iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XI **Month of publication:** November 2023

DOI: https://doi.org/10.22214/ijraset.2023.56642

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XI Nov 2023- Available at www.ijraset.com

Overview on Efficient Resource Provisioning Approach Using Cloud Computing Environment

Snehal Kolte[1], Hrutik Sawant[2], Roshan Kumar[3], Rajnish K. Singh[4], Pratik Shinde[5]
[1, 2, 3, 4, 5] Department of Computer Engineering, All India Shree Shivaji Memorial Society College of Engineering, Pune, India

Abstract: The goal of this study is to examine the importance of effective resource management across different fields and the critical role that cloud computing apps have in accomplishing this effectiveness. To show how these various approaches and strategies can be used in real-world scenarios, the article looks at case studies and application usage. Optimizing resource utilization and performance in cloud computing requires effective scheduling, resource allocation, and other algorithms Using Cloud Simulation (cloud sim) Tool and other cloud environment tools. By adjusting to changing workloads, these algorithms improve fault tolerance, cost effectiveness, and load balancing. Furthermore, maintaining cloud service integrity depends heavily on algorithms for data processing, network optimization, and fault tolerance. Sufficient research in this area is essential to fully utilize cloud computing.
Keywords: Cloud simulation (cloudsim), Scheduling Algorithms, Resource Allocation, Efficient task scheduling, Cloud Computing.

I. INTRODUCTION

In the ever-evolving landscape of cloud computing, the optimization of scheduling algorithms and resource allocation techniques holds the key to unlocking greater efficiency and performance. At its core, cloud computing is a dynamic and ever-expanding eco-system, characterized by a myriad of applications, workloads, and user expectations. To thrive in this environment, it is essential to ensure that cloud resources are allocated and managed with the utmost precision and cost-effectiveness. Efficiency is at the heart of cloud computing's success, driven by the development of advanced scheduling and resource allocation algorithms. This research focuses on creating and comparing new algorithms with existing ones to demonstrate superior efficiency. What sets this research apart is its emphasis on user-friendly graphical interfaces, enabling a practical showcase of algorithm performance. As cloud computing becomes increasingly integral to various industries, optimizing these algorithms is critical for achieving cost-effectiveness, resource utilization, and user satisfaction. This research endeavor embarks on a mission to not only build innovative algorithms but also rigorously compare them against existing counterparts to conclusively demonstrate their superior efficiency. The litmus test for these algorithms lies in their real-world applicability, measured against a backdrop of diverse workloads, dynamic user demands, and the ever-escalating expectations of cloud consumers.

In the dynamic realm of Cloud Computing, effective Task Scheduling is paramount for optimizing resource utilization and minimizing make span. This paper delves into the critical aspects of task scheduling, emphasizing the need to address energy consumption as a pivotal parameter in Cloud environments. Introducing a novel approach, the proposed task scheduling algorithm leverages Whale Optimization, intricately calculating priorities for tasks and virtual machines. By conducting simulations on Cloud Sim, this study demonstrates the algorithm's superiority over existing methods, showcasing significant reductions in energy consumption and power cost. The introduction encapsulates a forward-looking perspective, underscoring the evolving landscape of Cloud Computing and the role of innovative scheduling algorithms [3]. In the quest to facilitate a myriad of applications across diverse domains, the necessity for expansive cloud data centers has become indispensable. Cloud computing, offering access to virtually limitless computing resources on demand, has garnered widespread popularity. However, conducting real-world experiments in an actual cloud environment proves economically daunting. This research paper meticulously reviews simulation tools tailored for cloud computing, presenting a detailed analysis of notable ones such as Cloud Sim, Cloud Sim Plus, Cloud Analyst, iFog Sim, and Cloud Reports. The paper delves into the parameters guiding the evaluation of these tools and conducts comprehensive 5-parameter tests, showcasing the effectiveness of the proposed simulation system [5].

