EAI Endorsed Transactions
on Internet of Things
Review Article EAI.EU

A Systematic Review on Various Task Scheduling Algorithms in Cloud Computing

Mallu Shiva Rama Krishna[1], Sudheer Mangalampalli[1,*]

[1] School of Computer Science and Engineering, VIT-AP University, Amaravati, AP, India

Abstract

Task scheduling in cloud computing involves allocating tasks to virtual machines based on factors such as node availability, processing power, memory, and network connectivity. In task scheduling, we have various scheduling algorithms that are nature-inspired, bio-inspired, and metaheuristic, but we still have latency issues because it is an NP-hard problem. This paper reviews the existing task scheduling algorithms modelled by metaheuristics, nature-inspired algorithms, and machine learning, which address various scheduling parameters like cost, response time, energy consumption, quality of services, execution time, resource utilization, makespan, and throughput, but do not address parameters like trust or fault tolerance. Trust and fault tolerance have an impact on task scheduling; trust is necessary for tasks and assigning responsibility to systems, while fault tolerance ensures that the system can continue to operate even when failures occur. A balance of trust and fault tolerance gives a quality of service and efficient task scheduling; therefore, this paper has analysed parameters like trust and fault tolerance and given research directions.

Keywords: Task Scheduling, Machine Learning, Cloud Computing, nature-inspired algorithms

Received on 14 September 2023, accepted on 19 November 2023, published on 06 December 2023

doi: 10.4108/eetiot.4548

*Corresponding author. Email: sudheerkietmtech@gmail.com

1. Introduction

Cloud computing is a technology that has changed the way organizations store and access data. It enables the provision of computing services over the Internet, eliminating the need for traditional data centres that require hardware, software, licenses, infrastructure, and teams of experts. Public, private, and hybrid clouds are the three types of cloud computing deployment models, each with its advantages. Public clouds are accessible to everyone and cost-effective, while private clouds are more secure and offer greater control. Public and private clouds are combined in hybrid clouds. With rapid development of processing, storage technologies, and advancements of Internet, computing resources become more affordable, durable, globally available than before. Cloud computing is a popular service technology in the business world today. Cloud infrastructure provides users with flexible online access to computing resources available over the Internet on a pay-as you go basis. A scheduling method is predetermined in a typical cloud environment, and it is applied to each request which might not always be the best option for a given request. This results in a game of chance where the selected algorithm might be the best for some user requests.

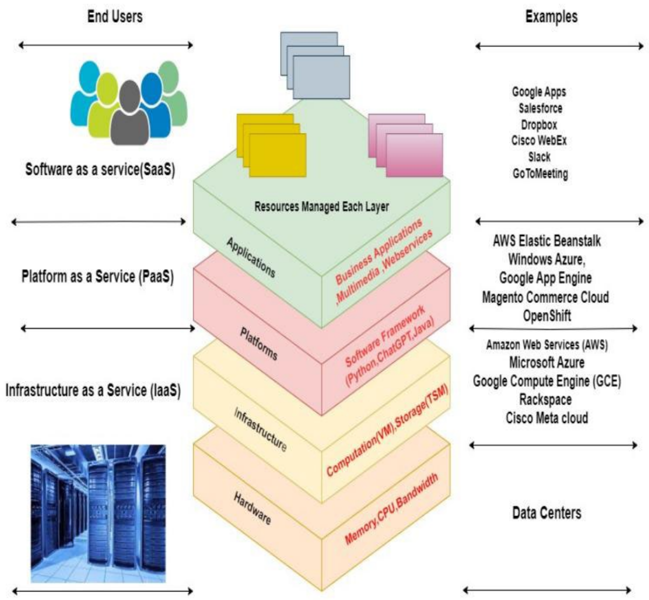

Figure 1: Cloud Computing Architecture

The many cloud computing architecture models resemble a tiered cloud computing model. Customers can choose from a different deployment model based on business needs and security considerations. SaaS gives access to software, databases based on SLA Chosen by Customer. while infrastructure, platforms are managed by service providers. PaaS provides access to customers for development of applications. Finally, IaaS customers get external support in the form of, servers, storage VMs networks. [1].

