International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Unleashing the Power of QoS: A Comprehensive Study and Evaluation of Services-based Scheduling Techniques for Fog Computing

[1]Meena Rani, [2]Kalpna Guleria[*], [3]Surya Narayan Panda

Submitted: 05/09/2023 **Revised**: 21/10/2023 **Accepted**: 06/11/2023

Abstract: Fog computing, a special worldview, has become famous for applications or technology that should be area and dormancy delicate. It is a powerful expansion to distributed computing that makes it conceivable to offer assets and administrations near-end gadgets that are not in the cloud. The presence of various heterogeneous, possibly cell phones in a fog system raised worries about QoS. A few QoS factors are considered, and QoS-aware methodologies are introduced in different pieces of the fog system. Despite the implication of quality of service in fog computing, there is currently no comprehensive focus on QoS-aware techniques. Subsequently, this study looks at ongoing discoveries that have utilized reliable energy to guarantee QoS in fog computing. To enhance the technological capabilities with the presentation of the IoT worldview, various computing parts require various alterations to help the QoS. Continuous reaction to time-delicate positions is advanced by the Nature of the Administration point of support. Any QoS boundaries ought to be eliminated and managed to enhance the quality of life of a human being. Fog computing was acquainted in 2012 with further developing QoS in existing systems with an end goal to address QoS issues welcomed on by utilizing distributed computing alone. Improving QoS is currently the principal accentuation or innovation of fog computing. Hence, the fundamental target of this study is to audit and survey the writing on the endeavors made to improve different QoS parts.

Keywords: *Fog Computing, Resource use efficiency, Quality of Service, Scheduling Algorithms, IoT (Internet of Things)*

1. Introduction

The location-aware software, the Internet of Everything (IoE), and the Internet of Things are being used more frequently, which has sped up the development of the aforementioned data and applications. For example, Forbes anticipated in 2015 that 25 trillion contraptions, sensors, and chips will be used in the following five years, and how much-created information would add up to 50 trillion gigabytes. These applications require extra particulars notwithstanding area awareness and application awareness, like defer awareness, elevated degrees of safety, and confirmation (Bonomi et al. 2012). Because of the centralization of gatherings and the huge span between them, the distributed computing model experiences enormous idleness in information movement and response time even though it has a high handling and stockpiling limit. Albeit the organization's data transfer capacity has fundamentally expanded, the handling power has not kept pace. The transmission capacity is in this manner seen as a bottleneck in distributed computing. Conventional registering strategies are in this way unfit to fulfill these necessities. The nearby handling has been involved more because of the expanded handling force of brilliant gadgets and simultaneously falling expenses of computer systems, giving far to satisfy these requirements (Li et al. 2020).

Making estimations locally rather than sending them to the cloud is the main arrangement. If fundamental, certain information will be moved to the cloud, where handling will occur. Cloudlets have been proposed in writing as an answer to this. When combined with the best offloading techniques, these cloudlets could prompt diminished communication and handling costs. The greatest inconvenience of Cloudlets is that they have a more modest inclusion region since they exclusively use Wi-Fi passageways. This computing model couldn't consequently uphold ubiquitous computing. Furthermore, in contrast with cloud computing, its asset requirements limit the accessibility of administrations and assets. The Mobile Cloud Computing (MCC) plan has been advanced to offer cell phone clients every one of the advantages of cloud computing (Pereira et al. 2019).

Fog computing consolidates gadgets at the organization's and server farms, offering a successful method for getting past the limitations illustrated previously. The organization edge's numerous heterogeneous gadgets are connected together in this dispersed computing model to offer adaptable computing, correspondence, and capacity abilities. The neighbourhood asset pool near end clients is utilized for related processes in fog computing, which diminishes how much information is moved and the time it takes to move it. Fog computing isn't a swap for cloud computing, yet rather a strong expansion that empowers handling at the edge and

[1]*Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India*
E-mail: meena.rani@chitkara.edu.in
[2]*Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India*
[*]*Corresponding Author: guleria.kalpna@gmail.com*
[3]*Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India*
E-mail: snpanda@chitkara.edu.in"

cloud communication. It additionally makes it workable for the mentioned administrations and applications to run on equipment like entryways, switches, passages, vehicular-to-vehicular doors, Street Side Units, and set-top boxes (Das and Inuwa 2023). Fog computing assists with eliminating energy utilization. Furthermore, minimal expense, somewhat strong handling gadgets can be utilized in closeness and with high dependability to reduce expenses and computing delays and streamline processes by understanding the way that specific applications work. In fog computing, we maximize the utilization of the capacities and assets present in nearby gadgets; be that as it may, when proficient assets are absent, we are compelled to utilize more costly cloud assets. Further using nearby assets will bring about lower costs, more limited stand by times, more prominent privacy and security, less organization traffic, and a greener computing method (Jamil et al. 2020).

While suggesting new designs, it's critical to design out the undertakings so they can be finished speedily and with the current most ideal utilization of the assets. The most effective method to oversee task execution, or to pick which occupations are done in the end-client layer, fog layer, along with cloud layer, is a vital test that emerges in cloud-fog frameworks. To fabricate novel applications like thoughts for checking electronic wellbeing, savvy homes, and brilliant traffic frameworks as well as decrease expenses and handling/correspondence delays, work in these models should be booked appropriately (Uppal et al. 2021). The fundamental objective of assignment booking is to accelerate execution and abbreviate task execution times. Such booking plans should be utilized by cloud specialist co-op associations to fulfil client needs and lift asset adequacy. The presentation of such successful booking strategies might prompt the development of green computing (Faticanti et al. 2020). A new worldview called FC has been put on a mission to help distributed computing. FC makes sense in this part. Also, QoS and its connected elements are presented as huge FC points.

2. Fog Computing

A distributed worldview called FC (Fog Computing) is utilized to get cloud-like services at the organization's edge. The expression "fog" in FC conveys similitudes to fog in reality. Fog happens among mists and the earth in the real world, and FC uses a similar thought. As such, fog hubs are set up between the end client's gadgets and the cloud (Fu, Liu, and Srivastava 2019). Notwithstanding, there are a few distinctions between the fog and cloud paradigms, despite the fact that the two of them offer services like controlling, registering, imparting, and putting away information that are practically indistinguishable (Jamil et al. 2020). A fog hub is located in a specific region of the world. Fog is likewise especially helpful for applications that demand a continuous response and negligible dormancy. Then again, the cloud is unified and regularly remote from clients, which has an effect on execution for applications that rely on area and inertness. Similar to distributed computing, FC can make use of the following service models: SaaS (software as a service), PaaS (platform as a service), and IaaS (infrastructure as a service). FC differs from other computer paradigms thanks to the attributes. Some of the key features are:

- Since fog nodes understand their logical place in the system and the expenses of connecting with other nodes, FC provides the lowest latency. Since fog nodes are located in close proximity to end devices, data produced by these devices may be processed and responses sent considerably more rapidly than with a cloud service.
- Applications as well as services in a distributed geographic area are generally candidates for FC rather than the unified cloud.
- FC offers the capacity to keep up with and process different types of information by using different types of organisational correspondence skills.
- Supplier collaboration is expected to help explicit administrations like continuous real time features. Along these lines, administrations should be united across all spaces, and FC parts should have the option to help out each other.
- As opposed to batch processing, in real time interactions, fog applications are used.

In recent times cloud computing capabilities have been extended to edge networks with the help of fog computing which has proven itself to be a promising paradigm in this field. It is done by bringing computational resources nearby the source of data which reduces the overall latency rate and helps the data to be processed in real-time with the help of cloud computing. These features make fog computing suitable in being used for applications for example IoT, edge analytics, as well as mission-critical systems. However, in the scenario of distributed and dynamic computing networks, effective utilization of fog resources and provision of guarantees of Quality of Service (QoS) still remains a big challenge (Fu, Liu, and Srivastava 2019) (Wang et al. 2019). In fog computing systems , optimum allocation of resources and timely execution of tasks heavily relies upon scheduling. The scheduling techniques which are based on the quality of service try to address these challenges while considering various QoS metrics for example utilization of network bandwidth, efficiency of energy being used, response timing, and reliability (Fu, Liu, and Srivastava 2019). These methods aid in the effective allocation of resources to satisfy the needs of the user and their applications with regard to quality of service. To enhance the scheduling technology for fog computing, a detailed evaluation of the present QoS-based scheduling techniques is required. This review paper's goal is to give a comprehensive overview of the most current research and advancements in this area. By examining the benefits, drawbacks, and developments of different approaches, we hope to identify gaps and potential for more study and innovation (Cheng, Lim, and Hui 2022).

