

A Review of Diverse Optimization Strategies for Task Scheduling in Cloud Computing

Sheetal Antony¹, Dr. Sujatha SR² and Dr. Chanchal Antony³

^{1,2}Sri Siddhartha Institute of Technology/CSE, Tumakuru, India

Email: antony89sheetal@gmail.com

³AJ Institute of Engineering & Technology/AI ML, Mangaluru, India

Email: antonychanchal@gmail.com

Abstract—Applications requiring large amounts of computation are increasingly being run on heterogeneous distributed computing platforms. When it comes to pay-per-use computing and storage resources, cloud computing is a paradigm for on-demand provisioning. Data centers are responsible for managing these resources, which are dynamically assigned to users based on criteria including availability, demand, and quality standards. Cloud computing manages a wide range of virtualized resources. Clients use hundreds of virtual resources for each job. By reducing time loss, task scheduling seeks to maximise performance. Several studies have been conducted in this area, resulting in an extensive list of tactics and measurements specific to cloud environments. In this study, scheduling approaches are examined, along with related problems and constraints. Arranged according to techniques, uses, and metrics based on parameters, the review of literature offers a comprehensive perspective. It also indicates potential directions for cloud- computing scheduling research.

Index Terms— task scheduling, optimization, cloud computing, virtual machine, virtualization and load balancing.

I. INTRODUCTION

An emerging information technology model called cloud computing has become a popular corporate strategy for providing IT applications, components, and infrastructure. It upsets the conventional IT-as-a-product paradigm by changing the model of IT delivery from one that is product-centric to one that is service-centric. Since its emergence cloud computing has completely changed how IT services are created, developed, deployed, scaled, updated, maintained, and paid for. It gives people and businesses the ability to use IT resources as a metered service, anytime, anywhere, on any device. [1] Cloud computing facilitates rapid ideation and development by providing easy access to a multitude of technologies. From sophisticated technologies like the Internet of Things, machine learning, data lakes, and analytics to infrastructure services like computers, storage, and databases, users may quickly deploy resources. Technology services may be quickly deployed, greatly speeding up the concept-to-implementation process. This flexibility promotes improved client experiences and distinction by enabling experimentation, innovation, and corporate transformation. [2]

A virtual machine (VM) is quite similar to a regular physical computer. Virtual machines (VMs) are thought of as software-defined computers that live within physical servers and exist primarily as code, in contrast to real computers, where hardware components are tangible. Creating a software-based, or "virtual," image of a

computer with allotted CPU, memory, and storage resources from a remote server, like one in a cloud provider's data centre, or from a local host computer, like your own device, is known as virtualization. A virtual machine, often called an image, mimics the actions of a real computer. As observed in many work configurations, it runs in a distinct computing environment, frequently a window, and permits the operation of other operating systems or offers a full computing experience. Interestingly, the virtual machine runs independently of the host system, guaranteeing that programmes running inside the VM cannot interfere with the host computer's main operating system.

A cloud system has many jobs running at once, hence it needs a good task scheduler to make the system work well. In order to satisfy user needs regarding Quality of Service (QoS) metrics like execution time and cost, this algorithm should arrange tasks in a specific order. Energy efficiency becomes a primary consideration in modern job scheduling efforts. In cloud computing, where settings need to support several users and workloads, the creation of an effective scheduling algorithm is critical. Cloud computing application scheduling is a complex operation that requires task schedulers to effectively match impending workloads to existing resources. For a variety of workloads, efficient scheduling methods that minimise makespan, maximise resource utilisation, and take energy usage and migration time into account are crucial. [3]

Numerous probabilistic approaches, including Ant Colony Optimisation (ACO), Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), and greedy algorithms, are being investigated in relation to cloud computing job and resource distribution.[4] Even though these algorithms improve scheduling and lower costs and execution times, problems still exist. Although they are complicated and prone to local optima, genetic algorithms provide quick global search. Greedy algorithms may produce just approximations yet prioritise local solutions. Current algorithms overlook smaller businesses in favour of massive server clusters. [5] For these kinds of enterprises, optimising operations and expenses in single-server cloud platforms requires a very effective virtual machine task-scheduling algorithm.[6]

II. RELATED WORKS

In this research, the SOA-based Task Scheduling (SOATS) algorithm is presented which aims to efficiently balance five main objectives: waiting time, cost, energy consumption, makespan, and load balancing, with the least number of iterations possible. This proposed technique outperforms other meta-heuristic algorithms in task scheduling, as demonstrated by our experimental results, which are benchmarked against them. These algorithms include Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), Ant Colony Optimisation (ACO), and Whale Optimisation Algorithm (WOA). Additionally, SOATS is contrasted with other well-known scheduling techniques as Fuzzy-GA (FUGE), Moth Search Algorithm-based Differential Evolution (MSDE), and Cost-based Job Scheduling (CJS). Especially in highly loaded environments, SOATS delivers a notable 25% cost savings and 10% energy usage reduction [7].