An efficient and dynamic task scheduler is needed to manage these types of requests and distribute resources to various cloud users when users request services from the cloud environment at the same time and these requirements are made by customers from various heterogeneous, diversified resources. [2]. By offering extra resources like additional memory, storage, or processors, virtualization in cloud computing also enables us to set up access control over the resources to safeguard them and enable optimal resource usage. [3]. Task scheduling is essential for enhancing cloud computing systems' performance. Most of the literature in this area has a common objective that can be summed up as follows: to maximize resource benefit, cut expenses, and enhance performance. a history of recent work on improving and optimizing the Existing task scheduling algorithms for cloud computing, considering different parameters, is provided in this study. [4]. Task scheduling is essentially an NP-hard problem [5]. The study directly addresses the scheduling of tasks via cloud computing, which has a direct impact on service quality metrics such as the makespan, rate of task rejection, resource utilization, energy consumption. VMs are instances of an operating system which runs as a partition inside a physical machine. Multiple VMs runs on a single physical Host [6]. When Workload on a physical host increases it should dynamically migrate its workload to another VMs with negligible downtime to users this is called migration.

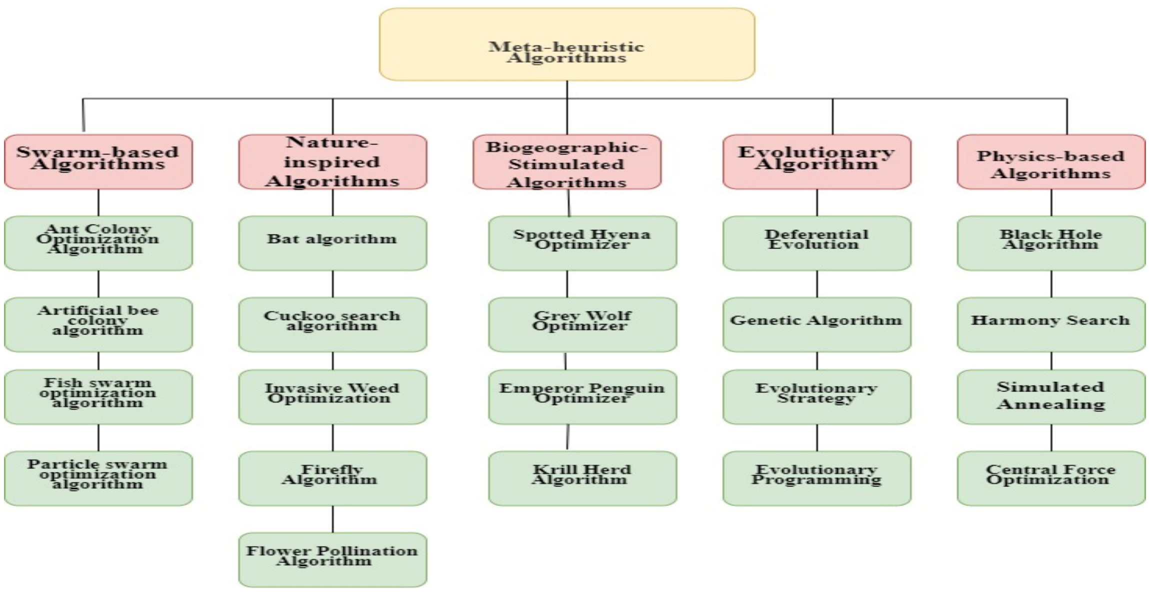

Figure 2: Classification of Metaheuristic Algorithms

Task scheduling is an NP-hard problem. Many researchers believe there are non-polynomial time algorithms for solving NP-hard problems. Contribution in this paper represents below:

- Review of various task scheduling algorithms modeled by earlier authors
- An effective analysis was done on existing task scheduling algorithms and parameters.

- Research directions are given for effective task scheduling in cloud paradigm, as it is a highly dynamic issue.

In this section, we discussed the importance of cloud computing, task scheduling. rest of the paper is Previous researchers used metaheuristic algorithms. Metaheuristic optimization deals with optimization problems using metaheuristic algorithms. Figure 2 shows the different metaheuristic algorithms.

