DOI: https://doi.org/10.52756/ijerr.2023.v30.011 *Int. J. Exp. Res. Rev.*, Vol. 30: 100-108 (2023)

Review Article | Peer Reviewed | Open Access

International Journal of Experimental Research and Review (IJERR)

ISSN: 2455-4855 (Online)
www.iaph.in

A systematic review of workflow scheduling techniques in a fog environment

Aakriti Jain[1] and Ranjit Rajak[2*]

[1,2]Department of Computer Science and Applications
Dr. Harisingh Gour Central University Sagar, Madhya Pradesh, India
E-mail/Orcid Id:
AJ, aakritijain1095@gmail.com, https://orcid.org/0009-0001-8529-2983; *RR,* ranjit.jnu@gmail.com, https://orcid.org/0000-0003-2746-3278

Article History:
Received: 27th Feb., 2023
Accepted: 28th Mar., 2023
Published: 30th Apr., 2023

Keywords:
Cloud computing, cloudsim, DAG, fog computing, metrics

Abstract: Various recent trends in computer science include machine learning, block chain technology, IoT, Cloud computing, etc. Fog computing is one of the research areas used everywhere in science or other fields. Due to it providing very fast service in the heterogeneous platform, more security, and low latency. Workflow scheduling is one of the current research areas in the fog computing platform. Workflow scheduling allocates the different jobs into the available fog server and cloud servers. In this paper, we have identified some methods based on workflow scheduling in a fog environment and compared these methods based on tools and performance metrics.

Introduction

Today is the era of technology, and many people rapidly connect with technology via the internet. Since the number of internet users is growing day by day, they generate voluminous data. This number of users may reach 1.2 trillion in 2030, and its annual economic effect is possibly $15 trillion (Chowdhury et al., 2019). These data are stored and processed on the cloud, which is so far from the end user. So there is a problem that arises as high latency, and network congestion, so it decreases the QoS services. A new model known as fog computing has emerged to address the issue. It was first introduced by Cisco in 2012 (Bonomi et al., 2012). According to the Cisco system, fog computing does not replace cloud computing but rather behaves as a mediator between cloud computing and sensor devices. It stretches cloud computing from the network's centre to its edges and intends to deliver networking, storage, and computing services to consumers (Dastjerdi et al., 2016).

Cloud computing refers to a replica that contains a resource pool and provides a fully virtualized environment. It is based on paying for a usage service which means the user only pays for the service which they needed only. National Institute for Standard and Technology (NIST) (Mell and Grance, 2011; Miyachi, 2018), according to its definition, cloud computing refers to a paradigm that makes its resources available to customers as needed and releases them when the customer is done using them. It offers two models, the service and deployment models (Diaby et al., 2017). The deployment model is focused on the end user's requirements for accessibility, infrastructure, and location. It is further categorized into sub-categories as Public Model (managed and used by CSP and end user respectively), Private Model (offers the infrastructure to a particular organization for their unique needs, and prohibited users are prevented from accessing this infrastructure), Community Model (offers the infrastructure that is used by several entities with related needs.), and Hybrid Model (it is a mixture of many cloud types). The service Model is focused on providing various services to end users. It offers services such as SaaS, PaaS, and IaaS. Figure 1 depicts the cloud computing architecture (Tsai et al., 2010). This architecture is separated into two ends: the front and back. These ends communicate with one another through the Internet. The user side end is known as the front end and the virtualized end where the computation is performed is known as the backend.

In cloud computing, all the user requests are computed at a geo-centralized data centre. These data centres contain huge servers, applications, and networking

***Corresponding Author:** ranjit.jnu@gmail.com

facilities but suffer from high latency, network congestion, and security issues. So, to overcome these drawbacks, perform modifications in cloud architecture by applying a middle layer between the cloud and the end user (Hu et al., 2017). This is shown in Figure 2 (Hu et al., 2017). This layer contains many fog nodes. Each fog node consists of its virtual machine which is closer to IoT devices and takes care of networking, computation, and storage facilities. These fog nodes are geographically distributed and aim to offer localized services to minimize the networking bandwidth and computational cost and extend QoS (Sarkar and Misra, 2016; Luan et al., 2016).

