Research Reports on Computer Science
https://ojs.wiserpub.com/index.php/RRCS/

Research Article in Special Issue: Selected Papers from the 4th International Conference on Machine Learning, Image Processing, Network Security and Data Sciences (MIND-2022)

Cloud-based Cost Effective IIoT Model Towards Industry 5.0

Satyajeet Rajaram Shinge*, Urmila N. Shrawankar

Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur (MH), India
E-mail: satyajeet89@gmail.com

Received: 8 March 2023; **Accepted:** 28 March 2023

Abstract: Companies across industries increasingly depend upon cloud computing to manage their Industrial Internet of Things (IIoT) technology. Machines are connected over a network in the IIoT. Cloud computing plays an essential role by connecting people, devices, work processes, and buildings to deliver cloud services in industries. But cloud computing faces a problem with task scheduling, high latency delay, and memory management, affecting the overall cost of industries using cloud services. A major concern in the cloud computing field is task scheduling which is essential for achieving cost-effective execution and improving resource usage. It refers to assigning available resources to user tasks. This problem can be solved effectively by improving task execution and increasing the use of resources. The waiting time between a client's sent request and a cloud service provider to give a response, known as latency, is another issue in cloud environments. In cloud computing, this delay can be significantly higher. As a result, users of various cloud services may incur increased expenses due to this delay. Finally, among the most significant topics in cloud computing is efficient memory management, which handles integrated data and optimizes memory management algorithms. This paper proposes a cloud model for IIoT, which provides task scheduling, helps reduce latency, and optimizes memory management. This proposed model helps to reduce the cost of using cloud computing in IIoT.

Keywords: task scheduling, cloud model, cost-effective, IIoT, Industry 5.0, effective resource sharing, energy optimization

1. Introduction

Cloud computing is an internet-based computing service that provides services, like infrastructure, data storage, and applications remotely. Various computing resources, like storage, processing power, databases, networking, analytics, artificial intelligence, and software applications, are available through cloud computing. Companies do not need to purchase need and maintain a physical, on-site Information technology (IT) infrastructure on their premises because they may outsource these resources and use computing assets whenever required. Customers can access new or scale existing resources as required by paying charges based on the number of resources used. Users pay for their services the same way as home utilities. Cloud computing resources are divided dynamically and assigned on demand. As illustrated in Figure 1, a cloud service provider offers computing resources to users. These computing resources are available on demand and are divided and assigned dynamically.

DOI: https://doi.org/10.37256/rrcs.2320232632

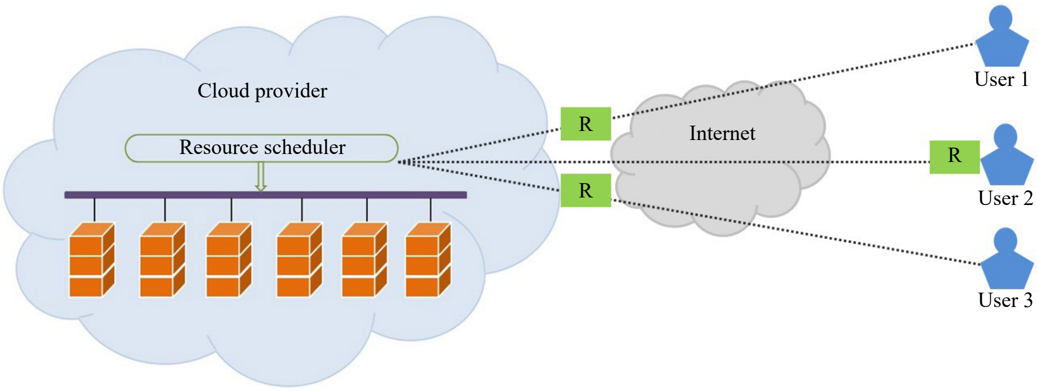

Figure 1. Resource allocation in cloud computing

The Industrial Internet of Things (IIoT) refers to networked sensors, instruments, and other resources used in manufacturing and energy management that are integrated with industrial computer applications (Figure 2). This connectedness makes the ability to collect, trade, and analyze data possible, which may also positively affect the economy. The IIoT is an expansion of a distributed computing system that enhances and optimizes process controls using cloud computing and allows a higher level of automation.