2.Related Works

This section gives an overview of several studies to explore the feasibility of task scheduling in cloud computing. Authors in [7] proposed the Whale Optimization Algorithm and a double adaptive Whale Optimization Algorithm, which decreased makespan while increases resource utilization. It mainly addressed the parameters execution time, resource use, makespan of computational cost, and completion time. All simulations have been performed on CloudSim. and it evaluated against baseline approaches. The results revealed increased parameters such as overload, peak demand, and resource utilization. For this approach, the authors in [8] proposed a task-scheduling algorithm based on decision tree is proposed in heterogeneous environment. The author introduces a new Task Scheduling-Decision Tree approach for allocating executing application's tasks. algorithms by reducing makespan, improving resource utilization, and improving load balancing, more performance parameters could be affected. The [9] authors for cloud scheduling, load balancing, swarm intelligence-based meta-heuristic algorithms are more appropriate. In this paper, we provide an adaptive task scheduling method based on particle swarm optimization that decreases task execution time while increasing throughput, average resource utilization. The proposed scheduling algorithm has dominated the existing baseline approaches by specified parameters. in [10] The CBT Scheduling Problem, formulated by authors which addresses task duration, energy consumption. A heuristic was added to h-DEWOA to improve resource assignment. Real-world cloud workloads from CEA-Curie, HPC2N are utilized to assess how well scheduling methods perform using the CloudSim simulator. authors in [11] proposed workflow scheduling algorithms that address QoS Parameters i.e., time, cost, resource utilization. A Hybridized approach used by the authors to tackle scheduling problems. a hybridized approach includes PFA and OBL Algorithms proposed scheduler puts particular emphasis on following parameters: resource utilization, cost, and overall execution time. The workflowsim toolset is used to run simulation experiments to gauge how effective the approach suggested entails. From Simulation results its shows s a significant impact on SOTA Algorithms. authors in [12] proposed a deadline aware scheduling algorithm using Firefly-based optimization that used its high performance to optimally schedule tasks among the Green Data Centres. It was implemented on Cloudsim. In [13], authors aimed to develop a dynamic scheduling methods in cloud paradigm based on MFO. it was implemented on CloudSim and evaluated over existing baseline approaches. From results, it shown huge impact on parameters i.e resource utilization, and makespan. In [14], BCSV scheduling algorithm proposed to tackle load balancing in cloud computing. Nit was implemented on CloudSim and compared against existing scheduling algorithms and results shows that The BCSV outperforms SOTA approaches. In [15] introduced a novel combination of the gravitational emulation local search algorithm with the genetic algorithm for cloud computing. hybrid weighted current ant colony optimization technique, is suggested to be used. in [16] authors propose HWACO which is a nature inspired algorithm uses to achieve convergence in short time. Entire experimentation conducted on CloudSim and this approach evaluated over existing scheduling algorithms. From results it evident that HWACO is more dominant towards QoS Parameters. In [17], the authors focused on a DAMPA to solved task.

scheduling in cloud paradigms. Because task scheduling in cloud computing directly influences quality of service (QoS) metrics including makespan, energy consumption, resource utilization rate, task response time, task rejection rate, this study concentrates on the first layer. It is an NP-hard problem to map set of tasks to an appropriate virtual resource when QoS parameters are present. In this paper, the author describes how the DAMPA algorithm provides the best optimal task scheduling in terms of computational cost, energy consumption, makespan, computational time, cost, resource utilization. It is implemented in CloudSim. In [18] proposed task scheduler algorithm is a firefly optimization algorithm used to model our TAFFA, entire simulation was carried out on CloudSim. Authors created jobs using various randomized task distributions. To evaluate the effectiveness of TAFFA, authors used the benchmark work logs for the HPC2N and NASA computing clusters. In this research, the optimal parameters are energy consumption, execution makespan, and execution cost.

Table 1. parameters addressed by varied existing algorithms

Ref. No.	Scheduling parameter	Finding	Limitations
[7]	Computation Cost, Makespan and Execution Time.	Decreasing makespan by increasing resource utilization	Energy consumption, optimization response time, and workflow scheduling

			were not considered
[8]	Cost and Task completion time	Reduced cost and makespan. Number of the task completed within the deadline and maximizing resource utilization	Fault Tolerance and Scalability are not considered
[9]	CPU time, Makespan, load balancing, stability, and throughput	optimized task-to-resource mapping strategy	response time not Optimized
[10]	Makespan and completion time	optimize both makespan and energy consumption	Complexity issue
[11]	Makespan and increasing the resource utilization	identifies the appropriate VM for reducing the makespan	It was hard to find the alternate VM.
[12]	Time to completion, average response time, and Quality of Service parameters	Improve algorithm performance by managing quality of service (QoS)	Stochastic service system theory for improving cloud computing in resource management
[13]	Makespan and energy consumption	reduce the makespan and execution time.	Long-time consumption and complexity
[14]	Execution time	minimize execution time, minimize execution cost, and maximize resource utilization	workload limit for each VM
[15]	Resource utilization cost	Improved response	Complex task jobs
		time and accuracy	were difficult to handle
[16]	Execution of tasks, completion time, energy consumption, fault tolerance, and resource use	maximizing cloud computing's performance.	This method did not consider task priority, memory usage, and overhead
[17]	Degree of imbalance Makespan, and CPU time,	Energy consumption and lower makespan and greater throughput	Not addressing the metrics of Network utilization memory utilization,
[18]	Makespan	Improve the Quality of service and reduced the turnaround time	Unable to predict upcoming tasks from different Users
[19]	Makespan, Energy Consumption	Improved Task Execution time by decreasing energy consumption	Overall Quality of service need to be improve
[20]	Makespan, Degree of Imbalance, VM Failure rate	Decreasing makespan by increasing resource utilization	Author unable to implement in real world environment

Previous researchers used metaheuristic algorithms. Metaheuristic optimization deals with optimization problems using metaheuristic algorithms. Figure 2 shows the different metaheuristic algorithms. From Table 1. The optimum algorithm is proposed for each of the following four criteria: cost, degree of imbalance, makespan, and throughput, for a given set of tasks.

