Research Inventy: International Journal of Engineering And Science
Issn (e): 2278-4721, Issn (p):2319-6483, PP: 64-74
www.researchinventy.com

Optimal Scheduling in Fog Environment: Opportunities and Obstacles

Navjeet Kaur[1], Ashok Kumar[1,*], and Rajesh Kumar[2]
[1]*Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab,India.*
[2]*Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India.*

Abstract
The emergence of the Internet of Things (IoT) brought fog computing into the existing paradigm, initiat- ing many discussions among researchers and the business community. Fog computing is one of the promising solutions for real-time data processing in the IoT environment, still faces various issues in scheduling tasks within the deadline. This survey paper investigates the basics of Fog computing, its key features, and its comparison with a similar paradigm. Further, the paper target the critical area of task scheduling in Fog scheduling, its need, and significant research gaps behind task scheduling in Fog computing. The paper analyzes and presents a comparative summary of the existing state-of-the-art scheduling strategies for the Fog environment. Further, this paper provides detailed insight on task scheduling open issues that are still uncovered or need improvements like mobility, security, storage, and fairness of resource allocation.
Keywords— *Cloud Computing, Fog Computing, Task Scheduling, Optimal Task-resource*

I. Introduction

The advancement of technology empowers higher functionalities with emergent demand to shrink the size of the IoT devices [1]. This demand often deals with the limited computation capacity of stand-alone devices. Further, the conventional cloud model also fails to deliver the required computation due to its centralized approach and outlying nature [2, 3]. Here, Fog computing comes into consideration that complements the existing cloud computing model by stationing services closer to the data generating sources. Fog computing enhances cloud computing by shifting its capabilities down in the form of intelligent Fog nodes [4]. These Fog nodes are installed near the network's edge and equipped with varying storage capacity, computation, and processing. These Fog nodes can process and execute high functionality dynamic tasks in real-time after analyzing requirements and emergencies. Therefore, almost every sector today needs Fog computing in direct and indirect ways to address many issues related to IoT developments in order to achieve low latency, bandwidth, network traffic, and high scalability [5, 6].

Instead of all convenience offered by Fog computing, it is not easy to implement due to the dynamic and limited configuration of Fog nodes with varying task demands [7, 8]. The paper provides a detailed survey of Fog computing, its advantages, and challenges in scheduling tasks among different Fog nodes.

1.1 Fog Computing

To address the challenges of IoT applications in the conventional cloud, Cisco coined the concept of Fog comput- ing in 2012 [9]. Fog computing is introduced as an extension to classic cloud computing services by decentralizing computing infrastructure and bringing the power of the cloud closer to the edge network where data is generated [10].

The Fog computing layer is depicted in Figure 1 that complements cloud computing by enabling short-term analytics at the edge network with the help of traditional networking components like switches, routers, Base Station (BS), etc. The Fog layer is expanded from edge to core networks, which comprises a Fog network layer. Further, cloud network is often far located and performs resource-intensive, longer-term analytics. The components at the fog layer are provided with disparate capabilities of computing and networking to support the execution of real-time applications timely. Further, these networking components create a large geographical distribution of cloud-based services that perform expeditiously in terms of a service delay, energy consumption, network traffic, etc. Fog computing also facilitates mobility support, real-time interactions, scalability, and interoperability, making it a better option than solely using the far located cloud.

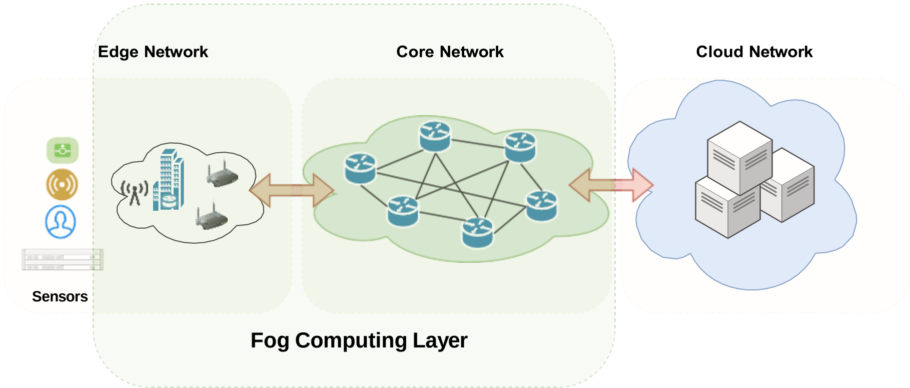

Figure 1: Fog Computing Layer in Basic Cloud Model

The motivation behind doing this work is to improve the task scheduling process of Fog computing so that time-sensitive applications get processed in real-time. Also, from the literature review, the current status of the work is quite limited and needs many improvements.

1.2 Fog Computing Architectures

Fog architecture necessitates using traditional networking components (switches, routers, multiplexers, etc) for computing, storing, and processing IoT applications. Fog architecture forms an extensive network of inter- connecting devices using various physical and logical elements, i.e., software and hardware. The distribution and layering of the Fog nodes is a crucial feature in architecture design. This section discussed various Fog architecture, including the standard reference architecture and other state-of-the-art designs for Fog computing.

1.2.1 OpenFog Reference Architecture

OpenFog Consortium Architecture Working Group [11] given the most detailed architecture of Fog computing named as OpenFog RA and covered significant aspects of good Quality of Service (QoS), which is adopted as an IEEE standard and named as IEEE 1934 in June 2018 [12]. The architecture envisaged high throughput, low latency, and reduced network cost through Fog nodes to be installed near IoT devices rather than on gateways of networks like edge computing. The main idea is to perform local processing via Fog nodes so that tasks can be validated, processed, and executed that are sensitive and require a prompt reply. They also presented four different high-level deployment models considering application-to-application complexity levels. The OpenFog RA operates on eight core pillars that intend to guide the definition of the standard reference architecture. The pillar represents the key characteristics that a system should deploy for the uniform distribution of computing, storage, networking, and security functions closer to the data generating source.