Fog Computing (Bonomi et al., 2012; Ashi et al., 2020) offers the capability of heterogeneity (fog node belongs to different form factors and these nodes are deployed in a different environment), low latency (quick response because fog node is closer to the source of data generator), multi-tendency (it refer as accessing the same resource by different cloud provider's customer), interoperability (which means running an application on the different cloud), Geographical distribution (since fog is geographically distributed so it provides better services). With all these capabilities, it faces some challenges as well, like synchronization, discovery, standardization, and management (Gill et al., 2022). Along with some capability and challenge, it also offers some advantages like saving the bandwidth of the network, minimizing response time due to geo-distributed fog nodes, and increasing security because the fog node is available nearer to the end user (Firdhous, 2014; Chakraborty, 2019).

Since fog computing is an advancement of previously available techniques (Huet al., 2017; Ghobaei-Arani et al., 2020), It mostly consists of storage, communication, and computing techniques to fulfill the user's requirements for resource management, privacy, fast response, etc. The storage technique is based on pre-caching and augmentation in the storage approach, which results in a fast response. The communication technique can be categorized into three ways of connections: (i) between the fog node and sensor device as a wireless connection (ii) between the fog node and another fog node as a wireless or wired connection (iii) between the fog node and the data center as a wireless or wired connection. Some wireless approaches are named WiFi, 3G, 4G, 5G, Zigbee, SDN, CDN, and NFV. With the use of these techniques, communication may be possible. SDN (software-defined network) offers network virtualization. NFV (Network Function Virtualization) offers quick deployment due to fully shareable resources. Computation techniques offer two approaches: latency control and offloading computing.

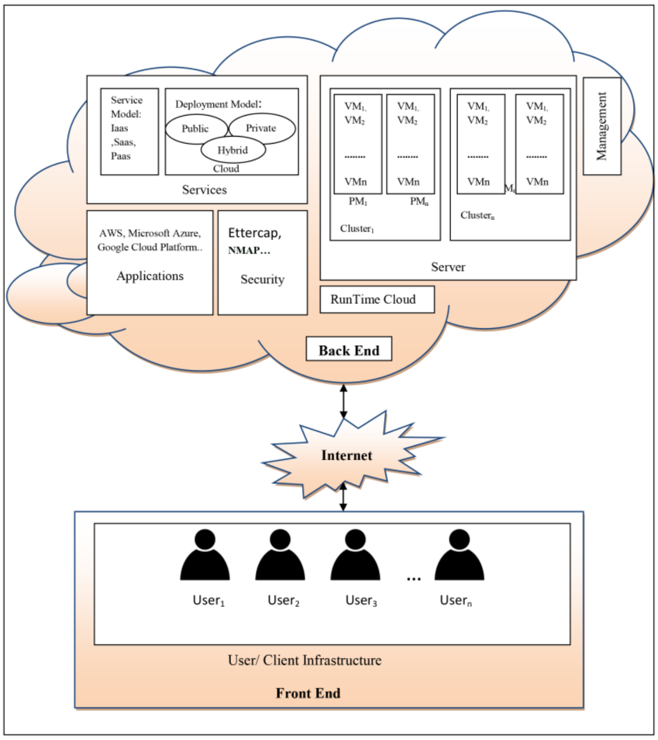

Figure 1. Cloud Computing Architecture: works into two ends that communicate through the internet (Tsai et al., 2010)

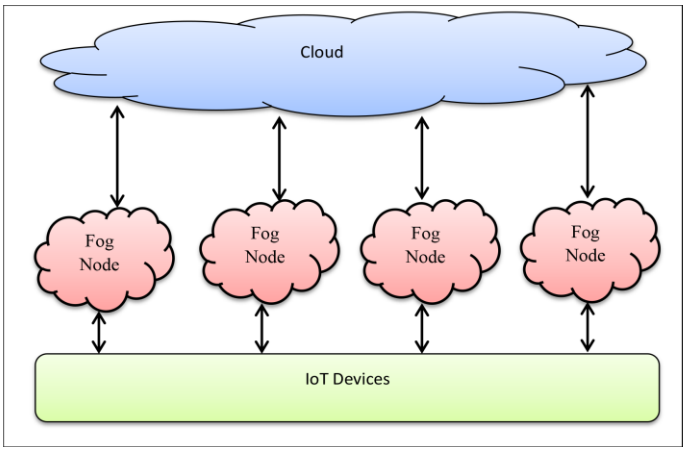

Figure 2. Fog Computing Model: Involves three layers where the lowest layer makes requests to the upper layers for computation, and after computation respond are given for the lowest layer (Hu et al., 2017)

These performance metrics make it possible to compare the effectiveness of the many algorithms in the Fog computing framework. These parameters provide QoS in the Fog system as described below (Subramoney and Nyirenda, 2022; Xie et al., 2019).

