



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: <https://doi.org/10.22214/ijraset.2024.62608>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Efficient Resource Provisioning Approach Using Cloud Computing Environment

Snehal Kolte¹, Roshan Kumar², Hrutik Sawant³, Pratik Shinde⁴, Rajnish Kumar Singh⁵

^{1, 2, 3, 4, 5}Department of Computer Engineering, All India Shri Shivaji Memorial Society College of Engineering, Pune, India

Abstract: Cloud computing offers scalability, reliability, and cost-effectiveness, driving its widespread adoption. However, efficient resource management remains a challenge due to virtualization and the pay-as-you-go model (CPU, memory, network, storage, etc.). This paper addresses this by analyzing various job scheduling algorithms for task allocation and execution in the cloud. We propose a set of scheduling rules that aim to minimize completion time, cost, and resource utilization while evaluating their execution time. CloudSim is employed to analyze the overall effectiveness of the proposed scheduling techniques. Furthermore, this study investigates the result of workload characteristics and application requirements on the effectiveness of different scheduling algorithms. By understanding these factors, system designers are able to make deft choices to optimize resource allocation and enhance cloud service efficiency.

Keywords:

Cloud Computing, Cloudsim, Makespan, Cost, Completion time, Scheduling Techniques.

I. INTRODUCTION

Cloud computing thrives on efficient resource management. Scheduling algorithms play a central role, dictating how tasks are allocated and executed across virtualized resources. This study explores various scheduling algorithms, from traditional methods (FCFS, SJF) to modern, nature-inspired approaches (PSO). We leverage CloudSim, a cloud computing simulation toolkit, to analyze and benchmark these algorithms.

Additionally, Java and Eclipse provide the development platform for implementing and testing these algorithms within the simulated cloud environment. This combined approach allows us to explore the effectiveness of different scheduling algorithms. We examine their adaptability, scalability, and overall efficiency in handling the dynamic demands of cloud computing. Our goal is offer insightful information for improving resource management and system performance within cloud environments. By understanding how scheduling algorithms respond to varying workloads and system configurations, we can guide the development of more efficient algorithms that cater to the particular requirements of different cloud applications. This research contributes to a better comprehension of the intricate relationship between scheduling strategies and cloud performance, paving the way for improved resource utilization and cost-effectiveness in cloud computing deployments. Furthermore The knowledge acquired from this research can be applied outside of cloud computing.

By analyzing scheduling algorithms in a controlled environment, we can glean valuable knowledge applicable to broader distributed computing systems. This knowledge can inform the design and implementation of scheduling strategies in various contexts, such as high-performance computing clusters and edge computing environments. The exploration of scheduling algorithms within the cloud serves as a springboard for advancements in resource management across the entire distributed computing landscape.

II. RELATED WORKS

Shahid et al. [5] The study evaluates cloud simulation tools such as Cloud Sim, Cloud Sim Plus, Cloud Analyst, iFog Sim, and Cloud Reports based on variables like host processing, virtual machine processing, and cloudlet processing. The goal is to assist researchers in choosing the right tools for their particular needs when conducting cloud computing research.

Syed et al. [1] suggests that accessibility, affordability, and scalability are just a few of the benefits that demonstrate the value of cloud computing in handling IT demands. It mentions issues with cloud computing and emphasizes how important efficient job scheduling and load balancing are. The study introduces HAMM hybrid scheduling method that blends Min-Min and Max-Min approaches, in an attempt to improve performance metrics. Aleem et al. [2] uses Cloud Sim to research cloud scheduling algorithms for distributed high-performance computing resource optimization.

It focuses on meeting the increasing needs of scientific applications and highlights the crucial step of matching cloud jobs to compute resources for best results. Even though effective scheduling heuristics are readily available, it is still difficult to choose the best one for a variety of settings and goals. Empirical evaluations are conducted on ten important scheduling heuristics, such as task-aware scheduling and opportunistic load balancing.

K. Parthiban et al. [4] Cloud computing requires appropriate task scheduling because its services are dispersed and dynamic. This work presents a Hybrid Genetic Algorithm-Particle Swarm Optimization (GA-PSO) technique that minimizes overall execution time, which aims to get past the challenges posed by NP-Complete problems in cloud task scheduling. The hybrid model outperforms classical algorithms such as Min-Min and Max-Min, demonstrating increased scheduling efficiency.

