Journal of Computers, Mechanical and Management
Journal homepage: www.jcmm.co.in
e-ISSN: 3009-075X
AAN PUBLISHING

Volume 2 Issue 5

Article Number: 23074

Intelligent Cloudlet Scheduling for Optimized Execution Time in Cloud Computing Environments

Anamika Yadav*, Hridesh Varshney, and Sarvesh Kumar

Department of Computer Science and Engineering, BBD University, Faizabad Road, Lucknow, Uttar Pradesh, India 226028

Abstract

Cloud computing has become a cornerstone of modern IT infrastructure, offering scalable and flexible resources. However, efficient resource management, particularly cloudlet scheduling, presents a significant challenge due to its NP-hard nature. This paper introduces a novel heuristic-based cloudlet scheduling algorithm aimed at minimizing execution time and improving load balancing in cloud computing environments. We detail the development and implementation of the algorithm, along with a simulation setup using the CloudSim toolkit to evaluate its performance against existing methods. Results from extensive simulations demonstrate that the proposed algorithm consistently reduces turnaround times, thus optimizing resource allocation. The findings suggest that our approach can significantly impact cloud computing efficiency, paving the way for improved service provider offerings and user satisfaction. The implications of these advancements are discussed, alongside potential directions for future research in dynamic cloud environments.

Keywords: Cloud Computing; Task Scheduling; Cloudlets; Virtual Machines (VMs); Load Balancing

1 Introduction

The landscape of computational technology has been profoundly transformed by the advent of cloud computing, which has redefined the paradigms of resource allocation, scalability, and computing on demand [1–3]. As businesses and individuals increasingly rely on cloud services for a wide range of applications, the efficiency and effectiveness of these services have become paramount [4]. Enterprises such as Amazon, Google, and Microsoft have pioneered this domain, yet despite their advancements, significant challenges persist [5]. Cloud computing environments are characterized by their ability to offer flexible and cost-effective resources, shifting from traditional capital expenses to an operational expenditure model [6–8]. This shift necessitates sophisticated strategies to manage the balance between cost and performance, requiring innovations in cloud resource scheduling and management. The scheduling of computational tasks, commonly referred to as cloudlets, within a cloud environment is a complex and critical issue [9]. The execution time of these cloudlets directly impacts the overall system performance and user satisfaction [10, 11]. The inherent difficulty of this scheduling problem, classified as NP-Hard, demands efficient heuristics to navigate towards optimal solutions [12, 13]. Current literature on cloudlet scheduling strategies reveals various approaches, ranging from static algorithms that prioritize quick and straightforward deployment to dynamic algorithms that adapt to changing system states [14, 15]. While progress has been made, there remains a gap in developing an algorithm that can consistently minimize execution times across diverse cloud environments.

*Corresponding author: anamikayadav@bbduniversity.ac.in
Received: 20 July 2023; **Revised:** 16 August 2023; **Accepted**: 29 September 2023; **Published:** 31 October 2023

DOI: 10.57159/gadl.jcmm.2.5.23074.

This research seeks to bridge this gap by proposing a novel heuristic-based cloudlet scheduling algorithm. Our approach is distinguished by its consideration of both the heterogeneity of cloud resources and the varying requirements of cloudlets. The algorithm's design aims to enhance task execution efficiency while ensuring a balanced distribution of the computational load. The validation of this algorithm is conducted through extensive simulations using the CloudSim toolkit, which provides a controlled environment to evaluate its performance against established scheduling methods [16–18]. The anticipated contributions of this research are multifaceted. Primarily, it introduces an innovative scheduling algorithm that demonstrates improved performance in execution time reduction. Additionally, it offers insights into the economic and operational benefits that such an algorithm can provide for cloud service providers. By enabling providers to optimize resource allocation, the proposed algorithm may lead to more competitive service offerings and higher levels of customer satisfaction. Furthermore, the study's findings have the potential to influence future research directions in cloud resource management, particularly in dynamic and heterogeneous computing environments. THus, the present article contributes a novel perspective to the field of cloud computing by addressing a critical challenge with a unique solution, setting the stage for further advancements in cloud resource scheduling.

2 Related Work

The landscape of cloudlet scheduling is marked by diverse methodologies and algorithms, each tailored to meet specific performance metrics and operating environments. This section reviews several notable scheduling strategies that have influenced the current study.