As the need for large-scale remote sensing data collection grows, this study suggests a TSCD-TSA dynamic task scheduling algorithm that combines PSO-based scheduling with BP neural network prediction. With an emphasis on increasing the speed at which data is collected, the algorithm's effectiveness is demonstrated using comparative experiments with Sentinel2 data.

The dynamic task scheduling algorithm MAX-MAX-PSO performs better than the others, as evidenced by its smaller fitness value and faster convergence. This work is important because it offers a useful framework for resource optimization in a dynamic environment to individuals, businesses, and research institutions that collect large amounts of remote sensing data [10].

II. LITERATURE REVIEW

Shahid et al. [5] Based on variables including host processing, virtual machine processing, and cloudlet processing, the study assesses cloud simulation tools like Cloud Sim, Cloud Sim Plus, Cloud Analyst, iFog Sim, and Cloud Reports. Helping researchers select the appropriate instruments for their unique requirements in cloud computing investigations is the aim.

Syed et al. [1] proposes scalability, affordability, and accessibility are just a few of the advantages mentioned in the text that highlight the significance of cloud computing in managing IT demands. It highlights the necessity for effective job scheduling and load balancing while mentioning cloud computing problems. In an effort to enhance performance metrics, the study presents HAMM, a hybrid scheduling method that combines Min-Min and Max-Min techniques.

A. Keivani et al. [8] explores the topic of cloud computing and highlights how important it is to schedule tasks well in order to reduce downtime, control workload, and increase throughput. The investigation looks at several job scheduling algorithms in the context of cloud computing, taking into account factors like cost, throughput, and completion time. The cloud model, categorized into public, private, and hybrid clouds, is examined, along with service models such as SaaS, PaaS, and IaaS. The paper evaluates and compares multiple task scheduling algorithms proposed by researchers, highlighting their impact on computation time, load balancing, resource utilization, cost, and quality of service in cloud computing.

Aleem et al. [2] investigates cloud scheduling algorithms for optimizing resource utilization in distributed high-performance computing using Cloud Sim. Focused on addressing the growing demands of scientific applications, it emphasizes the critical process of mapping cloud jobs to compute resources for optimal performance. Despite the availability of efficient scheduling heuristics, selecting the right one for diverse environments and objectives remains challenging. Ten key scheduling heuristics are empirically evaluated, including opportunistic load balancing and task-aware scheduling. The study, conducted on CloudSim, reveals that while Suffrage and task-aware algorithms minimize make span, they struggle to fully utilize cloud virtual machines. The paper underscores the persistent issue of load imbalance and its adverse effects on resource utilization.

G. Akanmu et al. [11] study presents the user-priority improved version of the Load Balance Improved Min-Min (LBIMM) scheduling method, called User-Priority Aware Load Balance Improved Min-Min (PA-LBIMM), specifically for cloud computing. LBIMM handles problems with load imbalance in Min-Min scheduling, but PA-LBIMM gives VIP tasks priority. Performance evaluations show that PA-LBIMM is superior in reducing completion times for VIP tasks, and both methods are adequately described.

Mangalamplli et al. [3] introduces cloud computing as on-demand access to configurable computational resources, emphasizing the convenience of performing tasks from various devices with internet connectivity. It draws attention to the difficulty of scheduling tasks in cloud computing and the requirement for power and energy cost optimization. In order to reduce energy consumption and power costs in data centers, the paper presents a task scheduling algorithm based on the whale optimization algorithm to prioritize tasks and virtual machines (VMs) for effective mapping.

K. Parthiban et al. [4] Appropriate task scheduling is needed in cloud computing, as services are scattered and dynamic. In order to overcome the obstacles presented by NP-Complete problems in cloud task scheduling, this work presents a Hybrid Genetic Algorithm-Particle Swarm Optimization (GA-PSO) technique that minimizes overall execution time. The hybrid model shows improved scheduling efficiency by outperforming traditional algorithms like Min-Min and Max-Min.

R. Pratap et al. [6] Cloud computing offers on-demand, scalable, and flexible services with pay-per-use. Parallel computing is used for fast execution, and scheduling is crucial for ensuring service quality. 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 environment in order to address the NP-Complete task scheduling challenge.

