Grenze International Journal of Engineering and Technology, June Issue

Enhancing Task Scheduling in Cloud Computing with an Improved PSORL Algorithm

Pavithra V[1] and Dr. Sindhu S[2]
[1,2]Computer Science and Engineering, NSS College of Engineering Palakkad, Kerala, India
Email: pavithravelayudhankutty@gmail.com, sindhus@nssce.ac.in

***Abstract*—Cloud computing is a popular infrastructure for conducting operations with virtual computers. Task scheduling is one of the most basic problem associated with cloud computing. Makespan is the entire amount of the time needed to complete all of the scheduled jobs so optimizing makespan is the primary goal of task scheduling. Existing system includes nature inspired methods, Traditional methods like Round Robin, Shortest Job First, First Come First Serve were used to reduce makespan but found to lack efficiency. Traditional methods was implemented and compared with PSO and found that PSO's makespan is comparatively better, but PSO could not adapt to the dynamic nature of cloud environment. Therefore to overcome the problems, the paper proposes a hybrid method using PSO and RL. Implementation of PSORL based task scheduling method to optimize the makespan. The hybrid method determines optimal mapping of tasks to virtual machines across multiple datacenters. This solution ensures in reducing the makespan.**

***Index Terms*— Cloud computing, Task scheduling, Particle Swarm Optimization (PSO), Reinforcement Learning (RL), Hybrid algorithms, Makespan minimization, CloudSim.**

I. INTRODUCTION

Cloud computing is a contemporary computer technology that uses virtualized infrastructure to offer end users dependable and secure services in a complex setting. Cloud computing has received a lot of attention as a computing model since it offers crucial information technology (IT) services including computing resources in the form of virtual machines (VMs). Infrastructure as a Service (IaaS), which offers virtualized computer resources via the internet, is one type of cloud computing, which is a new approach to parallel processing and large-scale distributed computing, Pay per usage is how it offers computing as a utility service. Applications, data, memory, bandwidth, and IT services are among the dynamic services that cloud computing currently offers online. Task scheduling is one of the many variables that affect cloud services' dependability and performance. The effectiveness and efficiency of cloud computing services are constantly reliant on how well users complete the tasks they submit to the cloud system. One important factor in enhancing cloud service performance is task scheduling [1]. Unreasonable task scheduling in a large-scale distributed cloud manufacturing environment will result in issues such as decreased cloud server resource usage, deteriorated system performance [2]. The amount of time needed to finish a series of actions from beginning to end is known as makespan, and it can be decreased by scheduling tasks optimally to cut down on idle time, enhance resource efficiency, and distribute the workload across available resources. By using efficient task-scheduling algorithms, these problems can be lessened and cloud computing systems can operate more efficiently.

Grenze ID: 01.GIJET.11.2.583

As a result, One essential element of any cloud architecture is the work scheduling algorithm. [6]. In the cloud computing resource management system, task scheduling is essential for ensuring overall performance in areas like response time, total execution time (Makespan) [10]. This dynamic nature affects makespan, or the overall amount of time needed to do a collection of tasks, because load balancing, task dependencies, and resource fluctuations impact how well tasks are planned, which could result in an increase in makespan if resources are not allocated or managed properly. Many authors have already worked on different algorithms inspired by nature to address scheduling problems. Numerous previous studies have addressed workflow scheduling through the use of metaheuristic, bio-inspired, and nature-inspired algorithms nevertheless, all of these methods provide almost ideal results when assessing parameters. Since scheduling in cloud computing is a very dynamic situation, these kinds of methodologies are not flexible enough to adjust to changing workflows in cloud environments and might not always produce schedules in an optimal way. Traditional scheduling methods, such as Round-Robin, Min-Min, and Max-Min, attempt to achieve this goal but are often limited by their simplistic decision-making processes. These methods struggle to adapt to dynamic workloads, leading to inefficient resource utilization and extended execution times. Additionally, metaheuristic approaches such as genetic algorithms (GA) and (PSO), while more sophisticated, often suffer from difficulties in handling the dynamic nature of slow convergence and cloud atmosphere. These boundaries highlight the need for a more adaptive and intelligent scheduling mechanism. To solve these challenges, a novel proposes the hybrid approach, The PSORL Scheduler, which combines (PSO) with (RL). By taking advantage of the PSO's global search ability and RL's adaptive decision making ability, the method dynamically acts for VMS, adapting for both execution time. The PSORL scheduler introduces a fitness assessment strategy that is efficient scheduling in a real -time cloud environment.

The arrangement of this paper follows: Section 2 presents an thorough analysis of related work, discussing limitations of existing task scheduling techniques in cloud environments. Section 3 presents the proposed PSORL Scheduler, detailing its algorithmic framework, task to VM mapping strategy, and fitness evaluation metrics. Section 4 presents experimental results, comparing the PSORL Scheduler with traditional methods in terms of makespan minimization. Finally, Section 5 summarizes our findings, highlights the contributions of the study.