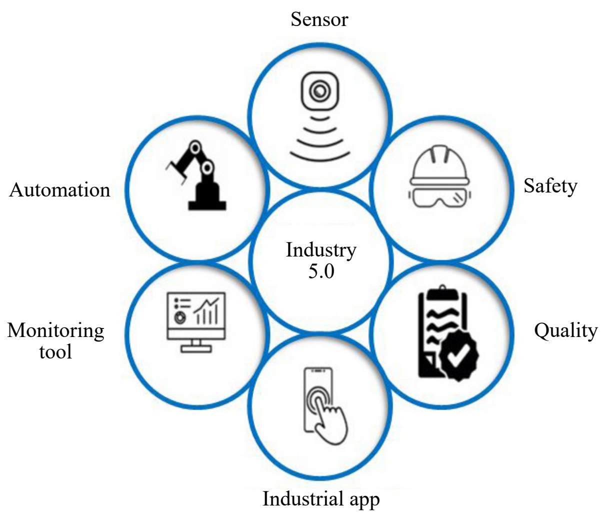

Figure 2. IIoT model

Due to the energy shortage and change in the global climate, power consumption is a hot issue in every field where power is used. In cloud computing, thousands of resources are working simultaneously, but some of them are used by clients or users for their requests, and many are just idle. Due to this, cloud computing can have over-utilized and under-utilized resources, which are called resource imbalances. To minimize the power consumption of these idle or under-utilized resources, we should try to balance the load by scheduling the task to these under-utilized resources.

Figure 3 shows the overall development in cloud computing from 2016-17 to till date. Techniques like mobile cloudlets, dynamic programming-based offline micro service coordination, deep reinforcement learning-based resource allocation (DRLRA), delay-dependent priority-aware task offloading, and auction-based virtual machine (VM) resource allocation are used in cloud computing to solve the problem of resource allocation, high latency, and memory management. Although load balancing, cloud computing has a problem with memory management and increased latency. Due to this problem, the cost of cloud services is rising. We can overcome this problem by developing a cost-effective cloud model for IIoT. This model can schedule the cloud resources and reduce resource imbalance. This model can also optimize memory management and reduce cloud computing latency. And it will have an immediate impact on the cost of cloud service.

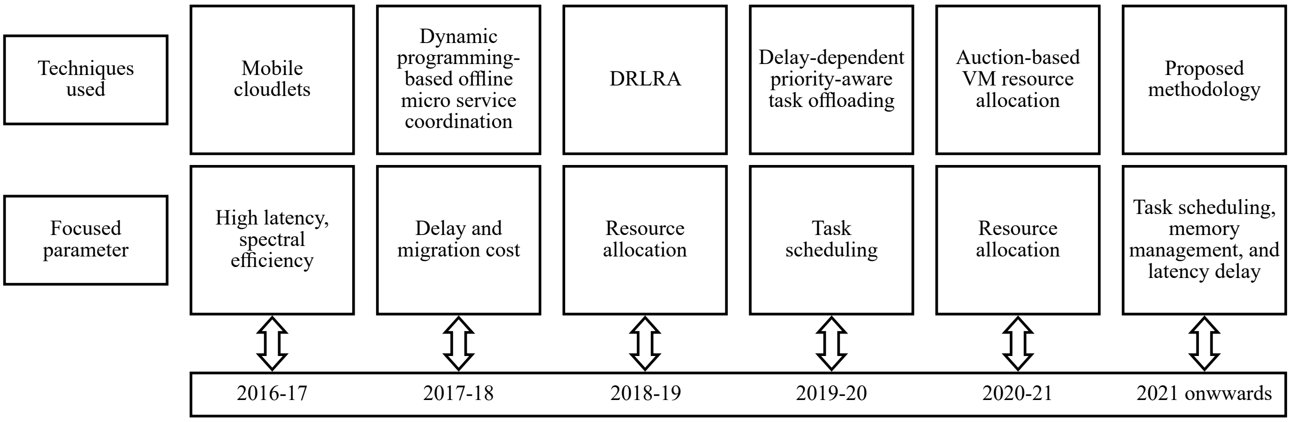

Figure 3. Overall development in cloud computing

2. Related works

Table 1 shows the techniques used, major findings, and limitations of various research related to cost-effective cloud computing. Table 2 evaluates the technique used in IIoT.