1.2.2 Other Architectures

Although there are many proposed Fog computing architectures, no single architecture still follows all the eight pillars of standard Fog architecture, also called open Fog reference architecture. Hao et al. [13] given software computing architecture for the Fog. This architecture is designed in stack form, having four layers. However, the model does not consider an encryption policy for sensitive data in the proposed model, which is entirely possible when data is communicated from one layer to another. Chen et al. [14] presented a three- layered, demand-based, multi-level model, claiming various features like availability of resources, scalability, and interoperability. Although, the model undiscussed any agreement criteria for Fog node composition. Further, end-to-end security, user mobility, energy efficiency, and network cost aspects are not considered. Luo et al.

[15] presented a container-based Fog model that illustrates it profoundly with three tiers: access tier, control tier, and infrastructure tier. Oma et al. [16] represent a general Fog workflow structure that shows significant improvement in terms of fast tasks processing as each Fog node process task of small size that can be processed and integrated quickly. The model follows the tree-based approach for information processing, considering that each Fog node must have a parent node and one or more child Fog nodes. Tuli et al. [17] proposed a lightweight, cross-platform application deployment model that supports the management of resources and provides flexibility in accessing various IoT services. The architecture also used blockchain technology to secure sensitive data from unauthorized updates. Sharma et al. [18] three-layer architecture provides a secure mechanism by implementing a blockchain-based software-defined

network (SDN) at the Fog layer. Chang et al. [19] architecture overcome the challenge of additional network cost of creating Fog infrastructure by using Consumer as a provider (CaP) service model. The model is three-layered, and the main computing layer is the Fog layer, where the Fog server will offload the traffic from the client and perform the local processing of data. Cerina et al. [20] four-layered architecture is based upon Field-programmable gate array technology to be embedded at the Fog layer for better communication.

1.3 Task Scheduling in Fog Computing

The necessity of Fog computing arises when enormous data is generated by time-sensitive applications that require low-latency computing. These applications include smart wearables, IoT systems, mobile devices, real- time navigation, smart vehicles, etc. Fog computing furnishes an extra layer of Fog nodes to reduce the latency between IoT devices and the cloud computing infrastructure. The task scheduling is the prime responsibility of a Fog scheduler that further ensures QoS to users [21, 22]. The task scheduling in Fog computing allocates available resources for the task execution. The limited computing capacity of Fog nodes in Fog computing is crucial for efficient task scheduling due to its stringent delay requirements [23]. Additionally, the limited availability of resources to serve numerous user requests is also a challenge. It is, therefore, desirable to serve tasks according to their priorities. Otherwise, an inefficient Fog node allocation could lead to a loss for end users. Hence, task scheduling in Fog computing is essential to assign efficient Fog nodes to achieve the maximum profit. [24, 25].

Task scheduling in Fog computing finds an optimal task-node pair among available Fog nodes. Moreover, selecting a capable resource-efficient Fog node, deploying an efficient scheduling strategy to minimize delay, and scheduling tasks before deadlines are some of the primary responsibilities in the Fog scheduler task scheduling. A practical task scheduling in Fog computing schedules the task(s) before its deadline and provides seamless services to the users. Task Scheduler in Fog computing guarantees this efficiency, also termed Fog Manager (FM) or broker who makes intelligent decisions at Fog nodes. The FM works on finding optimal task-node pairs with the least processing time.

The existing literature presents the Fog task scheduling through various phases and focus on numerous QoS parameters like response time [10,16,25,28-30,31,32,34,39,43, 53,61], cost reduction [9,16,28-30,34,40,45,47,53],
energy consumption [8,16,25,20, 32,33,37,50-52,54], security [13,14,30,56], load balancing [7,11,26,31,33,36, 37,41, 49,57,63,64] and fairness parameter [49, 50,60] to schedule task among Fog nodes.

1.4 Contribution

The major contribution of the paper is as follows:

- The paper briefs about the overview of Fog computing, its key characteristics, major entities and Fog com- puting reference architecture. The paper further compares Fog computing with foundation technologies such as cloud and edge computing.
- The paper covers task scheduling, its need, process, scheduler responsibilities, significant issues, and discussion about quality parameters in Fog computing.
- Finally, it covers a detailed literature review on task scheduling techniques, QoS analysis and identi- fies research gaps. The paper further highlights the identified challenges and existing solutions in task scheduling.

1.5 Paper Organisation

The paper is divided into five sections, where Section 2 covers the detailed literature review of scheduling algorithms in fog computing. Section 3 provides results analysis on various parameters. Section 4 discusses the identified challenges and existing solutions. Finally, the final section summarizes the findings and the concluding remark.

II. Scheduling Algorithms

This section presents a detailed survey of the task scheduling algorithm in a Fog environment where Ahmed et al. [26] proposed a fuzzy clustering task allocation approach to solve scheduling problems to minimize delay, cost, and energy. Tychalas et al. [27] proposed a method that executes some tasks on fog nodes and some on cloud nodes in order to reduce delays and costs of the system. The scheduling system makes several suppositions, such as independent tasks, all the fog nodes able to execute jobs, etc. A threshold value is also a station to represent the usage of the fog node. Boveiri et al. [28] proposed a meta-heuristic Min-Max Ant System based upon Ant Colony Optimization (ACO) to schedule tasks. Another ACO-based dynamic task scheduling technique is presented by Singh et al. [29], where the pheromone value is replaced with the usability index of the resources. A Virtual Machine (VM) allocation algorithm is introduced to execute tasks using link optimization. The algorithm further finds the optimal path to resources. Luo et al. [15] focuses on energy-efficient task scheduling in Fog computing. A threshold value for the resource requirement is considered. If its resource requirement crosses a given threshold, the tasks are scheduled on the cloud node; otherwise, the Fog node executes the tasks. Nguyen et al. [30] introduced a time-cost-aware task scheduling algorithm for optimal dispersal of fog resources considering time and cost. The authors use genetic algorithms for optimal fog node selection.