Ahmad et al. [7] The Raspberry Pi, the building block of the system, is utilized to run the Raspbian (Linux) operating system via a micro-SD card. By facing the camera, the camera will take a picture and then send it to the Raspberry Pi, which has been configured to use the Local Binary Patterns algorithm (LBPs) to conduct face recognition. Using the dataset of 11 human photos, the accuracy of the system is 95%.

Nzanywayingoma et al. [9] study reviews cloud computing resource management in detail, highlighting the difficulties brought on by the growing complexity of cloud infrastructure and user needs. With an emphasis on Infrastructure as a Service (IaaS), it offers a thorough review of methods including resource provisioning, monitoring, allocation, and migration strategies.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XI Nov 2023- Available at www.ijraset.com

The paper emphasizes how the growth of cloud resources makes manual resource management difficult. The National Institute of Standards and Technology's (NIST) basic requirements, the distinctions between physical and virtual resource management, and the function of logical resource management are among the important topics discussed. The paper explores a spectrum of resource management techniques, incorporating diverse methodologies such as bin-packing algorithms, game theory-based market policies, and virtualization strategies. In addition to resource management, the paper delves into critical aspects of virtualization, including server, storage, network, client, and operational framework virtualization. It thoroughly examines virtual machine migration techniques and storage migration solutions, encompassing various methods like LVM mirroring, Oracle RMAN-based migration, and Fast Copy-based data migration.

Wang et al. [10] introduces a novel task scheduling algorithm, TSCD-TSA, for large-scale remote sensing data collection. Combining BP neural network prediction and PSO-based scheduling, the algorithm addresses challenges like limited node resources and user access restrictions. Experimental results, particularly with MAX-MAX-PSO, demonstrate enhanced efficiency. The study underscores the importance of cloud technologies and notes limitations in platforms like Google Earth Engine. Key methods include a multi-objective task scheduling model considering time, energy, and cost constraints. The PSO optimization algorithm is improved using FCFS, SJF, and MAX_MAX algorithms to accelerate optimization. TSCD-TSA incorporates a BP neural network to predict node speeds, facilitating dynamic scheduling. Comparative results highlight MAX-MAX-PSO's superiority in fitness value and convergence speed. It presents an innovative task scheduling approach, showcasing its efficacy through comprehensive experimentation in the realm of large-scale remote sensing data collection.

Based on the works relevant in literature [5] [6] [7] [8] in cloud systems, scheduling techniques are often divided into three categories: task scheduling, process scheduling, and resource scheduling. Figure 1 shows the classification as a whole. Workflow scheduling is used to arrange the workflows that make up a project in an appropriate sequence, while resource scheduling maps virtual resources among physical machines. Methods for scheduling tasks might be dispersed or centralized. Depending on whether the tasks are dependent or independent, it can be carried out in a homogenous or heterogeneous setting.