Table 1. Summary of related works for cost-effective cloud computing

Technique used	Findings	Limitations
Optimal resource provisioning (ORP) [1-3]	In these publications, the Cloud Assisted Mobile Edge (CAME) system is used to increase the applicability of Multi-access Edge Computing (MEC) while handling requests from mobile devices with varying time stamps. Some cloud instances have resolved source provisioning issues despite particular quality of service (QoS) requirements for mobile queries. When comparing the proposed ORP algorithms to minimize machine cost while meeting QoS requirements, the Optimal Resource Provisioning with Hybrid Strategy (ORP-HS) method has repeatedly shown the most excellent performance.	It deals with the problems of resource allocation and workload planning when investigating multiplexing gain.
Delay-dependent priority-aware task offloading [4, 5]	In hierarchical fog-based cloud architecture, this paper examines the task of degenerating strategy with a multi-level feedback queuing model. This research [4] aims to meet deadlines while consuming less processing time overall. The results demonstrate that, based on a prioritized scheduling strategy, both the minimal queuing waiting time and the offloading time positively benefit meeting the mission deadline.	High-reliability constraints.
Dynamic programming-based offline micro service coordination [6, 7]	This paper [6] selects the best edge clouds to run microservices when a mobile user moves. An offline approach for determining the ideal alignment of microservices is introduced when possible system knowledge is available, adding to the overall cost of migration that exceeds the time restriction. When possible system knowledge is available, and an offline technique for determining the best alignment of microservices is added. The overall cost of exceeding the time limit migration.	Load balancing between microservices in the multiuser mobile edge computing approaches.
Energy optimization [8-10]	Many cloud enthusiasts and vendors are curious about how the design might be more effective in the case of energy conservation by increasing the cloud-related infrastructures, starting from entirely centralized to completely separate. The topic was covered in this essay. They utilize their communications network and reliable computing infrastructure and use multi-stranded technology to produce products that satisfy consumer needs.	The model cannot be used for users who are dispersed differently. Do not support other technologies, architectures, or virtual network functions (heterogeneous telecommunication networks, mobile networks).
DRLRA [11, 12]	In a mutative MEC setting, this study looks into the problem of resource allocation. We considered this issue in two different approaches. First, several tests were run to establish the effectiveness of DRLRA. The experimental results showed that our proposed DRLRA achieved much greater efficiency than the default Open Shortest Path First (OSPF) algorithm under various circumstances.	Too much reinforcement can lead to an overload of states which can diminish the results.

Table 1. *Continued*

Technique used	Findings	Limitations
Auction-based VM resource allocation [13, 14]	In this article [13], the problem of assigning varied VM resource requirements to globally dispersed edge cloud nodes with bandwidth constraints to maximize total social welfare is addressed. VM resources require globally distributed edge cloud nodes with bandwidth restrictions to maximize overall social welfare.	Dynamic pricing system, termination time with capital cost can be examined by including further QoS.
Edge and fog computing [15, 16]	Through this context, our work clarifies certain principles as an output in what can be called the very first doc to be read for those beginning research in computing fog and edge. In addition, a summary of the open challenges and possible directions for research in Internet of Things (IoT), cloud, fog, and edge computing was also presented as fuel for thinking.	Although heterogeneity is a problem for both the fog and the edge, it is expected that the capabilities of fog devices will be more consistent than those of edge devices.
Mobile cloudlets [17, 18]	This paper [17] thoroughly introduces the classic state-of-the-art technologies, such as mobile cloudlets, edge computing cloudlets, etc., that comprise cloud computing. The differences between fog computing and mobile edge computing are illustrated from the perspective of radio access networks and fog-based computing radio access networks.	The scheduling and calculating of deployments introduce entirely new difficulties. Now, various remote execution locations must be considered, either inside the cloudlet or in different cloudlets, as opposed to a single identified surrogate.