Three objectives are set forward in this review study. We will start by doing a thorough literature study to find and categorize the QoS-based scheduling approaches suggested for fog computing (Yoghourdjian et al. 2021). We will also look at and analyse these practices in light of several standards, such as their underlying algorithms for scheduling, metrics of QoS that need consideration, strategy for resource allocation, and assessment techniques. This examination will highlight the merits and drawbacks of various approaches as well as their suitability for various scenarios based on fog computing. Finally, we will discuss the knowledge gaps and potential future directions for QoS-based scheduling techniques used in fog computing. This will consist of finding out various novel algorithms while considering the heterogeneous, heuristics-based and dynamic nature of the fog environment that incorporates machine learning, and various techniques for optimization while addressing challenges associated with reliability and scalability (Khattar, Sidhu, and Singh 2019). By performing this thorough analysis, we intend to offer academics, practitioners, and system designer's helpful insights into the present status of QoS-based scheduling strategies for fog computing. The goal of this study is to facilitate the development of efficient, trustworthy, and scalable fog computing systems that can meet the diverse QoS demands of new applications. In the sections that follow, we will look at how existing fog computing QoS-based scheduling techniques are categorized, analysed, and compared. Finally, by outlining their salient features, benefits, and drawbacks, we will open the door for further developments in this quickly evolving sector. With the growing use of fog computing, more efficient and effective scheduling techniques are required. Fog computing's effectiveness depends on being able to allocate resources as efficiently as possible in light of the special qualities and difficulties of the fog environment. These challenges include the necessity for real-time processing, device heterogeneity, changeable network circumstances, and computing power limitations (Fu, Liu, and Srivastava 2019).

We will examine and evaluate a range of QoS-based scheduling techniques suggested for fog computing. The evaluation will take into account performance indicators such as resource allocation tactics, scalability, flexibility, and trade-offs between different QoS criteria (Wang et al. 2019). By carefully analysing these strategies, we seek to highlight their benefits, shortcomings, and possible areas for development. Based on their methods, algorithms, and objectives, the existing QoS-based scheduling systems will be categorized into pertinent categories for evaluation. We will look at a variety of scheduling strategies, including task offloading, load balancing, priority-based scheduling, and deadline-driven scheduling (Fu, Liu, and Srivastava 2019). We will also look into QoS parameters, which are used in scheduling algorithms. Response time, throughput, reliability, energy efficiency, and resource use are a few examples of these measures. Understanding the effects and trade-offs of various QoS measures will be extremely beneficial for fog computing systems that satisfy the unique requirements of varied applications. We will take into account various evaluation methods used in the writing, such as simulation-based investigations, conduct comparative analyses, experimental evaluation, and mathematical modelling, to ensure a thorough examination. This will allow us to test the performance and efficiency of scheduling algorithms for several scenarios and settings (Seth et al. 2022). By reviewing the current literature, we will also pinpoint typical issues and knowledge gaps in QoS-based scheduling for fog computing. These flaws might include issues associated with scalability, adaptation to dynamic networking conditions, and dealing with heterogeneity in fog devices while considering various privacy and security concerns and incorporating various techniques for optimization and machine learning in the process of scheduling. By emphasizing these gaps, we aim to stimulate more research to solve these problems and improve the discipline. The findings of this paper will serve as practical guidance for system designers, scientists, and engineers in addition to advancing our understanding of QoS-based scheduling in the case of fog computing. By enabling more informed choices, the knowledge obtained from this study will help in the selection of suitable scheduling techniques and the optimization of fog computing systems for certain application areas (Kimani, Oduol, and Langat 2019).

We shall examine the categorization, analysis, and comparison of already existing QoS-based scheduling techniques used in fog computing cases in subsequent sections of this paper. Each method's major traits, advantages, disadvantages, and evaluation findings will be presented in depth. By conducting such a thorough investigation, we seek to illuminate the existing landscape of QoS-based booking for cloudy processing and pave the way for future developments in this vital and quickly growing field.

2.1 Background

A common pool of reconfigurable computing assets, which includes services, servers, storage, applications, and networks can be immediately provisioned as well as delivered with service provider interaction or through little effort by management, as per the NIST "National Institute of Standards and Technology". Cisco guarantees that fog computing is an improvement in cloud computing in that it offers administration from the centre to the fringe of the organization. As per (Bonomi et al. 2012), cloud computing offers a virtualized environment for putting away information, handling it, and offering network administration among different cloud-based servers and gadgets.

Conversely, fog computing is known to utilize countless heterogeneous and fluctuating confined gadgets that connect and cooperate by means of an organization to store and handle information without the assistance of any external association. Just key low-level organization capabilities as well as primary applications which work in an enclosed climate can be upheld by the errands framed above, for example, putting away and handling (Jamil et al. 2020). Fog nodes quickly convey assets and administrations, i.e., they work at the organization's edge. Fog computing utilizes two gadgets with high limits, for example, those utilized in cloudlet and IOx, and gadgets with low limits, for example, set-top boxes, passageways, switches, switches, and base stations. Notwithstanding having a tremendous limit, fog nodes process information and deal with administrations at the organization's edge (Fu, Liu, and Srivastava 2019). As opposed to regular cloud computing stages, fog computing offers a virtual climate for handling, putting away, and conveying network administration between different gadgets and server farms. Both cloud and fog computing depend vigorously on handling, putting away, and organizing administrations.

2.2 Fog Architecture

A various levelled FC design is displayed in Figure 1. Three layers make up this design: cloud, fog, and user device layer (Fu, Liu, and Srivastava 2019):

a) **Cloud Layer:** This layer has strong storage as well as computing servers that, when required, can store huge amounts of information and do top to bottom PC examinations for different application administrations.

b) **Fog Layer:** The fog layer is present at the edge of the network. A thick network of physical or virtual fog nodes makes up a fog layer. These fog nodes are generally scattered among end gadgets and mists. They might move around or stay fixed. Fog nodes are prepared to process, send, and briefly store information. Constant application examination and administration conveyance occur in the fog layer. Moreover, fog nodes can possibly associate and interface with each other.

c) **User Device Layer:** It incorporates IoT gadgets and sensors and is the layer that is nearest to the end client and the actual climate. Every sensor or end gadget in this design is associated with a fog node utilizing a link association or a wireless innovation like Wi- Fi, 3G, 4G, Bluetooth, wireless neighbourhood, or ZigBee. Fog nodes can likewise communicate with each other wirelessly or over wires.

2.3 Fog Nodes

The key component of fog engineering is the fog hub. Servers, switches, passages, and switches are instances of actual fog nodes. Virtual fog nodes incorporate virtualized switches and virtual machines. These nodes can trade data and offer their computing power with end gadgets. The geological dispersion and legitimate place of a fog hub inside the bunching zone are both known to it. On the off chance that it is essential, fog nodes can interface end gadgets to cloud computing assets or give information to the board, or provide correspondence services between end gadgets and FC services. Fog nodes in FC can run as unified or decentralized frameworks.

All in all, fog nodes are united in a bunch or can run independently and speak with different nodes on a case-by-case basis. Fog nodes should have the option to deal with at least one of the accompanying elements to utilize FC abilities:

a) The ability to act autonomously and as indicated by nearby judgement at the node or node cluster level.

b) The presence of numerous components and the limits with regards to development in different extensive environments.

c) The capacity to deal with progressive designs with different layer arrangements to deliver different assistance capability assortments.

d) The capacity for complex frameworks to naturally oversee and arrange themselves.

e) The capacity to be modified at different levels by different partners, like end clients, network administrators, or gear providers.

