Journal of Turkish

Operations Management

A review of workload challenges in fog computing environment

Omar Anwer Nafea[1*], Turkan Ahmed Khaleel[2]

[1]Department of Computer Engineering, College of Engineering, University of Mosul, Mosul, Iraq
e-mail: omar.21enp5@student.uomosul.edu.iq, ORCID No: https://orcid.org/0009-0002-5966-3602
[2]Department of Computer Engineering, College of Engineering, University of Mosul, Mosul, Iraq
e-mail: turkan@uomosul.edu.iq, , ORCID No: https://orcid.org/0000-0002-4047-8100

Article Info

Article History:
Received: 29.03.2023
Revised: 15.04.2023
Accepted: 26.04.2023

Keywords

Fog Computing,
Workload,
Cloud,
Quality of Service,
Load-Balancing,
IoT.

Abstract

Users nowadays in environments with fog computing require applications that respond quickly to their requests for everything they want to access and work quickly and require to increase in the Quality of Service metrics such as minimum energy consumption, bandwidth efficiency, and reduction latency in a fog network, resulting in an improvement in the system's performance, that is done by getting to know the workload on the network and how to deal with it. In this paper, the various fog computing workloads are described, along with where each one should be executed, in addition, discuss the load-balancing techniques and strategies count as a very important issue and one of the important challenges in the fog computing environment, that play a significant role in resource management like resource provisioning, task offloading, resource scheduling, and resource allocation this will be done based on reviewing previous research and discussing the most important concepts in it.

1. Introduction

Globally, the number of Internet of Things (IoT) devices is constantly growing day by day worldwide and is forecast to almost triple from 9.7 billion in 2020 to more than 29 billion IoT devices in 2030 (Vailshery 2022) as shown in figure 1. Each IoT device has sensors that gather information about the environment in real time. At the IoT layer, huge amounts of data are produced and for processing, this data is sent to a cloud computing system. However, in some applications that are sensitive to time such as health, military operations, and fire control systems a quick response is essential and latency is a major factor. So, to get over this restriction, fog computing is placed between the two layers (the cloud and the end-user layer). In this case, only necessary data will go to the cloud after being quickly processed and responded to by fog computing. fog computing uses edge devices for computing, communication, and storage (Lyu et al. 2018).