Makespan

The overall execution time required to complete a task. It is mathematically defined as follows.

$$Msp = MAX\{Ft_{x_i}, x_i \,\varepsilon\, X\} - MIN\{St_{x_i}, x_i \,\varepsilon\, X\}$$

Where Ft_{xi} ,St_{xi} denotes the finish and starting time of a task X= $\{x_1, x_2, \ldots\ldots x_i\}$ denote the number of task

Table 1. Brief of Workflow Scheduling Methods

Name of Articles	Notation / Paper Name	Brief Description
Saif et al., 2023	WSA1	• They invented the MGWO (Multi-Objectives Grey Wolf Optimizer) algorithm to reduce energy consumption and QoS delays. • It schedules the task by using the queue theory. • It works in the hierarchy and divides into four (wolf) steps: alpha, beta, delta, and omega. • The alpha (α), also known as the dominant wolf, works in the upper section of the pack and is in charge of guiding and making decisions for the others. • Beta (β), takes care of discipline and controls the Delta and Omega. • Delta (δ), is giving to α,β and commanding the omega (ω). • Omega (ω) works at the lowest level and aids layers above it. • Since it's a meta-heuristic approach, the fitness function is used to get energy-aware scheduling and improve QoS.
Yin et al., 2023	WSA2	• It developed an algorithm for scheduling the resources by incorporating the benefits of the GA and ACO algorithms. • So, it gets named the new genetic ant colony optimization (NGACO) algorithm. • They improve pheromone generation with the use of the roulette algorithm. • The experiment proves that this algorithm reduces total cost, economic cost, and makes span compared to the ACO algorithm.
Mehta et al., 2023	WSA3	• It invented a district heuristic algorithm for real-time scenarios by taking low latency as a QoS parameter. • It also minimizes the application response time by 11% compared to other algorithms. • It views fog nodes (FN) as being in a hierarchical order, and as you move up the hierarchy, the nodes' capacity may grow. • The fog nodes decide when to schedule events using RMS by considering resource availability. • In terms of computations, it states that fog nodes at distinct levels are heterogeneous, while those at the same level are homogenous. • Same-level nodes are linked by the same ancestor, which creates a cluster. • A node can be a member of only one cluster at the moment. • These nodes communicated through the Constrained Application Protocol and other web protocols.
Yadav et al., 2022	WSA4	• It developed an algorithm for scheduling jobs based on a hybrid of meta-heuristic and heuristic methods. • They used FWA and HEFT as a meta-heuristic and heuristic methods, respectively. • This technique uses both exploration and exploitation to fast-track the search in solution space. • It reduces the makespan and cost.

Kaur and Aron, 2022	WSA5	• It developed an approach based on resource utilization when scheduling the workflow. • It evenly distributes the workload to ensure efficient resource use. • It combines the techniques of water cycle optimization, plant growth optimization, and simulated annealing. • This PSW-Fog Clustering approach executes the task by considering scientific workflow like Cyber shakes workflow. • It minimizes the wastage of resources and system overhead and maximizes the execution speed of tasks.
Ahmed et al., 2021	WSA6	• They proposed the DMFO-DE approach, which is based on hybrid discrete opposition. • It initially produces the MFO algorithm (Moth-Flame Optimization) with the capability of discrete and opposition-based learning. • For enhancing convergence speed and avoiding the problem of local optima, this MFO is merged with the DE algorithm (Differential Evaluation). • Then it is employed by using DVFS.
Stavrinides and Karatza, 2021	WSA7	• To deal with real-time tasks in Fog Senior, they investigate the use of partial computations. • They also deal with input timing errors since if a job is only half completed, it will affect its immediate children and subsequent successors as well. • To address this issue, they invented a scheduling approach. • It schedules the task in two steps: task priority and VM selection. • The "Earliest Deadline First" (EDF) policy assigns priorities. • The VM selection by the fog scheduler (orchestrator) depends on the EFT (earliest estimated finish time).
Abdel-Basset et al., 2021	WSA8	• It created IEGA (enhanced elitism genetic algorithm) to schedule jobs in a fog environment and ensure QoS. • There are two steps to it. • The first step finds a near-optimal permutation by manipulating the crossover and mutation rates. • The second step is to find the optimal solution by mutating various options. • It generates improved results concerning fitness function, makespan, flow time, and energy consumption.
Hosseinioun et al., 2020	WSA9	• They proposed an energy-aware task scheduling algorithm by using the DVFS approach in a cloud-fog environment. • A combination of invasive weed optimization and evolutionary culture algorithms (IWO-CA) generates appropriate task sequences. • This IWO-CEA approach minimizes energy consumption and enhances the utilization of resources at the fog node.
Jamil et al., 2019	WSA10	• This mainly focused on scheduling the task and assignment of resources in a fog environment. • So that, the utilization of resources gets enhanced. • It proposed a scheduler that optimizes the delay and energy consumption. • Tasks were processed according to their length, so those with the shortest lengths were initially given to the fog node for execution. • It also minimizes network congestion due to the use of the SJF-Scheduler.
Choudhari et al., 2018	WSA11	• It invented an algorithm that uses the priority concept for scheduling jobs in a fog environment. • To fulfil the deadline constraint, it executes the jobs with greater priority.

