International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I7P230

Volume 70 Issue 7, 291-300, July 2022

*Review Article*

# A Study on Resource Allocation and Scheduling Methods in Fog Computing Environment

Monika[1], Harkesh Sehrawat[2] ,Vikas Siwach[3]

[1, 2,3] *Computer Science and Engineering, UIET, Maharshi Dayanand University, Rohtak, India*

[1]monikacse1920@gmail.com

Received: 22 May 2022 Revised: 10 July 2022 Accepted: 19 July 2022 Published: 25 July 2022

***Abstract*** *- In recent years, distributed networks have been taking charge in numerous organizations, applications, and environments for organizational control, service distribution, centralized storage, and transactions. Cloud computing is the most accepted technology in such environments, providing centralized control, storage, and management. IoT and real-time interactive applications are getting popular to improve the quality of life. These applications, devices, environments, and users demand instant services. The centralized cloud system could not provide such services instantly for a distant device or user. Fog computing resolved this problem by minimizing the distance between devices, servers, and end users. In Fog computing, the IoT applications are controlled and implemented by the intermediate smart nodes called fogs. These fogs work as a layer between the centralized cloud system and data centre IoT devices. This technology has capabilities, limitations, architectural strengths, and functional methods. In this paper, some common aspects related to fog computing are explored and discussed. The paper includes the architectural and functional description of the fog computing model. The features and limitations of this distributed environment are also explored. Fog computing accepts the user requests through computing fog nodes and faces the problem of resource allocation and scheduling. The paper has identified various issues, features, and methods associated with resource allocation and task scheduling. A detailed study on scheduling methods and resource allocation challenges is also provided in this paper.*

***Keywords*** *- Fog Computing, Scheduling, Resource Allocation, Cloud Computing, Distributed Network.*

## 1. Introduction

The emergence of the Internet of Things (IoT) can be seen in daily life. Each application domain adapts IoT services to automate the services, equipment, and life. Smart homes, offices, grids, classrooms, and smart cities connect each individual with IoT devices. Various IoT-integrated devices are a part of our day-to-day life, including cars, surveillance cameras, smart watches, smartphones, etc. Using IoT devices in a domain or network results in huge data communication. It gives various issues, including data storage, management, and security. As the IoT devices work in real-time, this data should be in real-time. It means the user wants instant results on his queries. It gives the emergence of fog computing and edge computing [1][2][31][32].

Before fog computing, cloud computing was used as a well-known distributed architecture and was deployed on a larger scale with centralized data centres. But processing this huge centralized dataset is not efficient and practical in real-time. Even as the distance of a user increases from the centralized data centre, transmission delay and response time also occur. Fog computing answers all these problems [2][5]. Fig 1 shows the transformation of Cloud to fog computing with added functionality. With the start of IoT devices, the functionality of cloud computing was upgraded to attain its adaption to IoT devices. At that time, resource profiling, user customization, real-time support, service performance updation, and various embedded features were integrated. As the network size increases with many IoT devices, fog computing starts taking charge under a cloud environment. The functional adaption of fog with IoT includes dynamic cloud formation, VM-based clouds, service coordination, vertical management, resource customization, and time-sensitive operations [49][5][27]. Fog computing is getting popular in IoT-based data analytics.