Wang et al. [31] introduced a hybrid-heuristic algorithm by combining and improving the features of Par- ticle Swarm Optimization (PSO) and ACO. The work also removes checks related to terminal devices in Fog computing. Stavrinides et al. [32] proposed a hybrid heuristic method for task scheduling to be deployed on Fog or cloud, based on their communication and computation demand.

Sujana et al. [33] proposed a task scheduling technique that focuses on security issues related to Fog computing. They layout a trust-based stochastic scheduling algorithm to detect an optimal task-resource pair for efficient scheduling in the Fog environment. A non-preemptive real-time security-aware scheduling is proposed by Auluck et al. [34]. Choudhari et al. [35] proposed work in two main phases where the task is allocated to a Fog server in the first phase, and task priorities and subdivision of tasks is considered in the second phase. Wu et al. [36] proposed Energy Minimization Scheduling (EMS), a heuristic approach based on ILP (Integer Linear Programming). Pham et al. [37] proposed work calculates the priority level of the task where the task with the highest priority is scheduled according to the set utility function.

Bitman et al. [38] proposed a Bee Life Algorithm (BLA) for creating a population or jobs for task scheduling and their search for food as a strategy to deploy an appropriate Fog node to execute the jobs. Wan et al. [39] proposed the energy-aware load balancing and scheduling algorithm that ensures minimum consumption of energy and uniform workload among Fog nodes. Hoang et al. [40] proposed a heuristic region-based Fog computing scheduling method. Zhang et al. [41] proposed a task offloading scheme based on a fairness metric to select Fog nodes and then offload the task to those efficient Fog nodes to minimize the delay. Yang et al. [42] proposed algorithm uses the Lyapunov optimization technique to schedule tasks in order to minimize service delay and enhance energy efficiency.

A heuristic-based scheduling approach is proposed by Nazir et al. [43], based on the egg-laying nature of a lazy bird cuckoo called the cuckoo optimization algorithm. The technique further ensures efficient load balancing and energy efficiency. Liu et al. [44] introduced a cross entropy-based task scheduling strategy for the multiuser and multi-Fog network. Abreu et al. [45] proposed a ranking method for Fog nodes among total available nodes based on divisions, score, and rounds. The proposed approach schedules the tasks based on the deadline. An adaptive double fitness genetic task scheduling technique is proposed by Liu et al. [46] that is based on biological phenomena to reduce the delay and cost in the Fog network. The idea behind the proposed scheduling method is to use genetic evolution where an individual in a population set is optimized based on selection, crossover, and mutation operation. Tellez et al. [47] proposed a meta-heuristic method of Integer Linear Programming (IPL) for scheduling in the Fog network called tabu search. The main idea behind tabu search is to find an optimal solution from one potential solution by exploring each solution’s neighborhood and gradually progressing.

Liu et al. [48] proposed a Dispersive Stable Task Scheduling (DATS) algorithm based on stable matching mathematical theory to identify beneficial helper nodes for offloading of the task among them. Bittencourt et al. [49] proposed a scheduling strategy that takes scheduling decisions to execute tasks on different resources like Fog or cloud nodes as per the resource demand. Zhang et al. [50] proposed Delay Optimal Task Scheduling (DOTS), which aims at minimizing the delay rate based on the proposed Capability Report Ratio (CRR) of the Fog nodes for optimal scheduling. Fizza et al. [51] proposed Privacy-Aware Scheduling in a Heterogeneous Fog Environment (PASHE), a task scheduling algorithm that allocates sensitive tasks among heterogeneous Micro Data Center(MDC) and Cloud Data Center(CDC).

Zhang et al. [52] proposed offloading model that works around task nodes and an additional idle Fog node for offloading the task. The task is divided into two sub-tasks and offloaded to the allocated Fog node. Table 1 is the comparative chart of Task Scheduling Algorithms in Fog Computing.

III. Review Analysis

This section provides the quantitative summary of various analysis performed on the literature review. Figure 2 represents the analysis over the different scheduling techniques.

Further, QoS parameters are analyses and shown in Figure 3, which elaborates time as the most focused parameter in Fog task scheduling in the reviewed research with 38.5 % and cost, energy with load balancing being the second most target issue in QoS. The results also showed optimal resource searching with 9.0 %, security 5.1 %, and memory utilization with 2.6 %.

Note that in Figure 3, time is considered a common term for all different types of time like response time, delay time, deadlines, execution time, and completion time. Figure 4 represents a quantitative summary of

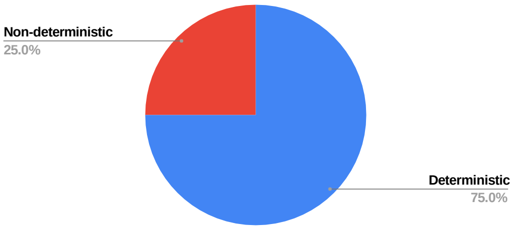

Figure 2: Scheduling Algorithms

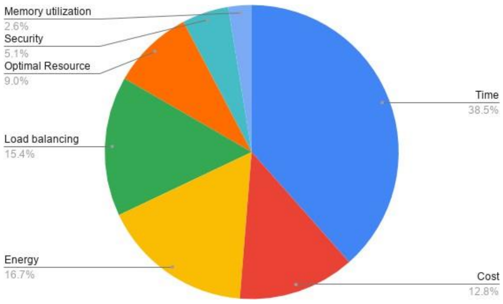

Figure 3: QoS Analysis

different times taken by various researchers where improving the response time is the main focused parameter.

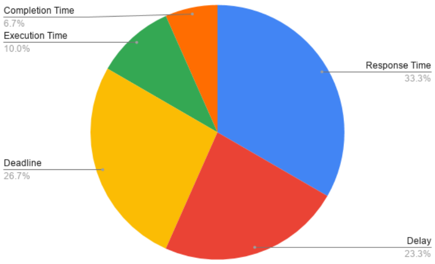

Figure 4: Different type of Time Analysis

IV. Identified Challenges and Existing Solutions - A Discussion

In task scheduling, two entities play an important role i.e., Fog nodes and requested task. The fog node is where a scheduling algorithm is implemented to process the requested task. The review in the above section helps in the identification of some major challenges, which are as follows:

- Limited processing capacities of Fog nodes: Fog network is a composition of various heterogeneous Fog nodes like a router, tablet, or maybe a different computer in terms of its configuration act like a scheduler to process data. Managing these heterogeneous devices with limited storage and computing capacity across the network is a tedious task. Existing literature suggests solutions with a container or docker approach for efficient memory usage [8,43].
- Heterogeneity and coordination among Fog nodes: The cloud-Fog system consists of various nodes called Fog nodes situated at different ownership and policies. So, it is complicated to build a policy for com- munication on standard terms and conditions. Also, Fog nodes are spread over a network with different configurations, so it becomes a challenge to coordinate among them on standard terms and conditions. Existing literature suggests solutions with DAG, fuzzy clustering, designation of a Fog broker, and helper Fog nodes to manage coordination among Fog nodes [26, 53, 37, 54, 50].
- Fog nodes Mobility: The Fog nodes are not static. They can move and register themselves in another area. A mapping is required which keep track of nodes leaving and entering the network. Existing research on this issue is quite limited where [49] suggests a solution by keeping track of the

present time of the Fog nodes.

- Data protection and Fog node security: Fog nodes can be public and private with different security policies. There are several difficulties in using them on common security policy as some user wants to execute sensitive data with privacy concern. There are authors [34, 33, 55], who work on Fog security, but its hierarchical nature is not considered. Security is one of the foremost concerns in Fog computing that ensures data integrity at various levels of task scheduling. Existing literature suggest solutions with trust models by [33], authentication mechanism by [55] and private, public Fog node divisions by [34, 51].
- Dynamic user requirements: The fog environment continuously deals with dynamic changing requirements of the user(s) with time. The Fog system needs continuously update the data to reorganize its priority queue in case of emergency requirements. Existing literature mainly simulates their work on a static environment with no so complex systems like [56] considered only one task per processor and VM. So, it is a great advantage to simulate work in a complex dynamic environment to achieve better realistic results. Naha et al. [57] contribute by creating a dynamic system with various phases of task scheduling.
- Deciding task priority: Fog computing is introduced to deal with time-sensitive priority data. The fun- damental issue in scheduling tasks is deciding the priority of a task with other tasks. Authors suggest various parameters on this issue [58, 41, 51, 49] like priority queue, deadlines of a task, and completion time of a task.
- Task sub-division and offloading policy: A job cannot be executed on a single node due to the limited re- source capacity of the Fog nodes. The application processes need to be scattered over multiple Fog devices for processing that needs efficient synchronization of devices and policy for final result compilation [40].

[59] suggest a solution through DRL, [60] suggests the Kubernetes approach that performs the orchestra- tion of Fog nodes. [61] introduce Fog functions, which are independent, loosely coupled functions in an application that can efficiently run on different Fog nodes, which minimum communication requirement.

- Load balancing among Fog nodes: One of the significant responsibilities of Fog task scheduling is to make optimal use of resources out of the available pool of resources. The resources' optimal use is essential be- cause they are available with limited capacities and data unpredictably changing its requirement. Wastage of resources will have a profound impact on the performance of Fog [62, 63, 29, 64]. Migration of resources in the Fog network is one of the solutions for handling faults and failures among Fog nodes. However, it also introduces the overhead of managing different roles [65].
- Optimal task-machine pairing: The optimal task-resource pairing decision is an important aspect and challenge in Fog computing. Moreover, management and connections between diversely distributed Fog nodes are open issues in Fog. These Fog nodes range from high-performance servers to gateways, access points, and base stations. The infrastructure is also comprised of different wired and wireless connections. Existing literature work done by [32, 33, 66, 26] did not consider the heterogeneity and hybrid nature of tasks and resources.
- VM Migration in Fog nodes: Migration is one of the best features in the Cloud-Fog virtualization envi- ronment, allowing virtual machines to move seamlessly. The feature also faces many challenges in terms of faults and failures of hardware devices on which it relies. As the Fog system is more geographically distributed than cloud computing, VM issues are more prominent. So, this topic is of great importance for the Fog research field. There is some solution in terms of VM framework and policies provided by [67, 68, 69].
- Optimal ranking of Fog nodes: A resource management system takes a task as input and returns the list of resources available within a pool of available resources. To make an optimal pairing of the highest priority task with Fog Node, we need a method to rank these resources, i.e., ranking Fog nodes. Various authors suggest various strategies like [57] rank resources in terms of processing capacity, [45] rank based on workload, and [70] use hop count.
- Time-sensitive data: The major concern of introducing Fog computing is to manage emergency data. This will be done by processing data and generating information locally which is previously managed by cloud task scheduling [10,16,25,28,29,30,31,32,34,39,43,53,61].

Table 1: Task Scheduling Algorithms in Fog Computing

Author	Problem Focused	Type	Solution Proposed	Implementation	Research Gap
Luo et al. [15]	Energy consump-tion and Resource demand	Deterministic	The author focuses on energy efficient scheduling of tasks and define a theta value to limit the energy consumption of the Fog devices.	Linkpack software, MySQL is used to evaluate the performance of VM and containers.	Average no of transaction handled is almost same when no of container/VM increases.
Ahmed et al. [26]	Time, Cost and Energy	Deterministic	The author proposed two-tier Bi-partite Graph with Fuzzy Clustering Task Allocation Approach (2tBiFTA) to solve META considering minimization of delay, cost and energy.	CloudSim is used to evaluate the performance	Single VM is consid-ered/Fog node and Fog broker is used making approach centralized.
Boveiri et al. [28]	Optimal Route/path to reach resources	Non-deterministic	The author proposed a meta-heuristic graph-based scheduling approach based upon ACO.	Random task graphs are generated to evaluate the proposed approach using Microsoft Visual Basic 6.0 programming language	The author doesn't consider the case if the state (Fog node) visited by elite ant is not available any-more after the updation.
Singh et al. [29]	Shortest path	Deterministic	The authors introduced a link optimization based optimal VM allocation algorithm.	iFogsim simulation done with 50 nodes and 50 virtual machines.	Only one virtual machine per Fog node is consid-ered.
Wang et al. [31]	Optimal path and Search	Non-deterministic	The author proposed a hybrid-heuristic algorithm by combin-ing and improving the features of ACO and PSO.	HH, IPSO, IACO perfor-mances evaluated through Simulation on MATLAB	-.
Stavrinides et al. [32]	Workload	Non-deterministic	The author introduced commu-nication overhead and computation demand as two parame- ter for efficient task scheduling Higher the value of communication overhead higher the chances to execute task Fog.	They implemented their own discrete-event simula-tion program in C++.	Case not taken up for the same values of introduced parameters.
Sujana et al. [33]	Searching optimal VM	Deterministic	The author proposed an algo-rithm is to find an optimal task VM pair to provide efficient scheduling of task in Fog environ-ment.	Random graph genera-tor is used for analysis of trust value and make span.	VM current workload evaluation while assigning a new priority task is not considered.

Continuation of Table 1					
Author	Problem Focused	Type	Solution Proposed	Implementation	Research Gap
Auluck et al. [34]	Security and Deadline	Deterministic	The authors introduced a RT-SANE algorithm which is based upon security and deadline as the two main parameters to schedule tasks.	They used iFogSim simulation experiment on scientific cloud from Czechoslovakia called CERIT-SC system.	-.
Choudhari et al. [35]	Deadline and Delay	Deterministic	The proposed scheduling strategy is based on priority levels, deadline and delay factor to schedule tasks at FS.	CloudAnalyst simulator is used to evaluate perfor-mance on response time and cost.	Large Task subdivision and distribution criteria is missing.
Wu et al. [36]	Energy consump-tion and Exe- cution time	Deterministic	They proposed Energy minimized scheduling strategy which further used two scheduling algorithm IEF and WEF running one after the other, reducing energy consumption and execution time among Fog nodes.	EMS algorithm is evaluated using the benchmark programs in SPEC CPU2006 with workload of 10 and 20 benchmark programs	-.
Pham et al. [37]	Makespan and Mon-etary Cost	Deterministic	The author proposed a heuristic based algorithm based upon DAG and utility function to schedule the task efficiently on Cloud and Fog nodes.	Cloudsim used to evaluate the performance on mem-ory, storage and cost	The performance on DAG is dependent on previous task to complete first, can not handle multiple events at a time.
Bitman et al. [38]	CPU execution Time and Memory	Non-deterministic	The author proposed a Bee Life Algorithm (BLA) inspired from Bees marriage and food searching strategy to schedule tasks in Fog Computing.	Simulation tests done according to different Fog computing infrastruc- tures.	The adopted approach doesn't consider dynamic job scheduling.
Pham et al. [53]	Cost, Makespan	Deterministic	They enhanced their own work in [37] by introducing a deadline	CloudSim is used to eval-uate the performance on	The approach doesn't con-sider multiple event han

	and Deadline		based task reassignment phase in which they assign critical task to better processing node to control their execution time.	memory, storage and cost.	dling at a time and energy consumption.

Continuation of Table 1

Author	Problem Focused	Type	Solution Proposed	Implementation	Research Gap
Wan et al. [39]	Energy and Load balancing	Deterministic	The author introduced Energy aware load balancing scheduling (ELBS) in which they introduce a energy consumption model and a load balancing approach based on multiagent System	Prototype implementation of proposed algorithm has been done.	Fast searching speed of PSO but accuracy is low. [31].
Yang et al. [54]	Energy Efficiency	Deterministic	They proposed MEETS, an energy efficiency algorithm that considers circuit, offloading and computation power as parameters for total energy consumption in their model. The motive is to bring quality in task offloading per energy consumption.	The opportunistic spectrum access is considered for simulating the proposed approach.	Energy consumption of sending back the computed output from the helper nodes to the task node is ignored.
Zhang et al. [41]	Offloading and Delay	Deterministic	The author proposed a task a fair task offloading scheme based on fairness metric to select Fog node for offloading task and then offload task to those efficient Fog nodes to minimized delay.	The performance is evaluated through simulation considering Task delay energy consumption and the Fog node fairness for energy consumption.	-
Yang et al.[42]	Delay and Energy	Deterministic	The author proposes delay energy-balanced task scheduling called DEBTS algorithm use lyapunov optimization technique to schedule average service delay to enhance energy efficiency.	Numerical Evaluation has been done to evaluate performance.	-
Nazir et al.[43]	Workload and Energy	Non-deterministic	The author proposed a heuristic based scheduling approach which is based on laying nature of a lazy bird cuckoo.	cloudAnalyst is used to simulate the proposed COA.	-

Continuation of Table 1

Author	Problem Focused	Type	Solution Proposed	Implementation	Research Gap
Liu et al. [44]	Latency and Energy consumption	Deterministic	The author introduced a cross entropy based task scheduling strategy for multiuser and multi Fog network.	The simulation done for the proposed approach on light, medium and heavy workload task.	Cross entropy is usually effected by cross entropy error as distributions with long tails can be modeled poorly with too much weight given to the unlikely events.
Abreu et al.[45]	Resource ranking	Deterministic	The author proposed a ranking method of Fog node/cloudlet among total available cloudlets on the basis of divisions, score and rounds.	CloudSim Plus is used to evaluate performance and compared with round robin scheduling	The mean latency between scheduling method is relatively small.
Tellez et al. [47]	Fast searching of Fog nodes	Non-Deterministic	They proposed a meta-heuristic method IPL for scheduling in Fog network called tabu search.	Experimental Prototype is built to evaluate performance.	In genetic algorithm, result is just an approximation, not exact solution.
Liu et al. [48]	Workload distribution	Deterministic	The author proposed a DATS algorithm which is based upon stable matching mathematical theory to identify beneficial helper nodes for offloading of the task among them.	Simulationexperiment done to evaluate performance	-
Zhang et al. [50]	Delay, Energy and Fairness	Deterministic	The author proposed DOTS, aims at minimizing the delay rate on the basis of the proposed CRR of Fog nodes.	Simulationexperiment is done to check energy consumption and fairness level among Fog nodes.	Author missed showing the biding criteria regarding nominations of voluntary Fog nodes.

Fizza et al. [51]	Security and Deadline	Deterministic	The author-proposed PASHE, a task scheduling algorithm which allocate sensitive tasks among heterogeneous MDC and CDC.	Simulation is done using iFogSim used 12 users, 3 MDC and 1 CDC with varying capacity	-.

V. Conclusion

In Fog computing, task scheduling is the structured way to assign available resources for the task execution. The limited computing capacity of Fog nodes in Fog computing is crucial for efficient task scheduling due to its stringent delay requirements. Additionally, the limited availability of resources to serve numerous user requests is also a challenge. In this paper, dominance and challenges of task scheduling in Fog computing is thoroughly investigated. It is found that tasks need to be served according to their priorities. Otherwise, allocation of an inefficient Fog node could lead to loss for end users. Hence, task scheduling in Fog computing is of great importance to assign efficient Fog nodes to achieve the maximum profit. Moreover, mobility and security are the major factors that influence the performance of task scheduling in Fog computing. Finally, there is always remains a scope for improvement and research directions still needs to be investigated to provide better and efficient solution in task scheduling in Fog computing.

References

[1]. Pengfei Hu, Sahraoui Dhelim, Huansheng Ning, and Tie Qiu. Survey on fog computing: architecture, key technologies, applications and open issues. Journal of network and computer applications, 98:27–42, 2017.

[2]. Mithun Mukherjee, Lei Shu, and Di Wang. Survey of fog computing: Fundamental, network applications, and research challenges. IEEE Communications Surveys & Tutorials, 20(3):1826–1857, 2018.

[3]. Ghazaleh Javadzadeh and Amir Masoud Rahmani. Fog computing applications in smart cities: A systematic survey. Wireless Networks, 26(2):1433–1457, 2020.

[4]. Gohar Rahman and Chai Wen Chuah. Fog computing, applications, security and challenges, review. Int.

[5]. J. Eng. Technol, 7(3):1615–1621, 2018.

[6]. Navjeet Kaur, Ashok Kumar, and Rajesh Kumar. A systematic review on task scheduling in fog com- puting: Taxonomy, tools, challenges, and future directions. Concurrency and Computation: Practice and Experience, 33(21):e6432, 2021.

[7]. Mir Salim Ul Islam and Ashok Kumar. Context-aware scheduling in fog computing: A survey, taxonomy, challenges and future directions. Journal of Network and Computer Applications, page 103008, 2021.

[8]. Prateeksha Varshney and Yogesh Simmhan. Demystifying fog computing: Characterizing architectures, applications and abstractions. In 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), pages 115–124. IEEE, 2017.

[9]. Karrar Hameed Abdulkareem, Mazin Abed Mohammed, Saraswathy Shamini Gunasekaran, Mo- hammed Nasser Al-Mhiqani, Ammar Awad Mutlag, Salama A Mostafa, Nabeel Salih Ali, and Dheyaa Ahmed Ibrahim. A review of fog computing and machine learning: concepts, applications, chal- lenges, and open issues. IEEE Access, 7:153123–153140, 2019.

[10]. Redowan Mahmud, Ramamohanarao Kotagiri, and Rajkumar Buyya. Fog computing: A taxonomy, survey and future directions. In Internet of everything, pages 103–130. Springer, 2018.

[11]. Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pages 13–16, 2012.

[12]. OpenFog Consortium et al. Openfog reference architecture for fog computing. Architecture Working Group, 2017.

[13]. IEEE Standard Association et al. Ieee 1934-2018-ieee standard for adoption of openfog reference architec- ture for fog computing, 2018.

[14]. Zijiang Hao, Ed Novak, Shanhe Yi, and Qun Li. Challenges and software architecture for fog computing.

[15]. IEEE Internet Computing, 21(2):44–53, 2017.

[16]. Nanxi Chen, Yang Yang, Tao Zhang, Ming-Tuo Zhou, Xiliang Luo, and John K Zao. Fog as a service technology. IEEE Communications Magazine, 56(11):95–101, 2018.

[17]. Juan Luo, Luxiu Yin, Jinyu Hu, Chun Wang, Xuan Liu, Xin Fan, and Haibo Luo. Container-based fog computing architecture and energy-balancing scheduling algorithm for energy iot. Future Generation Computer Systems, 97:50–60, 2019.

[18]. Ryuji Oma, Shigenari Nakamura, Dilawaer Duolikun, Tomoya Enokido, and Makoto Takizawa. An energy- efficient model for fog computing in the internet of things (iot). Internet of Things, 1:14–26, 2018.

[19]. Shreshth Tuli, Redowan Mahmud, Shikhar Tuli, and Rajkumar Buyya. Fogbus: A blockchain-based lightweight framework for edge and fog computing. Journal of Systems and Software, 2019.

[20]. Pradip Kumar Sharma, Mu-Yen Chen, and Jong Hyuk Park. A software defined fog node based distributed blockchain cloud architecture for iot. IEEE Access, 6:115–124, 2017.

[21]. Chii Chang, Satish Narayana Srirama, and Rajkumar Buyya. Indie fog: An efficient fog-computing infras- tructure for the internet of things. Computer, 50(9):92–98, 2017.

[22]. Luca Cerina, Sara Notargiacomo, Matteo GrecoLuca Paccanit, and Marco Domenico Santambrogio. A fog-computing architecture for preventive healthcare and assisted living in smart ambients. In 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), pages 1–6. IEEE, 2017.

[23]. Sadoon Azizi, Mohammad Shojafar, Jemal Abawajy, and Rajkumar Buyya. Deadline-aware and energy- efficient iot task scheduling in fog computing systems: A semi-greedy approach. Journal of Network and Computer Applications, page 103333, 2022.

