN KALYAN [2], UDATHALA RUDRA TEJA [3], CHILLAMCHARLA KEERTHI PRIYA [4], TATA SRI VISHAL SAI [5]

[1]Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

[2, 3, 4, 5] B.Tech, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

ABSTRACT— Simultaneous enhancement of Quality of Service (QoS) characteristics is difficult in the Cloud Computing (CC) environment if the overall QoS offered is inadequate for end users. In this research, the Smart Message Passing Interface Approach (SMPIA) is integrated with the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO) techniques for CC applications. The PSO-SMPIA and GA-SMPIA algorithms are described for task scheduling and resource allocation, with the goal of lowering makespan and overall execution time while enhancing resource utilization in CC. The primary contribution of this study is to compute the maximum cost for each transaction flow, which has not been addressed in prior research. This new multipurpose function covers the flow load, makespan load, virtual machine (VM) capacity, and execution speed parameters. Telecommunication deals and bids are classified according to the type of flow. The transaction flow allocation matrix to the VMs is calculated using appropriate values. The transaction flow is routed to the appropriate VMs based on the selected matrix. Furthermore, we investigate the impact of makespan and overall execution time on resource utilization. The findings indicate that PSO-SMPIA outperforms Optimized-SMPIA (O-SMPIA), Fuzzy SMPIA (FSMPIA), SMPIA, and GA-SMPIA in terms of average resource consumption. However, FSMPIA and O-SMPIA

outperform other algorithms in terms of improving makespan and overall execution time, respectively. The entire execution time and makespan are greatly lowered when compared to other methods, which improves the QoS offered to end users. Thus, GA-SMPIA outperforms other algorithms by simultaneously improving total execution time and resource consumption while also improving (minimizing) the makespan.

Index Terms – Cloud computing, Quality of Service (QoS), task scheduling, resource allocation, Smart Message Passing Interface Approach (SMPIA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), virtual machines (VMs), makespan optimization, execution time, resource utilization, metaheuristic optimization.

I. INTRODUCTION

Cloud computing has become one of the most widely adopted paradigms for delivering computing services due to its flexibility, scalability, and cost-effectiveness. By enabling on-demand access to shared computing resources such as servers, storage, networks, and applications, cloud computing allows users to deploy and execute applications without the need for maintaining physical infrastructure. However, the rapid growth of cloud users and the increasing complexity of cloud-based applications have introduced significant challenges in resource management, task scheduling, and Quality of Service (QoS) assurance. Efficient task scheduling plays a crucial role in cloud environments, as improper allocation of tasks to virtual machines can lead to increased execution time, higher makespan, resource underutilization, SLA violations, and poor user satisfaction. Cloud service providers must simultaneously consider several conflicting parameters such as execution time, makespan, operational cost, resource utilization, virtual machine capacity, and workload distribution. Traditional scheduling algorithms, including First Come First Serve (FCFS), Round Robin, and simple heuristic-based methods, are insufficient to address these challenges effectively. These approaches generally optimize a single objective and fail to adapt to dynamic workloads and heterogeneous cloud infrastructures. As a result, cloud systems often experience inefficiencies such as load imbalance, excessive processing delays, and increased operational costs, ultimately degrading the overall QoS delivered to end users. To overcome these limitations, metaheuristic optimization techniques have gained significant attention in recent years