Table 2. Evaluation in IIoT

Year	The technique used in IIoT	University/Center/Organization
1968	Dick Morley's programmable logic controller (PLC), created in 1968, was employed by the General Motors division that made automatic transmissions.	General Motors
1975	Two of the earliest Distributed Control Systems (DCSs) ever presented, the TDC 2000 and the CENTUM system, were created by Honeywell and Yokogawa.	Honeywell and Yokogawa
1983	After ethernet was developed, people started looking at the idea of a network of intelligent devices. For instance, the first internet-connected appliance was a modified Coke machine at Carnegie Mellon University that could report its inventory if freshly filled drinks were cold.	Carnegie Mellon University
1994	More extraordinary industrial applications were predicted. In an article for IEEE Spectrum, Reza Raji defined the idea as integrating and automating anything from home appliances to factories by sending little data packets to many nodes.	Menlo Park, California
1999	The Auto-ID Center at Massachusetts Institute of Technology (MIT) and associated market analysis publications helped the IoT concept gain popularity in 1999.	MIT
2002	The introduction of Amazon Web Service (AWS) solidifies the use of cloud technology.	Amazon
2006	A secure remote communication protocol called OPC Unified Architecture (UA) is made available to provide secure communication between devices, data sources, and applications.	Microsoft
2011	IT standards such as Message Queuing Telemetry Transport (MQTT), REpresentational State Transfer (REST), Hypertext Transfer Protocol (HTTP), etc. enter into industrial automation.	Opto 22

3. Methodology for the cost-effective cloud model in IIoT

A series of researches are done in cloud computing on task scheduling, latency, and memory allocation to obtain cost-effective cloud computing. This paper proposes a cloud model for IIoT which combines a solution of task scheduling, latency, and memory allocation to minimize the computational cost of the industry.

Figure 4 shows the proposed system for an optimized cloud model to achieve better performance by adopting the task scheduling algorithm, memory management techniques, and cloud-edge nodes to reduce latency. The adaptive model employs the data skew technique on IoT application data and employs the cloud model to allocate the resource by migrating the resources based on the load. The proposed method organizes the optimization technique to achieve load balancing; this proposed cloud model employs an optimization technique to minimize the load and improves the computation efficiency.

Various software and hardware components comprise a computational cloud environment, including computing nodes, storage systems, databases, network resources, file systems, etc. It is made up of four main parts: the cloud user, the cloud resources, the cloud resource broker (CRB), and the cloud information service (CIS). When using the cloud, a user first communicates with a resource broker to transmit a task for computation. Following that, resource finding, scheduling techniques, and task processing are carried out. Finally, the CIS was employed as an agent. It gathers all pertinent data, like resource availability, node capacity, etc., and gives it to the resource breaker so that they may decide when to schedule things. The interactions between various clouds components are done using multiple steps which are as follows:

a. The cloud user runs their application and after analysis and specifying their requirement, submits their jobs to CRB.
b. The CRB collects all resource information and performs resource discovery.
c. After authorizing the user and resource(s), the CRB schedules the job to the appropriate resource(s) or computing nodes.
d. Resource(s) execute the job and return the computational result to CRB.
e. The CRB collects the result and provides it to the cloud user.

With reference to the above-proposed model, this paper identifies different task scheduling algorithms used to perform task scheduling and compare their result to find the algorithm which gives minimum response time as compared to other algorithms.