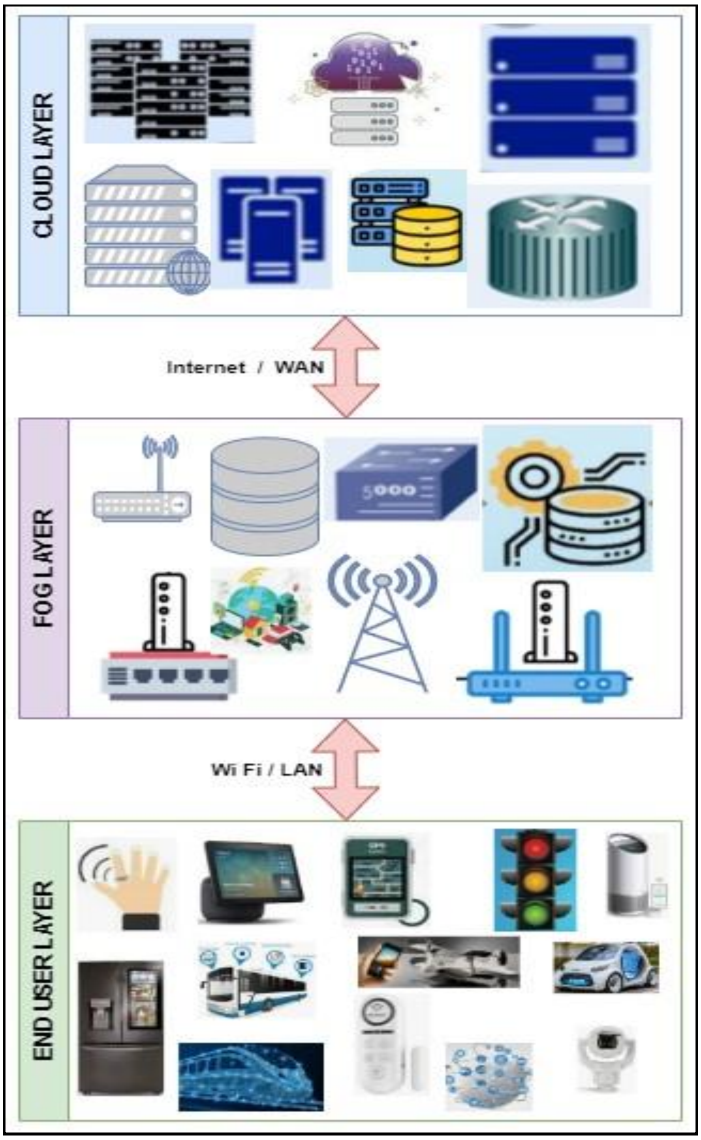


Fig 1. "The Hierarchical Architecture of Fog Computing"

2.4 Quality of Services in Fog Computing

Fog computing, or computing performed at the organization's edge, might provide new services and applications that decrease communication wait times. Therefore, higher QoS is given by means of fog computing. It is separated into four properties, like network, unwavering quality, limit, and postponement, and should be accomplished while offering fog administrations. The provision of sufficient QoS is a crucial concern in FC. FC's unique properties—including its heterogeneity, mobility, and distribution—mean that not all of cloud computing's QoS provisions apply to it. Supporting real-time applications is one of the main goals of fog design. For a successful system design, fog-based systems take into account a number of QoS parameters. Fourteen categories of QoS variables are defined in the literature (security, scalability, availability, connectivity, deadline, reliability, capacity, response time energy consumption, execution time, delay, cost, resource utilization, , and throughput) (Jamil et al. 2020).

a) **Connectivity:** In a different or heterogeneous highlights like organization handing-off, division, and bunching present new open doors that consider cost reserve funds, information decrease, and expanded association (Li et al. 2020).

b) **Reliability:** The limit of a framework's ability to perform its important capabilities under foreordained conditions and at a foreordained time. Fog computing's necessities for unwavering quality are equivalent to those of cloud, matrix, and group computing. The errand that fizzled can be rescheduled, the hubs can be recreated to execute the activity in equal measure, or the unwavering quality can be expanded by checking the time it takes to continue the undertaking after disappointment consistently (Pereira et al. 2019). Unfortunately, fog computing isn't fit for rescheduling and checking occasionally for inability to continue the undertaking in light of the fact that both reason idleness and the failure to adjust to changes. Replication may sometimes succeed, however at least two Fog hubs should coordinate for this to occur.

c) **Capacity:** The two fundamental factors that should be considered while deciding the limit are network data transmission and capacity limit. Knowing where the information is arranged inside the fog network is critical for accomplishing these two highlights; thus, information limitation is vital. This turns into a critical obstruction for fog computing. To moderate organization and limit the inactivity, the reserve ought to be reconfigured to utilize impermanent areas and bigger inclusion (Das and Inuwa 2023).

d) **Delay:** Applications that require fog computing that are dormancy delicate, like streaming, mining, and occasion handling, can't utilise group handling, but rather should stream continuously. Fleeting occasion handling can be utilised to forestall dormancy issues. As per Hong, the framework expects shoppers who frequently utilise the application's inquiry locales and performs occasion handling to make the information available when the clients search for it. To reuse calculations and save assets, an alternate handling worldview known as RECEP exploits information cross-over and mistaken results. It further develops the versatile CEP framework's adaptability and diminishes dormancy (Jamil et al. 2020).

e) **Throughput:** Throughput refers to the amount of data that may be sent and received in a given amount of time. Throughput is the rate at which messages are delivered successfully to their destination on average. In contrast to estimating the likelihood of packet delivery, throughput provides a relevant assessment of the rate at which packets are actually delivered. The maximum rate of desired help that the system can accommodate (Faticanti et al. 2020).

f) **Deadline:** The last conceivable time for a solicitation to be satisfied. Deadline refers to the predetermined time limit within which a task or activity must be completed. It is a crucial parameter in scheduling and resource allocation, ensuring timely execution and delivery of services. By adhering to deadlines, efficiency, productivity, and customer satisfaction can be enhanced.

g) **Response Time:** The time it takes for the computer to execute the query and deliver back the result to the terminal is included into the total response time. Response time is a popular statistic used to evaluate the efficiency of an interactive system. The amount of time that passes between when a customer makes a request and when they hear back from the company.(Fu, Liu, and Srivastava 2019).

h) **Resource Utilisation:** Resource allocation is the process of choosing the appropriate resources for various tasks, while resource utilisation examines the effectiveness of your resources. The best utilisation of a framework's assets as they are currently open (Wang et al. 2019).

i) **Cost:** By considering the cost implications, QoS scheduling techniques strive to optimize the allocation of resources, minimize communication expenses, and make informed data storage decisions to achieve desired QoS levels at a reasonable cost.

j) **Execution Time:** The amount of time that has passed between the first processor starting an execution and the final processor finishing it is referred to as the execution time of a parallel programme. A processor is either processing or communicating while it is being used. How much time it takes for a programme to wrap up (Fu, Liu, and Srivastava 2019).

k) **Energy Consumption:** The total energy utilized by an asset to offer support. By integrating IoT devices with smart grids and energy distribution systems, energy consumption can be optimized to minimize peak loads, reduce wastage, and improve overall energy efficiency. (Cheng, Lim, and Hui 2022).

l) **Availability:** The capacity of a framework to ensure that the mentioned assets are accessible and proceeding as expected and this feature enhances the accuracy, reliability, and usability(Cheng, Lim, and Hui 2022).

m) **Scalability:** A computing framework has the ability to keep up with framework execution in any event when the volume of administration solicitations or asset applications develops (Yoghourdjian et al. 2021).

n) **Security:** Secure strategies are utilized to safeguard accessible information in the fog or cloud climate and this is providing Authentication and Access Control, Data Encryption, and Intrusion Detection and Prevention Systems. (Kimani, Oduol, and Langat 2019).

QoS (Quality of Service) refers to the presentation characteristics and guarantees provided for satisfying the presumptions of users and applications. In fog computing case, QoS has a significant role to play in ensuring the competent and reliable delivery of services, information management, and resource utilization. The phrase QoS refers to a number of metrics and factors that are crucial for assessing and maintaining the operation of fog computing system (Guleria et al. 2021). Depending on the needs of the programme as well as user expectations, these metrics may change. Achieving optimal performance usually requires making trade-offs between different metrics depending on the specific requirements of the fog computing environment and applications due to the interrelated nature of these QoS factors. To maximize performance and user experience in fog computing systems, effective scheduling algorithms that prioritize and optimize these QoS factors must be developed and put into action (Bhatt and Bhensdadia 2017). When evaluating scheduling algorithms in the fog computing environment in relation to Quality of Service (QoS), there are a lot of variables and metrics to consider. Table 1 lists the thorough analysis of fog computing scheduling methods based on how they affect different QoS parameters.

TABLE 1. FOG COMPUTING PARAMETERS AND SCHEDULING METHODS

Response Time	
Robin Round Scheduling	Response time is not taken into account, and it may vary depending on the sequence in which activities are carried out.
Priority-based scheduling	Reaction times for routine tasks can be sped up by allocating more need to time-sensitive tasks.
Deadline-driven scheduling	Setting priorities for projects with shorter deadlines will speed up responses to time-sensitive assignments.
Load balancing	Effective load balancing may spread the work and speed up reaction time by preventing certain devices from over-burden.
Task offloading	Delegating activities to more capable resources can reduce response times, especially for computationally heavy jobs.
Reliability	
Round-Robin Scheduling	In the case of Round Robin scheduling reliability is not considered explicitly.
Priority-based scheduling	Reliability can be enhanced by assigning high priority to critical tasks which may ensure that they are executed timely.
Deadline-driven Scheduling	Meeting task deadlines improves reliability, especially for applications that depend on time.
Load Balancing	Two possible advantages of load balancing are the prevention of device overload and system dependability.
Task Offloading	Task offloading might increase reliability by utilizing even more impressive and reliable fog or cloud resources.
Throughput	
Round-Robin Scheduling	While throughput may be balanced between the tasks, there is no specific method for increasing throughput overall.
Priority-based scheduling	More resources are allocated to high-priority task fostering unwavering quality and improves throughput of a system.
Deadline-driven Scheduling	Creating a schedule based on due deadlines allows for the timely completion of tasks, which boosts throughput.
Load Balancing	Good load balancing improves system performance by maximizing resource utilization.
Task Offloading	Transferring tasks to more capable resources can increase overall system throughput [50].
Energy Efficiency	
Round-Robin Scheduling	Round Robin scheduling does not specifically take energy efficiency into account.
Priority-based scheduling	Concentrating on tasks might not automatically increase energy efficiency unless lower priority tasks are delayed lowering the energy loss.
Deadline-driven Scheduling	Meeting deadlines successfully might indirectly increase energy efficiency by decreasing the amount of work that is done that is not essential.

Load Balancing	By distributing tasks evenly and preventing device overload, load balancing can help reduce energy consumption.
Task Offloading	Delegating tasks to resources that utilize less energy allows devices with limited resources to conserve energy.
Utilization of Network Bandwidth	
Round-Robin Scheduling	Round Robin scheduling does not specifically consider network bandwidth utilization.
Priority-based scheduling	Unless bandwidth allocation is expressly regulated, network bandwidth utilization may not be directly optimized through priority-based scheduling.
Deadline-driven Scheduling	Successfully meeting task deadlines can indirectly optimize network bandwidth utilization by lowering the amount of needless data transmissions.
Load Balancing	To efficiently utilize network bandwidth, load balancing algorithms might consider how much of it is being used while distributing jobs.
Task Offloading	To ensure optimal resource utilization, decisions on task offloading may take into account the efficacy and availability of network bandwidth.
Security	
Round-Robin Scheduling	Security issues cannot be particularly handled by round-robin scheduling.
Priority-based scheduling	Security can be increased by devoting more reliable or secure resources to higher-priority jobs.
Deadline-driven Scheduling	Meeting deadlines successfully might indirectly improve security by minimizing the amount of time that crucial tasks are exposed to security hazards.
Load Balancing	Security considerations can be included in load balancing algorithms by separating important processes from unreliable equipment or putting crucial workloads on more secure hardware.
Task Offloading	It allows for the consideration of security issues, ensuring that sensitive tasks are delegated to reliable and secure resources.

The fog computing environment, workload characteristics, and system settings may all affect how effectively scheduling algorithms function in regard to certain QoS criteria. An in-depth research and assessment of these algorithms can offer helpful insights for choosing the scheduling algorithm that is most appropriate for a given scenario, taking into account the specific QoS needs of the applications and the limitations of the fog computing environment.

3. Literature Review

Over the years, to improve human life quality, many IoT applications have been developed (Ghobaei-Arani, Souri, and Rahmanian 2020). These IoT applications generate large volumes of data using actuators, sensors and mobile devices. Traditional cloud computing has issues like latency, and network bandwidth which does not make it suitable for latency-sensitive IoT applications (Souri, Asghari, and Rezaei 2017). Cisco created a new distributed computing architecture in 2012 called Fog Computing to address this problem; it serves as a bridge between the cloud and the devices and sensors that make up the Internet of Things and is designed to meet the needs of latency-sensitive applications. Industry and academics alike have embraced fog computing as a means to satisfy the processing needs of Internet of Things devices and sensors (Miah, Schukat, and Barrett 2018). Fog computing supports location-awareness, mobility (Jo and Kim 2019), content-awareness, heterogeneity, scalability to the IoT devices/sensors to satisfy latency-sensitive applications requirements (Kertesz, Pflanzner, and Gyimothy 2019), which includes routers, switches, gateways, proxy servers (Miah, Schukat, and Barrett 2018). Fog computing is highly heterogeneous and dynamic. Therefore, an efficient resource management technique to manage these resource-constrained devices has to be proposed (Deng et al. 2016).

3.1 Scheduling Objectives

Scheduling is the process of finding the best solution for scheduling some set of processes or tasks across some machines. For successfully solving the scheduling problems, scheduling parameters proves to be very efficient. Depending on the service approach, there are two groups of scheduling parameters: service providers and consumer services (Haghi Kashani, Rahmani, and Jafari Navimipour 2020).

3.2 QoS-aware Scheduling in a Fog-Cloud Environment

(Bitam, Zeadally, and Mellouk 2018) proposed the Bees Life Algorithm, a bio-inspired optimization technique, to address the challenge of work scheduling in a fog computing environment (BLA).The methodology for handling users' inordinate demands for computing resources is predicated on optimizing the allocation of a set of jobs across fog nodes. This method also aims to reduce CPU execution time as well as energy consumption. Furthermore, for the optimum utilization of resources, three approaches are proposed by the (Intharawijitr, Iida, and Koga 2016) in a different work. In the first approach, fog nodes are chosen at random to carry out various tasks as they arrive. In the second approach, the authors concentrated on the fog devices with the smallest lag time. Last but not least, the third approach prioritise the fog resources that provide the maximum available capacity. After that, a mathematical model is used for comparing all the three approaches. To improve the user experience by minimizing the tasks' overall completion time, a joint image placement algorithm and optimization task scheduling is introduced by the (Zeng et al. 2016). Workload balancing between computational servers and client devices is the first sub-problem that was studied. The authors also investigated a second sub-problem involving the distribution of task images among storage servers, and a third involving the distribution of I/O interrupt requests between storage servers. They analyse how application performance is impacted by the user mobility as well as in the fog computing process they discussed application scheduling. Various scheduling policies have been implemented, and depending on the kind of application request, the cloud or cloudlets may be used for its execution. Concurrent policy is one type of policy, and it applies to all requests that are either processed by the cloudlet or assigned to it without regard to consumption statistics. FCFS (First Come First Server) follows the traditional model of processing requests in the order in which they were received up until all available resources were used up. The third strategy prioritised requests based on their tolerance for delay, with the least urgent requests being processed first and is known as Delay-priority policy. Also, in a fog-cloud environment, a workload allocation framework for balancing the power consumption and computational latency was proposed.

(Qu et al. 2020) performed a comprehensive survey of energy saving and QoS optimization in IoT models, fog computing, edge computing, along with cloud computing. These authors reviewed five primary issues and with the help of existing work proposed solutions were analysed. The authors focused on the issues related to QoS as well as resource management. They also concluded that for above-mentioned issues the efficient solution was VM management and reasonable resource scheduling, respectively. Managing the scarce resources in the fog layer of a fog computing system is a topic that (Apat et al., 2020) looked into. The importance and features of the fog orchestration node in the generic three-tier architecture have been discussed; this node is primarily built for the efficient distribution of workloads, taking into account factors like the number of required resources, the number of virtual machines, and so on, so that they can be run in the unpredictable IoT setting. Further, they also discussed a few methodologies for improving QoS resource management. Also, they introduced resource management challenges which should be considered for providing improved QoS to the customers. Task scheduling is investigated in depth in a Fog computing setting (Murtaza et al. 2020). To reduce power consumption (from fog devices) and increase quality of service, the authors suggest LRFC (Learning Repository Fog-Cloud), an intelligent and adaptable job scheduling method (which includes processing time and response time of tuples).Also, a smart soft layer between Fog nodes and IoE/IoT-devices was propose by the authors which can be used for implementing different policies based on learning. It has been concluded that scalability is exhibited by the proposed deployment model and therefore, performance bottlenecks are avoided. (Badotra and Panda 2021) for a dynamic and quick resource allocation, authors, in real time fog environment, had implemented a QTCS model. Authors performed a thorough survey and concluded that during resource allocation, improved QoS parameters are needed. The authors have evaluated QoS metrics for example the energy consumption, response time, and resource allocation in simulation results of their proposed model. According to their experiment's results, they concluded that dynamic resource allocations can be done efficiently with their proposed method, even if there are high numbers of user requests. QoS criteria, such as energy use and response time, also benefit from the QTCS method's enhancements.

In the fog computing, (Singhrova and Anu 2022) represented the resource allocation's systematic literature analysis in this paper. In a fog computing environment resource allocation's current status is considered to have various categories which includes meta-heuristic techniques, heuristics techniques and auction-based techniques, etc. The authors in this paper presented resource allocation's methodological analysis that depends on the meta-heuristic approaches. In this paper, (Alaghbari et al. 2022) investigated fog task scheduling's various challenges, and issues are along with the research done by other authors. In addition to this, fog computing environment authors have proposed a new task scheduling model. With the help of this model paper, task scheduling's various aspects were also solved. For a fog computing environment, the proposed task scheduling model mainly focused on task scheduling's significant aspects such as optimal task resources, fog node capacities, task priorities, and security. The primary aim is achieving energy and minimum delay. (Kaur, Kumar, and Kumar 2021) in their paper investigated the fog computing environment's various challenge's existing solutions. A meta-analysis is presented by authors on QoS parameters as well as tools used for Fog task scheduling algorithms' implementation. The systematic review presented by the authors is able to help others in identifying research issues so that scheduling efficiency can be enhanced in the future. According to the authors, in task scheduling the most focused parameters are load balancing, energy, cost, and time, whereas the least focused parameters are mobility, security, and optimal resource searching. Authors have also mentioned that in comparison to the real-time dataset, the synthetic dataset is used by most of the authors. In addition to this, the authors also stated that CloudSim and iFogSim simulators are used extensively for fog computing.

(Arikumar and Natarajan 2020) in his paper has proposed a framework for Fog based IoT environment which can help in minimizing the consumption of energy while ensuring QoS. In this framework use of both centralized as well as distributed allocation of resources is recommended in a hybrid manner. (Dash et al. 2019) offer an architecture that combines fog computing with edge caching to enhance QoS-aware video streaming. A QoS-aware architecture for fog computing that is suitable for IoT in industry has been proposed (Singh, Singh, and Gill 2021). Quality of Service (QoS) in IoT-based industrial applications is promised by this architecture's mix of fog nodes and cloud servers. Allotment of resources, security, privacy, load balancing, and scheduling are just few of the many QoS-aware fog computing concerns that (Shukla et al. 2019) have attempted to examine. For optimal scheduling in fog nodes, a QoS-aware framework using a genetic algorithm has been developed (Naik et al., 2021). Using a reinforcement learning strategy in fog computing, they offer a system for load balancing between fog nodes (Maiti et al., 2019). (Maiti et al. 2019) have tried to introduce a fault tolerance scheme where a combination of migration and replication will be used to ensure that application keeps on running even in the case of event failures in fog computing. Authors have proposed a security framework for QoS-aware fog computing where fog nodes and application will be protected by a combination of authentication, encryption, and authorization. (Rani, Guleria, and Panda 2021) in their paper have proposed a framework for preserving the privacy of QoS-aware fog computing applications with the help of various methods such as encryption, anonymization, and using differential privacy to protect privacy of the data of the users. (Yassine et al. 2019) in their paper, have talked about how in a fog or cloud-based IoT systems a theoretical model of the middleware layer can be used to establish efficient and adaptive scheduling between fog nodes and edges.

In their study, (Huang et al. 2023), the authors offer an open-source, QoS-aware IoT service placement technique for fog computing. This method considers not only the resources at hand but also the capabilities of fog nodes and the Quality of Service requirements of IoT applications. In their research, (He et al. 2018), the authors suggest a fog service orchestrator (Q-FSO) to facilitate the distribution of IIoT software. To guarantee that IIoT application QoS requirements are satisfied, Q-FSO is known to use a two-level QoS architecture. A broker management system applicable to fog, cloud, and IoT contexts has been presented in a recent work (Abdel-basset et al., 2021). This broker management system was created with the aim of improving quality of service in fog computing use cases. To do this, proper administration of cloud resources and fog nodes is required. In their study (Rohinidevi et al. 2022) the authors propose enhancing the learning-based QoS-aware scheduling framework as a means of enhancing the quality of service (QoS) of applications running on fog computing infrastructure. It indicates that this framework may be used to sustain the quality of service (QoS) of applications despite the dynamic changes in the fog-based environment. (Sodhro et al. 2018) have suggested that the QoS fog computing application can be effectively improved using a game theory-based QoS aware scheduling framework designed for fog computing. It claims that the framework can help in the participation of fog nodes in the process of scheduling and are able to guarantee QoS. (Islam, Kumar, and Hu 2021) in their paper suggest efficient improvement of QoS in fog computing applications by using a hierarchical QoS- aware scheduling framework built for fog computing with resource constraints. This framework enables a reduction in latency, an increase in bandwidth utilization, and a reduction in energy usage while taking into consideration the resource limits of the fog nodes. (Rohinidevi et al. 2022) suggested that QoS of fog computing applications can be effectively enhanced by using a dynamic QoS-aware scheduling framework built for fog computing with security features. The framework can help in adapting to the various dynamic changes taking place in a fog environment and thus help in maintaining the required QoS of the applications while keeping in mind the security needs of the applications. (Huang et al. 2020) have tried to propose a QoS-aware scheduling framework having multiple objectives and is sustainable in nature as a result can efficiently enhance the QoS of the fog computing applications. The framework may take into account the applications' demand for sustainability while minimizing latency, maximizing bandwidth utilization, and consuming the least amount of energy feasible. The framework does this by considering how fog computing affects the environment, including the amount of energy utilized and the carbon dioxide released. The proposed framework also keeps account of the financial impact of fog computing by calculating fog nodes' cost and energy being consumed. The framework uses a genetic algorithm to identify an optimal scheduling option that satisfies the QoS, sustainability, and economic requirements of applications.

The suggested framework was assessed using a simulated study. The simulation research depicts that the framework may successfully raise the QoS of fog computing applications while also taking into account the budgetary and sustainability needs of the applications. The framework is a strategy that has promise for improving the QoS for fog applications while also taking the financial and sustainability requirements of the applications. This framework might have a huge influence on fog computing even if it is still in the development stage.

3.3 Resource Scheduling Technique

Fog computing's resource scheduling approaches may be broken down into three distinct types: static, dynamic, and hybrid.

TABLE 2. METRICS CONSIDERED OF CURRENT SCHEDULING ALGORITHMS IN FOG ENVIRONMENT

Authors and references	Make span	Cost	Resource utilization	Reliability	Stability	Energy consumption	Allocated memory	Latency	Failure rate	Response time
Urgaonkar et al. 2015	×	□	×	□	□	×	□	□	×	□
Sarangi et al. 2016	□	□	×	□	×	×	□	×	×	□
Pham and Huh 2016	□	×	□	×	□	□	×	□	×	□
Ghobaei-Arani, et al 2019	×	□	□	×	×	□	×	□	×	□
Souri et al. 2017	×	×	×	□	□	×	×	×	×	×
Sun and Zhang 2017	□	×	□	×	□	□	×	□	×	□
Chen and Wang 2017	□	×	□	□	×	□	□	×	□	×
Rahbari and Nickray 2017	×	□	□	×	×	□	□	□	×	□
Kabirzadeh et al. 2017	□	×	□	□	×	□	×	□	×	□
De Benedetti et al. 2017	×	□	□	×	□	×	□	×	□	□
Miah et al. 2018	×	□	×	×	×	□	×	×	×	□
Deng et al. 2018	□	×	×	×	□	×	□	×	×	□
Lin and Yang, 2018	×	□	×	×	□	×	□	□	×	□
Fan et al. 2018	×	□	□	×	×	□	×	□	×	□
Kertesz et al. 2019	×	□	×	□	×	×	□	×	×	×
Jo and Kim 2019	×	×	□	×	□	×	×	□	×	□
Mahmud and Buyya 2019	×	□	×	□	×	□	×	□	□	□
Haghi Kashani, M. et al 2020	×	×	□	□	×	□	×	×	×	□
Bitam et al., 2017	×	□	□	×	×	×	□	×	×	×
Zhiguo et al., 2020	×	□	×	□	□	×	×	□	×	□
Apat, H. K., 2021	×	×	□	□	□	×	×	×	×	×
Murtaza, F. et al, 2020	□	□	□	×	□	×	□	×	□	×