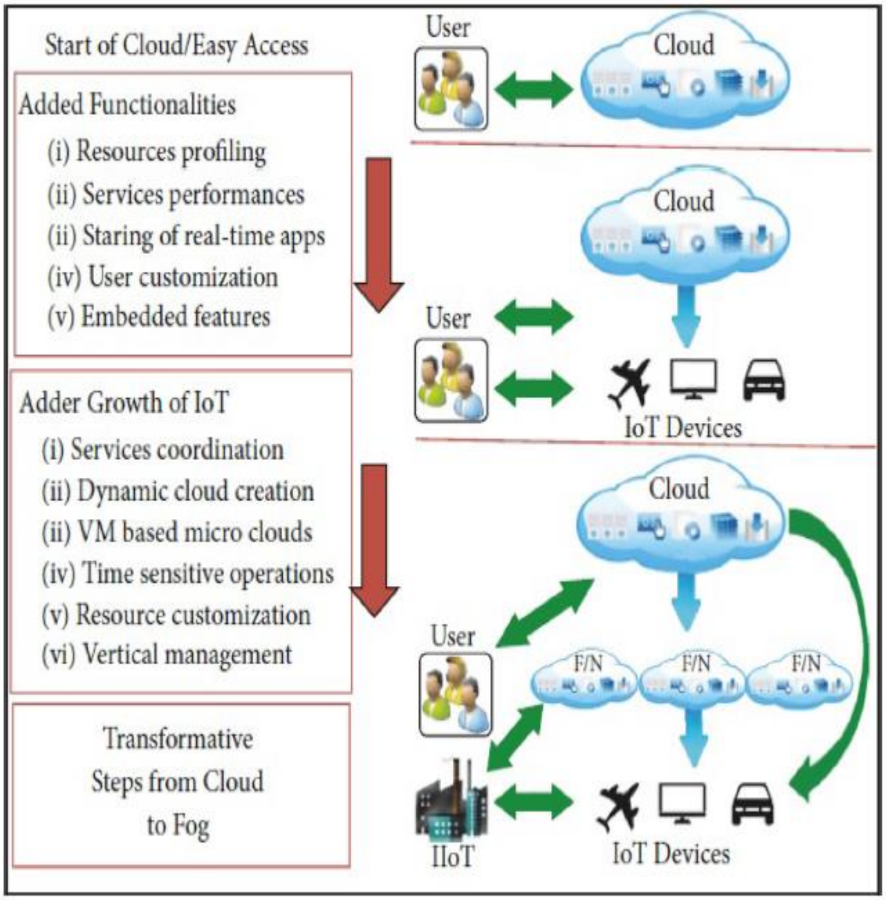

**Fig. 1 Cloud to Fog Transformation[49]**

Fog computing is an advancement over cloud computing by managing and controlling the data at the edge. Even though it is a separate network environment and cannot replace cloud computing. It is an extension of cloud computing specifically for IoT-based networks. As the fog exists nearer to the client, it reduces the latency, data transformation cost, and bandwidth requirement. It can respond in real-time with minimum storage and computation [3][23]. There are many challenging and real-time applications where cloud computing is not suitable. Vehicular networks [34] and grid computing [37] are such environments that require real-time and IoT-based support. Vehicular networks are highly mobile and hybrid as vehicle nodes change regions very frequently. Fog computing can be adopted in such networks to ensure seamless service delivery. The information about geographical information, traffic, and communication information can be processed effectively using fog computing architecture [34].

## 2. Fog Computing

Fog computing [7] [8] [9] [10] is a lightweight distributed architecture that is composed of software, hardware, and network elements. It is a layered architecture with three main layers: the Edge, Fog, and Cloud. The edge layer is where data resides and is present in a customer's physical position. This layer contains diverse IoT devices, including sensors, mobile phones, vehicles, smart appliances, etc. All these devices define a geographical circle and identify the segment data of physical elements and events. As some event or process is executed, the transmission begins to the upper layer connectivity-driven communication is established fog Fog layer is situated just above the edge layer. This layer contains innumerable centres using gateways, switches, hubs, and other controller devices. These fog hubs can establish connectivity to the base station, explicit servers, and another centralized system. This layer is the intermediate layer that manages the interconnection between end devices and the Cloud. The cloud computing layer is the top layer of this architecture. In this layer, the centralized and server-based system is established and configured. This layer has capacity gadgets, application services, transportation systems, and superior servers. The capacity and conclusive load-based data processing are performed on this layer. The centralized control of the architecture lies within the cloud computing layer. Fig 2 shows the layered architecture of fog computing with the specification of inclusive components and associated tasks[11][12][13].