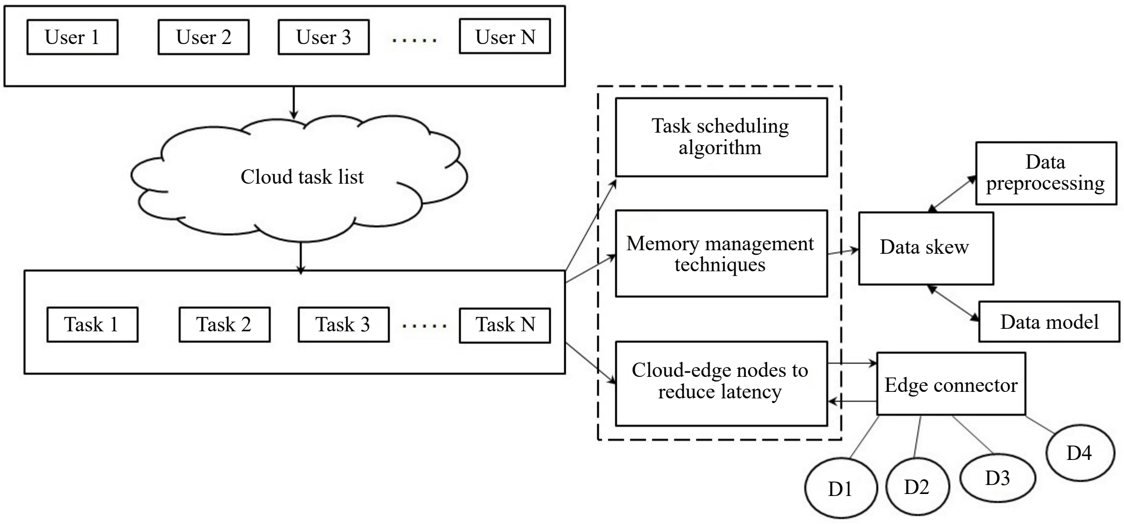

Figure 4. Block diagram of the cost-effective cloud model for IIoT

4. Implementation details for IIoT

4.1 *Task scheduling algorithms in cloud computing*

Algorithms for task scheduling are used to utilize resources effectively. Task scheduling variables include waiting, response, processing, and makespan. An efficient scheduler must control these factors to ensure that the scheduling policy is as effective as possible. The complexity increases in a dynamic environment. Real-time tasks with deadlines must arrange in advance according to priority.

4.2 *Task scheduling architecture*

The task scheduling process in cloud computing is depicted in Figure 5 [19]. The data center broker, who serves as an intermediary between the user and the data center, determines how the requested users submit their tasks to the data center. The service level agreement (SLA), which is maintained between the user and the data center broker, is specified by the user when the tasks are placed in the data center. The tasks are then compiled into a task pool and transmitted to the data center as a queue. There are several hosts in a typical data center, each with one or more processor cores. In the cloud concept, the host is mapped to one or more VMs, where the work is actually completed utilizing the recommended policy techniques. The data center broker supervises the task scheduling procedure.

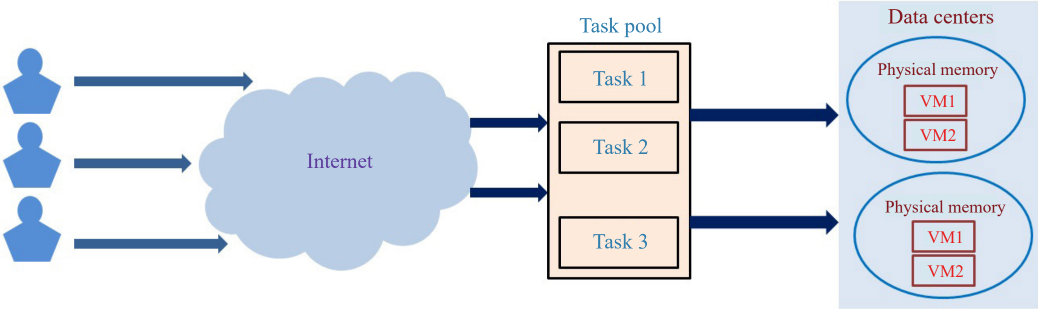

Figure 5. Task scheduling architecture in cloud computing

4.3 *Classification of task scheduling algorithms*

The various task scheduling algorithms used in cloud computing are depicted in Figure 6 [19]. In cloud computing, different task scheduling methods, including Round Robin (RR), First Come First Serve (FCFS), Max-Min, Min-Min, Particle Swam Optimization (PSO), Ant Colony Optimization (ACO), and Intelligent Water Drop Cloud Scheduling (IWDC) are used. The specifics of each algorithm are shown in Table 3.