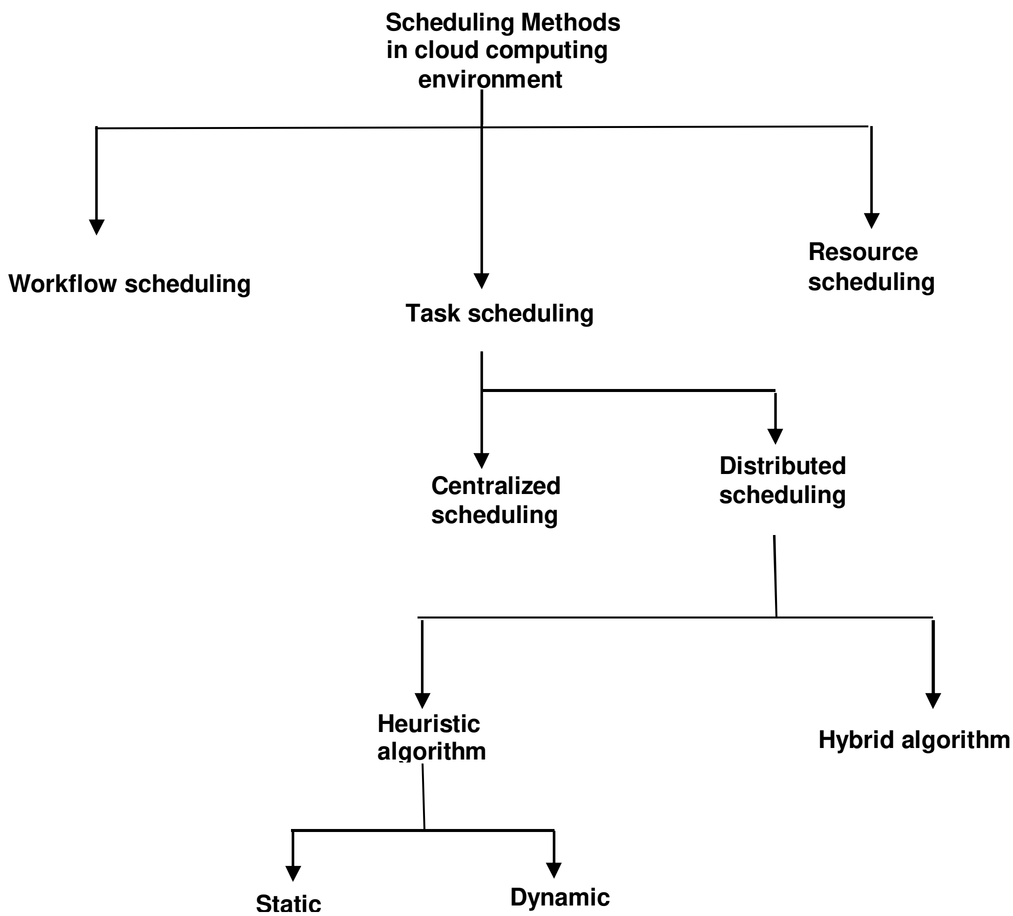


Fig. 1. Classification of Scheduling Methods in Cloud Computing Environment

A. *Heuristic Algorithms in Scheduling*

Ten well-known static cloud scheduling heuristics—MCT, OLB, Min-Min, Max-Min, Suffrage, RASA, PSSLB, PSSELB, LBIMM, and TASA—are compiled and assessed in this work. Below is a description of how these heuristics operate.