In this paper, Hamed et al. introduce a task scheduling method that uses the Genetic method (GA) to allocate and complete different jobs in an effective manner. The main goal of the algorithm is to maximise resource utilisation while minimising job completion time and execution cost. The method is tested on two scenarios with varying amounts of tasks and processors to determine how effective it is. Ten tasks and four processors are used in the first scenario, and the computation costs are produced at random. In the second scenario, each task on a different processor has randomly generated computing expenses. There are eight processors and a range of tasks from twenty to seventy. This analysis shows that the suggested method performs noticeably better in reaching ideal outcomes for the three goals of completion time, execution cost, and resource usage. [8]

This research by proposes an upgraded ant colony optimisation algorithm-based work scheduling optimisation technique for cloud computing. First, motivated by the concepts of work scheduling in cloud computing, a scheduling model is provided that uses the enhanced ant colony algorithm to avoid local optimisation. Then, a task scheduling satisfaction function incorporates goals such as minimising waiting times, balancing resource loads, and cutting expenses associated with task completion. Refinement of the pheromone updating rules with the use of reward and punishment coefficients speeds up the solution process. The volatility coefficient's dynamic updates maximise the overall performance of the strategy. When compared to other methodologies, experimental analysis utilising Cloudsim shows greater performance in terms of convergence speed, completion time, load balancing, and virtual machine resource utilisation. [9]

In order to optimise job scheduling, this study presents MGGS, a unique method that combines a greedy strategy and a modified Genetic method (GA). In contrast to traditional techniques, MGGS requires less iterations to reach optimal answers. Based on QoS criteria, average response time, and total completion time, performance

evaluation is done against current algorithms. The outcomes of the experiments show how well MGGS performs in comparison to other task scheduling algorithms. [10]

The Enhanced Sunflower Optimisation (ESFO) algorithm is presented in this research with the goal of improving the efficiency of the existing job scheduling techniques. It effectively accomplishes polynomial time optimal scheduling. The superiority of ESFO over current approaches is demonstrated by experimental data. In particular, when compared to its best-performing counterpart, ESFO improves makespan and energy consumption by 0.73% and 2.24%, respectively. [11]

This research presents a fuzzy self-defense algorithm-based multi-objective task scheduling optimisation for cloud computing. Reducing task completion expenses, reaching resource load balance, and minimising completion time are among the goals. The success of multi-objective task scheduling is measured using a mathematical model, and an objective function is created in accordance with the model. The results of the experiments show that the suggested methodology outperforms conventional methods in terms of multi-objective maximum completion time, deadline violation rate, and virtual machine resource utilisation. [12]

The purpose of this project is to combine cloud computing and the Ant Colony Optimisation Algorithm (ACO) for resource allocation and work scheduling. In light of the constraints of ACO and the present issues with cloud computing resource allocation, the paper presents Q-ACO optimised for cost and time. Task completion time, data transfer time, task completion cost, and user satisfaction are among the evaluation criteria that are used to compare Q-ACO with RR, MM, and TCLB-EACO algorithms experimentally. The results show that, with α set to 3 and β set to 4.5, there is a considerable impact of these values on algorithm performance. With more than 500 tasks, Q-ACO performs better than other algorithms, demonstrating increased task completion speed and cost efficiency. [13]

Creating a trustworthy, effective communication environment that puts dependability, minimal delays, and energy saving first requires developing an ideal task scheduling algorithm. This paper suggests MoHHOTS, a Multi-objective Harris Hawks Optimisation (HHO)-based method designed specifically for cloud-fog computing networks, to address this problem. MoHHOTS seeks to improve task scheduling by reducing energy usage and delay at the same time. It is implemented in MATLAB and rigorously tested against industry-leading benchmarks such as cloud-fog cooperation techniques and MOGWO. MoHHOTS provides a spectrum of Pareto-optimal solutions by utilising the robust convergence and stochastic characteristics of HHO, balancing iteration methods between intensification and diversity. The simulation results demonstrate the usefulness of MoHHOTS, showing gains in transmission latency and energy usage optimisation of up to 25% over similar algorithms. [14]

In order to address the mentioned problem, this study presents an Improved Particle Swarm Optimisation (IPSO) technique with the goal of shortening the original PSO algorithm's execution time for cloud computing task scheduling.[15] The proposed Multi Adaptive Learning for Particle Swarm Optimisation (MALPSO) reduces population diversity and increases the likelihood of reaching a local optimum by using a multi-adaptive learning strategy to distinguish between ordinary particles and locally best particles during the initial population phase. Using the CEC 2017 benchmark, the paper assesses this strategy against other algorithms in four areas: makespan, load balancing, stability, and efficiency. The results show that the recommended strategy performs better than the others in terms of solving problems more quickly and producing better results for the majority of criteria. [16]

The Artificial Bee Colony Algorithm, Q-learning, and Multi-objective task scheduling optimisation are combined in this study to present the MOABCQ technique. Through concurrent limitation resolution, scheduling, resource utilisation, and load balancing amongst virtual machines (VMs) are optimised with the goal of improving cloud computing. The MOABCQ dataset produced better results in the performance evaluation, which was carried out using CloudSim and multiple methods such as Max-Min and Q-learning across Random, GoCJ, and Synthetic datasets. Compared to previous methods, it successfully decreased makespan, cost, and imbalance degree while increasing throughput and average resource utilisation.[17].