Table 3. Task scheduling algorithms in cloud

Scheduling algorithm	Description	Advantages	Disadvantages
FCFS [19]	According to arrival time, jobs are ranked in the queue and distributed to resources in a first come first serve fashion. When there is no deadline restriction, this approach is used.	This method can be used when a deadline will not be an important constraint during the execution of the task.	Not preferable for complex systems.
RR [19]	A time slot is available for each resource and the task is allocated to each resource during an allocated time slot.	Each task will get a fair allocation of resources.	It gives poor makespan value.
Min-Min [20]	This algorithm selects the smallest task from the available tasks and assigns it to a resource with less execution time.	It gives good makespan value when most tasks have a small execution time.	For the competition of all little jobs, the longest task must wait.
Max-Min [20]	Max-Min chooses the longest task from a pool of tasks that have the potential to take the longest amount of time to complete and assigns it to the resources that will require the shortest amount of time to complete.	This algorithm will be useful when we have to give priority to the execution of large tasks first.	The smallest task must wait a long time for all other long tasks to be finished, starving the smaller tasks.
PSO [19]	PSO is a meta-heuristic algorithm that considers both computation task and data transmission time. It is used for workflow applications by varying its computation and communication costs.	Simple and effective algorithm with low computational cost.	This algorithm is easy to fall into the local optimum in high-dimensional space and it has a low convergence rate in the iterative process.
ACO [19]	The ant foraging behavior is replicated in ACO. When ants are looking for food, they leave pheromones along the way so that other ants can easily follow their trail or locate the quickest path. This approach works well for finding a solution in computations with local variables.	It increases task completion time.	Not considering load balancing.
IWDC [21]	The three steps of the IWDC algorithm are: initializing path parameters, giving them values, and allocating tasks following path creation.	Comparing this algorithm to other heuristic algorithms, it performs and uses resources well. Additionally, it offers superior outcomes compared to various meta-heuristic algorithms like PSO, ACO, and genetic algorithms.	Energy consumption of resources is not considered.

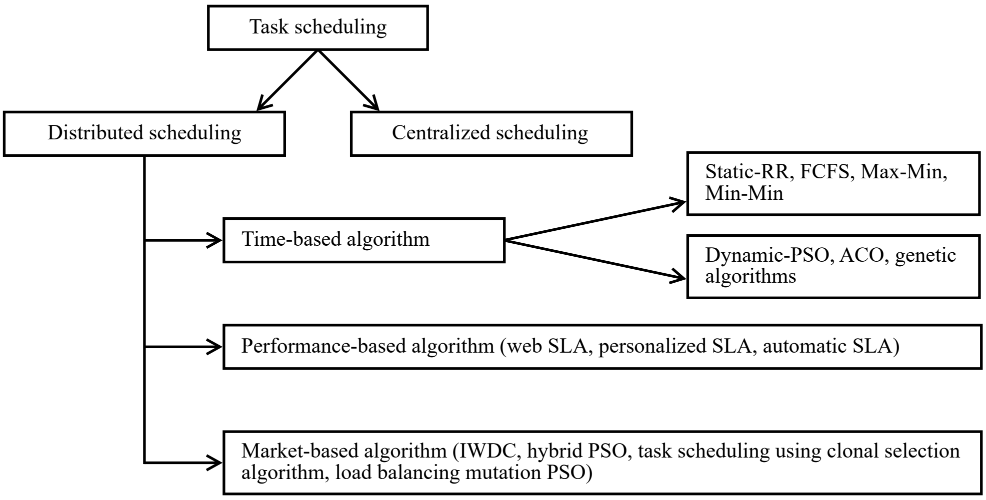

Figure 6. Classification of task scheduling algorithms

5. Results and discussion

Table 4 gives the comparative analysis of all task scheduling algorithms. These algorithms are compared based on makespan value, resource utilization, response time, energy consumption, and throughput. Effective task scheduling algorithms should have less makespan, balanced resource utilization, less response time, low energy consumption, and more throughput value.

The comparison shows that the IWDC algorithm gives a better makespan value as compared to other algorithms. It also gives good resource utilization, minimum response time, and large throughput. So, this algorithm can be used in the presented model to develop a cost-effective cloud model.

6. Conclusion

This paper presents a cost-effective cloud model for IIoT. This proposed model can help to minimize industrial computational cost by providing efficient task management, reducing latency delay, and optimizing memory allocation This paper identifies various work scheduling techniques applied in cloud computing and compares them all. Results and discussion show that each algorithm has distinct advantages for parameters like makespan, response time, resource utilization, energy consumption, and throughput. But the IWDC algorithm gives good results for each parameter. If the IWDC algorithm is used for task scheduling, then it helps to reduce the cost of computing in IIoT. If the cost of computation in industries is reduced, it benefits society by reducing the cost of products or services. Society can get services and products from industry at minimum cost.