Batch Mode Scheduling: One of the predominant approaches in task scheduling is batch mode. The Resource-Aware Scheduling Algorithm (RASA) is particularly noteworthy, reducing makespan effectively in grid environments [19, 20]. Similarly, the Reliable Scheduling Distributed in Cloud Computing (RIDC) algorithm has been developed to optimize processing time within cloud settings [21–23]. Furthermore, an Optimal Model for Priority-based Service Scheduling Policy has been proposed, targeting high Quality of Service (QoS) and throughput, showcasing the algorithm's aptitude for enhancing cloud computing services [24]. Another contribution in batch mode scheduling is the Extended Max-Min scheduling, which integrates Petri Net for task execution and fosters efficient load balancing in cloud environments [25, 26]. In addition, the Improved Cost-Based Algorithm for task scheduling has been introduced, which enhances the computation-communication ratio, an essential factor in cloud task scheduling [27, 28]. Moreover, investigations into gang scheduling have revealed improvements in performance and cost by incorporating job migration and addressing starvation in cloud systems [29].

Dependency Mode Scheduling: Departing from batch mode, the Dependency mode scheduling focuses on task interdependencies [30, 31]. A notable algorithm in this category is the Priority-Based Job Scheduling Algorithm, which aims to minimize the finish time of tasks, demonstrating its efficiency in cloud environments [**?**]. Another significant work is the Optimistic Differentiated job scheduling system that harmonizes the QoS demands of cloud users with the profit maximization for service providers, representing a balance of user and provider interests [32]. These varied approaches highlight the dynamic nature of cloudlet scheduling. Each algorithm brings a unique perspective to solving the complex problem of scheduling in cloud computing, with an emphasis on different aspects such as QoS, load balancing, cost-effectiveness, and processing time optimization. The proposed algorithm in this study draws inspiration from these methodologies, aiming to further the advancement of cloudlet scheduling through an innovative heuristic-based approach.

3 Methods

3.1 Overview of the Proposed Approach

The proposed approach seeks to address the intricate challenge of cloudlet scheduling in cloud computing, which stands as a pivotal component in the orchestration of cloud resources. Efficient scheduling is crucial as it directly impacts the performance and cost-effectiveness of cloud services, influencing both provider revenues and user satisfaction. The core objective of this research is to minimize the total execution time of cloudlets, which in turn can significantly improve operational efficiency and achieve a more balanced load across cloud resources. In pursuit of this goal, the approach introduces a novel scheduling algorithm that transcends traditional methods by incorporating advanced heuristics. These heuristics are designed to account for the heterogeneity of cloudlets and virtual machine capabilities, optimizing the allocation process in a manner that aligns with real-world cloud performance metrics. This optimization not only seeks to enhance individual task execution but also aspires to harmonize the cumulative load distribution, thereby mitigating the risk of resource bottlenecks and underutilization. The anticipated outcome is a robust, adaptable scheduling framework that delivers tangible improvements in cloudlet processing times while maintaining a high level of resource utilization efficiency. This approach is expected to contribute to the field of cloud computing by offering a scalable solution to the cloudlet scheduling problem, ultimately facilitating the management of increasingly complex and demanding cloud environments.

3.2 Algorithm Details

The core of the proposed approach is the cloudlet scheduling algorithm, delineated in Algorithm 1, which systematically minimizes the execution time of cloudlets on a pool of virtual machines (VMs). The algorithm leverages a heuristic that prioritizes the assignment of cloudlets to the VMs based on the estimated turnaround time, thereby optimizing the overall scheduling process. The pseudocode presented in Algorithm 1 outlines the sequential steps undertaken by the scheduling mechanism. The process begins with the initialization of the minimum time variable, which is set to infinity, and a null assignment to the selected VM for each cloudlet. It then iterates over each VM to compute the projected turnaround time for the cloudlet on that VM. The cloudlet is assigned to the VM that offers the shortest turnaround time, ensuring that the cloudlet's execution is as efficient as possible. After each assignment, the state of the VM is updated to reflect its new load, thereby preparing it for the next iteration of scheduling.