Iyapparaja, M.et al., 2022	×	□	□	×	×	□	×	□	×	□
Anu A. S. et al., 2022	□	□	□	□	□	×	×	×	×	□
Ranganathan, G. et al., 2021	□	□	×	□	×	□	×	□	×	□
Kaur N. et al., 2021	×	□	×	×	×	□	×	×	×	□
Zhang et al, 2023	×	×	×	×	×	□	×	□	×	×
Islam M. et al, 2022	×	×	□	×	×	□	×	□	×	×
Zhang X. et al., 2023	×	×	×	×	×	□	×	□	×	×
Zhang Y. et al., 2023	×	□	×	×	×	□	×	□	×	×
Iyapparaja M. et al., 2022”	×	×	×	×	×	□	□	×	×	□

(Lin and Yang 2018), describes static scheduling methods for resources in a fog environment by all the information available on resource requirements and availability of resources before scheduling. The fog computing is highly dynamic and heterogeneous in nature, so it is not possible to have all the information of resources before scheduling. The works of (Ghobaei-Arani, Khorsand, and Ramezanpour 2019) describes dynamic scheduling methods, where the availability and requirement of the resources are not known earlier. Thus, the dynamic scheduling schedules the task that arrival time is not known earlier. In their paper, (Mahmud and Buyya 2019), describes the hybrid scheduling methods in a fog environment that combines different criteria to process single or batch jobs. (Sun and Zhang 2017) proposed a NSGA-II (Non-Dominated Sorting Genetic Algorithm II) to enhance execution time as well as lessen latency. The authors (Kabirzadeh, Rahbari, & Nickray, 2018) presented a bio-inspired based Bees Life Algorithm for maximizing the efficiency of work scheduling in fog services with respect to both time and memory requirements. Hyper-heuristic algorithms were suggested by the authors to reduce power consumption and program runtime. JARVSIS, a distributed task scheduler for IoT applications in a fog environment, was created (De Benedetti et al., 2017). It uses lightweight messages to automate the execution of many tasks. Pham and Huh (2016) presented an ant colony algorithm that was inspired by nature and was designed to work under time and material restrictions. The authors suggested a two-tiered, heuristics-based scheduling algorithm in which the first tier is utilized to set the priority and the second tier chooses the earliest start and completion time for each needed node. For the vehicular network, (Chen and Wang 2017) suggested two distinct dynamic scheduling approaches, one based on reaction time and another on queue length. A Markov decision model was presented (Urgaonkar et al., 2015) for service migration to accommodate user performance expectations.

4. Discussion

The discussion section presents a comparison and discussion on various scheduling algorithms that are used in a fog computing environment.

4.1 Scheduling Problems

In addition, the five key scheduling challenges of workflow scheduling, job scheduling, resource allocation, resource scheduling, and task scheduling are all taken into account in fog computing. There is a lack of development in fog computing scheduling field at the present moment. In table 3, we have compared some scheduling algorithms proposed by other authors that covered some issues such as: algorithm (used by authors), problem (which includes resource allocation, resource scheduling, and workflow/job/task scheduling), various metrics that algorithm has used, environment where the algorithm is implemented (both fog-cloud or fog only), and the various applications that were used by those algorithms.

TABLE 3. SCHEDULING ALGORITHM'S DETAILED COMPARISON IN THE FOG COMPUTING ENVIRONMENT

Authors	Algorithm	Problem	Metrics	Environment	Applications
Urgaonkar et al. 2015	Markov decision	Task scheduling	Latency Energy consumption	Cloud-fog	Cloud Devices
Sarangi et al. 2016	IoT	Task Scheduling	Failure rate; Energy consumption	Fog	Healthcare
Pham and Huh 2016	Heuristics scheduling	Resource allocation	Sensor lifetime; Execution cost; Energy consumption	Fog	Large-scale offloading applications
Rahbari and Nickray 2017	Heuristic algorithms	Resource allocation Task scheduling	Make-span cost for using the cloud resource	Fog	Security systems
Sun and Zhang 2017	Crowd-Funding	Task scheduling Resource allocation	Resource utilization Make-span Latency	Cloud-fog	Gaming
Kabirzadeh et al. 2017	Bees Life Algorithm	Task scheduling Workflow scheduling	Energy consumption Latency; Load balancing rate; Scheduling time; Response time	Fog	Scheduling systems
De Benedetti et al. 2017	JarvSis	Task scheduling Job Scheduling	Energy consumption; Response time; Make-span	Fog	Robotics
Chen and Wang 2017	Dynamic scheduling	Task scheduling	Communication cost Make-span	Fog	Vehicular networks
Lin and Yang, 2018	DMGA	Resource Scheduling	Reliability; Make Span; User satisfaction	Fog	Locations of gateways, fog devices
Fan et al. 2018	LAB	Resource scheduling	Stability Latency	Fog-Networking	IoT Devices
Mahmud and Buyya 2019	iFogSim	Resource scheduling Task scheduling	Response time; Energy consumption; Make-span	Fog	NA
Ghobaei-Arani et al 2019	Fuzzy decision tree	Task scheduling	Reliability; Energy consumption; Make-span	Cloud-fog	Massively Multiplayer Online Games
Murtaza F. et al., 2020	LRFC	Task Scheduling	Energy consumption; Processing time; Response time	Fog	NA

Vijayalakshmi, R. et al 2020	Fog and Cloud computing	Task Scheduling	Make-span Resource allocation	Fog	traffic congestion, public safety
Alsmadi, A. M., et al, 2021	Weighted Round-Robin	Task Scheduling	Resource allocation Latency	Fog	Community service
Abd Elaziz, M. et al., 2021	Salp Swarm Algorithm	Task Scheduling	Make-span Throughput	Cloud-fog	IoT Devices
Iyarapparaja M. el al., 2022	QTCS	Resource Scheduling	Energy consumption; Processing time; Resource allocation	Fog	NA
Kumar N. et al., 2022	Broker Management	Resource Scheduling	Resource allocation	Fog	Scheduling systems
Zhang X. et al., 2023	Genetic Algorithm	QoS Aware scheduling	Latency; Bandwidth utilization; Energy consumption	Fog	NA
Saif, F. A., et al, 2023	Grey Wolf Optimizer	Task Scheduling	Energy consumption Delay	Cloud-fog	Artificial Intelligence

In the below Figure 2, we have presented scheduling problems' statistical comparison. In this work, we have mainly focused on five significant scheduling issues which are: workflow scheduling, resource scheduling, job scheduling resource allocation, and task scheduling. Among these scheduling problems, with 45% of literature usage, task scheduling problems are found to have scheduling algorithms' highest percentage. Resource scheduling has 26%, resource allocation has a percentage of 16%, workflow scheduling problem has 5%, and job scheduling has 5% usage in fog computing (Alsmadi et al. 2021). Further, the performance metrics' analytical report has been shown in Figure 3 which is used for scheduling algorithms' evaluation. The analytical report showed that in scheduling algorithms, the most used performance metric with 26% is make-span which is followed by 20% of energy consumption [58]. Following this is the time and the latency that has 12%. Next is the cost which is 10%. Next in the report is resource utilization and reliability with 6% and 4% respectively. Further, failure rate, user satisfaction, allocated memory, deadline miss ratio, and stability has the same percentage of 2% (Saif et al. 2023). While concluding this discussion, we must say that a significant part is played by the fog computing in IoT scheduling algorithms' implementation. Our findings from this literature analysis indicate that the issue of task scheduling is fertile ground for future study in the context of fog computing. Further, future research may focus on developing novel scheduling methods that combine various metric optimization with other scheduling issues. Energy consumption and make-span are crucial metrics in fog computing scheduling algorithms. The make-span is a crucial metric for fog computing because it is used in IoT applications where latency is extremely sensitive. Fog computing presents new difficulties, such as potential battery drain and energy wastage. Each fog node's energy can be conserved by employing an effective data scheduling technique. Energy consumption in fog nodes, migration of virtual machine, applications with hard or soft deadlines, heterogeneous fog nodes, task migration among edge nodes, periodic tasks, and dynamic task scheduling are just some of the applications and metrics that researchers should consider when developing their new proposed scheduling algorithm.