DOI: https://doi.org/10.52756/ijerr.2023.v30.011

Rahbari et al., 2017	WSA12	• It invented heuristic algorithms which use the data mining technique for scheduling the module. • Its main focus is optimizing bandwidth, resource utilization, and security aspects. • It enhances the cost and energy consumption with a rate of 44.71% and 63.27% as compared to ACO, PSO, and SA algorithms.
Pham and Huh, 2016	WSA13	• It invented cost- a make span-based scheduling algorithm concerned with balancing task execution and the necessary expense of cloud resources. • It produced a better result in terms of deadline and QoS constraints.
Verma et al., 2016	WSA14	• It developed an algorithm for scheduling the real-time load considering the deadline constraints. • This maximized the utilization of the network as well as throughput.
Cardellini et al., 2015	WSA15	• They proposed a scheduler that is capable of scheduling the data stream application in a fog scenario. • Upgrading functionality for the real-time environment while optimizing application performance.

Table 2. Analysis of performance metrics

Algorithm	WSA1	WSA2	WSA3	WSA4	WSA5	WSA6	WSA7	WSA8	WSA9	WSA10	WSA11	WSA12	WSA13	WSA14	WSA15
MakeSpan	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	No	No	No	Yes	No	No
Speedup	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Cost	No	Yes	No	Yes	Yes	No	No	No	No	No	Yes	Yes	Yes	No	No
Response Time	No	No	No	No	No	No	Yes	No	No	No	Yes	No	No	No	No
Energy Consumption	Yes	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	No	No

DOI: https://doi.org/10.52756/ijerr.2023.v30.011

Resource Utilization	No	Yes	Yes	No	Yes	No	No	No	No	No	No	No	Yes	No	Yes
Load Balancing	No	Yes	No	No	No	No	No	No	No	No	No	No	No	Yes	No
Latency	No	No	Yes	No	No	No	No	No	No	Yes	No	No	Yes	Yes	Yes

Table 3. Critical Analysis based on simulator used

Algorithm	**WSA1**	**WSA2**	**WSA3**	**WSA4**	**WSA5**	**WSA6**	**WSA7**	**WSA8**	**WSA9**	**WSA10**	**WSA11**	**WSA12**	**WSA13**	**WSA14**	**WSA15**
iFogSim	No	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	No	Yes	Yes	No	No
Matlab	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Cloudsim	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	No
Cloud Analyst	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No
Other	No	No	No	No	No	No	C++ Programming Language	Java Programining Language	C# language	No	No	No	No	No	Apache Storm 0.9.3

DOI: https://doi.org/10.52756/ijerr.2023.v30.011

Cost

We discussed two types of cost: computation cost and communication cost.

Communication Cost

It is defined as the cost of transferring data from one task to another task (DT_{ij}=<x_i,x_j>) which is a positive value. It is mathematically defined as follows:

$$Ccost_{i,j} = UC_{i,j} * DT_{i,j}$$

Where $UC_{i,j}$=<xi,xj> denotes the unit cost of the transaction if the task is processed in different resources and UC_{ij} =0 when the task is processed in the same resource.

