Improving the Processing Speed of Task Scheduling in Cloud Computing Using the Resource Aware Scheduling Algorithm over the Max-Min Algorithm

P. Sushma Priya and S. John Justin Thanjaraj
Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India

Keywords: Cloud Sim, Grid Sim, Max-Min Algorithm, Min-Min Algorithm, Novel Resource Aware Scheduling, Processing Speed, Resources, Cloud Computing.

Abstract: The study aimed to bolster the efficiency of task scheduling in cloud computing through the novel Resource Aware Scheduling Algorithm (RASA) and contrasted its performance with the Max-Min Algorithm. Cloud computing, with its decentralised nature, distributes internet-based resources, handling a diverse array of demands. For this research, data for task scheduling was sourced from the Cloudsim Tool. The cloud infrastructure was analysed, designed, and implemented to test both RASA and the Max-Min Algorithm, utilising 120 samples in two distinct groups. The processing speed outcomes, analysed via IBM SPSS, showed RASA to be notably superior. The independent sample t-test had a significance of 0.007($p < 0.05$), highlighting the distinctiveness of the algorithms. Impressively, RASA's processing speed outstripped that of the Max-Min Algorithm.

1 INTRODUCTION

The processing speed of task scheduling improves by organising incoming requests in a particular way, leading to a reduction in execution time (Nakum, Ramakrishna, and Lathigara 2014; Peng and Wolter 2019). Cloud computing offers numerous advantages, including quality of service (QoS), job scheduling, data storage, productivity improvements, and robust security measures. Effective job scheduling and system load management have become pivotal issues in the cloud computing domain (Heidari and Buyya 2019). These challenges can be adeptly tackled through the deployment of appropriate task scheduling algorithms. The Max-Min algorithm, for instance, functions within the cloud computing framework by estimating the expected completion time of tasks based on the extant resources. Task scheduling aims to diminish execution times and costs, thereby optimising performance. Analyses indicate that task scheduling can enhance processing speed (Kakaraparthi and Karthick 2022; AS, Vickram et al. 2013). This boosts the populace's quality of life by providing enhanced security and flexibility. Using the same dataset, the performance of the resource-aware scheduling algorithm is compared to a traditional scheduling method, specifically the max-min algorithm. Gridsim is utilised as a simulation framework for modelling and simulating distributed systems. This comparison unequivocally shows the superior performance of the new resource-aware scheduling algorithm over the max-min algorithm (Bandaranayake et al. 2020). Cloud computing applications encompass online data storage and services like Gmail (Sen 2017).

Over 100 articles have been showcased on ResearchGate. The Dynamic Adaptive Particle Swarm Optimisation (DAPSO) technique, designed to enhance the original PSO algorithm's functionality, reduces a task set's makespan and augments resource utilisation (Rosić et al. 2022). Due to its user-friendliness and efficacy across numerous applications, Particle Swarm Optimisation (PSO) has seen rising popularity. This study's aim is to assess cloud computing's impact on individuals' quality of life. A novel task scheduling algorithm is introduced to address historical challenges associated with the min-min and max-min algorithms (Aref, Kadum, and Kadum 2022; Vijayan et al. 2022). The RAS (Resource Aware Scheduling Algorithm) algorithm alternately applies the Max-Min and Min-Min tactics for task allocation to resources. Within the Gridsim

Priya, P. and Thanjaraj, S.
Improving the Processing Speed of Task Scheduling in Cloud Computing Using the Resource Aware Scheduling Algorithm over the Max-Min Algorithm.
DOI: 10.5220/0012544200003739

In *Proceedings of the 1st International Conference on Artificial Intelligence for Internet of Things: Accelerating Innovation in Industry and Consumer Electronics (AI4IoT 2023)*, pages 589-594
ISBN: 978-989-758-661-3

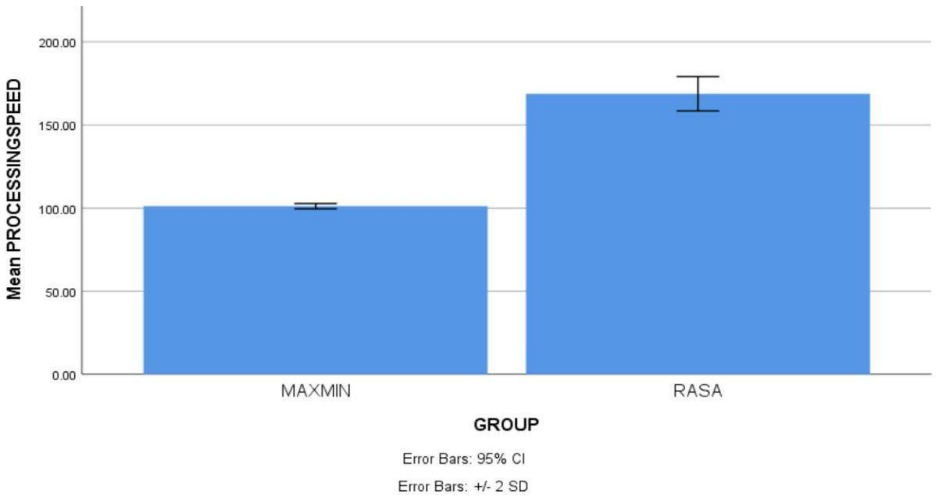

Figure 1: Bar chart showing the comparison of mean Processing Speed and standard errors.

framework, the Resource Aware Scheduling Algorithm (RASA) is employed. Practical Gridsim demonstrations highlight the merits of using RASA to schedule independent tasks within grid environments. The ACO algorithm capitalises on the generic algorithm's global search function to pinpoint the optimal solution, subsequently translating it into the ACO's initial pheromone. Simulation results suggest this algorithm outpaces genetic algorithms and ACO under comparable conditions, outshining them in larger-scale contexts. It's a potent job scheduling strategy for cloud computing (Aref, Kadum, and Kadum 2022; Guo 2017). The Improved Max-Min, a refined max-min method, was conceived by this algorithm. Rather than choosing the largest task, this method selects tasks of median or close sizes, which are then delegated to the resource with the shortest completion time. Consequently, this algorithm diminishes the overall makespan whilst ensuring workload is evenly distributed across resources (Aref, Kadum, and Kadum 2022; Guo 2017; Ming and Li 2012).

Previous task scheduling algorithms have suffered from slow processing speeds. An innovative algorithm known as RASA utilises a blend of Max-Min and Min-Min tactics in an alternating fashion to distribute tasks amongst available resources. There's a discernible research gap due to the scant focus on energy-efficient scheduling algorithms and diverse workload patterns. Moreover, there exists a demand for scheduling algorithms adept at handling multiple objectives, including accelerating response times, conserving energy, and guaranteeing equitable resource allocation. The study's primary goal is to craft the RASA Algorithm to boost processing speed.

2 MATERIALS AND METHODS

The research presented in this article was undertaken at the cloud computing laboratory within the Department of Computer Science and Engineering at the Saveetha Institute of Medical and Technical Sciences in Chennai. Initial assessments were made using pre-tests with a G-power set at 80%, a significance threshold of 0.05%, and a 95% confidence interval (Zhang et al. 2021).

The study comprised two groups, each containing 120 participants, making a total of 240 participants. Group 1 applied the RASA Algorithm, whereas Group 2 employed the Max-Min Algorithm. The experimental setup consisted of the Eclipse IDE for Java, SPSS version 26.0.1, and a laptop equipped with 8GB RAM, an Intel Gen i5 Processor, and a 4GB Graphics card. The study accommodated various types of data, encompassing documents, images, and videos. Additionally, Gridsim was utilised, a tool designed for modelling and simulating elements in parallel and distributed computing systems. This tool covered a range of entities, including resources, applications, users, and resource brokers or schedulers (Udayasankaran and Thangaraj 2023).

2.1 Group 1: Novel Resource Aware Scheduling Algorithm

The Novel Resource Aware Scheduling Algorithm (RASA) is a cutting-edge approach in cloud computing, cleverly oscillating between the Max-Min and Min-Min strategies to proficiently allocate tasks among the available resources. By taking into account aspects such as the size of the task and the time a resource takes to complete it, RASA strives to maximise processing speed. The goal is to curtail the overall duration tasks remain in the system, ensuring a balanced workload distribution, thereby elevating the efficacy of task execution in cloud ecosystems. The practical efficacy of RASA, especially when orchestrating standalone tasks within grid systems, has been substantiated through real-world Gridsim demonstrations.

To implement this, one should follow the below steps:
Step 1: Procure and install the necessary software packages.
Step 2: Integrate your dataset into the chosen coding environment.
Step 3: Extract the requisite files from Gridsim.
Step 4: Initiate the execution of the program.
Step 5: Incorporate the Resource Aware Scheduling Algorithm (RASA).
Step 6: Ascertain the parameters needed for the model to ensure an optimal fit.
Step 7: Delve deeper into the analysis, especially focusing on gauging processing speed metrics.

2.2 Group 2: Max-Min Algorithm

In the realm of distributed systems, this algorithm is frequently utilised. For each task, the anticipated completion time is determined considering the available resources. The algorithm allocates tasks to these resources based on their projected completion times, with a central objective of achieving an even distribution of workload across the resources. The primary aim is to amplify overall efficiency by curtailing the completion time of the most substantial tasks, thereby diminishing the collective processing duration.

Step 1: Set out the objective function.
Step 2: Initialise with starting values.
Step 3: Ascertain the maximum value.
Step 4: Identify the minimum value.
Step 5: Compute the intermediary values.
Step 6: Undergo repeated iterations.
Step 7: Distribute the resources appropriately.

3 STATISTICAL ANALYSIS

The study utilised IBM SPSS version 26.0.1, a statistical analysis software, to evaluate the data (George and Mallery 2021). After ensuring the datasets were normalised, they were structured into arrays. The most appropriate number of clusters was pinpointed and then assessed in conjunction with the extant algorithms. It was observed that as the number of tasks allocated to the RASA algorithm rose, there was a decline in the error rate, leading to an enhancement in processing speed. Within the dataset, the task size acted as an independent variable, whilst both task bandwidth and size were treated as dependent variables. The independent T-Test was employed to scrutinise the findings of the research.

4 RESULTS

Figure 1 presents a bar chart comparing the Processing Speed of two algorithms across varying sample sizes. The x-axis designates the algorithms, and the y-axis delineates the Processing Speed. Evidently, the Novel RASA algorithm surpasses the Max-Min Algorithm in this aspect.

Table 1 provides a breakdown of statistical outcomes based on all iteration variables. The RASA Algorithm consistently outshines the Max-Min Algorithm. The standard deviation for RASA stands at 1.3984, while for the Max-Min Algorithm it is 4.62161.

Table 1: Number of epochs taken for the RASA and Max-Min Algorithms are 20. Mean value for Group 1 is 3.7430 and Group 2 is 8.4090.

	ALGORITHM	N (Number of Epochs)	Mean	Standard Deviation	Standard Mean Error
PROCESSING SPEED (MIPS)	RASA	10	3.7430	1.39834	1.44219
PROCESSING SPEED (MIPS)	MAX-MIN	10	8.4090	4.62161	1.46148

The standard error mean for RASA is recorded at 0.44219 and it's 1.46148 for the Max-Min Algorithm. Subsequent findings from a random sample test can be perused in Table 2. Subjecting the dataset to an Independent Sample T-Test at a 95% confidence interval reveals the Novel Resource Aware Scheduling Algorithm's distinct edge over the Max-Min Algorithm.

Resource Aware Scheduling Algorithm. Resource Aware Scheduling Algorithm is better than Max-Min in terms of mean processing speed and standard deviation. X-Axis: Resource Aware Scheduling Algorithm vs Max-Min Algorithm Y-Axis: Mean Processing Speed of detection ± 2 SD.

When comparing processing speeds between RASA and the Max-Min algorithm, both exhibit a mean difference of 4.6 and a similar disparity in standard deviation. The Max-Min Algorithm's 95%

Table 2: Independent sample T-Test is applied for dataset fixing confidence intervals as 95%. The independent sample t test significance shown after the test is 0.007 ($p<0.05$) (Resource Aware Scheduling Algorithms appear to perform better than Max-Min Algorithm).

		Levene's test for equality of variables	
		F	sig
Processing speed (MIPS)	Equal variances assumed	21.781	.000
Processing speed (MIPS)	Equal variance not assumed		
		Sig(2-tailed)	
Processing speed (MIPS)	Equal variances assumed	0.07	
Processing speed (MIPS)	Equal variance not assumed	0.07	
		T-test for equality of Means	
		Mean Difference	Std.error Difference
Processing speed (MIPS)	Equal variances assumed	-4.66	1.52
Processing speed (MIPS)	Equal variance not assumed	-4.66	1.52
		95% Confidence Interval of Difference	
		Lower	Upper
Processing speed (MIPS)	Equal variances assumed	-7.87	-1.45808
Processing speed (MIPS)	Equal variance not assumed	-4.66	-1.29113

confidence interval showcases a range reaching 7.873. Following the analysis, the significance level of the independent sample t-test emerges as 0.007 ($p < 0.05$), pointing to a pronounced difference in variance, which strengthens RASA's standing. In terms of processing speed, a comparative examination between the two algorithms has been undertaken. The algorithms have also undergone distinct evaluations based on bandwidth parameters.

It is conclusive that the Novel RASA significantly surpasses the Max-Min Algorithm in performance.

5 DISCUSSION

The insights gleaned from this study confirm that the Resource Aware Scheduling Algorithm is more proficient than the Max-Min Algorithm, especially in terms of processing speed. The independent sample t-test's significance post-analysis stands at 0.007 ($p < 0.05$), underscoring the Novel Resource Aware Scheduling Algorithm's marked superiority over the Max-Min Algorithm. Parallel findings corroborate the advancements in task scheduling's processing speed. For instance, an article details the novel resource-aware scheduling algorithm underpinning task scheduling (Buakum and Wisittipanich 2022). The evaluation juxtaposes the optimized TSA with other algorithms such as Existing Min-Min, Existing Max-Min, RASA, Improved Max-Min, and Enhanced Max-Min, employing Java 7 for the simulation (Aref, Kadum, and Kadum 2022). The time each task is anticipated to take, based on available resources, is meticulously calculated (Gandomi et al. 2020). Within this framework, the scheduler's responsibility is to methodically organise all tasks within the earmarked meta-tasks in light of the resources on hand (Kamalam and Sentamilselvan 2022). As the volume of tasks and samples swells, processing speed follows suit, while the error rate inversely contracts. The propounded algorithm judiciously deploys one of these strategies, contingent on the context and computed parameters, thereby amalgamating the merits of other seasoned task scheduling methodologies (Aref, Kadum, and Kadum 2022). Cloud computing not only offers cost-effectiveness, particularly beneficial to small and medium businesses, but also augments the quality of life. RASA endeavours to strike a balance, mitigating the pitfalls of both approaches while capitalising on their strengths (Bandaranayake et al. 2020).

Factors such as makespan, load balancing, and cloud computing — which inherently deals with an expansive array of resources and requests — are pivotal determinants in task scheduling (Etminani and Naghibzadeh 2007; Tang 2018). Deploying a dataset abundant with tasks could bolster the assertion of time efficiency, all the while preserving an elevated processing speed. For future ventures, transplanting the proposed method directly into a genuine cloud computing milieu (via cloud sim) for myriad empirical evaluations, including scalability, resilience, and accessibility, could further refine task scheduling (Gandomi et al. 2020).

6 CONCLUSION

In the article under consideration, the study underscores the efficiency of the Novel Resource Aware Scheduling Algorithm, which boasts a processing speed of 3.7430 MIPS, outpacing the Max-Min Algorithm that operates at 8.4090 MIPS. The empirical results derived from the Gridsim simulator unequivocally establish RASA's dominance over the Max-Min algorithm, especially when deployed within extensive distributed systems.

REFERENCES

Aref, Ismael Salih, Juliet Kadum, and Amaal Kadum. (2022). “Optimization of Max-Min and Min-Min Task Scheduling Algorithms Using G.A in Cloud Computing.” 2022 5th International Conference on Engineering Technology and Its Applications (IICETA). https://doi.org/10.1109/iiceta54559.2022.9888542.

AS, Vickram, Raja Das, Srinivas MS, Kamini A. Rao, and Sridharan TB, (2013). "Prediction of Zn concentration in human seminal plasma of Normospermia samples by Artificial Neural Networks (ANN)." Journal of assisted reproduction and genetics 30: 453-459.

Bandaranayake, K. M. S. U., K. M. S. Bandaranayake, K. P. N. Jayasena, and B. T. G. Kumara. (2020). “An Efficient Task Scheduling Algorithm Using Total Resource Execution Time Aware Algorithm in Cloud Computing.” 2020 IEEE International Conference on Smart Cloud (SmartCloud). https://doi.org/10.1109/smartcloud49737.2020.00015.

Buakum, Dollaya, and Warisa Wisittipanich. (2022). “Selective Strategy Differential Evolution for Stochastic Internal Task Scheduling Problem in Cross-Docking Terminals.” Computational Intelligence and Neuroscience 2022 (November): 1398448.

Etminani, K., and M. Naghibzadeh. 2007. “A Min-Min Max-Min Selective Algorihtm for Grid Task Scheduling.” (2007) 3rd IEEE/IFIP International Conference in Central Asia on Internet. https://doi.org/10.1109/canet.2007.4401694.

Gandomi, Amir H., Ali Emrouznejad, Mo M. Jamshidi, Kalyanmoy Deb, and Iman Rahimi. (2020). Evolutionary Computation in Scheduling. John Wiley & Sons.

George, Darren, and Paul Mallery. (2021). IBM SPSS Statistics 27 Step by Step: A Simple Guide and Reference. Routledge.

Guo, Qiang.(2017). “Task Scheduling Based on Ant Colony Optimization in Cloud Environment.” AIP Conference Proceedings. https://doi.org/10.1063/1.4981635.

Heidari, Safiollah, and Rajkumar Buyya. (2019). “Quality of Service (QoS)-Driven Resource Provisioning for Large-Scale Graph Processing in Cloud Computing Environments: Graph Processing-as-a-Service (GPaaS).” Future Generation Computer Systems. https://doi.org/10.1016/j.future.2019.02.048

Kakaraparthi, Aditya, and V. Karthick. (2022). “A Secure and Cost-Effective Platform for Employee Management System Using Lightweight Standalone Framework over Diffie Hellman’s Key Exchange Algorithm.” ECS Transactions 107 (1): 13663–74.

Kishore Kumar, M. Aeri, A. Grover, J. Agarwal, P. Kumar, and T. Raghu, “Secured supply chain management system for fisheries through IoT,” Meas. Sensors, vol. 25, no. August 2022, p. 100632, 2023, doi: 10.1016/j.measen.2022.100632.

Kamalam, G. K., and K. Sentamilselvan. (2022). “SLA-Based Group Tasks Max-Min (GTMax-Min) Algorithm for Task Scheduling in Multi-Cloud Environments.” Operationalizing Multi-Cloud Environments. https://doi.org/10.1007/978-3-030-74402-1_6.

Ming, Gao, and Hao Li. (2012). “An Improved Algorithm Based on Max-Min for Cloud Task Scheduling.” Recent Advances in Computer Science and Information Engineering. https://doi.org/10.1007/978-3-642-25789-6_32.

Nakum, Sunilkumar, C. Ramakrishna, and Amit Lathigara. 2014. “Reliable RASA Scheduling Algorithm for Grid Environment.” Proceedings of IEEE International Conference on Computer Communication and Systems ICCCS14. https://doi.org/10.1109/icccs.2014.7068183.

Peng, Guang, and Katinka Wolter. (2019). “Efficient Task Scheduling in Cloud Computing Using an Improved Particle Swarm Optimization Algorithm.” Proceedings of the 9th International Conference on Cloud Computing and Services Science. https://doi.org/10.5220/0007674400580067.

Rosić, Maja, Miloš Sedak, Mirjana Simić, and Predrag Pejović. (2022). “Chaos-Enhanced Adaptive Hybrid Butterfly Particle Swarm Optimization Algorithm for Passive Target Localization.” Sensors 22 (15). https://doi.org/10.3390/s22155739.

Sen, Jaydip. (2017). Cloud Computing: Architecture and Applications. BoD – Books on Demand.

Tang, Ling. (2018). "Load Balancing Optimization in Cloud Computing Based on Task Scheduling." 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS). https://doi.org/10.1109/icvris.2018.00036.

Udayasankaran, P., and S. John Justin Thangaraj. (2023). "Energy Efficient Resource Utilization and Load Balancing in Virtual Machines Using Prediction Algorithms." International Journal of Cognitive Computing in Engineering 4 (June): 127–34.

Vijayan, D. S., Mohan, A., Nivetha, C., Sivakumar, V., Devarajan, P., Paulmakesh, A., & Arvindan, S. (2022). Treatment of pharma effluent using anaerobic packed bed reactor. Journal of Environmental and Public Health, 2022.

V. P. Parandhaman, (2023)"An Automated Efficient and Robust Scheme in Payment Protocol Using the Internet of Things," Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICONSTEM56934.2023.10142797.