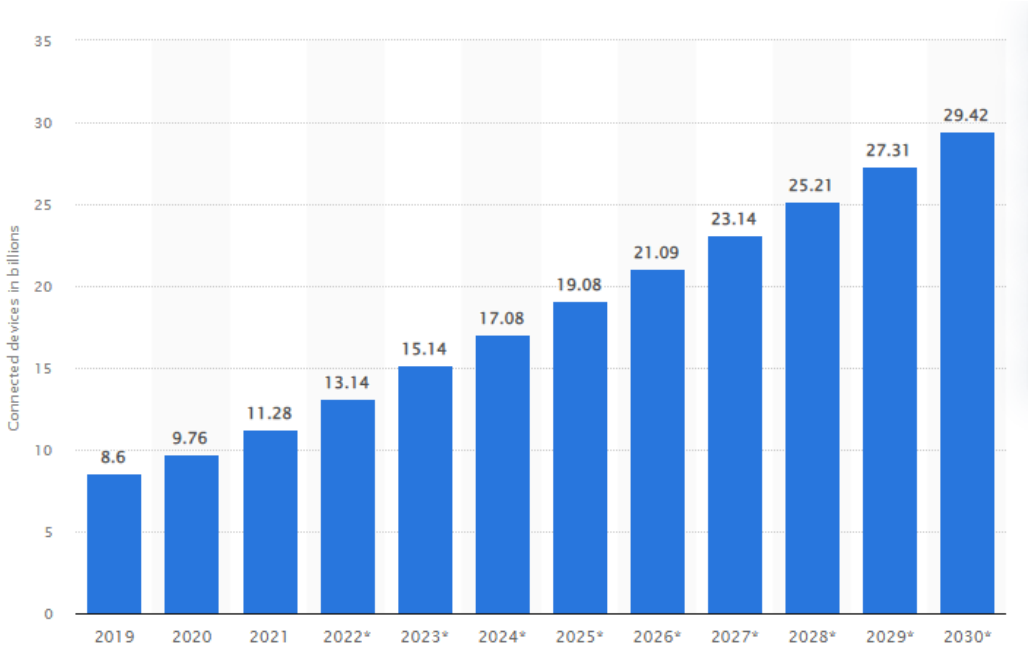

Figure 1. No. of IoT Devices (Vailshery 2022)

In the fog computing layer, the fog nodes receive data that IoT devices have sensed, and with the increased rate of data production, some fog nodes get overloaded. Due to this, processing tasks (data processing) take longer to compute, which affects delivery times. To resolve this problem, fog nodes must successfully collaborate to offload tasks to less overloaded nodes, and the workload should also be distributed through an algorithm of load balancing or approach to the less overloaded nodes. This decreases response time, reduces resource consumption, and improves resource utilization. The load balancing function of fog nodes spreads the workload over all of their resources. It is used to increase resource usage and user satisfaction while ensuring that no single node is overloaded, which enhances system performance as a whole.
The remaining sections are arranged as follows: Section 2 describes fog computing (Characteristics, Architecture, Services), and describes the Challenges with fog computing. Computing workload and Classification of workloads in the cloud and fog are presented in section 3. Section 4 of the paper reviews techniques and strategies of load balancing used in fog computing.

2. Fog computing

A brief description of the fog computing environment, its characteristics, architecture, and the services it provides will be given in this section:

2.1 The Characteristics of Fog Computing

As a compute layer that is located closer to the user layer, where the IoT devices are placed, fog computing provides computing, networking, and storage capabilities. The following features and characteristics of the fog computing layer are presented compared to the cloud computing and edge-network layers to provide these services and achieve the requirements of IoT systems (Costa et al. 2022) (Tim Mell 2009) (Rahul and Aron 2021) (Kumari, Singh, and April 2017):

- **Geographical Distribution:** Unlike cloud computing, fog computing architecture requires geo-distributed deployment and administration of services and applications (He et al. 2018).
- **low latency:** In comparison to activities performed by a cloud service, since fog computing nodes are located closer to the users, they can analyze and respond to generated and requested data more quickly (Shi et al. 2018).
- **Heterogeneity**: The ability to collect and process data from various sources and through a variety of network connection methods.

- **Real-time interactivity**: Unlike cloud computing, where batch processing is the norm, real-time interactions are possible due to the proximity of devices and fog nodes.
- **Scalability:** supports resource flexibility by enabling quick detection of changes in network and device conditions and variations in workload response times.
- **Support for mobility**: Many fog applications demand direct communication with mobile devices; therefore, they must support mobility techniques.
- **Improved service quality (QoS):** When compared to cloud computing numerous parameters including reliability, bandwidth, and connection health fared better in fog computing (Mahmud et al. 2019). As a result, the fog computing technique hence improving the Quality of Service (QoS).
- **Increased security:** fog computing offers increased security. It is possible to create policies and procedures to safeguard the network's fog nodes (P. Zhang et al. 2018).

2.2 Fog Computing Architecture

Fog computing expands on cloud computing by offering computational resources to execute services closer to the end users or the end device layer (IoT). Architecture with three layers is one of the common architectures which include (IoT or end device layer, fog computing layer, and cloud computing layer), where the fog layer comprises everything between the cloud and the end users (Habibi et al. 2020) (Rahimi, Songhorabadi, and Kashani 2020):

- **Tier 1:** In this lowest level layer (End devices layer) you can find all IoT devices that can store and communicate unprocessed data to its upper layer.
- **Tire 2:** It's the middle layer (fog layer), is made up of numerous network devices that can process and temporarily store data, including computing devices, switches, and routers, these units are linked to the cloud network and will keep sending data there regularly.
- **Tire 3:** The top layer (cloud layer) has numerous servers and data centers. which are capable to handle a high amount of data and capacity to store it too. Figure 2 shows the fog computing architecture with its three layers.