1) *OLB Algorithm:* It is a dynamic load balancing technique used in computer networking to efficiently distribute network traffic and workloads across available resources or paths. Unlike traditional load balancing methods that often rely on predefined rules or static configurations, OLB adapts to real-time network conditions and takes advantage of opportunities to optimize traffic distribution. [6]
2) *MCT Algorithm:* It is a popular search and decision-making algorithm used in artificial intelligence and game-playing. It combines elements of random sampling and tree-based exploration to find optimum or near-optimum results in complex decision spaces. MCTS is known for its adaptability to various domains, efficient resource utilization, scalability, and the ability to handle situations with incomplete or uncertain information. It has achieved remarkable success in playing games like Go and chess, as well as applications in robotics, planning, recommendation systems, and more [6].
3) *Min-Min Algorithm:* As a heuristic scheduling algorithm, the this algorithm works in grid and cloud computing environments to minimize the total execution time of a set of tasks on a set of heterogeneous processors. A common application for task scheduling in distributed systems, it is an expansion of the popular Min algorithm. By efficiently assigning tasks to available resources, the algorithm reduces the amount of time needed to complete tasks. Locating the task that takes the least amount of time to complete among all the tasks and assigning it to the resource that takes the least amount of time to complete among all the available resources is the basic idea behind mentioned algorithm [1] [6] [8].
4) *LBMM Algorithm*: The Min-Min algorithm, while straightforward and effective in minimizing the make span, may not fully capitalize on resource efficiency. To address this limitation, the Load-Balanced Min-Min (LBMM) algorithm was introduced. This two-stage algorithm combines the simplicity of Min-Min with an additional step aimed at enhancing resource utilization. This algorithm uses the Min-Min approach in its first stage to determine the lowest execution time possible for every task. The tasks are then assigned that produces resources in the shortest completion time after being chosen in a sequential manner based on their minimal execution times. Make span minimization is given priority in this first step. The second stage of the LBMM algorithm focuses on refining resource utilization. After the Min-Min scheduling, tasks are reassigned to resources in a manner that maximizes efficiency. This involves a reconsideration of the initial task-resource assignments to ensure a more balanced distribution, thereby enhancing overall resource utilization [8].
5) *MAX-MIN Algorithm:* This algorithm is a examining in optimization and scheduling problems, particularly in the context of task allocation and resource allocation. It is applied to distribute resources among a set of tasks in a way that aims to maximize the minimum resource allocation to any task. This algorithm is often used in situations where fairness or preventing resource starvation for individual tasks is a primary concern. The Max-Min algorithm is often employed in scenarios where it is important to ensure that no task suffers from resource starvation, even in the presence of varying workloads or changing resource demands. It contributes to a more equitable distribution of resources, which can be crucial in systems where fairness is a primary consideration. It's worth noting that while the Max-Min algorithm prioritizes fairness, it might not always lead to the most efficient utilization of resources [1] [6].
6) *RASA:* The continuous monitoring and analysis of data produced by applications and systems as events take place is known as real-time application and system analytics. Real-time insights on user behavior, system performance, and other pertinent indicators are sought after in order to facilitate prompt decision-making and proactive management. While the term "Real-Time Application and System Analytics (RASA)" does not refer to a specific algorithm. Real-time analytics apps should have quick reaction times and high availability in order to be immediately relevant. They ought to be capable of managing massive data volumes, up to and including terabytes. However, they ought to still respond to inquiries in a matter of seconds [2].
7) *ACO:* This is a meta heuristic algorithm influenced due to foraging act of ants. Developed based on the collective intelligence observed in ant colonies, ACO is particularly effective for solving combinatorial optimization problems. In this approach, the problem is represented as a graph, with nodes symbolizing solution components and edges representing connections or paths. ACO employs a colony of artificial ants to construct solutions by probabilistically choosing paths based on both pheromone levels and heuristic information. ACO has demonstrated robustness and adaptability in solving various problems, including the TSP and Vehicle Routing Problem. While its convergence speed and sensitivity to parameter tuning are considered, ACO remains a powerful tool in optimization, illustrating the effectiveness of decentralized and swarm-based approaches in solving complex problems [6].

8) *PSO:* The collective behavior of swarms, such as group of birds or shoal of fish, serves as model for Particle Swarm Optimization (PSO), an algorithm inspired by nature. PSO, which Eberhart and Kennedy first introduced in 1995, has been widely used to solve optimization issues in a variety of fields. Particles within a multidimensional search space are used to represent possible solutions in PSO. An initial population of particles with random locations and velocities is created at the start of the process. Each particle iteratively adjusts its position and velocity in an attempt to converge towards the optimal solution, taking into account both its personal experience (the local best) and the collective understanding of the global best. Acceleration coefficients, random values, and the inertia weight determine the updates to the velocity and location [4].