due to their ability to explore large solution spaces and provide near-optimal solutions within acceptable time. Among these techniques, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have proven to be highly effective for solving complex scheduling and resource allocation problems in cloud computing. Genetic Algorithm, inspired by the principles of natural selection and evolution, is capable of exploring diverse scheduling solutions through operations such as selection, crossover, and mutation. PSO, inspired by the collective behaviour of bird flocks and fish schools, efficiently exploits the search space by guiding particles toward global and local best solutions. However, when applied independently, both GA and PSO have certain limitations, such as slow convergence or premature stagnation in local optima. In this project, a hybrid optimization framework is proposed by integrating Genetic Algorithm and Particle Swarm Optimization with a Smart Message Passing Interface Approach (SMPIA) to achieve QoS-aware and cost-aware task scheduling in cloud environments. The proposed system aims to simultaneously minimize makespan and total execution time while maximizing resource utilization and maintaining SLA compliance. A major contribution of this project is the computation of the maximum cost for each transaction flow, which has not been adequately addressed in existing research. The proposed multi-objective fitness function incorporates flow load, makespan load, virtual machine capacity, and execution speed to evaluate scheduling decisions more comprehensively. Transaction flows, particularly in cloud-based telecommunication environments involving deals and bids, are classified based on their flow characteristics. Based on this classification, a transaction flow allocation matrix is generated and used to map transaction flows to appropriate virtual machines. This intelligent mapping ensures that tasks are executed on suitable resources, thereby improving performance and reducing operational costs. To ensure fairness and priority handling among multiple transaction flows, the proposed system incorporates a Priority-Based Fair Scheduling (PBFS) algorithm. PBFS assigns priorities to tasks based on predefined criteria and ensures equitable distribution of cloud resources while preventing starvation. By combining PBFS with the hybrid GA–PSO SMPIA framework, the system achieves balanced workload distribution and enhanced QoS performance under varying load conditions. Additionally, the system supports dynamic job migration, allowing tasks to be reassigned

when a selected virtual machine becomes unsuitable due to capacity constraints or increased load. In conclusion, this project presents an efficient and intelligent cloud task scheduling framework that addresses the limitations of traditional approaches by adopting a hybrid metaheuristic optimization strategy. By jointly optimizing multiple QoS parameters and incorporating cost-awareness, fairness, and security, the proposed system offers a robust solution for modern cloud computing environments. The results confirm that hybrid GA–PSO based SMPIA scheduling significantly improves cloud performance and user satisfaction, making it a promising approach for future cloud resource management systems.

II. RELATED WORK

A. SG-PBFS: Shortest Gap–Priority Based Fair Scheduling Technique for Job Scheduling in Cloud Environment

With the rapid growth of cloud computing, efficient job scheduling has become a critical challenge to ensure fairness, reduced waiting time, and optimal resource utilization. This study introduces SG-PBFS (Shortest Gap–Priority Based Fair Scheduling), a novel scheduling technique designed to improve job scheduling performance in cloud environments. The proposed approach prioritizes jobs based on the shortest gap between job requirements and available resources, while also maintaining fairness among users. By combining priority-based scheduling with fairness constraints, the technique aims to minimize job starvation and reduce overall turnaround time. Simulation results demonstrate that SG-PBFS outperforms traditional scheduling algorithms in terms of average waiting time, response time, and resource utilization. The method shows improved QoS and better workload handling under dynamic cloud conditions. However, the approach may introduce computational overhead when dealing with highly heterogeneous workloads. Additionally, further optimization is required to address scalability issues in large-scale cloud infrastructures. Overall, the study highlights the effectiveness of intelligent scheduling techniques in enhancing cloud performance and fairness.

B. Priority-Based Job Scheduling Technique Utilizing Gaps to Increase the Efficiency of Job Distribution in Cloud Computing

The increasing demand for cloud services requires efficient job scheduling mechanisms to ensure balanced resource utilization and improved system performance. This study

proposes a priority-based job scheduling technique that utilizes resource gaps to enhance the efficiency of job distribution in cloud computing environments. The method assigns priorities to incoming jobs while intelligently identifying idle or underutilized resource gaps to schedule tasks more effectively. By filling these gaps, the proposed approach reduces resource wastage and improves overall throughput. Experimental evaluations show that the technique significantly lowers average waiting time and turnaround time compared to conventional scheduling algorithms. The approach also improves fairness among jobs with different priority levels and enhances Quality of Service (QoS). However, the scheduling strategy may face limitations when handling highly dynamic workloads. Additionally, the computational complexity can increase in large-scale cloud systems. Overall, the study demonstrates that gap-aware priority scheduling can substantially improve job distribution efficiency in cloud computing environments.

C. MTD-DHJS: Makespan-Optimized Task Scheduling Algorithm for Cloud Computing with Dynamic Computational Time Prediction

Efficient task scheduling is a major challenge in cloud computing due to varying workloads and unpredictable execution times. This study presents MTD-DHJS, a makespan-optimized task scheduling algorithm that incorporates dynamic computational time prediction to improve scheduling decisions. The proposed approach estimates task execution time dynamically and assigns tasks to appropriate virtual machines to minimize overall makespan. By accurately predicting computational time, the algorithm enhances load balancing and resource utilization in cloud environments. Experimental results show that MTD-DHJS significantly reduces makespan and improves throughput compared to traditional scheduling algorithms. The method also contributes to lower task completion time and better system performance under dynamic conditions. However, the accuracy of the model depends on reliable time prediction, which can be challenging in highly volatile environments. Additionally, the algorithm may introduce computational overhead when scaling to very large cloud infrastructures. Overall, the study demonstrates that integrating dynamic prediction into scheduling algorithms can effectively optimize cloud task execution.