3. Research Directions

Based on the literature we address, the majority of the scheduling algorithms are focused on quality of service, performance, scalability, energy consumption, load balancing, resource utilization, response time, and makespan. While observing the existing scheduling algorithms, trust and fault tolerance have scope to metrics include execution time, probability of failure, reliability. Performance metrics are measured using a Trust-Aware scheduling algorithm. The proposed trust-aware scheduling outperforms the various approaches [21]. performance metrics are measured across different types of tasks, probability of failure, reliability, execution time. Calculating the trust level of VMs, determining the priority level of tasks, scheduling with confidence. trust score for each VM is calculated based on performance and prioritizes tasks based on their resource requirements and associated prices. Trust-Aware algorithm is designed to consider fault-tolerant behaviour, reliability. trust rating of resources is divided into direct trust, reputation, and relative trust, the proposed trust-based scheduling algorithm depends on trust value of the virtual machines. improve. Machine learning can optimize task scheduling process and improve overall efficiency of task scheduling.

3.1. Trust Aware Scheduling Algorithm

In the scalable cloud environment, trust-based, dependent task scheduling is a prominent feature. We focus on the deployment of dependent tasks. Performance metrics include execution time, probability of failure, reliability. Performance metrics are measured using a Trust-Aware scheduling algorithm. The proposed trust-aware scheduling outperforms the various approaches. performance metrics are measured across different types of tasks, probability of failure, reliability, execution time. Calculating the trust level of VMs, determining the priority level of tasks, scheduling with confidence. trust score for each VM is calculated based on performance and prioritizes tasks based on their resource requirements and associated prices. Trust-Aware algorithm is designed to consider fault-tolerant behavior, reliability. trust rating of resources is divided into direct trust, reputation, and relative trust, the proposed trust-based scheduling algorithm depends on trust value of the virtual machines.

3.2. Fault tolerance scheduling Algorithm

Fault-tolerant scheduling was evaluated against a set of benchmark workflows, and the results showed its effectiveness in providing fault tolerance, reducing total execution time, and improving system scalability. Cloud computing provides a platform for large-scale workflow processing and allows tasks to be replicated for fault tolerance. In addition, it allows efficient storage and processing of large data sets. Identify potential errors and plan strategies to mitigate their impact. Implementation mechanisms for failure detection and recovery ensure that workflow tasks can continue to execute even if one of them fails. The fault-tolerant scheduling algorithm indicates that the user's QoS requirement is improved. A fault-tolerant resource scheduling strategy performs better than a non-fault-tolerant resource scheduling algorithm considering various performance metrics such as duration, average execution time, and number of completed jobs.

4. Conclusion

The purpose of the proposed framework is to develop a novel strategy for dynamically scheduling jobs in an infrastructure as a service cloud environment. As compared with earlier efforts, this strategy makes use of Machine learning approaches to to dynamically predicting the scheduling algorithm for the incoming request. The optimum algorithm is proposed for each of the following four criteria: degree of imbalance, makespan, cost and throughput, for a given set of tasks. Based on the parameters that need to be optimized, the task scheduler unit produces the optimal scheduling algorithm to be applied. In this paper, the newly modeled task scheduling algorithms in both cloud and cloud computing environments are studied, and different task scheduling parameters are used to compare existing algorithms. MFO-Based Task Scheduling Hybrid. A WCO algorithm, DAMP, TAFA Algorithm are proposed recently by authors. These algorithms are based on nature-inspired, bio-inspired, and meta-heuristics. From Figure 2, we can see that resource utilization and system performance are covered in 70 to 80 percent of the discussed algorithms for improvement. But there is still a lot of work required to improve trust and fault tolerance. Machine Learning based task scheduling fulfills all the requirements of the cloud computing paradigm. While we are using deep reinforcement learning in cloud computing, task scheduling helps optimize resource allocation and task execution.

Conflict of Interest

Authors declares that no conflicts of interest.

Acknowledgements

I would like to express my sincere gratitude to my Ph.D. research supervisor Dr. Sudheer Mangalampalli for his valuable and constructive suggestions during the planning and development of this research work.
Financially, this work does not support any organisation/funding agency.

References

1. Shetty, C., Sarojadevi, H., & Prabhu, S. (2021). Machine learning approach to select optimal task scheduling algorithm in cloud. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(6), 2565-2580.
2. Mangalampalli, S., Karri, G. R., & Kose, U. (2023). Multi Objective Trust aware task scheduling algorithm in cloud computing using Whale Optimization. Journal

of King Saud University-Computer and Information Sciences, 35(2), 791-809.
3. Kumar, M. S., & Karri, G. R. (2023). Eeoa: cost and energy efficient task scheduling in a cloud-fog framework. Sensors, 23(5), 2445.
4. Chen, D., & Zhang, Y. (2023). Diversity-Aware Marine Predators Algorithm for Task Scheduling in Cloud Computing. Entropy, 25(2), 285.
5. Li, X. (2023). An IFWA-BSA Based Approach for Task Scheduling in Cloud Computing. Journal of ICT Standardization, 45-66.
6. Chen, D., & Zhang, Y. (2023). Diversity-Aware Marine Predators Algorithm for Task Scheduling in Cloud Computing. Entropy, 25(2), 285.
7. Manikandan, N., Gobalakrishnan, N., & Pradeep, K. (2022). Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment. Computer Communications, 187, 35-44.
8. Mahmoud, H., Thabet, M., Khafagy, M. H., & Omara, F. A. (2022). Multiobjective task scheduling in cloud environment using decision tree algorithm. IEEE Access, 10, 36140-36151.
9. Nabi, S., Ahmad, M., Ibrahim, M., & Hamam, H. (2022). AdPSO: adaptive PSO-based task scheduling approach for cloud computing. Sensors, 22(3), 920.
10. Chhabra, A., Sahana, S. K., Sani, N. S., Mohammadzadeh, A., & Omar, H. A. (2022). Energy-aware bag-of-tasks scheduling in the cloud computing system using hybrid oppositional differential evolution-enabled whale optimization algorithm. Energies, 15(13), 4571.
11. Talha, A., Bouayad, A., & Malki, M. O. C. (2022). An improved pathfinder algorithm using opposition-based learning for tasks scheduling in cloud environment. Journal of Computational Science, 64, 101873.
12. Ammari, A. C., Labidi, W., Mnif, F., Yuan, H., Zhou, M., & Sarrab, M. (2022). Firefly algorithm and learning-based geographical task scheduling for operational cost minimization in distributed green data centers. Neurocomputing, 490, 146-162.
13. Radhika, D., Duraipandian, M., Kaliyapuram, C., & Nadu, T. Virtual Machine Task Classification Using Support Vector Machine and Improved MFO Based Task Scheduling.
14. Chiang, M. L., Hsieh, H. C., Cheng, Y. H., Lin, W. L., & Zeng, B. H. (2023). Improvement of tasks scheduling algorithm based on load balancing candidate method under cloud computing environment. Expert Systems with Applications, 212, 118714.
15. Praveen, S. P., Ghasempoor, H., Shahabi, N., & Izanloo, F. (2023). A hybrid gravitational emulation local search-based algorithm for task scheduling in cloud computing. Mathematical Problems in Engineering, 2023.
16. Chandrashekar, C., Krishnadoss, P., Kedalu Poornachary, V., Ananthakrishnan, B., & Rangasamy, K. (2023). HWACOA scheduler: Hybrid weighted ant colony optimization algorithm for task scheduling in cloud computing. Applied Sciences, 13(6), 3433.
17. Chen, D., & Zhang, Y. (2023). Diversity-Aware Marine Predators Algorithm for Task Scheduling in Cloud Computing. Entropy, 25(2), 285.
18. Mangalampalli, S., Karri, G. R., & Elngar, A. A. (2023). An Efficient Trust-Aware Task Scheduling Algorithm in Cloud Computing Using Firefly Optimization. Sensors, 23(3), 1384.
19. Kumar, M. S., & Kumar, G. R. (2023). EAEFA: An Efficient Energy-Aware Task Scheduling in Cloud Environment. EAI Endorsed Transactions on Scalable Information Systems.
20. Medishetti, S. K., & KARRI, G. R. (2023). An Improved Dingo Optimization for Resource Aware Scheduling in Cloud Fog Computing Environment. Majlesi Journal of Electrical Engineering, 17(3).
21. Kumar, M. S., & Karri, G. R. (2023, August). Parameter Investigation Study On Task Scheduling in Cloud Computing. In 2023 12th International Conference on Advanced Computing (ICoAC) (pp. 1-7). IEEE.