HEURISTIC ALGORITHMS	KEY FEATURES	DRAWBACKS	COMPLEXITY	TOOLS USED
MCT (Monte Carlo Tree Search)	MCTS can be used for various problems, like recommendation systems, and decision-making tasks. [12]	Randomness in simulations, Overloads faster VMs [12]	O(M . N)	tGSF Simulator, Cloud Sim Simulator [5]
OLB (Opportunistic load balancing)	Efficiently distribute network traffic and workloads across different resources or paths. [14]	imbalanced loads and unfair scheduling [14]	O(M)	Cloud Sim, NS Simulator [5]
Min-Min	Efficient grid dispatch optimizes the grid system, boosting application performance with m missions and n available units. [1]	Min-Min algorithm assigns tasks based on the smallest time, leading to load imbalance. [1]	O(M^3)	Mat lab Toolbox Grid Technology [2]
Max-Min	It balances the workload among resources, which can lead to improved resource utilization. [1]	smaller jobs are penalized, and the job pool with more larger jobs has an imbalance in load. [1]	O(M . N^2)	Mat lab and Java-based Simulation [2]
LBIMM (Load Balanced Improved Min-Min)	Improved resource utilization, reduced completion times, and a more balanced allocation of tasks. [12]	Certain lesser jobs are penalized in the process. Postponed [12]	O(M . N^2)	Mat lab [2]
RASA (Real-Time Application and System Analytics.)	larger and smaller jobs are treated equally. [4]	smaller jobs in a dataset with larger jobs are penalized.[4]	O(M . N^2)	Grid Sim [11]
ACO (Ant Colony Optimization)	It solves problems with large search spaces by locating globally optimal or nearly optimal solutions. [6]	Slow convergence speed in complex problems. [6]	O(M . N^2)	Cloud sim [5]

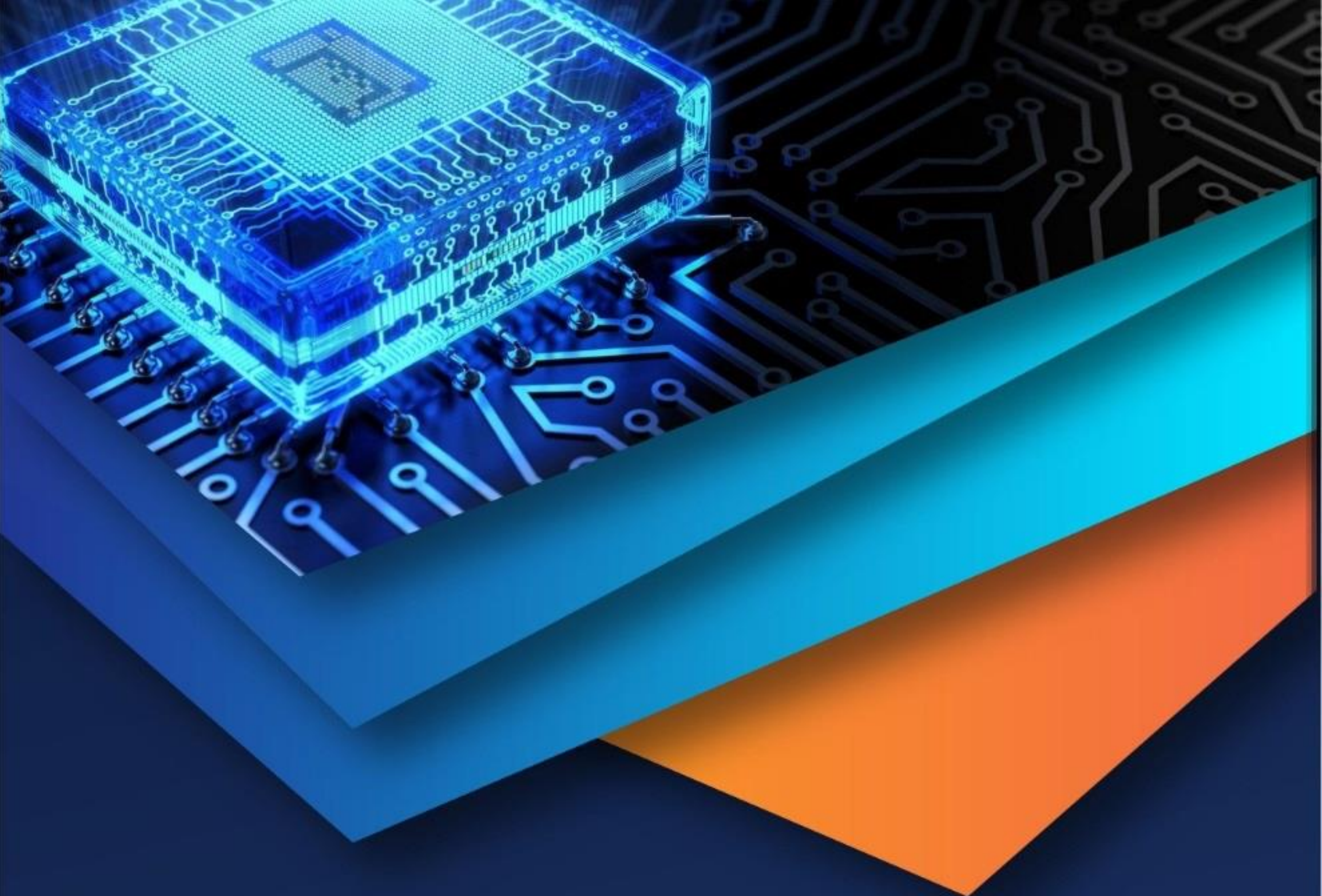
PSO (Particle Swarm Optimization)	Parallelism, Robustness, Efficiency, and Versatility [6]	Premature convergence, sensitivity to parameters, limited performance in dynamic environments.[6]	O(M* N * objective eval)	Cloud sim [5]
GA-PSO (Genetic Algorithm and Particle Swarm Optimization)	Reduced Premature Convergence, Performance on Complex Landscapes, and Hybridization Flexibility. [6] [7]	Parameter Sensitivity, Limited Memory. [6] [7]	O(O(g(nm + nm + n)))	Cloud sim [5]