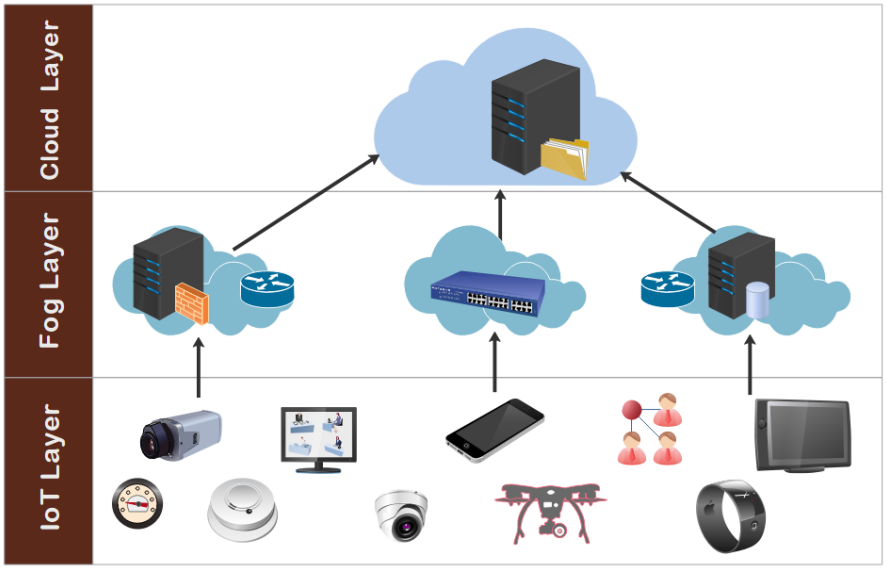

Figure 2. Fog Computing Architecture

2.3 Services in Fog Computing

The fog computing layer, being close to the Internet of Things-IoT layer, is used to provide many services that are an extension of cloud computing services, this section offers a summary of several services at the fog layer, which are divided into three categories: processing, storing, and network services(Negash et al. n.d.):

- **Storage Services**

The sensor nodes can produce enormous amounts of data. When you take into account the rate of data generation, the storage space offered by the devices at the perception layer is frequently insufficient to retain even a day's worth of data. As was previously mentioned, it is not required to transmit all data directly to the cloud, especially when there are redundant or irrelevant data. In these situations, it would be prudent to filter the data and temporarily store it in the intermediate fog layer (Rahmani et al. 2017).

- **Computing services**

The advent of remote processing techniques is a result of the computing capabilities of the devices in the perception layer being constrained. Processing at the fog layer is motivated by the need to better meet system requirements and maintain energy efficiency, as well as to provide local processing and a faster response, that reduces processing available in the cloud can be brought down into the fog computing layer (Hu et al. 2016) (Datta, Bonnet, and Haerri 2015).

- **Communication Services**

Wireless nodes are largely responsible for communication in the Internet of Things (IoT), because of resource limitations in the IoT layer these wireless protocols are optimized for low power operation, narrowband transmission, or greater range of coverage. The market currently offers a wide variety of alternative protocols (Sheng et al. 2013).