D. Multi-Objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud–Fog Computing

The integration of cloud and fog computing introduces new challenges in task scheduling due to distributed resources and diverse performance requirements. This study proposes a multi-objective task scheduling approach based on the Grey Wolf Optimizer (GWO) algorithm to efficiently manage tasks in cloud–fog computing environments. The proposed algorithm simultaneously optimizes multiple objectives, including execution time, energy consumption, and resource utilization. By mimicking the leadership hierarchy and hunting behavior of grey wolves, the algorithm effectively balances workloads between cloud and fog nodes. Experimental results demonstrate improved performance compared to traditional and single-objective scheduling algorithms, particularly in terms of reduced latency and better energy efficiency. The approach also enhances system scalability and Quality of Service (QoS). However, the algorithm may experience increased computational complexity as the number of tasks and nodes grows. Additionally, fine-tuning of algorithm parameters is required to achieve optimal results in highly dynamic environments. Overall, the study highlights the effectiveness of metaheuristic multi-objective optimization for task scheduling in cloud–fog systems.

III. PROPOSED METHODOLOGY

The proposed system presents a multi-objective, QoS-aware and cost-aware approach for efficient cloud resource management. The main goal of the system is to minimize operational cost while ensuring high Quality of Service (QoS) and strict SLA fulfillment. It calculates the maximum allowable cost for each transaction flow to support better financial control in cloud environments. The system simultaneously optimizes multiple parameters such as cost, performance, and resource utilization using a combined intelligent MPI-based method.

This approach is specifically designed for transactions and tenders in cloud telecommunications environments. To enhance optimization efficiency, the system integrates Genetic Algorithms (GA) with Particle Swarm Optimization (PSO). Priority-Based Fair Scheduling (PBFS) is used as the core scheduling algorithm to ensure fairness among tasks while respecting priority constraints.

The system also supports secure cross-domain authentication and key agreement mechanisms. New cross-domain certificates are generated based on local digital certificates to improve trust management. Smart certificate re-issuance avoids insecure

reuse of local blockchain certificates during cross-domain access. Overall, the proposed system enhances security, efficiency, and fairness in cloud resource allocation while maintaining high performance and reliability.

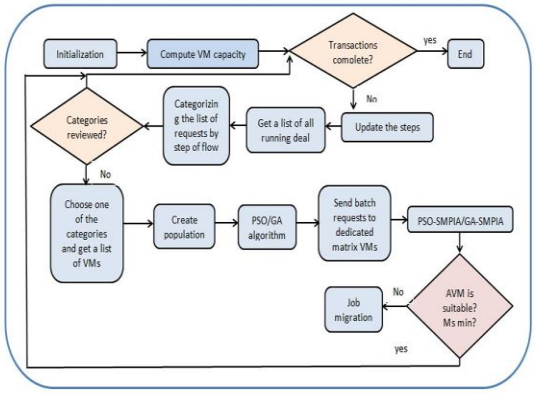
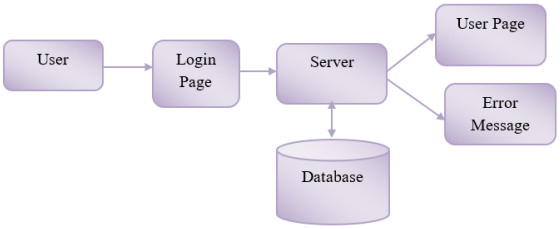


Fig. 1: System Overview

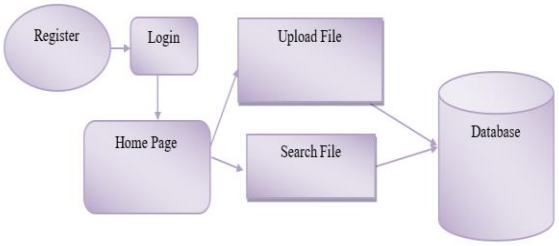
1. User Interface Design

To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password, Email id, City and Country into the server. Database will create the account for the entire user to maintain upload and download rate. Name will be set as user id. Logging in is usually used to enter a specific page. It will search the query and display the query.