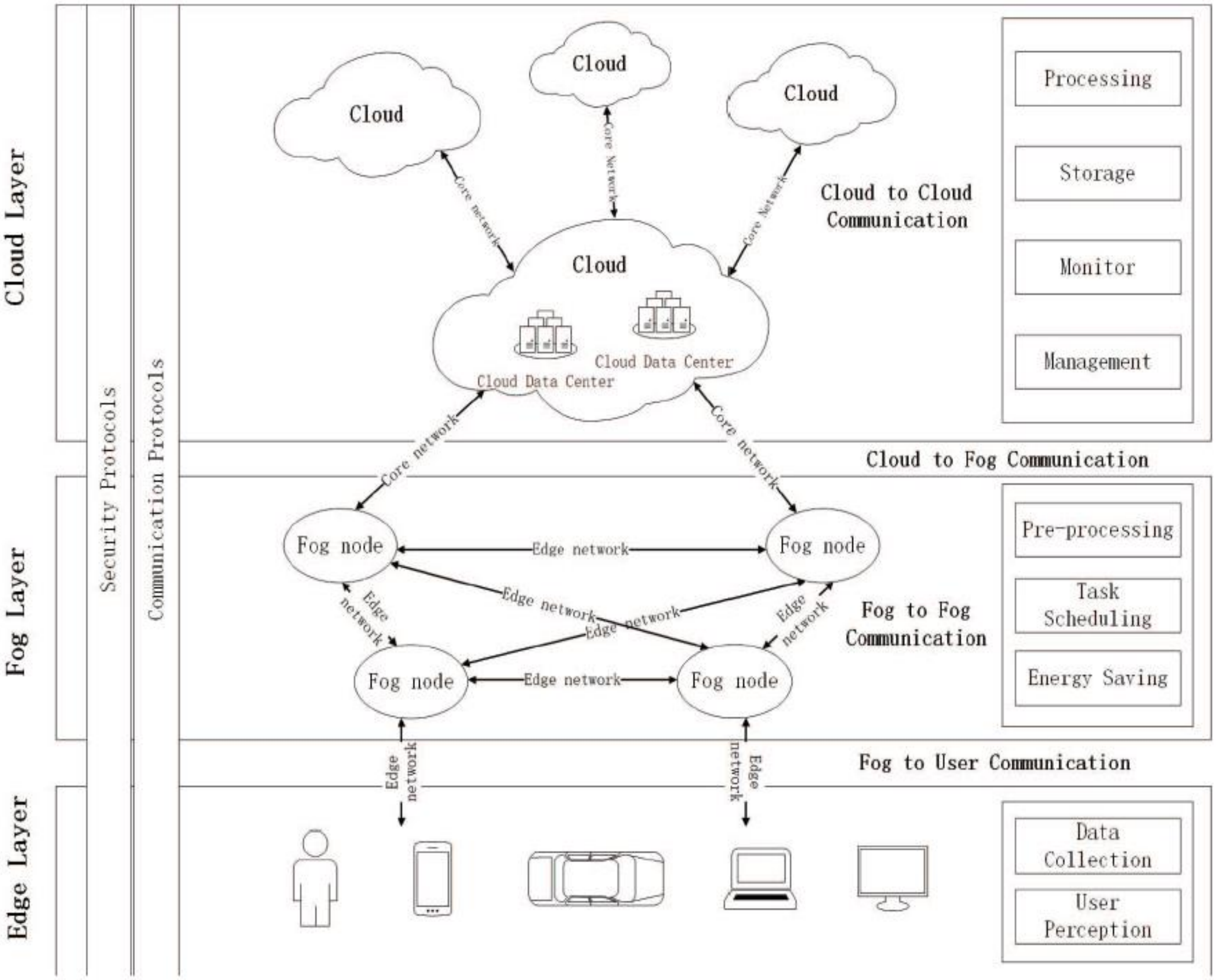

**Fig. 2 Standard Architecture of Fog Computing [13]**

Fig 2 shows that the edge layer collects the data through IoT devices and manages them based on user requirements and perceptions. This data is collected based on user activity and participation per defined schedule and characteristics. The Fog-to-user communication is performed between the fog and edge layer to track the user activity and avail services based on their requests. In this fog computing layer, the user requests are processed, analyzed, and executed to avail of these services. This layer controls the consumption of available resources and energy. The resource allocation and scheduling are also performed within the fog layer. Multiple fog nodes in this layer interact to reduce the wastage of resources and optimize the service execution in the fog environment. This layer also interacts with the cloud layer for submitting the information to a centralized server and availing the centralized services. The cloud layer is responsible for managing and monitoring the behaviour of the fog environment. It contains a centralized database including user, activity, and analytical information. Any configuration and architecture level control decisions are also taken within the cloud layer [11][12][13][30].

Fog computing with IoT[18] is the most adaptive environment and technological integration to provide reliable and effective services to clients. Fog computing brings a lot of advantages for IoT when it is implied in real and varied environments. IoT applications work on big data, and its size is amazingly increasing. But, if the centralized Cloud is used to manage and respond to this data, it must reach the Cloud. Executing the jobs within the deadline in a real-time environment won't be easy. Fog computing provides the solution for such IoT-based applications with a timely response. Fog computing targets mobile and dynamic users in a real-time network. These users are equipped with varied technology, and platforms are processed with different protocols. The users in a coverage region are dynamic, but the user count is always restricted. The servers are the controllers with the medium capability to handle a certain load as the expected density of requests is lesser. Even the controllers in the regions can be dynamic and established in indoor and outdoor environments. The real-time interaction is performed with medium power consumption and with lesser computation power. As fog computing involves dynamic and mobile users in a region, the user requests and load is unequal. This unequal and heterogeneous load problem is a challenge in fog computing in which the edge-placed servers or controllers are configured with restricted and limited capabilities. A dynamic load balancing mechanism is required to keep the homogeneous load on controllers or servers even in a heterogeneous environment. A parameter-driven analysis is required to identify and resolve the load issue in real-time. Pereira et al.[19] analyzed the availability of computational resources and proposed a policy for the effective allocation of tasks. The task priority was analyzed while allocating a fog node. The author compared latency, load, available memory, and several core features for optimizing the usage of available resources with the homogenous load. Fig 3 shows the component-based architecture of a fog node. Each fog node maintains two main local databases: Local DB and Network topology DB. The local DB manages the IoT services and application-specific services. Network topology DB maintains the data related to the network. The network connectivity, communication, and configuration information are maintained in this database.

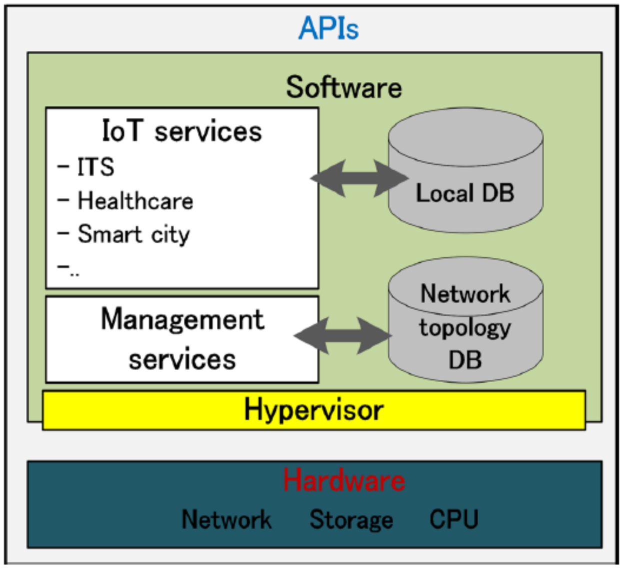

**Fig. 3 Components of Fog Node[4]**

The common, adaptive, and optimum features of fog computing are listed in section 2.1, and the commonly faced issues and challenges in the fog environment are listed in section 2.2:

### *2.1. Features of Fog Computing*

#### *2.1.1. Openness*

Fog computing is implemented in IoT based network that is accessible to different types of users based on the application domain. Such IoT networks can be a home, office, hospital, school, institute, parking, or city. Openness is the most effective feature of fog computing. The environment can be configured based on the business requirements. Various integrated qualities of the open nature of fog computing include support for multiple operating systems and its hierarchical architecture. Multiple operating systems can interact with different hardware and applications because the hierarchical feature can be integrated into the environment with different roles[4].