2.4 Challenges

Fog computing still faces numerous difficulties in its early stages, even though it brings cloud computing services closer to the user and offers several extra advantages over cloud computing. The following issues face fog computing, according to (Hao et al. 2017) (Singh et al. 2019):

- **Heterogeneity:** because fog computing nodes are made up of a variety of devices, including sensors, laptops, mobile phones, and desktop computers, heterogeneity is a challenge with how different devices communicate with one another to execute tasks(Cinzia Cappiello 2018).
- **Load balancing:** is an expression in cloud and fog computing that is essential to achieving a time- and energy-efficient system., we will discuss this in the next sections.
- **Resource management:** to enhance resource scheduling, resource management may require to be properly planned (Yi et al. 2016). where the performance of fog computing applications depends on resource discovery and sharing. For maximizing resource availability and reducing energy consumption, resource-sharing optimization algorithms can be developed (Yi et al. 2016) (Dastjerdi and Buyya 2016).
- **The policy of connection: a**dditional difficulties include how the fog nodes are interconnected and how to use the fog nodes that are available to handle the workload (Aazam, Zeadally, and Harras 2018).
- **Strategy for deployment:** The deployment plan is the largest obstacle. How are the workloads distributed across the fog nodes that make up the fog network? (Lin and Yang 2018).
- **Offloading of tasks**: Offloading of tasks is the process of carrying tasks from the cloud and fog nodes to another node. This choice of execution is based on several factors, such as the amount of computing power required for the task, which is offloaded to the cloud for execution, and the amount of computing power needed for the task, which is handled on the fog nodes. Task offloading can also take into account latency, with latency-sensitive tasks running on fog nodes and non-latency-sensitive tasks being sent to the cloud (Qiao et al. 2018) (G. Zhang et al. 2018).

3. Comparison between Fog and Cloud Computing Workload

Any program or application that is active on a computer is considered a workload in the field of computing. A workload can be as simple as a contact app running on a smartphone or as complex as an enterprise application that runs on one or more servers and communicates with several client systems over a large network. The workload can also be used to describe how much amount of work puts on the underlying computational resources. The amount of time and computing power needed to complete a certain activity or create an output from given inputs is generally referred to as an application's workload. A light workload uses few computing resources, such as processors, CPU clock cycles, storage I/O (input/output), and other components, to complete its specified tasks or performance objectives. Significant amounts of computing resources are needed to handle a big workload. It is challenging to group all workloads into a single set of consistent criteria because workloads are intended to execute a myriad of distinct jobs in a multitude of various ways, Workloads, for instance, could be classed as static or dynamic. An operating system (OS) is an example of a static workload that is constantly active. A dynamic workload is transient and only loads and runs when necessary. Numerous new workload classifications have been created as a result of the enormous diversification of software development (Bigelow n.d.).

3.1 The Workload in Cloud Computing

"Cloud workload" is the term used to describe the amount of work produced by a variety of apps and services deployed on cloud infrastructures. Or is a particular program, service, function, or amount of work that can be executed on a cloud resource? cloud workloads include virtual machines, databases, containers, and applications. The cloud computing paradigm has recently seen a substantial increase in popularity due to virtualization technologies. However, due to the heterogeneity and varying needs for resources in cloud applications, new difficult problems with workload allocation and scheduling solutions have emerged. cloud computing workload can be characterized by focusing on (Calzarossa, Maria Carla, Luisa Massari 2016):

- Virtual data centers
- cloud infrastructures
- Services for computers and storage

3.2 Classification of Workloads in The Cloud

Workloads must be categorized according to their design, resource requirements, and usage patterns to determine whether private, public, or hybrid cloud environments are the most appropriate for them. So, according to their resource requirements, cloud workloads can be divided into the following categories(Stephanie Vozza 2022):

- Workloads that don't require specialized computation and run on the cloud's default setup are referred to as general computing. Web servers, distributed data stores, common web programs, and containerized microservices are some of them.
- Workloads that require a lot of processing power and can manage several users at once are referred to as CPU-intensive. Deep learning applications and massively multiplayer online games that require processing-intensive tasks like video encoding, large data analytics, 3D modeling, etc. fall under this category.
- Workloads that require a lot of memory and processing capacity to execute, are referred to be memory-intensive. Caches, distributed databases, and real-time streaming data are examples.
- Some workloads that need GPU-accelerated computing, including seismic analysis, self-driving cars, navigation systems, and speech recognition, have extremely high processing demands. Do real-time operations, that require the strength of GPUs in addition to CPUs.
- Workloads like in-memory databases are storage optimized.