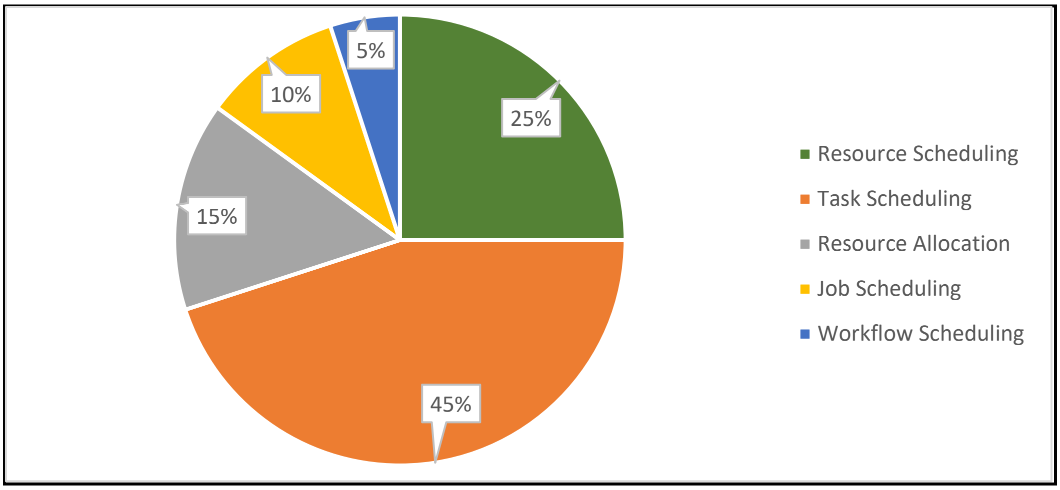
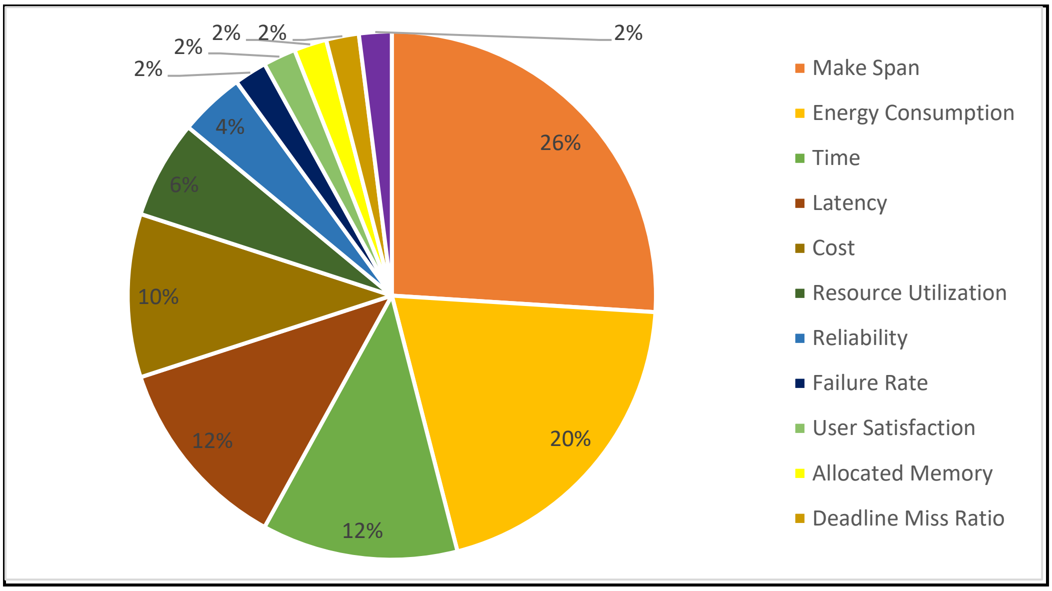


Fig 2. Fog computing's scheduling problems percentage solved by scheduling algorithms



"**Fig 3.** Percentage of performance metrics for evaluating scheduling algorithms"

5. Conclusion

In this paper, fog computing's recent scheduling algorithms have been comprehensively reviewed and analysed. After analysing the scheduling algorithms' papers, we have chosen the most used scheduling algorithm. In this survey, scheduling problems have been classified into five significant classes which are: workflow scheduling, resource allocation scheduling, resource scheduling, job scheduling, along with task scheduling. As it is clearly seen from comparison that with 45% of literature usage, task scheduling problems are found to have scheduling algorithms' highest percentage. The analytical report showed that in scheduling algorithms, the most used performance metric with 26% is make-span. Therefore, in a fog computing environment, the majority of scheduling problems are supposed to be research hotspots, and there is a need for more research in these areas. Furthermore, energy consumption as well as make span are considered to be most significant metrics in fog computing scheduling algorithms.

References

[1] Alaghbari, Khaled A., Mohamad Hanif Md Saad, Aini Hussain, and Muhammad Raisul Alam. 2022. "Complex Event Processing for Physical and Cyber Security in Datacentres - Recent Progress, Challenges and Recommendations." *Journal of Cloud Computing* 11(1), https://doi.org/10.1186/s13677-022-00338-x.

[2] Alsmadi, Ahmad Mohammad et al. 2021. "Fog Computing Scheduling Algorithm for Smart City." *International Journal of Electrical and Computer Engineering* 11(3): 2219–28, DOI: 10.11591/ijece.v11i3.pp2219-2228.

[3] Arikumar, K S, and V Natarajan. 2020. "FIoT: A QoS-Aware Fog-IoT Framework to Minimize Latency in IoT Applications via Fog Offloading." In *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 1*, Springer, 551–59, DOI:10.1007/978-981-15-5788-0_53.

[4] Badotra, Sumit, and Surya Narayan Panda. 2021.

"SNORT Based Early DDoS Detection System Using Opendaylight and Open Networking Operating System in Software Defined Networking."*Cluster Computing* 24(1):501–13, https://doi.org/10.1007/s10586-020-03133-y.

[5] Bala, Mohammad Irfan, and Mohammad Ahsan Chishti. 2019. "Survey of Applications, Challenges and Opportunities in Fog Computing." *International Journal of Pervasive Computing and Communications*, DOI:10.1108/IJPCC-06-2019-059.

[6] Bhatt, Chintan, and C K Bhensdadia. 2017. "Fog Computing: Applications, Concepts, and Issues." *International Journal of Grid and High Performance Computing (IJGHPC)* 9(4): 105–13, https://doi.org/10.4018/IJGHPC.2017100107.

[7] Bonomi, Flavio, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. "Fog Computing and Its Role in the Internet of Things." *MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*: 13–15, https://doi.org/10.1145/2342509.2342513.

[8] Cheng, Yew Leong, Meng Hee Lim, and Kar Hoou Hui. 2022. "Impact of Internet of Things Paradigm towards Energy Consumption Prediction: A Systematic Literature Review." *Sustainable Cities and Society* 78: 103624, https://doi.org/10.1016/j.scs.2021.103624.

[9] Das, Resul, and Muhammad Muhammad Inuwa. 2023. "A Review on Fog Computing: Issues, Characteristics, Challenges, and Potential Applications." *Telematics and Informatics Reports* 10(February):100049, https://doi.org/10.1016/j.teler.2023.100049.

[10] Deng, Ruilong et al. 2016. "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption." *IEEE Internet of Things Journal* 3(6): 1171–81, DOI: 10.1109/JIOT.2016.2565516.

[11] Faticanti, Francescomaria et al. 2020. "Throughput-Aware Partitioning and Placement of Applications in Fog Computing." *IEEE Transactions on Network and Service Management* 17(4): 2436–50, DOI: 10.1109/TNSM.2020.3023011.

[12] Fu, Weina, Shuai Liu, and Gautam Srivastava. 2019. "Optimization of Big Data Scheduling in Social Networks." *Entropy* 21(9): 1–16, doi:10.3390/e21090902.

[13] Ghobaei-Arani, Mostafa, Reihaneh Khorsand, and Mohammadreza Ramezanpour. 2019. "An Autonomous Resource Provisioning Framework for Massively Multiplayer Online Games in Cloud Environment." *Journal of Network and Computer Applications* 142(June): 76–97, DOI:10.1109/ACCESS.2021.3090366.

[14] Ghobaei-Arani, Mostafa, Alireza Souri, and Ali A Rahmanian. 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.

[15] Guleria, Kalpna et al. 2021. "An Enhanced Energy Proficient Clustering (EEPC) Algorithm for Relay Selection in Heterogeneous WSNs." *Ad Hoc Networks* 116, https://doi.org/10.1016/j.adhoc.2021.102473.

[16] Haghi Kashani, Mostafa, Amir Masoud Rahmani, and Nima Jafari Navimipour. 2020. "Quality of Service-Aware Approaches in Fog Computing." *International Journal of Communication Systems* 33(8), https://doi.org/10.1002/dac.4340.

[17] Huang, Xiaoge, Yifan Cui, Qianbin Chen, and Jie Zhang. 2020. "Joint Task Offloading and QoS-Aware Resource Allocation in Fog-Enabled Internet-of-Things Networks." *IEEE Internet of Things Journal* 7(8): 7194–7206, DOI: 10.1109/JIOT.2020.2982670.