2. Priority-Based Fair Scheduling

This module is responsible for managing and scheduling user requests for accessing uploaded data in a fair and organized manner. Each access request submitted by a user is assigned a priority based on factors such as request order and approval status. The system processes higher-priority requests first while ensuring that lower-priority requests are not ignored. The scheduling logic is implemented within the admin-related servlets, which retrieve request details from the database tables and update their status accordingly. This approach prevents request starvation and maintains balanced access control across multiple users. The module ensures orderly data access without increasing system complexity, making it suitable for academic and prototype-level implementation.



3. JSP Dashboard

Displays the uploaded encrypted data along with file details and timestamps. It also shows the transaction status, verification results, pending registration details, and overall project modules. The dashboard acts as the user interface for monitoring stored data, status, and user details etc.

4. MySQL Database

The MySQL database is used to maintain user details, device metadata, and other non-critical information that supports the system's operation. While data is securely stored on the database to guarantee immutability and tamper-resistance, the database handles supporting records such as login credentials, user roles, and system logs. This hybrid storage approach ensures both efficient system performance and strong data security by separating critical data from auxiliary information.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results show that the proposed GA-SMPIA and PSO-SMPIA algorithms improve QoS-aware task scheduling in cloud environments by reducing makespan and overall execution time while enhancing resource utilization. PSO-SMPIA achieved the best average resource utilization, whereas GA-SMPIA provided the best overall balance between execution time, makespan, and resource consumption. Compared with existing methods, the proposed hybrid approach significantly improves scheduling efficiency and cloud resource management.