[24]. Petar Krivic, Mario Kusek, Igor Cavrak, and Pavle Skocir. Dynamic scheduling of contextually categorised internet of things services in fog computing environment. Sensors, 22(2):465, 2022.

[25]. Saeed AL-Amodi, Sudhansu Shekhar Patra, Suman Bhattacharya, Jnyana Ranjan Mohanty, Vinay Kumar, and Rabindra Kumar Barik. Meta-heuristic algorithm for energy-efficient task scheduling in fog computing. In Recent Trends in Electronics and Communication, pages 915–925. Springer, 2022.

[26]. JJ Geetha, Jaya Lakshmi DS, and Keerthana Ningaraju LN. Consistent hashing and real-time task schedul- ing in fog computing. In Deep Learning Applications for Cyber-Physical Systems, pages 245–261. IGI Global, 2022.

[27]. Fatma M Talaat, Hesham A Ali, Mohamed S Saraya, and Ahmed I Saleh. Effective scheduling algorithm for load balancing in fog environment using cnn and mpso. Knowledge and Information Systems, pages 1–25, 2022.

[28]. Ahmed. A.A. Gad-Elrab and Amin. Y. Noaman. A two-tier bipartite graph task allocation approach based on fuzzy clustering in cloud–fog environment. Future Generation Computer Systems, 103:79–90, 2020.

[29]. Dimitrios Tychalas and Helen Karatza. A scheduling algorithm for a fog computing system with bag-of- tasks jobs: Simulation and performance evaluation. Simulation Modelling Practice and Theory, 98:101982, 2020.

[30]. Hamid Reza Boveiri, Raouf Khayami, Mohamed Elhoseny, and M Gunasekaran. An efficient swarm- intelligence approach for task scheduling in cloud-based internet of things applications. Journal of Ambient Intelligence and Humanized Computing, 10(9):3469–3479, 2019.

[31]. Simar Preet Singh, Anand Nayyar, Harpreet Kaur, and Ashu Singla. Dynamic task scheduling using balanced vm allocation policy for fog computing platforms. Scalable Computing: Practice and Experience, 20(2):433–456, 2019.

[32]. Binh Minh Nguyen, Huynh Thi Thanh Binh, and Bao Do Son. Evolutionary algorithms to optimize task scheduling problem for the iot based bag-of-tasks application in cloud–fog computing environment. Applied Sciences, 9(9):1730, 2019.

[33]. Juan Wang and Di Li. Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing. Sensors, 19(5):1023, 2019.

[34]. Georgios L Stavrinides and Helen D Karatza. A hybrid approach to scheduling real-time iot workflows in fog and cloud environments. Multimedia Tools and Applications, 78(17):24639–24655, 2019.

[35]. J Angela Jennifa Sujana, M Geethanjali, R Venitta Raj, and T Revathi. Trust model based scheduling of stochastic workflows in cloud and fog computing. In Cloud Computing for Geospatial Big Data Analytics, pages 29–54. Springer, 2019.

[36]. Nitin Auluck, Omer Rana, Surya Nepal, Andrew Jones, and Anil Singh. Scheduling real time security aware tasks in fog networks. IEEE Transactions on Services Computing, 2019.

[37]. Tejaswini Choudhari, Melody Moh, and Teng-Sheng Moh. Prioritized task scheduling in fog computing. In Proceedings of the ACMSE 2018 Conference, page 22. ACM, 2018.

[38]. Hsiang-Yi Wu and Che-Rung Lee. Energy efficient scheduling for heterogeneous fog computing architec- tures. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), volume 1, pages 555–560. IEEE, 2018.

[39]. Xuan-Qui Pham and Eui-Nam Huh. Towards task scheduling in a cloud-fog computing system. In 2016 18th Asia-Pacific network operations and management symposium (APNOMS), pages 1–4. IEEE, 2016.

[40]. Salim Bitam, Sherali Zeadally, and Abdelhamid Mellouk. Fog computing job scheduling optimization based on bees swarm. Enterprise Information Systems, 12(4):373–397, 2018.

[41]. Jiafu Wan, Baotong Chen, Shiyong Wang, Min Xia, Di Li, and Chengliang Liu. Fog computing for energy- aware load balancing and scheduling in smart factory. IEEE Transactions on Industrial Informatics, 14(10):4548–4556, 2018.

[42]. Doan Hoang and Thanh Dat Dang. Fbrc: Optimization of task scheduling in fog-based region and cloud. In 2017 IEEE Trustcom/BigDataSE/ICESS, pages 1109–1114. IEEE, 2017.

[43]. Guowei Zhang, Fei Shen, Yang Yang, Hua Qian, and Wei Yao. Fair task offloading among fog nodes in fog computing networks. In 2018 IEEE international conference on communications (ICC), pages 1–6. IEEE, 2018.

[44]. Yang Yang, Shuang Zhao, Wuxiong Zhang, Yu Chen, Xiliang Luo, and Jun Wang. Debts: Delay energy balanced task scheduling in homogeneous fog networks. IEEE Internet of Things Journal, 5(3):2094–2106, 2018.

[45]. Saqib Nazir, Sundas Shafiq, Zafar Iqbal, Muhammad Zeeshan, Subhan Tariq, and Nadeem Javaid. Cuckoo optimization algorithm based job scheduling using cloud and fog computing in smart grid. In International Conference on Intelligent Networking and Collaborative Systems, pages 34–46. Springer, 2018.

[46]. Zening Liu, Yang Yang, Ming-Tuo Zhou, and Ziqin Li. A unified cross-entropy based task scheduling algorithm for heterogeneous fog networks. In Proceedings of the 1st ACM International Workshop on Smart Cities and Fog Computing, pages 1–6. ACM, 2018.

[47]. David Perez Abreu, Karima Velasquez, Marcio Roberto Miranda Assis, Luiz Fernando Bittencourt, Marilia Curado, Edmundo Monteiro, and Edmundo Madeira. A rank scheduling mechanism for fog environments. In 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), pages 363–369. IEEE, 2018.