This paper also presents the evaluation done in Industries till the current year. Manufacturers are now including new technologies into their manufacturing facilities and operations, like the IoT, robotics, artificial intelligence, cloud computing, and machine learning to enable fast and efficient production of goods and services. The presented model will help towards Industry 5.0.

Table 4. Comparison of task scheduling algorithms used in cloud computing

Parameters / Algorithm	FCFS	RR	Min-Min	Max-Min	PSO	IWDC
Makespan	Quick tasks at the back of the queue must wait until a big task is finished, giving more makespan.	Due to more context switches, it gives more makespan value.	It gives good makespan value.	Results in better makespan.	It gives the minimum makespan compared to all algorithms.	IWDC has a better makespan than PSO.
Resource utilization	FCFS gives the worst load balancing.	Provides balanced utilization of resources.	In the Min-Min algorithm, the long tasks have to wait for smaller tasks to finish their execution. So, the workload can't be properly allocated and gives unbalanced utilization.	Larger tasks can complete on fast resources and small tasks can work parallel on other resources. So, it gives good resource utilization.	It offers the best solutions for allocating all the work among the pool of available resources.	It gives better utilization of resources and load balancing than PSO and other task scheduling algorithms.
Response time	The response time for smaller tasks at the back of the queue is more because of long tasks at the front.	Each task is allocated a quantum of time for execution; there is no need for the jobs to wait. So, it gives a good response time.	Response time for a task with minimum completion time is good. But, tasks having large execution time have to wait for allocation.	Reduces the waiting time for long tasks and gives minimum response time. But, for small tasks, it gives more response time.	When the search space is large, the algorithm speed is slow, resulting in a longer response time.	IWDC gives less response time as compared to other scheduling algorithms.
Energy consumption	FCFS has low energy consumption.	Due to more context switches, it consumes more energy.	Consumes more energy for heterogeneous tasks.	Consumes more energy for heterogeneous tasks.	PSO has good energy efficiency as compared to other algorithms.	IWDC algorithm consumes more energy.
Throughput	This algorithm provides fast execution of tasks.	It gives good throughput.	If the number of tasks having less execution time is more than throughput is high.	Throughput is low if tasks with more execution time are more in numbers.	The number of tasks executed is more.	Compared to other algorithms, this algorithm is faster. The result is excellent throughput.

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] Ma X, Wang S, Zhang S, Yang P, Lin C, Shen X. Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing. *IEEE Transactions on Cloud Computing*. 2019; 9(3): 968-980. https://doi.org/10.1109/TCC.2019.2903240

[2] Tong L, Li Y, Gao W. A hierarchical edge cloud architecture for mobile computing. In: *IEEE INFOCOM 2016- The 35th Annual IEEE International Conference on Computer Communications*. San Francisco, CA, USA: IEEE; 2016. p.1-9. https://doi.org/10.1109/INFOCOM.2016.7524340

[3] Zhao T, Zhou S, Guo X, Zhao Y, Niu Z. Pricing policy and computational resource provisioning for delay-aware mobile edge computing. In: *2016 IEEE/CIC International Conference on Communications in China (ICCC)*. Chengdu, China: IEEE; 2016. p.1-6. https://doi.org/10.1109/ICCChina.2016.7636891

[4] Adhikari M, Mukherjee M, Srirama SN. DPTO: A deadline and priority-aware task offloading in fog computing framework leveraging multilevel feedback queueing. *IEEE Internet of Things Journal*. 2019; 7(7): 5773-5782. https://doi.org/10.1109/JIOT.2019.2946426

[5] Zhang G, Shen F, Liu Z, Yang Y, Wang K, Zhou MT. FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks. *IEEE Internet of Things Journal.* 2018; 6(3): 4388-4400. https://doi.org/10.1109/JIOT.2018.2887229

[6] Wang S, Guo Y, Zhang N, Yang P, Zhou A, Shen X. Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach. *IEEE Transactions on Mobile Computing.* 2019; 20(3): 939-951. https://doi.org/10.1109/TMC.2019.2957804