II. RELATED WORKS

The section discuss the related works, limitations of the existing system.

In existing scheduling system for task scheduling and to minimize the makespan methods like Traditional methods (FCFS, Min-Max, Min-Min), Heuristic algorithms and Metaheuristic algorithms have been used. But these methods were not that effective because they face challenges like adaptability, scalability. As cloud environment is dynamic in nature the existing system cannot process task efficiently. To overcome the challenges which is faced by the existing system a hybrid method using PSO- RL have been proposed.

The study examines heuristic, hybrid, and energy-efficient work scheduling techniques, pointing out that most algorithms' efficiency and flexibility in maximizing cloud service performance can be limited by their predominance of one or two parameters [1]. Task scheduling in cloud computing is significant for optimizing resource utilization, with techniques like ACO, PSO, GA, and Fuzzy Logic categorized by their applications and performance metrics. Challenges include high complexity, scalability issues, and gaps in addressing all aspects of scheduling effectively [2]. To dynamically schedule activities with precedence relationships for cloud manufacturing, a deep reinforcement learning-based task scheduling algorithm (RLTS) is suggested. This algorithm uses a Deep-Q-Network to handle high-dimensional and complicated problems while minimizing execution time. The results of the experiments show that RLTS performs better in terms of computational efficiency and solution quality; however, more work is required to improve the algorithm's convergence speed and account for deadline and cost limitations [3].]. In order to increase task scheduling efficiency in cloud computing, a new hybrid differential evolution (HDE) task scheduler is suggested. It does this by improving the exploitation operator of classical differential evolution and dynamically modifying the scaling factor. According to experimental results, HDE is the most effective method for task scheduling since it performs better than other heuristic and metaheuristic algorithms in terms of makespan and overall execution time. [4]. Enhanced versions of the Heterogeneous Earliest Finish Time (HEFT) algorithm for task scheduling in cloud computing, improving the scheduling results by modifying rank generation and idle slot selection methods, while maintaining the same computational complexity as the original HEFT, with future work focusing on nature-inspired optimization and dynamic scheduling [5]. The NSGA-III algorithm is a multi-objective task scheduler for cloud computing that optimizes execution time, energy consumption, and cost, with better utilization of resources and efficiency in executing tasks compared to NSGA-II [6]. Future work can be on hybridizing the NSGA-III with metaheuristics or extending it to other domains like intrusion detection and big data. An end-to-end algorithm for CMfg-SP combines a multi-head attention (MHA) mechanism and deep reinforcement learning (DRL) to improve scheduling efficiency by modeling the problem as an MDP and using graph representations, outperforming eight mainstream DRL algorithms and two heuristic algorithms in an automobile part scheduling case. Future work will focus on multi-objective optimization, integrating unified enterprise-task features, and leveraging decision transformers for enhanced performance [7]. Other notable approaches include a task scheduling algorithm that uses a waiting time matrix with Fibonacci heap principles, outperforming BATS, IDEA, and BATS+BAR in waiting time and CPU efficiency [8]. Nature-inspired algorithms such as PSO and GA have also shown superiority over traditional methods in the case of resource underutilization and overutilization, with hybrid approaches improving makespan, utilization [9]. A systematic literature review of task scheduling in cloud computing, analyzing 63 articles from 2021 to 2024, categorizing various metaheuristic techniques, and highlighting the challenges of improving resource utilization, reliability, and performance, with suggestions for future enhancements and research in this evolving field [10].

III. Proposed Methodology

The section in detail discuss about the proposed PSORL Scheduler, its algorithmic framework, task-to-VM mapping strategy.

A. System Overview

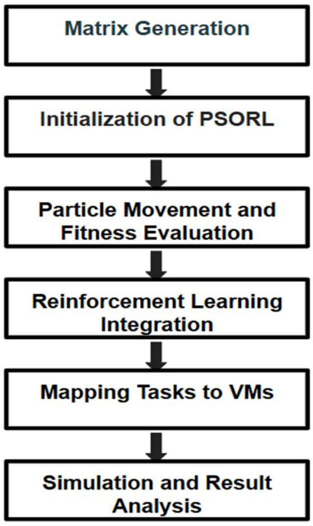

Figure 1. Architecture of Proposed Systems

The PSORL Scheduler is designed as a hybrid system combining (PSO) and (RL) to overcome the difficulty of efficient task scheduling in cloud environments. Fig.1 represents the task scheduling framework that combines (PSO) with (RL) to achieve efficient and adapted scheduling. The process begins with the matrix generation, where work and resource information is prepared. After this, after the commencement of PSORL, where the initial particles (representing potential scheduling solutions) are set. These particle goes through movement and fitness evaluation, where their positions are adjusted to optimize the allocation of work based on defined fitness norms. Strengthening learning integration increases this process by enabling the system to learn from previous scheduling decisions and to improve task mapping dynamically. Once the adaptation is completed, the tasks are mapped into the virtual machines (VMs), ensuring efficient resource usage. The framework then undergoes simulation and result analysis to evaluate the effectiveness of the scheduling approach under various scenarios. Based on these results, the final scheduling is determined, implements the allocation scheme adapted for execution. This hybrid approach takes advantage of the strength of PSO and RL to effectively handle complex and dynamic functioning challenges.

B. PSORL Scheduler

The PSO algorithm is enhanced with an adaptive inertia weight facility, which adjusts the convergence of the herd to maintain a balance between exploration and exploitation. The reinforcement learning (RL) is integrated into the system to refine the work-to-VM mapping mechanisms, improves the adaptability of the schedule over time. Fitness functions used, making a well-balanced trade-off between execution time and use, adds minimization and resource optimization. RL prescheduled contributes to learning from results, improves the quality of progressive mapping decisions.

C. Task-to-VM Mappig Strategy

The task-to-VM mapping logic considers connectivity and time consumption to ensure optimal performance. When possible, communication between VMs is reduced by having projects in highly interdependent groups on the same VM. Utilization time is calculated using historical data and dynamic monitoring, allowing schedulers to better prioritize tasks based on their computational demands and urgency NO OF TASKS (number of scheduled tasks), NO OF DATA CENTERS (data mean number of clouds), and PSO parameters such as swarm size, inertia load, acceleration coefficients These parameters, which are defined to guide the algorithm, are tuned based on the needs of the system to achieve the best balance between speed and accuracy in design. For example, a higher swarm size can improve the quality of the solution at the expense of increased computational time, while adaptive parameters ensure flexibility between tasks.

IV. Implementation Details

Section presents implementation details such as environment used for simulation and objective functions.

A. System Overview

The implementation was developed using Cloudsim, an open-source framework is widely used to simulate the cloud computing environment. Cloudsim was chosen for the ability to model various cloud elements including its flexibility, modularity and data centers, virtual machines (VMs), scheduling algorithms, and others. Support for Custom algorithm and accurate performance metrics ensured access to a suitable platform for the proposed PSORL Scheduled Testing. The simulation includes 30 functions (cloudlets) that need to be determined in 5 data centers. Each task represents a computational job with a specific execution requirement, and the data centers provide resources to execute these tasks. These functions are distributed based on their execution and communication requirements. Which is modeling using two major matrices execution time matrix (Execmatrix) and communication time matrix (Commmatrix). Execmatrix stores the execution time required for each task in each data center, in which the execution in various data centers with random generated values between 10ms and 500ms to simulate variability in time. Commmatrix represents the communication time required for data transfer between tasks and data centers, in which the network has random generated values to simulate delay. These matrix clouds capture computational and communication requirements of functions in the environment, ensuring accurate representation of the functioning scheduling scenarios.

B. Objective Function

The objective considering both the function performance time and communication time evaluate the efficiency of a given business data center mapping. Equation (1) is the objective is to reduce overall processing time and communication in all tasks.

$$f(\text{mapping}) = \sum_{i=1}^{n} (\text{ExecMatrix}[i][\text{mapping}[i]] + \text{CommMatrix}[i][\text{mapping}[i]]) \quad (1)$$

where:
n is the Number of the task
ExecMatrix[i][mapping[i]] is the execution time for task i in the assigned data center.
CommMatrix[i][mapping[i]] is the communication time for task i with the assigned data center Mapping[i] represents the data center assigned to task i.

V. Results and Discussion

The section presents the evaluation and analysis of the suggested model.
Results discuss the performance of the proposed PSORL scheduler against traditional scheduling methods such as RR, FCFS, SJF and PSO method. Makespan was calculated as primary metric to evaluate the effectiveness of each of these approaches for functioning scheduling. Traditional methods such as RR, FCFS and SJF, although easy to apply, complex workflows and dynamic clouds were shown boundaries. As a result of these methods, the makespan increased due to often different assertions and inability to adapt to the work preferences. On the other hand, the PSO showed significant improvement in Makespan by giving priority to the works. However, while the

PSO efficiently reduced the makespan, its static nature limited its adaptation ability to dynamic changes in the cloud environment and caused potential disabilities in real time scenario. To remove these boundaries of PSO and traditional methods, a new approach was developed, known as PSORL. This hybrid method balances the workload with the ability to find optimal solutions to quickly find the ability of PSO and learn reinforcement. Table1, provides a comparative analysis of the makespan obtained by separate different scheduling methods, which highlights the better performance of the PSORL. PSORL performed better in all other methods in the dynamic cloud environment with a consistent low makespan. Strengthening learning allowed the scheduler to adapt to real -time charge variations and environmental changes, ensuring more efficient resource usage and scalability under the increase in workload. This adaptability addresses PSO's inherent limitations, making PSORL a robust and efficient solution for cloud task scheduling. Although the hybrid model may lead to a longer calculation time for hybrid models based on the training mechanism, this outweighs benefits obtained in makespan reduction. Table I. represents Comparative analysis of makespan between PSO and PSORL, highlighting the improved efficiency of the PSORL algorithm. Fig.2 shows the graphical representation of Comparative analysis of makespan between PSO and PSORL highlighting the efficiency of the PSORL algorithm

TABLE I. COMPARATIVE ANALYSIS OF MAKESPAN ACROSS DIFFERENT TASK SCHEDULING METHODS

Methods	Makespan (ms)
FCFS	6871.4817
RR	6213.8404
SJF	5006.4356
PSO	2819.3468
PSORL	1814.9833

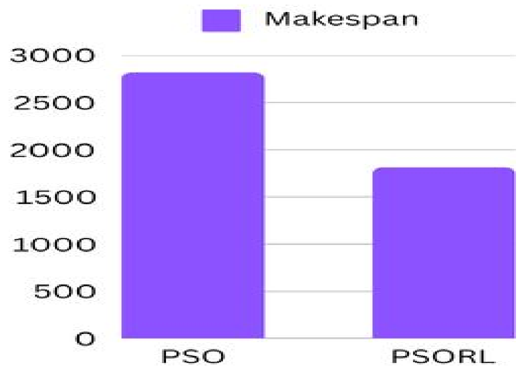

Figure 2. Comparative Analysis of Makespan Between PSO And PSORL Highlighting the Efficiency of The PSORL Algorithm

VI. CONCLUSION

The PSORL algorithm combines the basic idea of PSO and Reinforcement Learning so as to cope with several difficult problems in the process of scheduling for cloud computing systems. Since optimization is based on PSO, while RL manages adaptive decisions, the use of PSORL can remarkably minimize makespan by comparing the three conventional methods (FCFS, RR, SJF), together with PSO alone. This ensures that the system is able to adapt dynamically to changes in workload, thereby enhancing overall system performance through efficient execution of tasks. With enhanced efficiency and the ability to adapt, PSORL is particularly suited for realistic industrial applications in data-intensive industries such as healthcare, finance, and IoT. These fields require robust task scheduling and optimal resource allocation to manage complex and time-sensitive operations. PSORL not only minimizes the time taken to complete tasks but also optimizes the resource usage, which makes it a very useful tool that can be employed to improve performance in dynamic environments of resource constraint.

REFERENCES

[1] Mathew, Teena, K. Chandra Sekaran, and John Jose. "Study and analysis of various task scheduling algorithms in the cloud computing environment." 2014 International conference on advances in computing, communications and informatics (ICACCI). IEEE, 2014.

[2] Arunarani, A. R., Dhanabalachandran Manjula, and Vijayan Sugumaran. "Task scheduling techniques in cloud computing: A literature survey." Future Generation Computer Systems 91 (2019): 407-415.
[3] Dong, Tingting, et al. "Task scheduling based on deep reinforcement learning in a cloud manufacturing environment." Concurrency and Computation: Practice and Experience 32.11 (2020): e5654.
[4] Abdel-Basset, Mohamed, et al. "Task scheduling approach in cloud computing environment using hybrid differential evolution." Mathematics 10.21 (2022): 4049.
[5] Gupta, Sachi, et al. "Efficient prioritization and processor selection schemes for heft algorithm: A makespan optimizer for task scheduling in cloud environment." Electronics 11.16 (2022): 2557.
[6] Imene, Latreche, et al. "A third generation genetic algorithm NSGAIII for task scheduling in cloud computing." Journal of King Saud university-computer and information sciences 34.9 (2022): 7515-7529.
[7] Wang, Xiaohan, et al. "Solving task scheduling problems in cloud manufacturing via attention mechanism and deep reinforcement learning." Journal of Manufacturing Systems 65 (2022): 452-468
[8] Lipsa, Swati, et al. "Task scheduling in cloud computing: A priority-based heuristic approach." IEEE access 11 (2023): 27111-27126.
[9] Prity, Farida Siddiqi, Md Hasan Gazi, and KM Aslam Uddin. "A review of task scheduling in cloud computing based on nature-inspired optimization algorithm." Cluster computing 26.5 (2023): 3037-3067
[10] Abraham, Olanrewaju L., et al. "Task Scheduling in Cloud Environment—Techniques, Applications, and Tools: A Systematic Literature Review." IEEE Access (2024).