[18] Intharawijitr, Krittin, Katsuyoshi Iida, and Hiroyuki Koga. 2016. "Analysis of Fog Model Considering Computing and Communication Latency in 5G Cellular Networks." In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 1–4, DOI: 10.1109/PERCOMW.2016.7457059.

[19] Islam, Mir Salim Ul, Ashok Kumar, and Yu-Chen Hu. 2021. "Context-Aware Scheduling in Fog Computing: A Survey, Taxonomy, Challenges and Future Directions." *Journal of Network and Computer Applications* 180: 103008, https://doi.org/10.1016/j.jnca.2021.103008.

[20] Jamil, Bushra et al. 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.

[21] Jo, Dongsik, and Gerard Jounghyun Kim. 2019. "AR Enabled IoT for a Smart and Interactive Environment: A Survey and Future Directions." *Sensors (Switzerland)* 19(19), doi:10.3390/s19194330.

[22] Kaur, Navjeet, Ashok Kumar, and Rajesh Kumar. 2021. "A Systematic Review on Task Scheduling in Fog Computing: Taxonomy, Tools, Challenges, and Future Directions." *Concurrency and Computation: Practice and Experience* 33(21): e6432, https://doi.org/10.1002/cpe.6432.

[23] Kertesz, A., T. Pflanzner, and T. Gyimothy. 2019. "A Mobile IoT Device Simulator for IoT-Fog-Cloud Systems." *Journal of Grid Computing* 17(3): 529–51, DOI: 10.1007/s10723-018-9468-9.

[24] Khattar, Nagma, Jagpreet Sidhu, and Jaiteg Singh. 2019. "Toward Energy-Efficient Cloud Computing: A Survey of Dynamic Power Management and Heuristics-Based Optimization Techniques." *The Journal of Supercomputing* 75,

https://doi.org/10.1007/s11227-019-02764-2.
[25] Kimani, Kenneth, Vitalice Oduol, and Kibet Langat. 2019. “Cyber Security Challenges for IoT-Based Smart Grid Networks.” *International journal of critical infrastructure protection* 25: 36–49, https://doi.org/10.1016/j.ijcip.2019.01.001.
[26] Li, Qiang et al. 2020. “Capacity-Aware Edge Caching in Fog Computing Networks.” *IEEE Transactions on Vehicular Technology* 69(8): 9244–48, DOI: 10.1109/TVT.2020.3001301.
[27] Lin, Chun Cheng, and Jhih Wun Yang. 2018. “Cost-Efficient Deployment of Fog Computing Systems at Logistics Centers in Industry 4.0.” *IEEE Transactions on Industrial Informatics* 14(10): 4603–11, DOI:10.1109/TII.2018.2827920.
[28] Mahmud, Redowan, and Rajkumar Buyya. 2019. “Modeling and Simulation of Fog and Edge Computing Environments Using Ifogsim Toolkit.” *Fog and Edge Computing: Principles and Paradigms*: 433–165, DOI:10.1002/9781119525080.ch17.
[29] Maiti, Prasenjit, Jaya Shukla, Bibhudatta Sahoo, and Ashok Kumar Turuk. 2019. “Mathematical Modeling of QoS-Aware Fog Computing Architecture for Iot Services.” *Advances in Intelligent Systems and Computing* 814(November 2018): 13–21, DOI:10.1007/978-981-13-1501-5_2.
[30] Miah, Md Sipon, Michael Schukat, and Enda Barrett. 2018. “An Enhanced Sum Rate in the Cluster Based Cognitive Radio Relay Network Using the Sequential Approach for the Future Internet of Things.” *Human-centric Computing and Information Sciences* 8(1). https://doi.org/10.1186/s13673-018-0139-4.
[31] Pereira, Jorge et al. 2019. “Assessing the Reliability of Fog Computing for Smart Mobility Applications in VANETs.” *Future Generation Computer Systems* 94(August 2019): 317–32, DOI:10.1016/j.future.2018.11.043.
[32] Qu, Zhiguo et al. 2020. “Study QoS Optimization and Energy Saving Techniques in Cloud, Fog, EDge, and IoT.” *Complexity* 2020, https://doi.org/10.1155/2020/8964165.
[33] Rani, Meena, Kalpna Guleria, and Surya Narayan Panda. 2021. “Cloud Computing An Empowering Technology: Architecture, Applications and Challenges.” *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*: 1–6, DOI: 10.1109/ICRITO51393.2021.9596259.
[34] Rohinidevi, V Vasuki et al. 2022. “A Taxonomy towards Fog Computing Resource Allocation.” In *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, , 1–5, DOI: 10.1109/CISCT55310.2022.10046643.
[35] Saif, Faten A., Rohaya Latip, ZM Hanapi, and K. Shafinah. 2023. “Multi-Objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud-Fog Computing.” *IEEE Access* 11(January): 20635–46, DOI:10.1109/ACCESS.2023.3241240.
[36] Seth, Ishita et al. 2022. “A Taxonomy and Analysis on Internet of Vehicles: Architectures, Protocols, and Challenges.” Wireless Communications and Mobile Computing 2022, Article ID 9232784, 26 pages, volume 2022, https://doi.org/10.1155/2022/9232784.
[37] Singhrova, Anita, and Anu Anu. 2022. “Resource Allocation in Fog Computing Based on Meta-Heuristic Approaches: A Systematic Review.” 22: 503–14, DOI:10.22937/IJCSNS.2022.22.9.65.
[38] Sodhro, Ali Hassan et al. 2018. “5G-Based Transmission Power Control Mechanism in Fog Computing for Internet of Things Devices.” *Sustainability (Switzerland)* 10(4): 1–17, https://doi.org/10.3390/su10103626.
[39] Souri, Alireza, Parvaneh Asghari, and Reza Rezaei. 2017. “Software as a Service Based CRM Providers in the Cloud Computing: Challenges and Technical Issues.” *Journal of Service Science Research* 9: 219–37, DOI 10.1007/s12927-017-0011-5.
[40] Sun, Yan, and Nan Zhang. 2017. “A Resource-Sharing Model Based on a Repeated Game in Fog Computing.” *Saudi Journal of Biological Sciences* 24(3): 687–94. http://dx.doi.org/10.1016/j.sjbs.2017.01.043.
[41] Uppal, Mudita et al. 2021. “Cloud-Based Fault Prediction Using IoT in Office Automation for Improvisation of Health of Employees.” Journal of Healthcare Engineering 2021, Article ID 8106467, 13 pages, Volume 2021 https://doi.org/10.1155/2021/8106467.
[42] Wang, Haozhe et al. 2019. “Data-Driven Dynamic Resource Scheduling for Network Slicing: A Deep Reinforcement Learning Approach.” Information Sciences 498: 106–16, https://doi.org/10.1016/j.ins.2019.05.012.
[43] Yassine, Abdulsalam, Shailendra Singh, M Shamim Hossain, and Ghulam Muhammad. 2019. “IoT Big Data Analytics for Smart Homes with Fog and Cloud Computing.” Future Generation Computer Systems 91: 563–73, DOI:10.1016/j.future.2018.08.040.
[44] Yoghourdjian, Vahan et al. 2021. “Scalability of Network Visualisation from a Cognitive Load Perspective.” IEEE Transactions on Visualization and Computer Graphics 27(2): 1677–87, https://doi.org/10.48550/arXiv.2008.07944.
[45] Zeng, Deze et al. 2016. “Joint Optimization of Task Scheduling and Image Placement in Fog Computing Supported Software-Defined Embedded System.” IEEE Transactions on Computers 65(12): 3702–12, DOI: 10.1109/TC.2016.2536019.
[46] Krishna, K. S. ., Satish, T. ., & Mishra, J. . (2023).

Machine Learning-Based IOT Air Quality and Pollution Detection. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 132–145. https://doi.org/10.17762/ijritcc.v11i2s.6036

[47] Prof. Muhamad Angriawan. (2016). Performance Analysis and Resource Allocation in MIMO-OFDM Systems. International Journal of New Practices in Management and Engineering, 5(02), 01 - 07. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/44

[48] Kamble, S.D., Saini, D.K.J.B., Jain, S., Kumar, K., Kumar, S., Dhabliya, D. A novel approach of surveillance video indexing and retrieval using object detection and tracking (2023) Journal of Interdisciplinary Mathematics, 26 (3), pp. 341-350.