[7] Guerrero C, Lera I, Juiz C. Resource optimization of container orchestration: a case study in multi-cloud microservices-based applications. *The Journal of Supercomputing*. 2018; 74(7): 2956-2983. https://doi.org/10.1007/s11227-018-2345-2

[8] Ahvar E, Orgerie AC, Lebre A. Estimating energy consumption of cloud, fog, and edge computing infrastructures. *IEEE Transactions on Sustainable Computing*. 2019; 7(2): 277-288. https://doi.org/10.1109/TSUSC.2019.2905900

[9] Wei X, Tang C, Fan J, Subramaniam S. Joint optimization of energy consumption and delay in cloud-to-thing continuum. *IEEE Internet of Things Journal*. 2019; 6(2): 2325-2337. https://doi.org/10.1109/JIOT.2019.2906287

[10] Babu GP, Tiwari AK. Energy efficient scheduling algorithm for cloud computing systems based on prediction model. *International Journal of Advanced Networking and Applications*. 2019; 10(5): 4013-4018. https://doi.org/10.35444/IJANA.2019.10055

[11] Wang J, Zhao L, Liu J, Kato N. Smart resource allocation for mobile edge computing: A deep reinforcement learning approach. *IEEE Transactions on Emerging Topics in Computing*. 2019; 9(3): 1529-1541. https://doi.org/10.1109/TETC.2019.2902661

[12] Wu CG, Li W, Wang L, Zomaya AY. Hybrid evolutionary scheduling for energy-efficient fog-enhanced Internet of Things. *IEEE Transactions on Cloud Computing*. 2018; 9(2): 641-653. https://doi.org/10.1109/TCC.2018.2889482

[13] Talpur A, Gurusamy M. DRLD-SP: A deep-reinforcement-learning-based dynamic service placement in edge-enabled Internet of Vehicles. *IEEE Internet of Things Journal*. 2021; 9(8): 6239-6251. https://doi.org/10.1109/JIOT.2021.3110913

[14] Gao G, Xiao M, Wu J, Huang H, Wang S, Chen G. Auction-based VM allocation for deadline-sensitive tasks in distributed edge cloud. *IEEE Transactions on Services Computing*. 2019; 14(6): 1702-1716. https://doi.org/10.1109/TSC.2019.2902549

[15] Kumar RA, Kartheeban K. Resource allocation using Dynamic Pricing Auction Mechanism for supporting emergency demands in Cloud Computing. *Journal of Parallel and Distributed Computing*. 2021; 158: 213-226. https://doi.org/10.1016/j.jpdc.2021.07.016

[16] De Donno M, Tange K, Dragoni N. Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog. *IEEE Access.* 2019; 7: 150936-150948. https://doi.org/10.1109/ACCESS.2019.2947652

[17] Li C, Xue Y, Wang J, Zhang W, Li T. Edge-oriented computing paradigms: A survey on architecture design and system management. *ACM Computing Surveys (CSUR)*. 2018; 51(2): 1-34. https://doi.org/10.1145/3154815

[18] Ai Y, Peng M, Zhang K. Edge computing technologies for Internet of Things: a primer. *Digital Communications*

and Networks. 2018; 4(2): 77-86. https://doi.org/10.1016/j.dcan.2017.07.001

[19] Mallik P, Nayak AK, Dalei RK. Comparative analysis of various task scheduling algorithms in cloud environment. In: *2021 19th OITS International Conference on Information Technology (OCIT)*. Bhubaneswar, India: IEEE; 2021. p.37-41. https://doi.org/10.1109/OCIT53463.2021.00019

[20] Bharot N, Shukla S. A review on task scheduling in cloud computing using parallel genetic algorithm. In: *2020 International Conference on Computing and Information Technology (ICCIT-1441)*. Tabuk, Saudi Arabia: IEEE; 2020. p.1-4. https://doi.org/10.1109/ICCIT-144147971.2020.9213822

[21] Dubey K, Sharma SC. An extended intelligent water drop approach for efficient VM allocation in secure cloud computing framework. *Journal of King Saud University-Computer and Information Sciences*. 2022; 34(7): 3948-3958. https://doi.org/10.1016/j.jksuci.2020.11.001