III. CONCLUSION

This survey paper highlights the pivotal part of effective asset management in the dynamic landscape of cloud computing. Optimization of scheduling algorithms and resource allocation techniques is essential for achieving greater efficiency, cost-effectiveness, and user satisfaction. The research introduces innovative algorithms like HAMM and prioritized energy-efficient task scheduling, demonstrating their applications through real-world case studies. The evaluation of cloud simulation tools and a comparative analysis of task scheduling algorithms provide valuable insights for researchers. As cloud computing becomes increasingly integral to diverse industries, the significance of user-friendly graphical interfaces for algorithm performance is emphasized. The study underscores the ongoing importance of advancing scheduling algorithms to address challenges posed by dynamic workloads and escalating user expectations. In summary, optimizing these algorithms remains critical for achieving enhanced resource utilization, cost-effectiveness, and service integrity in cloud computing.

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] Ahmad, Md & Khan, Rafiqul Zaman. (2019). Cloud Computing Modeling and Simulation using CloudSim Environment. International Journal of Recent Technology and Engineering. 8. 10.35940/ijrte.B3669.078219.
[8] A. Keivani and J. -R. Tapamo, "Task Scheduling in Cloud Computing: A Review," 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), Winterton, South Africa, 2019, pp. 1-6, doi: 10.1109/ICABCD.2019.8851045
[9] Nzanywayingoma, Frederic, and Yang Yang. "Efficient resource management techniques in cloud computing environment: a review and discussion." International Journal of Computers and Applications 41.3 (2019): 165-182.
[10] 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.
[11] Huankai Chen, F. Wang, N. Helian and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), Bangalore, India, 2013, pp. 1-8, doi: 10.1109/ParCompTech.2013.6621389.
[12] Singh, Shyam & Kushwah, Virendra. (2017). A Genetic based Improved Load Balanced Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing. 10.1109/CICN.2016.139.
[13] Madni SHH, Abd Latiff MS, Abdullahi M, Abdulhamid SM, Usman MJ (2017) Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. PLoS ONE 12(5): e0176321. https://doi.org/10.1371/journal.pone.0176321
[14] E. K. Tabak, B. B. Cambazoglu and C. Aykanat, "Improving the Performance of IndependentTask Assignment Heuristics MinMin,MaxMin and Sufferage," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 5, pp. 1244-1256, May 2014, doi: 10.1109/TPDS.2013.107.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY