

Paper Type: Original Article



Efficient Task Scheduling for Cloud Computing: a Comparative Survey of State-of-the-Art Algorithms

Aziza Algarni*

Department of Preparatory Year–Basic Sciences, Umm Al-Qura University, Makkah, Saudi Arabia; amalgarni@uqu.edu.sa.

Citation:



Algarni, A. (2023). Efficient task scheduling for cloud computing: a comparative survey of state-of-the-art algorithms. *Big data and computing visions*, 3(2), 50-54.

Received: 03/01/2023

Reviewed: 05/02/2023

Revised: 15/04/2023

Accepted: 12/05/2023

Abstract

Cloud computing is an essential tool for sharing resources across virtual machines, and it relies on scheduling and load balancing to ensure that tasks are assigned to the most appropriate resources. Multiple independent tasks need to be handled by cloud computing, and static and dynamic scheduling plays a crucial role in allocating tasks to the right resources. This is especially important in heterogeneous environments, where algorithms can improve load balancing and enhance cloud computing's efficiency. This paper aims to evaluate and discuss algorithms that can improve load balancing in cloud systems.


Keywords: Cloud computing, Algorithms, Computer science, Software engineering.

1 | Introduction

The growth of cloud computing has been significant since its inception in 2007. It provides users a wide range of services through the Internet, including large infrastructure, storage, virtualization, and resource pooling [1]. The increasing demand for cloud computing services by Internet users has presented various technical challenges, such as high availability, fault tolerance, scalability, and server consideration, that must be addressed [2]. One of the critical challenges affecting cloud computing is load balancing. Load balancing ensures that workloads are evenly distributed among available nodes in a distributed system to prevent individual devices from being overloaded and thus affecting overall performance [3]. Load balancing in cloud systems is crucial to maintain quality service and helps to solve performance, economy, and availability problems.

1.2 | Cloud Computing

Cloud computing is a model that is widely used in parallel or distributed systems, where a shared pool of computing resources is configured together [4]. It provides on-demand network access and there is very minimal or no interaction from a service provider. This model allows users to access services over the Internet via a reliable data center [5]. A client can access the cloud to manage their information, while a data center stores different types of applications on several servers.

 Licensee **Big Data and Computing Visions**. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

Different types of cloud computing are categorized according to capability and accessibility. In the case of accessibility cloud computing can be categorized into private, public, and hybrid clouds [6]. Whereas depending on what services cloud computing has to offer it can be divided into the following:

- I. Software as a Service (SaaS): enables users to use and access the cloud provider's applications running on the provider's infrastructure from a thin client or program interface. For example, google documents and web-based e-mail [7].
- II. Platform as a Service (PaaS): offers users the capability to build or deploy applications using tools (i.e., programming languages, libraries, services) without maintaining the underlying infrastructure. Users control the applications themselves. For example, Salesforce, Heroku, and Google App Engine (GAE) [8].
- III. Infrastructure as a Service (IaaS): provides the users with the lowest levels of network infrastructure, including networking, storage servers, hardware, and virtualized computing resources [9].

1.3 | Load Balancing

Load balancing is a critical component in maintaining efficient operations in cloud computing. It helps to prevent system overload and ensures optimal resource utilization, thus minimizing response time and maximizing throughput [10]. Load balancing algorithms are introduced to avoid overloading and idleness of nodes in a cloud system. These algorithms ensure that all nodes are assigned equal amounts of workload, allowing for continuous services to users without any interruption [11]. With the increase in cloud computing platforms such as Windows Azure Platform, Amazon S3, etc., and usage in Artificial Intelligence (AI), load-balancing algorithms will become more essential.

Static load balancing algorithms work in a static environment, while dynamic load balancing algorithms perform efficiently in a dynamic environment [12]. In dynamic environments, the state of the system greatly influences the performance of load-balancing algorithms. Overall, load balancing plays a crucial role in ensuring that cloud computing systems operate effectively and efficiently [13].

2 | Load Balancing Algorithm in Cloud Computing

2.1 | Round Robin Algorithm

The Round Robin algorithm is a simple and effective way of load balancing in cloud computing. It utilizes a time-triggered scheduling policy and assigns jobs to devices using a round-robin method [14]. This algorithm randomly selects nodes for load balancing and relies on data centers as its main components. When an Internet user sends a request to the cloud system, the data center controller receives the request and passes it to the Round Robin algorithm. This algorithm is based on time-sharing and divides time into slices and quanta [15].

Initially, in the load-balancing process, all processors are stored in a circular queue. The scheduler assigns the server to the processes based on the predefined time slot. Whenever a new process arrives, it is added to the end of the queue [16]. The Round Robin algorithm randomly selects the first process from the queue, and when the process's time slot is over, it moves the process to the end of the queue [17]. If a process finishes before its defined time slot, the algorithm voluntarily releases it. As each process has a different loading time, some nodes may become overloaded while others are underutilized, which can decrease the load-balancing performance. To address this issue, a weighted Round Robin load balancing algorithm was introduced to provide a more efficient allocation technique [18].

The weighted Round Robin load balancing algorithm distributes jobs based on their prescribed weight values. This means that the algorithm assigns processors with higher weight values more tasks since they have greater processing ability [19]. As a result, servers with higher weight values handle more tasks, and as the weights come into balance, traffic to the servers becomes more consistent. The weighted Round

Robin algorithm ensures that workload distribution is proportional to the processing ability of each server, resulting in a more efficient load balancing process [20].



2.2 | Opportunistic Algorithm

This algorithm is static in nature which means that it will not take into consideration the current state or workload of each system. It distributes all uncompleted tasks randomly to each node without considering how long it will take to complete that task [21]. As a result, this algorithm may perform poorly in terms of load balancing as it fails to calculate the implementation time for each node. This can cause bottlenecks in the cloud system, especially if there are idle nodes that are not being utilized efficiently.

2.3 | Min-Min Algorithm

The Min-Min algorithm is a method of scheduling tasks based on their minimum completion time. It is a fast and simple algorithm that improves system performance. The process begins by calculating the minimum completion time for all the available tasks. The task with the smallest completion time is selected and scheduled on the machine accordingly. After updating the machine's current execution time with the task's completion time, it is removed from the available task set. This process continues until all the tasks in the set have been allocated to their respective machines [8].

2.4 | Max-Min Algorithm

This algorithm starts by first searching for the task with the minimum implementation time among all available tasks, and then it calculates the maximum value. Once identified, the algorithm selects a task with a high completion time and assigns it to the respective machine. After assigning the task, the algorithm updates the execution time of all remaining tasks, and upon completion, the task is removed from the list. This algorithm differs from the min-min algorithm as, here, only one long task runs in parallel with many shorter tasks [9].

2.5 | Active Monitoring Algorithm

This algorithm automatically assigns workloads to the least busy or idle virtual machines. The servers and requests are maintained in the server's index table by the controllers. This allows it to quickly identify which servers have available resources or we can say those servers which are least busy or idle. When a new request comes in, the algorithm assigns the task to the first available server. Here a first-come-first-serve approach is used in assigning the tasks. The server's state is then updated in the index table that reflects the added workload. As tasks are completed, the controllers are notified and the server's state is reduced in the index table. When a user sends a request, the load balancer checks the index table again and allocates the task to the most appropriate server.

2.6 | Equally Spread Current Execution Algorithm

This method of load balancing equally divides the workload among servers in a data center. It assigns priority to all queue processes, evaluating their capacity and size. The algorithm selects the server that can complete the task in the shortest time based on the estimated load and virtual machine capacity. The workload is then assigned to a server that can handle the job's size and capacity. This ensures that the workload is balanced across all servers for optimal performance. The above algorithms are compared by using the various measured parameters and are shown below.

Table 1. LB algorithms comparison chart.

Load Balancing Algorithms/ Performance Parameters	Throughput	Overhead	Fault-Tolerance	Response Time	Resource Utilization	Scalability	Performance
Round robin	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Opportunistic	No	No	No	Yes	No	No	No
Min-min	Yes	Yes	No	Yes	Yes	No	Yes
Max-min	Yes	Yes	No	Yes	Yes	No	Yes
Active monitoring	Yes	Yes	No	Yes	Yes	Yes	No

3 | Conclusion

Load balancing algorithms are essential in providing fast connectivity and improving the performance, scalability, fault-tolerance, throughput, and overhead of cloud systems. Different load-balancing algorithms can be applied to different types of tasks, and the Round and Robin (weighted Round Robin) algorithm was found to be the most suitable for both heterogeneous and homogeneous tasks. However, load balancing remains a challenging task in cloud computing, and further fine-tuning of algorithms is necessary to achieve better and more consistent results from different perspectives. This paper extensively evaluated the concepts of cloud computing, different types of cloud computing, and various load-balancing algorithms, focusing on the overall completion time of processes in the queue.