#### *2.1.2. Scalability*

As fog computing can be applied in a diverse environment, the size and requirements of these organizations and applications differ. The flexible and scalable nature of fog computing makes it capable of being implemented under different loads, costs, and performance needs. The scalability in an environment is required regarding the number of nodes, networks, and infrastructure devices. A fog computing environment can be extended by including additional fog nodes or infrastructure devices. Multiple networks can be merged based on the requirement to get a wider fog network[4].

#### *2.1.3. Programmability*

Fog computing also provides the flexibility to configure the deployment based on the customer or user requirement. The programmability feature of fog computing makes it more flexible to do such changes. The additional functions can be included in the behaviour and functioning of fog computing to provide support at different levels. The updated programs and services can be configured in the environment without affecting the whole system. The technological updation increases the strength of the cloud computing environment [4].

#### *2.1.4. Low Latency*

The key requirements and motivation of fog computing are to reduce data transmission's weight and response time. Fog computing is applied in real-time applications requiring an instant response. Fog computing has reduced this delay, enhancing its adaptation over existing distributed networks. In real-time and time-sensitive applications, fog computing is the preferred technology[6].

#### *2.1.5. Mobility*

Fog computing can be applied to dynamic and mobile networks. Fog computing is adaptive to vehicular networks. Even mobile users can use fog computing services in real-time. Even there is no additional cost or computation required for adapting such features. It is the integrated part of fog computing[6][28][29].

#### *2.1.6. Bandwidth*

Fog computing preprocesses the data before transmission, reducing the network traffic and enhancing the network's performance. The data filtering and aggregation are performed at the local level. It improves the execution speed of various tasks [6].

### *2.2. Issues/Challenges in Fog Computing*

The architecture of Fog computing is complex as it exists at the edge of the Internet and closer to the user. It is a heterogeneous, dynamic, and scalable network in which hybrid technologies and networks are integrated. Even with these complexities, the resources and capabilities of edge servers are restricted[20][26][29]. Some of the common restrictions and issues of the fog computing environment are given here:

#### *2.2.1. Hybrid Environment*

Fog computing provides a hybrid and dynamic environment in which various nodes, controllers, and technology exist. Even while selecting the virtualization technology, several possible techniques have different capabilities. The network design, flexibility, and capabilities also change with the adaption of virtualization technology. When we select container-based virtualization, it reduces flexibility. There is the requirement to identify the infrastructure, load, and objective of that application area while setting up the technological aspects in the network. Even though these factors can affect the network performance, security, reliability, and privacy in one way or another [21].

#### *2.2.2. Communication Delay*

A fog network is implemented in real-time situations that require instant decision and service delivery. But as the environment is dynamic and load is variable, the server should be ready to handle maximum load and expected load situations. This kind of load can cause anonymous delay and execution failures. Rescheduling and reprocessing tasks can solve the problem, but in the real-time environment, it can increase communication delay[21].

#### *2.2.3. Security and Privacy*

Fog network is implemented in public, personal and real-time situations and is available to various users. Intruders also target this open environment to read sensitive personal information or degrade the network's performance. Maintaining the data level, system level, and communication level security in a fog environment is a big challenge[21][25][26].

#### *2.2.4. Accountability*

In the fog environment, various users, organizations, and companies share resources to adapt the business model. There is a requirement for some standard and proper system to decide their incentives. The central control and assessment are required to decide the accountability in the system. If some security leak or request drops happen, there should be a mechanism to take responsibility and compensate with some incentives[22].

## 3. Resource Distribution and Management

The distributed fog environment provides lightweight data storage, management, and processing technology to perform IoT services. This IoT-based network's functional behavior is similar to a cloud environment. Various network, storage, and communication-based resources are available in different layers of Fog computing. Fog computing is more complex than cloud computing as real-time and faster communication is performed by IoT devices. Features of this network are heterogeneous, dynamic, and uncertain behaviour. These characteristics also increase the computational and functional complexity of fog computing. As discussed earlier in this paper, resource management is a critical challenge in fog computing. The researchers investigated various resource management approaches to optimize the fog environment's reliability. These approaches are resource scheduling, task offloading, resource allocation, application placement, and resource provisioning. In this section, resource allocation and management methods are discussed.

### *3.1. Resource Scheduling*

Multiple fog nodes in distributed fog can provide the requested service. But the performance, architectural configuration, capabilities, and load on these fog nodes can differ. Sometimes, the user request is submitted to a fog node that can be fully exhausted with a heavy load. The allocation of requests to such nodes can cause higher delay or even execution failure. In such a case, the user request can be placed to another fog node with lesser load and higher capability. Resource scheduling is used in such cases to arrange the available and feasible resources in order of their capabilities. Various QoS and capability features can be analyzed to find an optimal solution. Resource scheduling is an NP-hard problem and can be optimized using some meta-heuristic solution. These optimization methods can identify the near-optimal and feasible solution. We need to set some objectives and constraints to identify the optimal solution. Resource scheduling is an adaptive method that effectively utilizes available resources[24].

### *3.2. Task offloading*

Task offloading analyzes IoT devices' computational features and measures, including energy, storage, and processing capability. It identifies the resources with limited and lesser capabilities that need some outsourcing. The resources of these IoT devices are limited and require some additional backup and outsourcing to ensure the longer functioning of hardware devices. In a real-time fog environment, many applications require a rich number of resources with longer life and capabilities. These applications include user authentication, multimedia-based applications, augmented reality, and virtual reality-based applications. In such applications, external resources or entities can be used for executing the processing called task offloading. Offloading transfers the task from a low-resource source to a rich-resource destination node[24].