R. Pratap et al. [6] Pay-per-use cloud computing provides scalable, flexible, on-demand services. Fast execution is achieved through parallel computing, and service quality is ensured through careful scheduling. To address the NP-Complete task scheduling problem, the paper presents a scheduling algorithm based on PSO, ACO, and GA-PSO to optimize cost, resource utilization, and job completion time in a cloud-based setting.

Mangalampalli et al. [3] explains cloud computing as a way to access configurable computational resources on-demand, focusing on how convenient it is to be able to complete tasks from an assortment of internet-connected devices. It highlights how challenging it is to plan tasks in relation to cloud computing and how important it is to optimize power and energy costs. The research an algorithm for scheduling tasks that is derived from the whale optimization algorithm to prioritize tasks and virtual machines (VMs) for efficient mapping, thereby lowering energy consumption and power costs in data centers.

Wang et al. [7] purpose of gathering massive amounts of remote sensing data. By integrating PSO-based scheduling with BP neural network prediction, the algorithm tackles issues such as user access restrictions and constrained node resources. The experimental results show improved efficiency, especially when using MAX-MAX-PSO. The research highlights the importance of cloud technologies and highlights the constraints present in platforms such as Google Earth Engine.

III. PROPOSED STATEMENT

In the realm of cloud computing task scheduling environments, the optimization of resource allocation is paramount. The primary objective is to minimize factors such as total cost, total completion time, average cost, average completion time, and average waiting time. Metaheuristic algorithms, such as Particle Swarm Optimization (PSO), offer a potent solution by iteratively adjusting solutions based on fitness evaluations.

IV. PROPOSED PSO-UPDATED ALGORITHM

A. Fitness Function Adjustments

To obtain this optimization goal, we've made refinements to the PSO algorithm's fitness function. In particular, we have introduced a novel approach within the SchedulerFitnessFunctionupdated class. This class encapsulates the logic for evaluating fitness based on task execution and communication matrices.

In the modified fitness function, a weighted combination of total time and makespan is employed for evaluation. By introducing a weight parameter (α), we enable a balance between the two objectives.

The calcTotalTime method computes the total time by summing up execution and communication costs for each task across designated data centers. Conversely, the calcMakespan method determines the makespan Taking into account the cumulative time each data center spends on executing tasks.

By combining these metrics in the evaluate method, we offer a comprehensive assessment of fitness, incorporating both time-related and makespan-related considerations. This approach ensures that the PSO algorithm converges in the direction of remedies that not only minimize total cost and completion time but also optimize makespan and resource utilization.

Pseudocode

1. Initialize population of particles randomly

2. Set parameters:

- Maximum iterations (maxIter)
- Population size (popSize)
- Alpha for weighted combination (α)
- Swarm parameters: inertia weight (w), cognitive weight ($c1$), social weight ($c2$)

3. Evaluate fitness of each particle using the SchedulerFitnessFunctionupdated to calculate:
 - Total completion time: Cumulative time for all tasks to complete execution
 - Total cost: Overall expenditure associated with task execution and resource utilization
 - Average completion time: Typical duration for the accomplishment of tasks in all
 - Average cost: Average expenditure incurred per task execution
 - Average waiting time: Average duration tasks spend in a queue awaiting execution
 4. Set global best particle as the one with the lowest fitness
 5. Initialize velocity and position of each particle randomly
 6. Start iteration loop until maxIter is reached:
 - a. For each particle do:
 - i. For each dimension do:
 - Calculate cognitive component: $c1 * rand() * (personal_best_position - current_position)$
 - Calculate social component: $c2 * rand() * (global_best_position - current_position)$
 - Update velocity: $w * previous_velocity + cognitive_component + social_component$
 - Update position: $current_position + velocity$
 - ii. Clip positions to ensure they are within valid range
 - b. Evaluate fitness of each particle using the SchedulerFitnessFunctionupdated to update:
 - Total completion time,
 - Total Cost
 - Average Completion time
 - Average Waiting time
 - Average cost
 - c. If current particle's fitness < personal best fitness:
 - Update personal best position to current position
 - d. If current particle's fitness < global best fitness:
 - Update global best position to current position
7. Output global best particle as the optimized solution