References

- [1] Bal, P. K., Mohapatra, S. K., Das, T. K., Srinivasan, K., & Hu, Y. C. (2022). A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques. *Sensors*, 22(3), 1242. DOI:https://doi.org/10.3390/s22031242
- [2] 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. DOI:https://doi.org/10.1016/j.comcom.2022.01.016
- [3] Imene, L., Sihem, S., Okba, K., & Mohamed, B. (2022). A third generation genetic algorithm NSGAIII for task scheduling in cloud computing. *Journal of King Saud university-computer and information sciences*, 34(9), 7515–7529. DOI:https://doi.org/10.1016/j.jksuci.2022.03.017
- [4] Khan, M. S. A., & Santhosh, R. (2022). Task scheduling in cloud computing using hybrid optimization algorithm. *Soft computing*, 26(23), 13069–13079.
- [5] Gupta, S., Iyer, S., Agarwal, G., Manoharan, P., Algarni, A. D., Aldehim, G., & Raahemifar, K. (2022). Efficient prioritization and processor selection schemes for heft algorithm: a makespan optimizer for task scheduling in cloud environment. *Electronics*, 11(16), 2557. DOI:https://doi.org/10.3390/electronics11162557
- [6] Siddesha, K., Jayaramaiah, G. V., & Singh, C. (2022). A novel deep reinforcement learning scheme for task scheduling in cloud computing. *Cluster computing*, 25(6), 4171–4188. DOI:https://doi.org/10.1007/s10586-022-03630-2
- [7] Algarni, A. (2022). A study on deep learning based parking lot allotment to the vehicles. *Computational algorithms and numerical dimensions*, 1(1), 46–51. DOI:10.22105/cand.2022.159983
- [8] Mohapatra, H., & Rath, A. K. (2021). An iot based efficient multi-objective real-time smart parking system. *International journal of sensor networks*, 37(4), 219–232. DOI:10.1504/IJSNET.2021.119483
- [9] Mohapatra, H., & Rath, A. K. (2019). Fault tolerance through energy balanced cluster formation (ebcf) in wsn. In *Smart innovations in communication and computational sciences* (pp. 313–321). Singapore: Springer Singapore.

- [10] Bezdán, T., Zivković, M., Bacanin, N., Strumberger, I., Tuba, E., & Tuba, M. (2022). Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. *Journal of intelligent and fuzzy systems*, 42(1), 411–423. DOI:10.3233/JIFS-219200
- [11] Mangalampalli, S., Swain, S. K., & Mangalampalli, V. K. (2022). Multi objective task scheduling in cloud computing using cat swarm optimization algorithm. *Arabian journal for science and engineering*, 47(2), 1821–1830.
- [12] Abdullahi, M., Ngadi, M. A., Dishing, S. I., & Abdulhamid, S. M. (2023). An adaptive symbiotic organisms search for constrained task scheduling in cloud computing. *Journal of ambient intelligence and humanized computing*, 14(7), 8839–8850.
- [13] Prity, F. S., Gazi, M. H., & Uddin, K. M. (2023). A review of task scheduling in cloud computing based on nature-inspired optimization algorithm. *Cluster computing*, 1–31.
<https://link.springer.com/article/10.1007/s10586-023-04090-y>
- [14] Gad, A. G., Houssein, E. H., Zhou, M., Suganthan, P. N., & Wazery, Y. M. (2023). Damping-assisted evolutionary swarm intelligence for industrial iot task scheduling in cloud computing. *IEEE Internet of things journal*, 1. <https://ieeexplore.ieee.org/abstract/document/10171225/>
- [15] 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.
DOI:10.1109/ACCESS.2022.3163273
- [16] Iftikhar, S., Ahmad, M. M. M., Tuli, S., Chowdhury, D., Xu, M., Gill, S. S., & Uhlig, S. (2023). HunterPlus: AI based energy-efficient task scheduling for cloud–fog computing environments. *Internet of things*, 21, 100667. DOI:<https://doi.org/10.1016/j.iot.2022.100667>
- [17] Zhou, Z. (2023). Soil quality based agricultural activity through iot and wireless sensor network. *Big data and computing visions*, 3(1), 26–31. DOI:10.22105/bdcv.2022.332447.1056
- [18] Nabi, S., Ahmad, M., Ibrahim, M., & Hamam, H. (2022). AdPSO: adaptive pso-based task scheduling approach for cloud computing. *Sensors*, 22(3), 920. <https://www.mdpi.com/1424-8220/22/3/920>
- [19] Abualigah, L., & Alkhrabsheh, M. (2022). Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing. *The journal of supercomputing*, 78(1), 740–765.
- [20] 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. <https://doi.org/10.3390/en15134571>
- [21] Ghafari, R., Kabutarkhani, F. H., & Mansouri, N. (2022). Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review. *Cluster computing*, 25(2), 1035–1093.