[48]. Qianyu Liu, Yunkai Wei, Supeng Leng, and Yijin Chen. Task scheduling in fog enabled internet of things for smart cities. In 2017 IEEE 17th International Conference on Communication Technology (ICCT), pages 975–980. IEEE, 2017.

[49]. Nadim T´ellez, Miguel Jimeno, Augusto Salazar, and E Nino-Ruiz. A tabu search method for load balancing in fog computing. Int. Artif. Intell, 16(2):1–31, 2018.

[50]. Zening Liu, Xiumei Yang, Yang Yang, Kunlun Wang, and Guoqiang Mao. Dats: Dispersive stable task scheduling in heterogeneous fog networks. IEEE Internet of Things Journal, 6(2):3423–3436, 2018.

[51]. Luiz F Bittencourt, Javier Diaz-Montes, Rajkumar Buyya, Omer F Rana, and Manish Parashar. Mobility- aware application scheduling in fog computing. IEEE Cloud Computing, 4(2):26–35, 2017.

[52]. Guowei Zhang, Fei Shen, Nanxi Chen, Pengcheng Zhu, Xuewu Dai, and Yang Yang. Dots: Delay-optimal task scheduling among voluntary nodes in fog networks. IEEE Internet of Things Journal, 6(2):3533–3544, 2018.

[53]. Kaneez Fizza, Nitin Auluck, Omer Rana, and Luiz Bittencourt. Pashe: Privacy aware scheduling in a heterogeneous fog environment. In 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), pages 333–340. IEEE, 2018.

[54]. Guowei Zhang, Fei Shen, Zening Liu, Yang Yang, Kunlun Wang, and Ming-Tuo Zhou. Femto: Fair and energy-minimized task offloading for fog-enabled iot networks. IEEE Internet of Things Journal, 2018.

[55]. Xuan-Qui Pham, Nguyen Doan Man, Nguyen Dao Tan Tri, Ngo Quang Thai, and Eui-Nam Huh. A cost-and performance-effective approach for task scheduling based on collaboration between cloud and fog computing. International Journal of Distributed Sensor Networks, 13(11):1550147717742073, 2017.

[56]. Yang Yang, Kunlun Wang, Guowei Zhang, Xu Chen, Xiliang Luo, and Ming-Tuo Zhou. Meets: Maximal energy efficient task scheduling in homogeneous fog networks. IEEE Internet of Things Journal, 5(5):4076– 4087, 2018.

[57]. Dadmehr Rahbari, Sabihe Kabirzadeh, and Mohsen Nickray. A security aware scheduling in fog computing by hyper heuristic algorithm. In 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS), pages 87–92. IEEE, 2017.

[58]. Wei-Chang Yeh, Chyh-Ming Lai, and Kuan-Cheng Tseng. Fog computing task scheduling optimization based on multi-objective simplified swarm optimization. In Journal of Physics: Conference Series, volume 1411, page 012007. IOP Publishing, 2019.

[59]. Ranesh Kumar Naha, Saurabh Garg, Andrew Chan, and Sudheer Kumar Battula. Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment. Future Generation Computer Systems, 104:131–141, 2020.

[60]. Tajul Islam and MMA Hashem. Task scheduling for big data management in fog infrastructure. In 2018 21st International Conference of Computer and Information Technology (ICCIT), pages 1–6. IEEE, 2018.

[61]. Kunlun Wang, Youyu Tan, Ziyu Shao, Song Ci, and Yang Yang. Learning-based task offloading for delay- sensitive applications in dynamic fog networks. IEEE Transactions on Vehicular Technology, 68(11):11399– 11403, 2019.

[62]. Paridhika Kayal. Kubernetes: Towards deployment of distributed iot applications in fog computing. Com- panion of the ACM/SPEC International Conference on Performance Engineering, pages 32–33, 2020.

[63]. Bin Cheng, Jonathan Fuerst, Gurkan Solmaz, and Takuya Sanada. Fog function: Serverless fog computing for data intensive iot services. 2019 IEEE International Conference on Services Computing (SCC), pages 28–35, 2019.

[64]. Mithun Mukherjee, Mian Guo, Jaime Lloret, Razi Iqbal, and Qi Zhang. Deadline-aware fair scheduling for offloaded tasks in fog computing with inter-fog dependency. IEEE Communications Letters, 2019.

[65]. Yabin Wang, Chenghao Guo, and Jin Yu. Immune scheduling network based method for task scheduling in decentralized fog computing. Wireless Communications and Mobile Computing, 2018, 2018.

[66]. Jiafu Wan, Baotong Chen, Shiyong Wang, Min Xia, Di Li, and Chengliang Liu. Fog computing for energy- aware load balancing and scheduling in smart factory. IEEE Transactions on Industrial Informatics, 14(10):4548–4556, 2018.

[67]. Peng Zhang, Joseph K Liu, F Richard Yu, Mehdi Sookhak, Man Ho Au, and Xiapu Luo. A survey on access control in fog computing. IEEE Communications Magazine, 56(2):144–149, 2018.

[68]. Cosimo Anglano, Massimo Canonico, and Marco Guazzone. Online user-driven task scheduling for fem- toclouds. In 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), pages 5–12. IEEE, 2019.

[69]. Guowei Zhang, Fei Shen, Yueyue Zhang, Rong Yang, Yang Yang, and Eduard A Jorswieck. Delay min- imized task scheduling in fog-enabled iot networks. In 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), pages 1–6. IEEE, 2018.

[70]. Opeyemi Osanaiye, Shuo Chen, Zheng Yan, Rongxing Lu, Kim-Kwang Raymond Choo, and Mqhele Dlodlo. From cloud to fog computing: A review and a conceptual live vm migration framework. IEEE Access, 5:8284–8300, 2017.

[71]. Sonja Filiposka, Anastas Mishev, and Katja Gilly. Community-based allocation and migration strategies for fog computing. IEEE Wireless Communications and Networking Conference (WCNC), pages 1–6, 2018.

[72]. Anil Singh and Nitin Auluck. Load balancing aware scheduling algorithms for fog networks. Software: Practice and Experience.