Computation Cost

It refers to the cost of a task when it is processed on resources. It defines as follows:

$$EC_i = p * (Ft_i - St_i)$$

Where 'p' is the unit processing cost.St_i and $Ft_i$are starting and finishing time of the task.

Total cost

For processing 'y' task on 'z' resources is defined as a sum of communication cost and computation cost.

$$Tcost = \sum_{i=1}^{y} . \sum_{j=1}^{y} Ccost_{i,j} + \sum_{r=1}^{z} . \sum_{i=1}^{y} EC_i$$

Energy Consumption

It is defined in terms of working and idle state. When the task is executed on the resource is known as working energy consumption and is denoted as Eng_{work}. When there is no task in the resource for execution, resources enter sleep mode, known as idle energy consumption and denoted as Eng_{idle}.

$$Eng_{work} = \sum_{i=1}^{n} a * f *_i s_i * Vol^2 \ (Ft_i - St_i)$$

Where,

a = constant

f_i = Frequency

S = Power Supply

Vol= Voltage

$$Eng_{idle} = \sum_{l=1}^{n} \sum_{idel_{l,m} \epsilon IDLEMode_{l,m}} a * f_{minimum} * Vol^2{}_{minimum} * TS_{l,m}$$

Where,

$IDLEModel_{l,m}$= Number of idle slots at l^{th} resource.

$Vol_{minimum}$ = Least voltage supply

$f_{minimum}$ = Least frequency

$TS_{l,m}$= Time duration for $idle_{l,m}$

So, Total Energy Consumption for executing task is defined as follows:

$$Eng_{total} = Eng_{work} + Eng_{idle}$$

Load Balancing

It defines as a standard deviation of all virtual Machine (VM)'s load. So, this deviation should be minimized to allocate an equal load on the VM. Suppose there is R number of VM is presented in Cloud and the S number of VM is presented in Fog.

$$VM_{load} = \frac{Number\ of\ task\ executed\ by\ VM}{VM's\ capacity}$$

$$FOG_{load} = \sqrt{\sum_{j=1}^{S} \frac{(LS_j - ALS)^2}{S}}$$

Where, LS_j= Load of j^{th} VM on fog,

ALS = Entire Virtual Machine's average load on fog

$$Cloud_{load} = \sqrt{\sum_{j=1}^{R} \frac{(LR_j - ALR)^2}{R}}$$

Where LR_j= Load of j^{th} VM on cloud,

ALR = Entire Virtual Machine's average load on cloud

Critical Analysis of Workflow Scheduling Methods

This section discussed the critical analysis of fifteen algorithms based on schedule on the fog platform. Here are three tables, such as Table 1, consisting of a brief description of the scheduling algorithms, their year of publication, and a notation of the algorithms. Table 2 contains various performance matrices, and these metrics are used to analyse fifteen algorithms. Here, yes means it is related to performance metrics for that algorithm, and no means the performance parameter is not used in that algorithm. The third table is based on different simulators used for the experimental purposes of the fifteen algorithms, and their details are shown in the table. The details of the critical analysis of the algorithm are shown below in the tables.

Here, fifteen algorithms are discussed in tabular form. These algorithms describe their usage, advantages, techniques on which they are based, and goals that they achieve. These algorithms are then matured in terms of timeliness, cost, resource usage, speedup, energy consumption, etc., as WSA1 minimized the energy consumption, WSA2 reduced the makespan, cost, and energy consumption, utilized the resources, and also balanced the load on VMs. WSA3 achieved low latency and high resource utilization. WSA4 minimized the cost and makespan.WSA5 also reduces cost, reduces energy consumption, and increases resource utilization. WSA6, WSA8, and WSA9 minimised the makespan and energy consumption. WSA7 minimized the makespan and response time. WSA10 gained low energy consumption and low latency. WSA11 reduces the response time and cost. WSA12 minimized the cost and energy consumption.

WSA13 reduced the makespan, cost, and latency and maximized resource utilization. WSA14, balanced the load and achieved low latency. WSA15 minimized latency and maximized resource utilization. This discussion clearly shows that these algorithms play a significant role in different criteria. This result is achieved by implementing these algorithms in different simulators such as iFog Sim, Matlab, Cloud Sim, Cloud Analyst, C++, Java, and other platforms. From Table 3, it is shown that WSA1 is implemented in Matlab, while WSA2 is implemented in iFog Sim. Similarly, the remaining algorithms are implemented on different platforms, as shown in Table 3.