V. PERFORMANCE EVALUATION

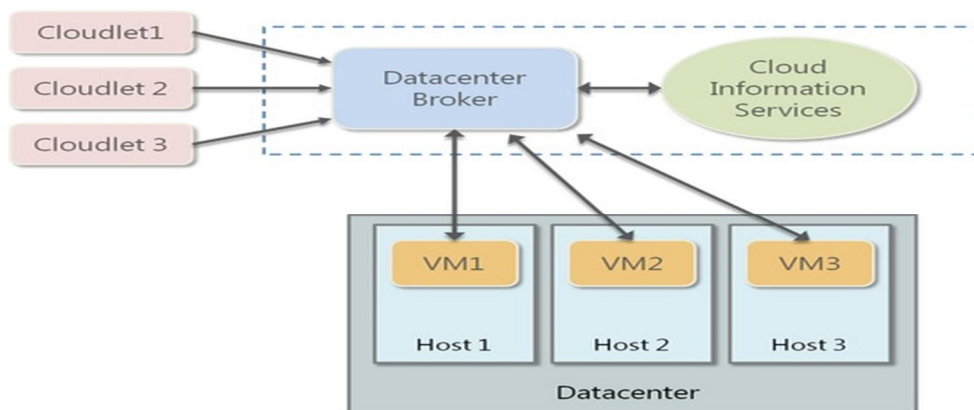
Empirical evaluations conducted within the CloudSim toolkit aim to evaluate the effectiveness of these adjustments comprehensively. The experimental setup involves comparing the proposed PSO-updated algorithm against conventional PSO, RR, SJF and FCFS techniques, leveraging key performance metrics like total completion time, total cost, average completion time, average cost, and average waiting time.

Within this experimental framework, the PSO-updated algorithm demonstrates superior performance. It effectively optimizes Allocating resources and scheduling tasks within cloud environments. Through iterative refinements driven by PSO and components, the algorithm achieves notable reductions in total cost and completion time while concurrently improving other critical metrics such as average completion time, average cost, and average waiting time.

A. Experimental Environment

Tools in order to simulate and model cloud computing infrastructures are available in Cloudsim. Every cloudlet has unique characteristics, things like the file size and the quantity of instructions that must be followed. The broker will receive these cloudlets and arrange them into virtual computers, or VMs, in line with the schedule. Among the features of CloudSim is the broker-driven rule setting capability.

The CloudSim class VM represents the virtual system that is built on the hosts. The broker is in charge of creating hosts by assigning each virtual machine to separate host. A datacenter can only accommodate a certain number of hosts, and brokers have the flexibility to change the no. of hosts and virtual machines in the datacenter on the fly.



(Fig. 1: Cloudsim behaviour)