III. FUTURE SCOPE

Subsequent investigations into task scheduling optimisation may examine many pathways towards enhancement. Further investigation of hybrid optimisation strategies, which combine several algorithms to improve performance, is one interesting avenue.[18] Moreover, attention may be directed towards developing adaptive scheduling algorithms that are able to instantly adapt to shifting resource conditions and workloads. Furthermore, work scheduling with deep learning and machine learning integrated holds potential for learning from past decisions for iterative strategy refinement. Furthermore, to effectively handle massive workloads

across numerous cloud nodes, scalable and distributed task scheduling techniques are essential. All of these projects have the potential to make a big impact on the field of cloud computing task scheduling optimisation.

IV. CONCLUSIONS

This study reviews in-depth cloud computing work scheduling techniques that draw inspiration from nature. In the context of cloud computing, it provides a comparative analysis of these techniques and presents a fresh taxonomy for classification. Based on scheduling algorithms, scheduling issue features, task kinds, primary scheduling objectives, task-resource mapping schemes, scheduling constraints, and testing settings, the taxonomy classifies nature-inspired scheduling techniques. It also provides ideas for future study, making it a useful tool for practitioners and scholars alike, and it welcomes newcomers who wish to further the topic of cloud task scheduling.

REFERENCES

- [1] Sunyaev, A. (2020). Cloud Computing. In: Internet Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-34957-8_7
- [2] Pawar, Shailaja Dilip, Dipali Navnath Argade, and Vijay Vitthal Thitme. "Cloud Computing: A Review." *International Journal for Research in Applied Science & Engineering Technology* 9 (2021): 1201-1204.
- [3] Mangalampalli, S., Karri, G.R. & Kumar, M. Multi objective task scheduling algorithm in cloud computing using grey wolf optimization. *Cluster Comput* **26**, 3803–3822 (2023). <https://doi.org/10.1007/s10586-022-03786-x>
- [4] AR. Arunarani, D. Manjula, Vijayan Sugumaran, Task scheduling techniques in cloud computing: A literature survey, *Future Generation Computer Systems*, Volume 91, 2019, Pages 407-415, ISSN 0167-739X, <https://doi.org/10.1016/j.future.2018.09.014>.
- [5] Singh, P., Dutta, M. & Aggarwal, N. A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowl Inf Syst* **52**, 1–51 (2017). <https://doi.org/10.1007/s10115-017-1044-2>
- [6] Z. Zhong, K. Chen, X. Zhai and S. Zhou, "Virtual machine-based task scheduling algorithm in a cloud computing environment," in *Tsinghua Science and Technology*, vol. 21, no. 6, pp. 660-667, Dec. 2016, doi: 10.1109/TST.2016.7787008.
- [7] Ghafari, R., F. Hassani Kabutarkhani, and N. Mansouri. "Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review." *Cluster Computing* 25.2 (2022): 1035-1093.
- [8] Hamed, Ahmed Y., and Monagi H. Alkinani. "Task scheduling optimization in cloud computing based on genetic algorithms." *Comput. Mater. Contin* 69.03 (2021): 3289-3301.
- [9] Wei, Xianrong. "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing." *Journal of Ambient Intelligence and Humanized Computing* (2020): 1-12.
- [10] Zhou, Zhou, et al. "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments." *Neural Computing and Applications* 32 (2020): 1531-1541.
- [11] Emami, Hojjat. "Cloud task scheduling using enhanced sunflower optimization algorithm." *Ict Express* 8.1 (2022): 97-100.
- [12] Guo, Xueying. "Multi-objective task scheduling optimization in cloud computing based on fuzzy self-defense algorithm." *Alexandria Engineering Journal* 60.6 (2021): 5603-5609.
- [13] Su, Yingying, Zhichao Bai, and Dongbing Xie. "The optimizing resource allocation and task scheduling based on cloud computing and Ant Colony Optimization Algorithm." *Journal of Ambient Intelligence and Humanized Computing* (2021): 1-9.
- [14] A. Ali et al., "Multi-Objective Harris Hawks Optimization Based Task Scheduling in Cloud-Fog Computing," in *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2024.3391024.
- [15] Amer, D.A., Attiya, G., Zeidan, I. et al. Elite learning Harris hawks optimizer for multi-objective task scheduling in cloud computing. *J Supercomput* 78, 2793–2818 (2022). <https://doi.org/10.1007/s11227-021-03977-0>
- [16] Pirozmand, Poria, et al. "An improved particle swarm optimization algorithm for task scheduling in cloud computing." *Journal of Ambient Intelligence and Humanized Computing* 14.4 (2023): 4313-4327.
- [17] X. Chen et al., "A WOA-Based Optimization Approach for Task Scheduling in Cloud Computing Systems," in *IEEE Systems Journal*, vol. 14, no. 3, pp. 3117-3128, Sept. 2020, doi: 10.1109/JSYST.2019.2960088.
- [18] Kruekaew, Boonhatai, and Warangkha Kimpan. "Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning." *IEEE Access* 10 (2022): 17803-17818.