Workload availability and the Volume of traffic are more significant. So, the following patterns can be used to categorize cloud workloads:

- **Static Workload:** In general, the requirements for resources, demand, and uptime are well defined. These include crucial enterprise services including CRM, ERP, and email.

- **Periodic workloads:** During specific hours of the day, week, month, or year, these see an increase in traffic, such as payment for bills or using accounting and tax software. The workloads that serverless computing, in which clients do not pay for flawless instances, is most suited to handle.
- **Unpredictable workloads:** Popular platforms and apps like online multiplayer games, video streaming websites, social networks, etc., might experience a rapid rise in traffic. clouds' auto-scaling capabilities can manage these spikes by dynamically adding instances as needed.

3.3 Fog Computing Workload

The workload is helpful when assessing a system's performance, the system's capacity to handle the burden can be translated into evaluating performance. A single computer or a network of computers can make up the system. The time it takes between a user request and the system's response is one way to evaluate the performance of such a system. The system's throughput, which shows how much work can be done each time, is another critical metric. Similar measurements like availability, reliability, etc. However, the user's needs are the factor that should be taken into account when choosing performance measures. There is no accurate definition of workload in the literature because different researchers have described and used different definitions to depict it. The workload has been popularly referred to as a task or a job. From the literature, it may be assumed that a workload can be viewed as an application or service that has been deployed to the cloud or fog. As a result, the workload could range from a simple single service to one that is enormous and made up of hundreds of micro-services cooperating.
The distribution of computing tasks among the available resources determines the effectiveness of the system as a whole. The workload itself affects the relationship between the job and the resources Thus, according to the task's requirements, workload classification plays a crucial part in identifying the proper resource allocation, which enhances QoS. As a result, in the fog and cloud computing environments, a systematic and structured approach to workload classification may aid in an accurate prediction of incoming resource requests that comply with QoS criteria. (Singh and Chana 2016).
According to Z. Raza, et al. (Raza and Jangu 2022) to understand the workload, it was divided into three classifications, according to their characteristics:

- **Quantitative attributes:** refer to the characteristics of request/workload that determine the number of resources required by them. These resources may include the computing, networking, or storage requirements necessary to finish or properly manage requests.
- **Qualitative attributes:** refer to the non-quantifiable characteristics that a workload holds, such as a deadline, sensitivity to delay, tolerance to latency, priority, and many more that specify the characteristics of a workload.
- **Non-functional requirements:** like performance, security, availability, and privacy that are connected to (SLA) and improve QoS.

4. Load Balancing

In computing networks, some nodes can occasionally carry all of the network's load while others can occasionally remain underloaded. The servers' uneven load may cause problems such as system failure, energy consumption, network failure, and longer execution times (Mishra, Sahoo, and Parida 2020) (Sultan and Khaleel 2022), as a result, Load-balancing becomes crucial to managing the load on computational nodes.

In general Load Balancing is a technique for evenly distributing incoming requests among several servers, so the workload is distributed equally(Kaur and Aron 2021). The employment of load Balancing methods helps to prevent overloaded servers. The load balancer is used to monitor traffic between servers and user requests as a traffic monitoring program. Incoming requests will be routed to the remaining available servers whenever a server goes offline, according to the load balancer, the load balancer will instantly forward requests to any new servers when they are introduced(Talaat et al. 2020). Utilizing Load Balancing methods has the following benefits: reduce waiting times; reduce response times; maximize resource use; boost throughput; enhance reliability; and improve performance.

The process of workload load-balancing in fog computing and the most important related algorithms will be the main topic of the next section.

4.1 Balancing Workload in Fog computing

As previously mentioned by the researchers, load balancing is required to control or to evenly spread the workload equally in computing nodes according to their capability, which leads to making effective use of all available resources, this guarantees that are no over-utilized or underutilized resources(Téllez et al. 2018) (Velde and Rama 2017). Among the few properties of load balancing, the Workload is evenly distributed across all nodes, resources are efficiently used, the system performs better, less energy is consumed, the user is more satisfied, and the response time is sped up (Sumathy and Manju 2019).

In a fog environment, load balancing enhances the throughput of task processing at fog nodes. fog nodes contain resources from both the network and end users and have distributed geographical locations; they preprocess the work before forwarding it to a cloud data center. It is the responsibility of the service broker to route user requests to the appropriate data centers, where they must be handled by their priority and related processing expense, The network resource at each location in this manner is the service broker (Chiang and Zhang 2016). While some applications need to complete the work at the lowest possible cost, others need to respond quickly regardless of the cost of processing. The data center nearby becomes overcrowded when a large number of requests originate from the same geographic region, hence load balancing measures are required to distribute the workload to prevent overload at the same data center, the same scenario in the fog environment, to evenly distribute the load, correct load distribution rules must be implemented whenever a user's work at a specific location becomes overwhelmed due to a particular fog cluster. The ultimate goal in a fog environment is real-time computing and fast response, thus the load-balancing strategy shouldn't take too long. A load balancing technique in a fog environment must be adaptive for any changes in resources of the fog computing environment (Sumathy and Manju 2019). Although it is possible to apply the load-balancing strategies used in the cloud to the fog computing environment, they must be changed to take into consideration the resources and tasks that are available there. fog networking and its architecture have drawn more scholarly attention as the number of Internet of Things devices has increased (Mao et al. 2017).

Figure 3 shows how a load balancer distributes workload to the compute servers by taking it from various users. The available servers on the network are routinely observed by a load-balancer, when it receives workload from several users, it first checks to see if resources are available before distributing the workload between all the computing resources to minimize network overload.

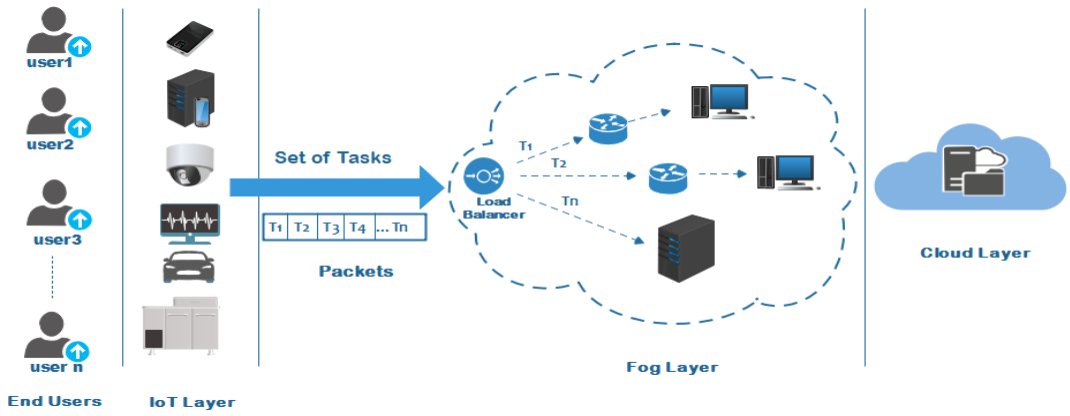

Figure 3. Load-Balancing in fog network

The load was separated into several categories, including CPU load, storage device load, and network load. Load balancing is the process of identifying overloaded and lightly laden nodes and then distributing the workload evenly across them all. System performance can be enhanced by using fog resources effectively. Resources used in fog can be physical (hardware) or virtual(Milani and Navimipour 2016). according to (Kaur and Aron 2021) The procedure of load balancing used in fog is shown in the flow diagram in Figure 4.

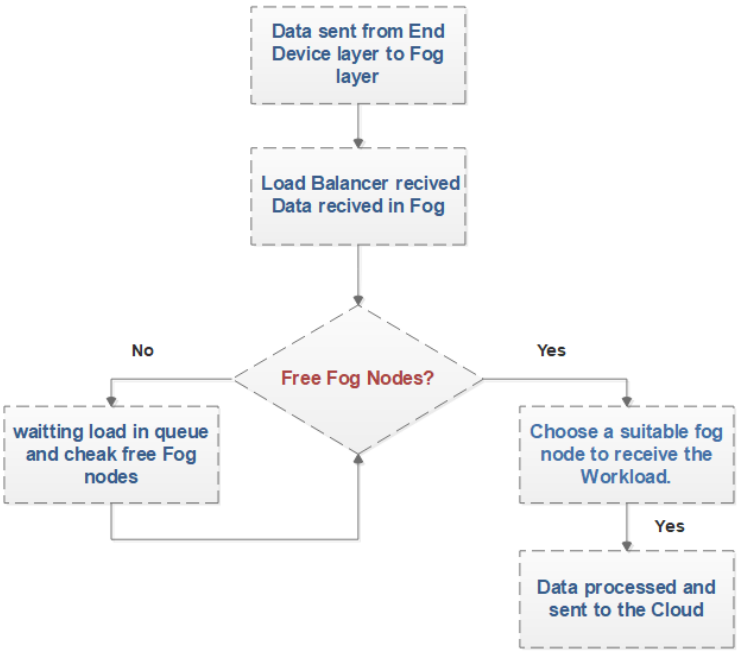

Figure 4. Load Balancing Flow Diagram in a fog Environment.

4.2 Load Balance Techniques in Fog Environment

The load balancing has a few objectives, such as traffic optimization, response time reduction, throughput maximization, reduction request processing times, optimize server-side resource consumption, and increase scalability in distributed environments(Adhianto et al. 2017), this led to improving the pace at which programs executed on resources due to the unpredictability of their execution times at runtime. In fog computing, load balancing is utilized on both virtual machines and physical nodes (fog nodes), and all of the processing nodes receive an equal share of the load. In fog networks different load balancing techniques can be used or done in two ways static and dynamic(Neghabi et al. 2018) (Verma, Bhardawaj,Yadav 2015):

- **static techniques** (Singh et al. 2020) (Baek et al. 2019)**:** In fog computing, the load is balanced by splitting the traffic into equal amounts and distributing them among the servers. The equal distribution of load among the servers in static load balancing provides prior knowledge of the applications, statistical data, and the system's resources that will be used. This kind of load balancing is not movable as soon as the process starts, it cannot be transferred to another machine while it is in operation because the tasks are not assigned to the processor until they have been created.
- **Dynamic techniques**(Singh et al. 2020) (Baek et al. 2019): In the case of dynamic balancing, the server that manages the least amount of load is checked for and chosen for load balancing. These algorithms make use of real-time network communication and consider the current system state to be in control of the network load. The dynamic load balancing method finds out how to look for servers that aren't overloaded before assigning the right amount of work to them. Real-time load balancing occurs between unused and overused fog nodes. The workload is distributed across the processors during operation. Although the algorithms used in this method are thought to be complex, their fault tolerance and overall performance are superior.

We will provide a brief review of load balancing techniques done by several researchers given as follows:

- M. Verma et al. (Verma 2016) developed a real-time efficient scheduling (RTES) load balancing algorithm in a fog computing environment, that has been suggested and implemented in the cloudSim tool. In comparison to other algorithms used in the fog computing environment, such as FCFS, Priority, and Multi-Objective Tasks Scheduling Algorithm, the results obtained after implementing the proposed architecture and algorithm are good. They have provided minimal execution time, quick response to client

requests, and completion of real tasks ahead of schedule, while maintaining data consistency and proper resource and bandwidth utilization. the suggested algorithm is 90% effective, and in the future, it can be further enhanced by adding other QoS aspects like security, etc.