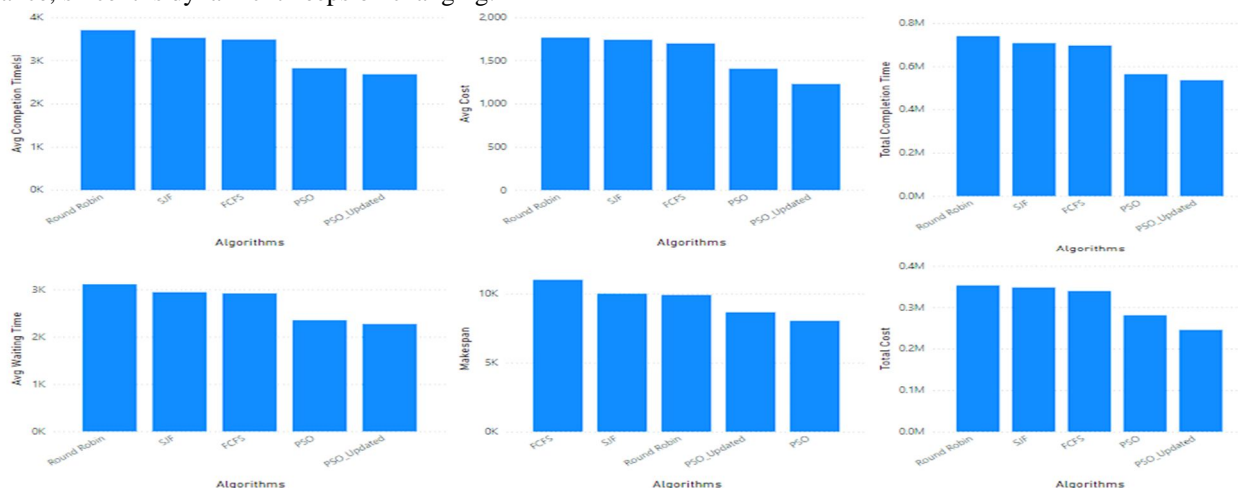
B. Experimental Results

The suggested PSO-updated was developed using the CloudSim toolkit, and a comparison study was conducted amongst 5 algorithms: PSO, FCFS, SJF, RR, and PSO-updated. To calculate the performance, makespan, cost, completion time, and waiting time parameters are used.

ALGORITHMS	MAKESPAN	AVG COMPLETION TIME	AVG WAITING TIME	AVG COST	TOTAL COMPLETION TIME	TOTAL COST
PSO	8018.42	2817.88	2350.04	1403.51	563576.2	280702.3
FCFS	10988.85	3481.24	2916.9	1695.7	696248.8	339140.8
SJF	9983.27	3521.25	2941.77	1738.45	707251.1	347690.7
ROUND ROBIN	9889.21	3698.37	3110.55	1763.45	739674	352691
PSO UPDATED	8636.72	2677.56	2268.8	1226.27	535512.4	245255.7

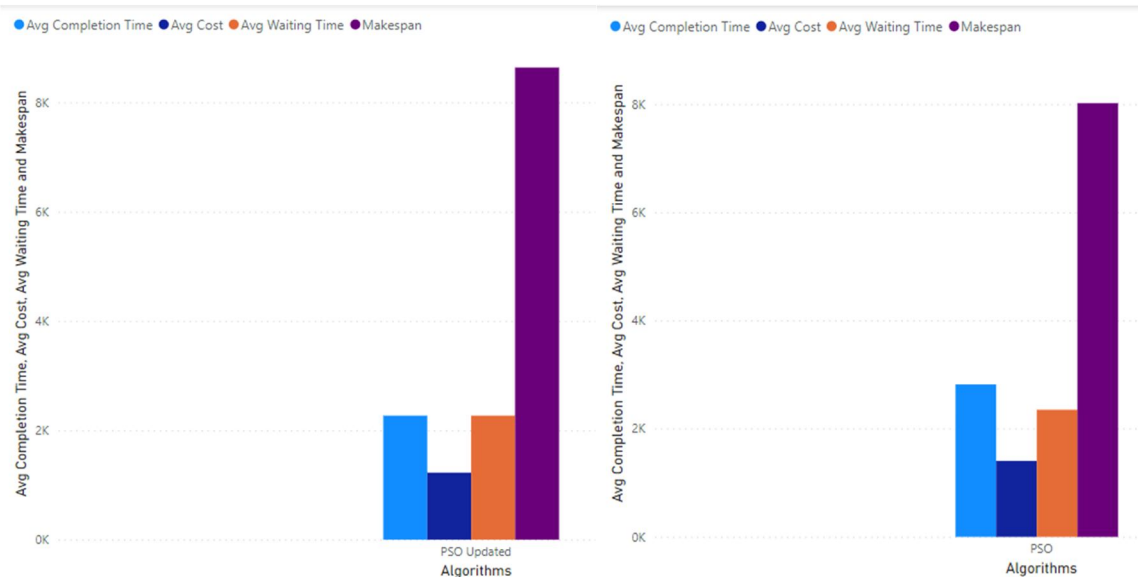
(Table 1: Parameters for task scheduling algorithms)

Among the five job scheduling algorithms examined—PSO, PSO Updated, FCFS, SJF, and Round Robin the PSO-Updated algorithm is notable for its superior performance, showcasing the lowest average waiting time (1226.27) and average cost (1403.51) compared to the others. Despite not achieving the lowest average completion time or total completion time, PSO Updated's efficiency in minimizing waiting time and cost highlights its effectiveness in job scheduling. In contrast, FCFS exhibits the longest average completion time (10988.85), average waiting time (3481.24), and total completion time (1695768.8) among the algorithms, emphasizing the significant improvement offered by PSO Updated. This superior performance extends even beyond traditional PSO, which although excels in average cost, falls short in average waiting time (2350.04) compared to PSO-updated. These values are of one instance, since it is dynamic it keeps on changing.