Algorithm 1 Cloudlet Scheduling Algorithm

```
procedure CloudletScheduling(Cloudlets, VMs)
    for each cloudlet ∈ Cloudlets do
        minTime ← ∞
        selectedVM ← null
        for each vm ∈ VMs do
            time ← CalculateTurnaroundTime(cloudlet, vm)
            if time < minTime then
                minTime ← time
                selectedVM ← vm
            end if
        end for
        AssignCloudletToVM(cloudlet, selectedVM)
        UpdateVMState(selectedVM)
    end for
end procedure
```

The *CalculateTurnaroundTime* function is essential to the proposed scheduling algorithm. It estimates the time required for a cloudlet to execute on a given virtual machine. This estimation takes into account the processing power of the VM, the length of the cloudlet, and the VM's current load. The function is defined as Algorithm 2:

Algorithm 2 Heuristic Estimation of Cloudlet Turnaround Time

```
function CalculateTurnaroundTime(cloudlet, vm)
    mips ← vm.MIPS
    length ← cloudlet.length
    currentLoad ← vm.currentLoad
    availableMips ← mips − currentLoad
    if availableMips ≤ 0 then
        return ∞                                    ▷ VM is currently fully loaded
    else
        executionTime ← length/availableMips
        return executionTime
    end if
end function
```

This function provides a heuristic estimate rather than an exact computation, reflecting the inherent uncertainty and variability in cloud computing environments. The actual turnaround time may vary based on network latency, VM state changes, and other runtime conditions not accounted for in the static simulation. The *AssignCloudletToVM* function formalizes the allocation of the cloudlet to the selected VM, and *UpdateVMState* revises the VM's status, accounting for its new workload. These functions work in concert to optimize the scheduling process and are detailed in the subsequent subsections.

3.3 Simulation Setup

To evaluate the effectiveness of the proposed cloudlet scheduling algorithm, a series of simulations were carried out using the CloudSim toolkit, a widely recognized framework for modeling and simulation of cloud computing infrastructures and services.

The simulation environment was meticulously configured to reflect a typical cloud computing setup with the following parameters:

- **Host Configuration**: The simulation environment included 5 hosts, each equipped with 4 processing elements (PEs). These PEs represent the individual CPUs within a host.
- **Processing Power**: Each processing element was assigned a capacity of 2000 MIPS (Million Instructions Per Second), totaling 8000 MIPS per host. This metric is crucial as it determines the speed at which each host can execute cloudlets.
- **Cloudlets**: The simulation managed 100 cloudlets, which are abstractions of cloud-based application workloads. Each cloudlet has its unique computational requirements and characteristics.
- **Virtual Machines (VMs)**: A diverse set of virtual machines was deployed to ascertain the algorithm's performance under various configurations. Each VM's specification, including its processing power, memory, and bandwidth, was systematically varied to simulate different scheduling scenarios.

This setup aimed to create a controlled yet versatile cloud environment to assess the algorithm's responsiveness to different workloads and resource availability scenarios. Such a comprehensive simulation framework is essential for the validation of the proposed scheduling algorithm's adaptability and efficiency.

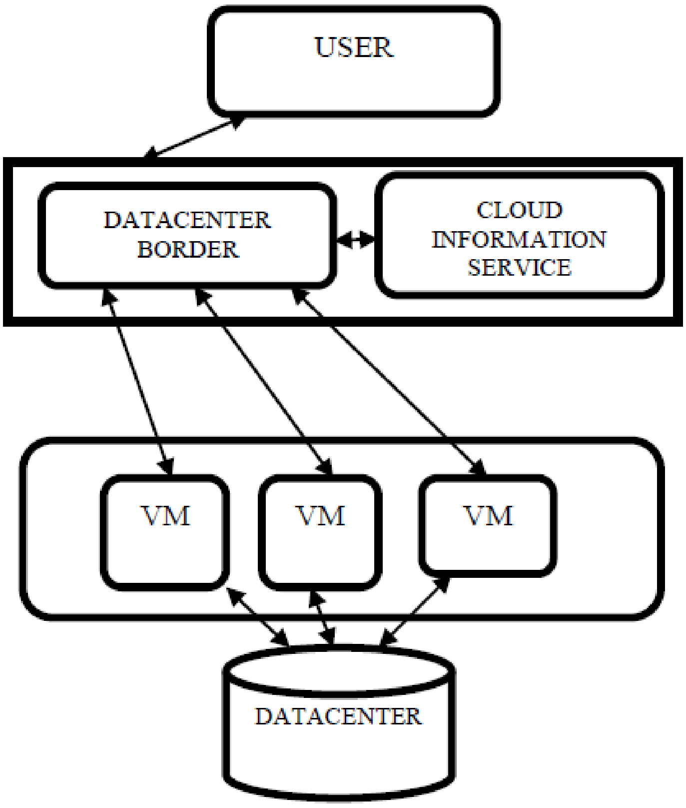

Figure 1: Cloud Computing Environment Architecture.