### *3.3. Device or Resource Allocation*

Resource allocation is a double-matching problem with two-level control in fog nodes and cloud servers. In this network, low-level IoT devices, users, and fog nodes are coupled with cloud servers. As multi-level resources exist in this environment, resource allocation is quite challenging. Double matching is applied to achieve optimization at the cloud and fog levels. In fog computing, market-based and auction-based resource allocation is very common. The resource cost is computed based on the market price in the market-based method. Based on this cost computation and resource requirement, the resources are allocated. In auction-based allocation, the cost of resources changes based on the user requirement and demand for the resource. The objective functions are defined based on cost minimization and QoS optimization. The main objective is to utilize the resources and satisfy the user demand[24].

### *3.4. Application placement*

In a fog environment, the Cloud is the centralized server, and it has to identify the virtual fog nodes where the services can be deployed. An optimal service placement method is required for resource utilization and QoS enhancement. Application placement uses a broker management approach for controlling and managing the services in a fog computing environment. The broker management models are divided into three main categories called centralized, decentralized, and hierarchical. The centralized broker collects and manages the information from users, IoT, and fog devices and performs global optimization. The centralized control-based decision-making ensures the optimized solution against execution overhead and fault occurrence. The decentralized broker-based method is highly scalable and requires high-level management to enhance the system's performance. The hierarchical system combines the advantages of both decentralized and centralized systems[24].

### *3.5. Resource Provisioning*

Fog is a dynamic network in which the users are mobile. Because of this dynamic nature, the number of users in a region changes regularly and causes an overload problem. This workload fluctuation is identified as an over and under-provisioning problem. More resources are allocated than users' expected or actual demand in over-provisioning. In such a case, the user has to bear the unnecessary cost. In the case of under-provisioning, fewer resources are allocated to services than expected or users' actual demand. In this case, the SLA violation, requests, drop, and loss of users happen. The fog computing environment should be capable of dealing with these problems while maintaining the QoS constraints. Dynamic resource provisioning is also adapted to handle load fluctuations[24].

## 4. Task Scheduling

In a fog computing environment, a large user count is connected to the environment, generating several job requests. Task scheduling is the functional technology that allocates the user requests to the particular server, controller, or fog device. Each controller or virtual machine can have several assigned tasks. In such a case, the task scheduling algorithms are incorporated within the system to set up the task execution sequence. Various features, priority measures, and mathematical computations are implied within the system to set up a schedule for allocated requests. Some common measures and parameters for assigning priority or order of task execution are delay, average delay, turnaround time, probability of failure, task criticality, etc. Its objective is to reduce the delay, execution time, cost, and power consumption. The scheduling method should ensure the maximum utilization of available resources and execute the task before the deadline. An effective scheduling technique benefits the service provider and the user. Scheduling methods improve the efficiency, scalability, and performance of the system. Some traditional scheduling methods include Shortest Job First (SJF), First come, first served (FCFS), etc. Later, researchers investigated various scheduling methods based on cloud and fog environments' application and architectural requirements. These methods adopt tasks' and resources' static and dynamic features to optimize task scheduling performance. In recent years, genetics, particle swarm optimization, and simulated annealing methods have been integrated by researchers to improve the effectiveness of task scheduling methods[16][35]. In multi-task and critical-tasks applications, the scheduling algorithms are more sensitive to adapt. In such an application, the task weightage, behaviour, and objectives are considered while setting up the task priority and schedule[36].

When the available resources are limited, and the particular resource or server cannot handle the assigned job, the process can be switched to another machine, called VM migration. For handling the load and optimizing resource utilization, the virtual machines can be moved from one physical host to other. Various measures and feature-based analyses are conducted to identify the migration machine. VM migration enhances fault tolerance and load balancing and reduces energy consumption and unusual delay[33]. In scheduling algorithms, VM is also an associated task to handle the heavy load or unusual delay situations. The researchers in this section proposed various VM migration and scheduling methods.

Workflow[15] is an extensive scheduling method in which tasks depend on each other and use distributed resources. The resources access, management, and control are utilized by scientific workflow to optimize storage and computing in a distributed environment. In this environment, priority assignment and clustering are performed to optimize task execution quality in terms of reliability, availability, interoperability, recoverability, and performance measures. Memari et al.[14] used the meta-heuristic algorithm that used latency key parameters to optimize the performance. The author combined the approximate fruit fly optimization (FOA) with the nearest neighbour (ANN) algorithms for optimizing the scheduling in a cloud-fog computing environment. The proposed method reduced the latency, execution time, cost, and allocated memory.