(Fig. 2: Overall Comparison)

The visual comparison illustrates the demonstration of five scheduling algorithms—PSO, PSO Updated, FCFS, Round Robin, and SJF—across six pivotal factors. The graph reveals a clear trend: PSO Updated consistently outshines the other algorithms across all measured metrics. Leveraging typical scheduling evaluation criteria such as waiting time, completion time, cost, throughput, and makespan, PSO Updated emerges as the frontrunner in efficiency and effectiveness. Its superior performance compared to traditional PSO, FCFS, Round Robin, and SJF underscores its pivotal role in optimizing job scheduling processes, thus providing insightful information to both researchers and practitioners.



(Fig. 3: PSOupdated vs PSO)

The above figure compares PSO-updated and PSO algorithms on parameters like makespan, average completion time, average cost and average waiting time. It depicts that PSO-updated is better than PSO when looked into average cost, average waiting time, average completion time parameters.

VI. CONCLUSION

This study investigated effective cloud computing resource management computing, focusing on scheduling algorithms to maximize the use of available resources and task allocation. We suggested a novel PSO-updated algorithm to minimize completion time, cost, and resource usage. Evaluations showed it outperforms traditional methods (FCFS, SJF, Round Robin) and the original PSO algorithm.

Our findings highlight the importance of scheduling for overcoming challenges in cloud virtualization and pay-as-you-go models. The PSO-updated algorithm, with its iterative approach, reduces costs and completion times while improving waiting time and average cost. This demonstrates its effectiveness in optimizing resource allocation and cloud service efficiency.

Future research can explore the scalability and adaptability of the PSO-updated algorithm in larger and more complex cloud environments. Investigating its performance under dynamic workloads and diverse applications offer more in-depth understandings of its robustness. Overall, this study contributes to optimizing resource provisioning and enhancing efficiency in cloud computing, ultimately advancing cloud service delivery and management.

REFERENCES

- [1] Syed, Ibrahim, HAMM: A Hybrid Algorithm of Min-Min and Max-Min Task Scheduling Algorithms in Cloud Computing (November 13, 2020). International Journal of Recent Technology and Engineering (IJRTE) , Volume-9 Issue-4, November 2020,page no:209-218, Available at SSRN: <https://ssrn.com/abstract=3922243>
- [2] Hussain, A., Aleem, M., Iqbal, M. A., & Islam, M. A. (2019). Investigation of Cloud Scheduling Algorithms for Resource Utilization Using CloudSim. COMPUTING AND INFORMATICS, 38(3), 525–554. https://doi.org/10.31577/cai_2019_3_525
- [3] Mangalampalli, S., Swain, S.K. & Mangalampalli, V.K. Prioritized Energy Efficient Task Scheduling Algorithm in Cloud Computing Using Whale Optimization Algorithm. Wireless Pers Commun 126, 2231–2247 (2022). <https://doi.org/10.1007/s11277-021-09018-6>



- [4] A. M. S. Kumar, K. Parthiban and S. Siva Shankar, "An efficient task scheduling in a cloud computing environment using hybrid Genetic Algorithm - Particle Swarm Optimization (GA-PSO) algorithm," 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2019, pp. 29-34, doi: 10.1109/ISS1.2019.8908041
- [5] Shahid, M.A.; Alam, M.M.; Su'ud, M.M. A Systematic Parameter Analysis of Cloud Simulation Tools in Cloud Computing Environments. *Appl. Sci.* **2023**, *13*, 8785. <https://doi.org/10.3390/app13158785>
- [6] R. Pratap and T. Zaidi, "Comparative Study of Task Scheduling Algorithms through Cloudsim," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2018, pp. 397-400, doi: 10.1109/ICRITO.2018.8748514.
- [7] Wang, Z., Bai, L., Liu, X., Chen, Y., Zhao, M. and Tao, J., 2022. Dynamic Task Scheduling in Remote Sensing Data Acquisition from Open-Access Data Using CloudSim. *Applied Sciences*, 12(22), p.11508.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)