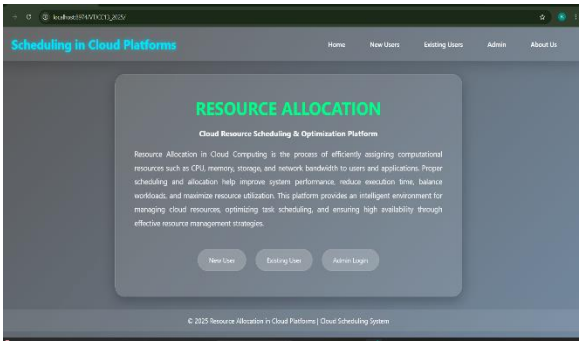


Fig. 2: Home Page

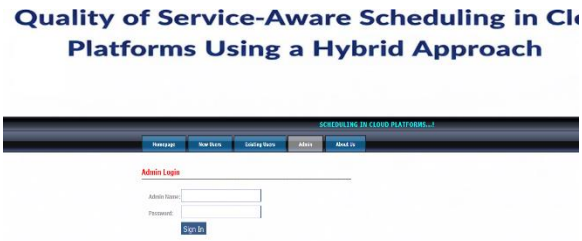


Fig. 3: Admin Login

Fig. 4: User Request

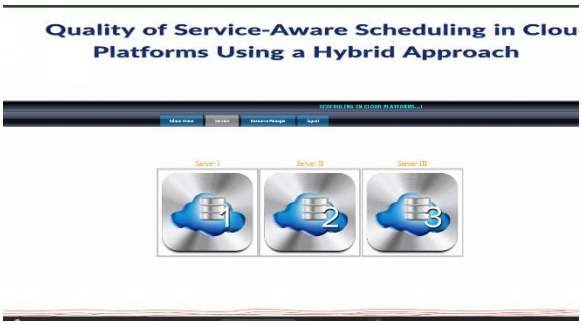


Fig. 5: Servers

Fig. 6: Application Identification

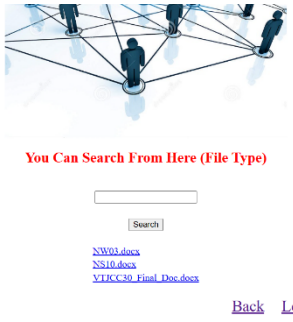


Fig. 6: Search File

V. CONCLUSION

This project successfully presents an efficient cloud task scheduling system that improves resource utilization and Quality of Service. The system manages user requests, file uploads, and server allocation through a structured scheduling process. By analyzing task size and server capacity, the system selects suitable resources for execution. Priority-based scheduling ensures fair allocation of resources among multiple users. The system continuously monitors server load and prevents overload through job migration. Administrative monitoring helps track user activity and system performance. Execution details are recorded for analysis and evaluation. The proposed approach reduces execution time and improves overall system efficiency. It also helps in better utilization of available cloud resources. The system provides reliable task execution even under heavy workloads. The implementation demonstrates effective coordination between users, servers, and scheduling logic. Overall, the project offers a practical solution for cloud resource management. It can serve as a foundation for advanced cloud scheduling systems in the future.

REFERENCES

[1]M.Mokhtari,P.Bayat,andH.Motameni,‘‘Multi-objectivetaskschedulingusingsmartMPI-basedcloudresources,’’Comput.Informat., vol.40,no.1, pp. 104–144, 2021.

[2] T. Ma, Y. Chu, L. Zhao, and O. Ankhbayar, ‘‘Resource allocation and scheduling in cloud computing: Policy and algorithm,’’ IETE Tech. Rev., vol. 31, no. 1, pp. 4–16, Jan. 2014, doi: 10.1080/02564602.2014.890837.

[3] A. M. Manasrah and H. Ba Ali, ‘‘Workflow scheduling using hybrid GA PSO algorithm in cloud computing,’’ Wireless Commun. Mobile Comput., vol. 2018, no. 1, Jan. 2018, Art. no. 1934784.

[4] A. K. Bardsiri and S. M. Hashemi, ‘‘QoS metrics for cloud computing services evaluation,’’ Int. J. Intell. Syst. Appl., vol. 6, no. 12, pp. 27–33, Nov. 2014.

[5] M.Mokhtari, H. Motameni, and P. Bayat, ‘‘Solving the task starvation and resources problem using optimized SMPIA in cloud,’’ Comput. Syst. Sci. Eng., vol. 42, no. 2, pp. 659–675, 2022, doi: 10.32604/csse.2022.021729.

[6]M.MokhtariandP.Bayat,‘‘Improvingcompletiontimeandexecutiontime using FSMPIA: Acasestudy,’’ Int. J. Nonlinear Anal. Appl., vol. 13, no. 1, pp. 3707–3721, 2022.

[7] M. Farid, R. Latip, M. Hussin, and N. A. W. Abdul Hamid, ‘‘A survey on QoS requirements based on particle swarm optimization scheduling tech niques for workflow scheduling in cloud computing,’’ Symmetry, vol. 12, no. 4, p. 551, Apr. 2020.

[8] N. Kumar, N. Chilamkurti, S. Zeadally, and Y.-S. Jeong, ‘‘Achieving quality of service (QoS) using resource allocation and adaptive schedul ing in cloud computing with grid support,’’ Comput. J., vol. 57, no. 2, pp. 281–290, Feb. 2014.

[9] M. H. Ghahramani, M. Zhou, and C. T. Hon, ‘‘Toward cloud comput ing QoS architecture: Analysis of cloud systems and cloud services,’’ IEEE/CAA J. Autom. Sinica, vol. 4, no. 1, pp. 6–18, Jan. 2017.

AUTHORS Profile

Dr. M. V. NARAYANA received his Bachelor of Engineering (B.E.) degree in Computer Science in 2004 from JNTU Kakinada, India. He completed his Master of Technology (M.Tech) degree in

Computer Science in 2010 from JNTU Hyderabad, India, and was awarded his Ph.D. in Computer Science in 2017 from JNTU Kakinada, India. With over 20 years of dedicated experience in academia and research, he currently serves as a Professor and Head of the Department (HoD) in the CSE Department at Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India. His research interests encompass various areas within computer science, and he has contributed significantly to the field throughout his career.

Mr. ARIGE PAVAN KALYAN pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Mr. UDATHALA RUDRA TEJA pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Ms. CHILLAMCHARLA KEERTHI PRIYA pursuing B.Tech deg-ree in Computer Science and Engineering at Guru Nanak Institutions Tech-nical Campus affiliated to JNTUH in 2022-2026.

Mr. TATA SRI VISHAL SAI pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.