**Table 1. Various Scheduling Techniques/Algorithms**

| Author | Algorithm/Model/ Method | Problem Addressed | Performance Results |
|---|---|---|---|
| Luo et al.[41] | Multi-cloud to multi-fog architecture with energy-based scheduling method | Unequal Load, Energy consumption | Achieve Power Balancing, High transmission rate, and High accuracy of executed tasks |
| Tychalas et al.[42] | Bag-of-Tasks-based workload Model | Response Time, cost of execution | Enhanced resource utilization, cloud cost ratio in different load conditions |
| Ghobaei-Arani et al.[43] | Moth-flame Optimization Model | Volume, Bandwidth requirement | Reduced Execution time and Makespan |
| Sun et al.[44] | Improved Non-dominated Sorting Genetic Algorithm (NSGA) II | Stability and Latency | Reduced the execution delay and achieved higher stability |
| Pham et al.[45] | Cost makespan aware of heuristic scheduling | Deadline, Processing Cost | Reduced cost and Makespan |
| Mukherjee et al.[46] | Deadline-aware fair scheduling | Delay, Deadline, and Job Density | Improved reliability and accuracy of task execution |
| Li et al.[47] | Fuzzy Clustering with Particle Swarm Optimization | Bandwidth, Computing capability, storage | Improved User satisfaction |
| Hosseinioun et al.[48] | Hybrid Invasive Weed Optimization and Culture (IWO-CA) evolutionary algorithm | Energy, Task Density, Cost | Improved resource utilization and reduced energy consumption |

Wadhwa et al.[17] observed and tracked the intensity-based evaluation to integrate the expectation maximization (EM) algorithm within the system to optimize the utilization of resources. The author utilized resource grading as a key feature within the proposed scheduling algorithm. The proposed work reduced the execution time, delay, and energy consumption compared with SJF, FCFS, and MPSO algorithms. Wan et al.[38] identified the energy restriction and heavy load problem issues. The author proposed an energy-aware and load balancing (ELBS) scheduling method to handle the workload problem. This research proposed a cluster-based method to optimize the performance of fog nodes. An improved particle swarm optimization-based scheduling method was proposed for balancing the load and reducing the energy consumption in a fog environment. Abdel-Basset et al.[39] proposed a marine predators algorithm (MPA) for optimizing the performance of task scheduling in a fog computing environment. This algorithm processed the energy, bandwidth, and response time parameter to reduce the task processing overheads in a fog environment. The proposed algorithm defined a ranking method to achieve the maximum utilization of available resources. The proposed method computed the local optima and identified the near-optimal solution. The results identified that the proposed model reduced the energy consumption, turnaround time, and flow time and improved the performance of task scheduling in a fog environment. A priority[40] based task scheduling method was proposed to execute the tasks before the deadline. The algorithm was applied to the middle layer of the fog layer and managed the resources based on this priority algorithm. The resource features were analyzed while assigning the requests to a fog node. A priority-based queuing method was defined to reduce the turnaround and wait time.

## 5. Conclusion

This paper provides a comprehensive survey on various aspects of fog computing technology. The requirement of computing and its advantages over cloud computing are discussed. The layered architecture of fog computing is discussed in detail. The functional behaviour and responsibilities of these layers are also defined. The interaction between IoT devices and the centralized cloud server is explored in this paper. Various challenges and characteristics of the fog computing environment and application are also explored. Fog nodes accept requests from IoT devices and avail the required services. The issues in the real environment are resource allocation, availability, and scheduling. The paper discussed these issues in detail and provided a survey on different methods, approaches, and techniques proposed by the researchers to resolve these issues.

## References

[1] Ghobaei-Arani, M., Souri, A. & Rahmanian, A.A, “ Resource Management Approaches in Fog Computing: A Comprehensive Review,” *J Grid Computing*, Vol. 18, Pp. 1–42, 2020. Https://Doi.Org/10.1007/S10723-019-09491-1

[2] Singh, S.P., Nayyar, A., Kumar, R. Et Al., “Fog Computing: From Architecture To Edge Computing and Big Data Processing,” *J Supercomput*, Vol. 75, Pp.2070–2105 2019. Https://Doi.Org/10.1007/S11227-018-2701-2

[3] Hurbungs, V., Bassoo, V. & Fowdur, T.P, “ Fog and Edge Computing: Concepts, Tools and Focus Areas,” *Int. J. Inf. Tecnol*, Vol. 13, Pp.511–522 , 2021. Https://Doi.Org/10.1007/S41870-020-00588-5

[4] Tran, Q.M., Nguyen, P.H., Tsuchiya, T. Et A, “Designed Features for Improving Openness, Scalability and Programmability in the Fog Computing-Based Iot Systems,” *SN COMPUT. SCI*, Vol. 1, Pp.194, 2020. Https://Doi.Org/10.1007/S42979-020-00197-W

[5] Kaur, M., Aron, R., “A Systematic Study of Load Balancing Approaches in the Fog Computing Environment,” *J Supercomput*, Vol.77, Pp.9202–9247 2021. Https://Doi.Org/10.1007/S11227-020-03600-8

[6] Gasmi, K., Dilek, S., Tosun, S. Et Al, “A Survey on Computation Offloading and Service Placement in Fog Computing-Based Iot,” *J Supercomput* , 2021. Https://Doi.Org/10.1007/S11227-021-03941-Y

[7] Mann, Z.Á., “Notions of Architecture in Fog Computing. Computing,” Pp.103, Pp.51–73, 2021. Https://Doi.Org/10.1007/S00607-020-00848-Z

[8] S. Delfin, N. P. Sivasanker., N. Raj and A. Anand, "Fog Computing: A New Era of Cloud Computing," *2019 3rd International Conference on Computing Methodologies and Communication ICCMC*, Pp. 1106-1111, 2019. Doi: 10.1109/ICCMC.2019.8819633.

[9] A. Khakimov, A. Muthanna and M. S. A. Muthanna, "Study of Fog Computing Structure," *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering Eiconrus*, Pp. 51-54, 2018. Doi: 10.1109/Eiconrus.2018.8317028.