Conclusion and future scope

This article begins by expanding on the concepts of cloud and fog computing. It covered the fundamentals of cloud computing, including its architecture, concepts, and models. This paper mainly focuses on studying different types of workflow scheduling in a fog computing environment. Every algorithm has some merits and demerits concerning performance metrics, simulators, and other parameters. This review paper illustrates fifteen algorithms in the fog computing platform in the three tables. The critical analysis of these algorithms is based on three factors, as shown in the table, which includes a brief description of the algorithms, performance metrics, and simulators used. Further, these fifteen algorithms are compared using numerical data based on different DAG models, and these data help compute different parameters of the algorithm. Upon this computing, we can do the performance analysis of the fifteen algorithms based on numerical data and draw graphs.

Conflict of interest

Nil

References

Abdel-Basset, M., Mohamed, R., Chakrabortty, R. K., & Ryan, M. J. (2021). IEGA: an improved elitism-based genetic algorithm for task scheduling problem in fog computing. *International Journal of Intelligent Systems*, *36*(9), 4592-4631. https://doi.org/10.1002/int.22470

Ahmed, O. H., Lu, J., Xu, Q., Ahmed, A. M., Rahmani, A. M., & Hosseinzadeh, M. (2021). Using differential evolution and Moth–Flame optimization for scientific workflow scheduling in fog computing. *Applied Soft Computing*, *112*, 107744. https://doi.org/10.1016/j.asoc.2021.107744

Ashi, Z., Al-Fawa'reh, M., & Al-Fayoumi, M. (2020). Fog computing: security challenges and countermeasures. *Int. J. Comput. Appl.*, *175*(15), 30-36. https://doi.org/10.5120/ijca2020920648

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* (MCC), pp. 13–16. doi/pdf/10.1145/2342509.2342513

Cardellini, V., Grassi, V., Presti, F. L., & Nardelli, M. (2015). On QoS-aware scheduling of data stream applications over fog computing infrastructures. IEEE. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, pp. 271-276. https://doi.org/10.1109/ISCC.2015.7405527

Chakraborty, M. (2019). Fog computing vs. cloud computing. *arXiv preprint, arXiv,1904.04026*.

Choudhari, T., Moh, M., & Moh, T. S. (2018). Prioritized task scheduling in fog computing. In *Proceedings of the ACMSE 2018 Conference*, pp. 1-8. doi/abs/10.1145/3190645.3190699

Chowdhury, A., Karmakar, G., & Kamruzzaman, J. (2019). The co-evolution of cloud and IoT applications: Recent and future trends. IGI Global. In *Handbook of Research on the IoT, Cloud Computing, and Wireless Network Optimization*, pp. 213-234.

Dastjerdi, A. V., Gupta, H., Calheiros, R. N., Ghosh, S. K., & Buyya, R. (2016). Fog computing: Principles, architectures, and applications. In *Internet of Things*, pp. 61-75.

Diaby, T., & Rad, B.B. (2017). Cloud computing: a review of the concepts and deployment models. *International Journal of Information Technology and Computer Science*, *9*(6), 50-58. https://doi.org/10.5815/ijitcs.2017.06.07

Firdhous, M., Ghazali, O., & Hassan, S. (2014). Fog computing: Will it be the future of cloud computing? *Proceedings of the Third International Conference on Informatics & Applications, Kuala Terengganu, Malaysia, 2014*, 8-15.

Ghobaei-Arani, M., Souri, A., & Rahmanian, A. A. (2020). Resource management approaches in fog computing: a comprehensive review. *Journal of Grid Computing*, *18*(1), 1-42. https://doi.org/10.1007/s10723-019-09491-1

Gill, S. S., Xu, M., Ottaviani, C., Patros, P., Bahsoon, R., Shaghaghi, A., & Uhlig, S. (2022). AI for next generation computing: Emerging trends and future directions. *Internet of Things*, *19*, 100514. https://doi.org/10.1016/j.iot.2022.100514

Hosseinioun, P., Kheirabadi, M., Tabbakh, S. R. K., & Ghaemi, R. (2020). A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm. *Journal of Parallel and Distributed Computing*, *143*, 88-96. https://doi.org/10.1016/j.jpdc.2020.04.008

Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of Network* *and Computer Applications*, *98*, 27-42. https://doi.org/10.1016/j.jnca.2017.09.002