The architecture of the cloud computing environment utilized for the simulations is depicted in Figure 1. This schematic representation outlines the interaction between the user, the datacenter border, the cloud information service, and the virtual machines (VMs) hosted within the datacenter, demonstrating the multi-layered structure of the cloud environment.

3.4 Assumptions

To ensure a controlled environment for the simulation and evaluation of the proposed scheduling algorithm, certain assumptions were adopted. These assumptions are necessary to simplify the model, allowing for a clear analysis of the algorithm's performance without the variability inherent in real-world cloud environments. The simulation assumes:

- **Static Cloud Environment**: The cloud infrastructure is considered to be static throughout the simulation process. This implies that the number of virtual machines (VMs) and hosts remains constant, with no additions or removals during runtime. This assumption is made to focus on the scheduling algorithm's performance without the confounding effects of a dynamic infrastructure.

- **Job Independence**: All cloudlets, representing discrete computational jobs, are treated as independent units of work. There is no inter-cloudlet communication or dependencies, which simplifies the scheduling process and ensures that the performance of each cloudlet can be evaluated in isolation.

These assumptions are standard in simulation studies for cloud scheduling, providing a clear baseline from which the efficacy of the scheduling algorithm can be assessed. However, it is acknowledged that real-world cloud computing environments are dynamic and cloudlets may have interdependencies, which could be considered in future work.

3.5 Performance Metric

The efficacy of the cloudlet scheduling algorithm is evaluated using a primary performance metric that reflects the scheduling process's objectives. Turnaround Time is the chosen metric, defined as the duration between the submission of a cloudlet and the completion of its execution on a VM. This metric is critical, as it captures the efficiency of the scheduling algorithm in terms of both execution speed and resource utilization. Minimizing the turnaround time indicates an effective scheduling strategy that enhances the throughput of the cloud infrastructure. The focus on Turnaround Time aligns with the core goal of the proposed scheduling algorithm, which is to minimize cloudlets' execution time. Future studies may explore additional metrics, such as cost, energy consumption, and overall resource utilization, to provide a more comprehensive evaluation of the scheduling performance.

4 Results and Discussion

4.1 Simulation Outcomes

The simulation environment was configured with 5 hosts, each having 4 processing elements and a capacity of 2000 MIPS. A total of 100 cloudlets were processed, with varying lengths and virtual machine configurations. The simulation results, summarized in Table 1, indicate a significant reduction in turnaround time when using the proposed algorithm compared to an existing algorithm.

Table 1: Comparison between existing and proposed algorithm based on the number of virtual machines.

No. of VM	Existing Algo	Proposed Algo
4	5000.1	4250.09
8	2600.09	2200.1
16	1400.1	1200.1
32	800.1	650.1

The proposed algorithm demonstrates an improved performance in terms of lower turnaround times, which suggests that the heuristic approach employed can effectively reduce the computational complexity of job matching from NP-hard to polynomial.

4.2 Extended Simulation Analysis

To further validate the performance of the proposed algorithm, additional simulations were performed with a fixed number of 8 VMs, varying the number of cloudlets. The new simulation environment included hosts with equal specifications as before, but VM power was divided with half having 1000 MIPS and the rest 2000 MIPS. The results, presented in Table 2, show that the proposed algorithm consistently outperforms the existing algorithm across different numbers of cloudlets. Figure 2 illustrates the comparative performance improvement of the proposed algorithm over the existing one.

Table 2: Comparison between existing and proposed algorithm based on the number of cloudlets.

No. of Cloudlets	Existing Algo	Proposed Algo
50	1400.1	1200.1
100	2600.09	2200.1
150	3800.1	3250.09
200	5000.1	4250.1

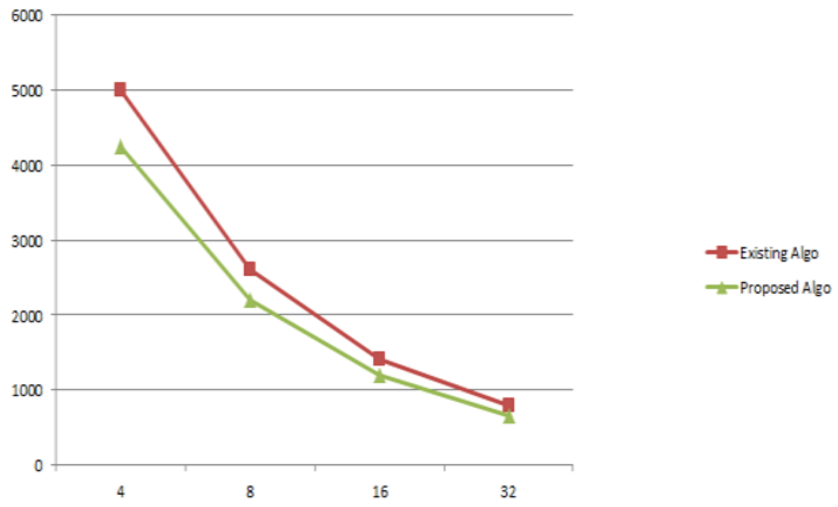

Figure 2: Graphical representation of performance comparison between existing and proposed algorithms.

These findings reinforce the superiority of the proposed algorithm in scenarios with both varying VM and cloudlet numbers, showcasing its robustness and scalability. The simulation results provide strong evidence that the proposed cloudlet scheduling algorithm is superior to the existing algorithm in terms of turnaround time. This suggests that the application of advanced heuristics can significantly optimize cloud resource allocation, which is particularly relevant in the context of cloud computing where efficient resource management is crucial for performance and cost management. The improved efficiency could have a notable impact on cloud service providers by enabling them to serve more users with the same amount of resources, potentially leading to increased revenues and customer satisfaction. It also opens the door to more sustainable cloud computing practices, as optimized resource usage directly correlates with energy consumption and environmental impact.

5 Conclusion

This study addressed the challenge of cloudlet scheduling in cloud computing, a critical factor in the performance and cost-effectiveness of cloud services. The proposed heuristic-based scheduling algorithm demonstrates significant improvements in execution times, as evidenced by the simulation results. These findings underscore the potential of heuristic approaches in overcoming the complexities of NP-hard scheduling problems within static cloud environments. By enabling more efficient resource utilization, the algorithm not only enhances the performance of cloud services but also contributes to the broader goal of sustainable and economical cloud computing. Future research directions include adapting the algorithm for dynamic cloud environments and exploring its applicability in real-world scenarios. The continued evolution of cloud computing demands such innovative solutions, and this work contributes a valuable perspective to the ongoing discourse in cloud resource management.

Declaration of Competing Interests

The authors declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Declaration

This research did not receive any grants from governmental, private, or nonprofit funding bodies.

Author Contributions

Anamika Yadav: Conceptualization, Writing – Original draft preparation; **Hridesh Varshney**: Methodology, Data curation, Investigation, Software, Validation, Writing - Reviewing and Editing; **Sarvesh Kumar**: Supervision, Data curation, Investigation, Software, Validation, Writing – Reviewing and Editing.

References

[1] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.

[2] S. Kumar, "Reviewing software testing models and optimization techniques: An analysis of efficiency and advancement needs," *Journal of Computers, Mechanical and Management*, vol. 2, no. 1, pp. 32–46, 2023.

[3] F. Faridi, H. Sarwar, M. Ahtisham, and K. Jamal, "Cloud computing approaches in health care," *Materials Today: Proceedings*, vol. 51, pp. 1217–1223, 2022.

[4] R. Buyya, S. N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, and E. e. a. Gelenbe, "A manifesto for future generation cloud computing: Research directions for the next decade," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–38, 2018.

[5] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—the business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.

[6] R. S. S. Dittakavi, "An extensive exploration of techniques for resource and cost management in contemporary cloud computing environments," *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 4, no. 1, pp. 45–61, 2021.

[7] T. Ellahi, B. Hudzia, H. Li, M. A. Lindner, and P. Robinson, "The enterprise cloud computing paradigm," in *Cloud Computing: Principles and Paradigms*, pp. 97–120, 2011.

[8] F. N. A. Sackey, *Strategies to manage cloud computing operational costs*. PhD thesis, Walden University, 2018.

[9] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends," *Swarm and Evolutionary Computation*, vol. 62, p. 100841, 2021.

[10] A. Badshah, A. Ghani, S. Shamshirband, G. Aceto, and A. Pescapè, "Performance-based service-level agreement in cloud computing to optimise penalties and revenue," *IET Communications*, vol. 14, no. 7, pp. 1102–1112, 2020.

[11] S. Shahryari, F. Tashtarian, and S.-A. Hosseini-Seno, "Copam: Cost-aware vm placement and migration for mobile services in multi-cloudlet environment: An sdn-based approach," *Computer Communications*, vol. 191, pp. 257–273, 2022.

[12] F. Xhafa and A. Abraham, *Meta-heuristics for grid scheduling problems*, pp. 1–37. 2008.

[13] R. Pellerin, N. Perrier, and F. Berthaut, "A survey of hybrid metaheuristics for the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 280, no. 2, pp. 395–416, 2020.

[14] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends," *Swarm and Evolutionary Computation*, vol. 62, p. 100841, 2021.

[15] R. Ghafari, F. H. Kabutarkhani, and N. Mansouri, "Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review," *Cluster Computing*, vol. 25, no. 2, pp. 1035–1093, 2022.

[16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[17] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities," in *2009 International Conference on High Performance Computing & Simulation*, pp. 1–11, IEEE, 2009.

[18] A. Sundas and S. N. Panda, "An introduction of cloudsim simulation tool for modelling and scheduling," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 263–268, IEEE, 2020.

[19] A. Narwal and S. Dhingra, "A systematic review of scheduling in cloud computing framework," *International Journal of Advanced Studies in Computers, Science and Engineering*, vol. 5, no. 7, p. 1, 2016.

[20] G. T. Abraham, *Group-based parallel multi-scheduling methods for grid computing*. PhD thesis, Coventry University, 2016.

[21] F. S. Gharehchopogh and B. Abdollahzadeh, "An efficient harris hawk optimization algorithm for solving the travelling salesman problem," *Cluster Computing*, vol. 25, no. 3, pp. 1981–2005, 2022.

[22] Z. Gao, Y. Fan, X. Li, L. Gu, C. Wu, and J. Zhang, "Discovery and analysis about the evolution of service composition patterns," *Journal of Web Engineering*, vol. 18, no. 7, pp. 579–625, 2019.

[23] A. S. Nielsen, *Scaling and resilience in numerical algorithms for exascale computing.* PhD thesis, EPFL, 2018.

[24] M. Dakshayini and H. S. Guruprasad, "An optimal model for priority based service scheduling policy for cloud computing environment," *International Journal of Computer Applications*, vol. 32, no. 9, pp. 23–29, 2011.

[25] K. Mishra and S. Majhi, "A state-of-art on cloud load balancing algorithms," *International Journal of Computing and Digital Systems*, vol. 9, no. 2, pp. 201–220, 2020.

[26] M. H. Moghadam and S. M. Babamir, "Makespan reduction for dynamic workloads in cluster-based data grids using reinforcement-learning based scheduling," *Journal of Computational Science*, vol. 24, pp. 402–412, 2018.

[27] S. Selvarani and G. S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," in *2010 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–5, IEEE, 2010.

[28] L. M. Mustafa, M. K. Elmahy, and M. H. Haggag, "Improve scheduling task based task grouping in cloud computing system," *International Journal of Computer Applications*, vol. 93, no. 8, 2014.

[29] I. A. Moschakis and H. D. Karatza, "Evaluation of gang scheduling performance and cost in a cloud computing system," *The Journal of Supercomputing*, vol. 59, pp. 975–992, 2012.

[30] D. Li, P. H. Chou, and N. Bagherzadeh, "Mode selection and mode-dependency modeling for power-aware embedded systems," in *Proceedings of ASP-DAC/VLSI Design 2002. 7th Asia and South Pacific Design Automation Conference and 15th International Conference on VLSI Design*, pp. 697–704, IEEE, 2002.

[31] D. G. Amalarethinam and F. K. M. Selvi, "The best-effort based workflow scheduling in grid computing-an overview," *International Journal of Research and Reviews in Computer Science*, vol. 2, no. 1, p. 48, 2011.

[32] K. Sutha and G. K. Nawaz, "Research perspective of job scheduling in cloud computing," in *2016 Eighth International Conference on Advanced Computing (ICoAC)*, pp. 61–66, IEEE, 2017.