[10] H. Wadhwa and R. Aron, "Fog Computing with the Integration of Internet of Things: Architecture, Applications and Future Directions," 2018 IEEE *Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications ISPA/IUCC/Bdcloud/Socialcom/Sustaincom*, Pp. 987-994, 2018. Doi: 10.1109/Bdcloud.2018.00144.

[11] S. M. Salman, T. A. Sitompul, A. V. Papadopoulos and T. Nolte, "Fog Computing for Augmented Reality: Trends, Challenges and Opportunities," *2020 IEEE International Conference on Fog Computing ICFC*, Pp. 56-63, 2020. Doi: 10.1109/ICFC49376.2020.00017.

[12] M. Jalasri and L. Lakshmanan, "A Survey: Integration of Iot and Fog Computing," *2018 Second International Conference on Green Computing and Internet of Things Icgciot*, Pp. 235-239, 2018. Doi: 10.1109/Icgciot.2018.8753010.

[13] C. Liu, F. Xiang, P. Wang and Z. Sun, "A Review of Issues and Challenges in Fog Computing Environment," *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress DASC/Picom/Cbdcom/Cyberscitech,* Pp. 232-237, 2019. Doi:10.1109/DASC/Picom/Cbdcom/Cyberscitech.2019.00050.

[14] Memari, P., Mohammadi, S.S., Jolai, F. Et Al, “ A Latency-Aware Task Scheduling Algorithm for Allocating Virtual Machines in A Cost-Effective and Time-Sensitive Fog-Cloud Architecture,” *J Supercomput* 2021. Https://Doi.Org/10.1007/S11227-021-03868-4

[15] Davami, F., Adabi, S., Rezaee, A. Et Al, “ Fog-Based Architecture for Scheduling Multiple Workflows with High Availability Requirement,” *Computing*, 2021. Https://Doi.Org/10.1007/S00607-021-00905-1

[16] Singh R.M., Awasthi L.K., Sikka G, “Techniques for Task Scheduling in Cloud and Fog Environment, “ A Survey. in: Singh P., Sood S., Kumar Y., Paprzycki M., Pljonkin A., Hong WC. Eds Futuristic Trends in Networks and Computing Technologies. FTNCT 2019. Communications in Computer and Information Science, Vol .1206. Springer, Singapore. Https://Doi.Org/10.1007/978-981-15-4451-4_53

[17] Wadhwa, H., Aron, R, “ TRAM: Technique for Resource Allocation and Management in Fog Computing Environment, “ *J Supercomput*, 2021. Https://Doi.Org/10.1007/S11227-021-03885-3

[18] Fersi, G, “ Fog Computing and Internet of Things in One Building Block: A Survey and An Overview of Interacting Technologies,” *Cluster Comput* , 2021. Https://Doi.Org/10.1007/S10586-021-03286-4

[19] Pereira E., Fischer I.A., Medina R.D., Carreno E.D., Padoin E.L., “A Load Balancing Algorithm for Fog Computing Environments. in: Crespo-Mariño J., Meneses-Rojas E. Eds High Performance Computing. CARLA 2019. Communications in Computer and Information Science, Vol.1087, 2020. Springer, Cham. Https://Doi.Org/10.1007/978-3-030-41005-65

[20] Yi, Shanhe, Cheng Li, and Qun Li, “A Survey of Fog Computing: Concepts, Applications and Issues,” *in Proceedings of the 2015 Workshop on Mobile Big Data*, Pp. 37-42, 2015.

[21] Yi, Shanhe, Zijiang Hao, Zhengrui Qin, and Qun Li, “Fog Computing: Platform and Applications,” *in 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies Hotweb,* Pp. 73-78, 2015.

[22] Vaquero, Luis M., and Luis Rodero-Merino, “Finding Your Way in the Fog: Towards A Comprehensive Definition of Fog Computing, “ *ACM SIGCOMM Computer Communication Review*, Vol. 44, No. 5, Pp. 27-32, 2014.

[23] Mukherjee, Mithun, Lei Shu, and Di Wang, “Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges,” *IEEE Communications Surveys & Tutorials* , Vol. 20, No. 3 , Pp. 1826-1857, 2018.

[24] Ghobaei-Arani, M., Souri, A. & Rahmanian, A.A. Resource Management Approaches in Fog Computing: A Comprehensive Review,” *J Grid Computing,* Vol.18, Pp.1–42 , 2020. Https://Doi.Org/10.1007/S10723-019-09491-1

[25] Mukherjee, Mithun, Rakesh Matam, Lei Shu, Leandros Maglaras, Mohamed Amine Ferrag, Nikumani Choudhury, and Vikas Kumar,”Security and Privacy in Fog Computing: Challenges, “ *IEEE Access* , Vol.5, Pp. 19293-19304, 2017.

[26] Zhang, Peiyun, Mengchu Zhou, and Giancarlo Fortino, “Security and Trust Issues in Fog Computing: A Survey,”Future Generation Computer Systems , Vol.88 , Pp. 16-27, 2018.

[27] Zhou, Yiqing, Lin Tian, Ling Liu, and Yanli Qi, “Fog Computing Enabled Future Mobile Communication Networks: A Convergence of Communication and Computing, “ *IEEE Communications Magazine*, Vol. 57, No. 5 , Pp.20-27, 2019.