Jamil, B., Shojafar, M., Ahmed, I., Ullah, A., Munir, K., & Ijaz, H. (2020). A job scheduling algorithm for delay and performance optimization in fog computing. *Concurrency and Computation: Practice and Experience*, *32*(7), e5581. https://doi.org/10.1002/cpe.5581

Kaur, M., & Aron, R. (2022). An energy-efficient load balancing approach for scientific workflows in fog computing. *Wireless Personal Communications*, *125*(4), 3549-3573. https://doi.org/10.1007/s11277-022-09724-9

Luan, T. H., Gao, L., Li, Z., Xiang, Y., Wei, G., & Sun, L. (2015). Fog computing: Focusing on mobile users at the edge. *arXiv preprint*, *arXiv,1502.01815*.10.

Mehta, R., Sahni, J., & Khanna, K. (2023).Task scheduling for improved response time of latency sensitive applications in fog integrated cloud environment. *Multimedia Tools and Applications*, pp. 1-24. https://doi.org/10.1007/s11042-023-14565-0

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. *National Institute of Standards and Technology Special Publication*, *53*, 1-7.

Miyachi, C. (2018). What is "Cloud"? It is time to update the NIST definition? *IEEE Cloud computing*, *5*(03), 6-11.

Pham, X. Q., & Huh, E. N. (2016).Towards task scheduling in a cloud-fog computing system. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1-4. IEEE. https://doi.org/10.1109/APNOMS.2016.7737240

Rahbari, D., Kabirzadeh, S., & Nickray, M. (2017). A security aware scheduling in fog computing by hyper heuristic algorithm.In *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, pp. 87-92. IEEE. https://doi.org/10.1109/ICSPIS.2017.8311595

Saif, F. A., Latip, R., Hanapi, Z. M., & Shafinah, K. (2023).Multi-objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud-Fog Computing. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3241240

Sarkar, S., & Misra, S. (2016). Theoretical modelling of fog computing: a green computing paradigm to support IoT applications. *Iet Networks*, *5*(2), 23-29. https://doi.org/10.1049/iet-net.2015.0034

Stavrinides, G. L., & Karatza, H. D. (2021). Orchestrating real-time IoT workflows in a fog computing environment utilizing partial computations with end-to-end error propagation. *Cluster Computing*, *24*(4), 3629-3650. https://doi.org/10.1007/s10586-021-03327-y

Subramoney, D., & Nyirenda, C. N. (2022). Multi-Swarm PSO Algorithm for Static Workflow Scheduling in Cloud-Fog Environments. *IEEE Access*, *10*, 117199-117214. https://doi.org/10.1109/ACCESS.2022.3220239

Tsai, W. T., Sun, X., & Balasooriya, J. (2010). Service-oriented cloud computing architecture. IEEE, In *2010 Seventh International Conference on Information Technology: New Generations*, pp. 684-689. https://doi.org/10.1109/ITNG.2010.214.

Verma, M., Bhardwaj, N., &Yadav, A. K. (2016). Real time efficient scheduling algorithm for load balancing in fog computing environment. *Int. J. Inf. Technol. Comput. Sci.*, *8*(4), 1-10. https://doi.org/10.5815/ijitcs.2016.04.01

Xie, Y., Zhu, Y., Wang, Y., Cheng, Y., Xu, R., Sani, A. S., Yuan, D., & Yang, Y. (2019). A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment. *Future Generation Computer Systems*, *97*, 361-378. https://doi.org/10.1016/j.future.2019.03.005

Yadav, A. M., Tripathi, K. N., & Sharma, S. C. (2022).A bi-objective task scheduling approach in fog computing using hybrid fireworks algorithm. *The Journal of Supercomputing*, *78*(3), 4236-4260. https://doi.org/10.1007/s11227-021-04018-6

Yin, C., Li, H., Peng, Y., Fang, Q., Xu, X., & Dan, T. (2023). An optimized resource scheduling algorithm based on GA and ACO algorithm. Pre-print (Version 1) available at Research Square, https://doi.org/10.21203/rs.3.rs-2559005/v1

How to cite this Article:

Aakriti Jain and Ranjit Rajak (2023). A Systematic Review of workflow scheduling techniques in a fog environment. *International Journal of Experimental Research and Review, 30*, 100-108.

DOI：https://doi.org/10.52756/ijerr.2023.v30.011