[28] Ning, Zhaolong, Jun Huang, and Xiaojie Wang, “Vehicular Fog Computing: Enabling Real-Time Traffic Management for Smart Cities,” *IEEE Wireless Communications* , Vol. 26, No. 1, Pp.87-93, 2019.

[29] Huang, Cheng, Rongxing Lu, and Kim-Kwang Raymond Choo, “Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges, “ *IEEE Communications Magazine*, Vol. 55, No. 11 , Pp.105-111, 2017.

[30] Byers, Charles C, “Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled Iot Networks,” *IEEE Communications Magazine,* Vol. 55, No. 8 , Pp.14-20, 2017.

[31] Kumari, Aparna, Sudeep Tanwar, Sudhanshu Tyagi, Neeraj Kumar, Mohammad S. Obaidat, and Joel JPC Rodrigues, “Fog Computing for Smart Grid Systems in the 5G Environment: Challenges and Solutions,” *IEEE Wireless Communications* , Vol. 26, No. 3 , Pp.47-53, 2019.

[32] Zahmatkesh, Hadi, and Fadi Al-Turjman, “Fog Computing for Sustainable Smart Cities in the Iot Era: Caching Techniques and Enabling Technologies-An Overview, “ *Sustainable Cities and Society*, Vol. 59 , Pp.102139, 2019.

[33] Osanaiye, Opeyemi, 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* , Vol.5 , Pp. 8284-8300, 2019.

[34] Sookhak, Mehdi, F. Richard Yu, Ying He, Hamid Talebian, Nader Sohrabi Safa, Nan Zhao, Muhammad Khurram Khan, and Neeraj Kumar, “Fog Vehicular Computing: Augmentation of Fog Computing Using Vehicular Cloud Computing,” *IEEE Vehicular Technology Magazine*, Vol. 12, No. 3 2017: 55-64.

[35] Yin, Luxiu, Juan Luo, and Haibo Luo, “Tasks Scheduling and Resource Allocation in Fog Computing Based on Containers for Smart Manufacturing,” *IEEE Transactions on Industrial Informatics*, Vol.14, No. 10 , Pp.4712-4721,2018.

[36] Mutlag, Ammar Awad, Mohd Khanapi Abd Ghani, Mazin Abed Mohammed, Mashael S. Maashi, Othman Mohd, Salama A. Mostafa, Karrar Hameed Abdulkareem, Gonçalo Marques, and Isabel De La Torre Díez, “MAFC: Multi-Agent Fog Computing Model for Healthcare Critical Tasks Management, “ *Sensor,* Vol. 20, No. 7 , 2020.

[37] He, Jianhua, Jian Wei, Kai Chen, Zuoyin Tang, Yi Zhou, and Yan Zhang, “Multitier Fog Computing with Large-Scale Iot Data Analytics for Smart Cities, “*IEEE Internet of Things Journal*, Vol. 5, No. 2 , Pp. 677-686, 2019.

[38] Wan, Jiafu, 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*, Vol.14, No. 10 , Pp.4548-4556, 2018.

[39] Abdel-Basset, Mohamed, Reda Mohamed, Mohamed Elhoseny, Ali Kashif Bashir, Alireza Jolfaei, and Neeraj Kumar, “Energy-Aware Marine Predators Algorithm for Task Scheduling in Iot-Based Fog Computing Applications,” *IEEE Transactions on Industrial Informatics*, Vol.17, No. 7, Pp. 5068-5076, 2020.

[40] Choudhari, Tejaswini, Melody Moh, and Teng-Sheng Moh, “Prioritized Task Scheduling in Fog Computing,” *in Proceedings of the ACMSE 2018 Conference*, Pp. 1-8, 2018.

[41] Luo, Juan, 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,* Vol. 97 , Pp.50-60, 2019.

[42] Tychalas, Dimitrios, 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,* Vol. 98 , Pp.101982, 2020.

[43] Ghobaei-Arani, Mostafa, Alireza Souri, Fatemeh Safara, and Monire Norouzi, "An Efficient Task Scheduling Approach Using Moth-Flame Optimization Algorithm for Cyber-Physical System Applications in Fog Computing," *Transactions on Emerging Telecommunications Technologies*, Vol.31, No. 2, Pp.E3770.

[44] Sun, Yan, Fuhong Lin, and Haitao Xu, "Multi-Objective Optimization of Resource Scheduling in Fog Computing Using An Improved NSGA-II," *Wireless Personal Communications,* Vol. 102, No. 2 , Pp.1369-1385, 2018.

[45] Pham, Xuan-Qui, 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,* Vol. 13, No. 11 2017.1550147717742073.

[46] Mukherjee, Mithun, 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*, Vol. 24, No. 2 , Pp.307-311, 2019.

[47] Li, Guangshun, Yuncui Liu, Junhua Wu, Dandan Lin, and Shuaishuai Zhao, "Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing, " *Sensors*, Vol. 19, No. 9 , Pp.2122, 2019.

[48] Jamil, Bushra, Mohammad Shojafar, Israr Ahmed, Atta Ullah, Kashif Munir, and Humaira Ijaz, "A Job Scheduling Algorithm for Delay and Performance Optimization in Fog Computing," *Concurrency and Computation: Practice and Experience*, Vol. 32, No. 7 , Pp. E5581, 2020.

[49] Anawar MR,Wang S,Azam Ziam, Jadoon AK, Akramu,Raza S , " Fog Computing: An Overview of Big Iot Data Analytics," *Wireless Commun Mob Comput.* Https://Doi.Org/10.1155/2018/7157192