- S. Hamrioui et al. (Hamrioui, Lorenz, and Grtc 2017)The suggested technique, known as "LBA-Ie (Load Balancing Algorithm for IoT Communications inside e-Health Environment)", is based on the integration of IoT communication characteristics in the flow management process provided by TCP. "LBA-Ie" is a self-organizing and adaptable algorithm that takes into account network changes, link conditions, and object properties. Energy efficiency and QoS (Quality of Service) are measured when evaluating "LBA-Ie". The simulation's output is contrasted with three alternative solutions' findings. By enhancing data reliability, "LBA-Ie" raises the quality of service (QoS) of IoT communications, which enhances e-health applications. LBA-Ie enables objects to use less energy overall, extending their typical lifespan in the process.
- D. Puthal, et al. (Puthal et al. 2018) suggested a novel load-balancing approach for "EDCs" (Edge Data Centers) in a fog computing environment. It is secure and long-lasting. The suggested load-balancing technique is essentially split into two parts, the first of which focuses on the secure authentication of the region's EDCs using cloud-initiated credentials, and the second of which focuses on the sustainable load balancing architecture by obtaining load information from the destination EDCs. Utilizing both theoretical analysis and experimental evaluation, the suggested solution has been assessed in two distinct methods. According to the performance evaluation and comparison results, the suggested method is safe and tenable because it uses the destination EDC's load during authentication.
- Q. Fan et al (Fan, Member, and Ansari 2018) suggested the load balancing "(LAB)" scheme for the fog network, to reduce the average latency of data flows from IoT devices, By connecting IoT devices to appropriate Base stations BSs/fog nodes, LAB accounts for both the distribution of computation and traffic loads. The IoT device association concentrates on balancing the traffic loads among BSs when the network's traffic demand is greater than its processing load. Similarly, to this, compute latency becomes the dominant factor in the average latency ratio when the network's computing demand is high, making the fog nodes the bottleneck. However, LAB may still lower the average latency by modifying the IoT device association to balance the traffic load and computation load simultaneously, this is accomplished by creating a distributed algorithm that iteratively finds the best solution.
- M. Jimeno et al. (Jimeno et al. 2018)suggested a "Tabu" Search method for the optimal load balancing across cloud and fog nodes that takes into account resource constraints. For job scheduling and execution, the suggested design supports the integration of fog and cloud Nodes. The architecture can be modified to support more coordinator nodes. Adopting the Tabu Search Method has the primary benefit that certain layers' online calculations are efficient, and jobs should be handled as they come in.
- H. Zhuang, et al. (Zhuang et al. 2018) present "SSLB", a self-similarity-based load balancing strategy for large-scale fog computing. It fully exploits the benefits of both centralized and decentralized systems. Even as the fog increases in size, load balancing overhead may be kept to a minimum with this structure. We suggest an adaptive threshold strategy that properly and dynamically determines the load threshold on each node to ensure SSLB's efficiency. Furthermore, work distribution and task grasping are two proposed scheduling techniques. Results from experiments indicate that SSLB performs better than conventional systems, particularly when the fog scale is quite large.
- A mathematical search optimization method called the HCLB (Hill Climbing Load Balancing) algorithm was proposed by K. Hassan, et al (Hassan et al. 2019). Finding accessible VMs relies on a random solution. This algorithm depends on the repetition of execution until the ideal answer to a problem is discovered. The loop in HCLB is increased until the closest available VM is identified. The best VM is then chosen and given the task of processing requests.
- According to S. Sumathy et al. (Sumathy and Manju 2019) the main objective is to divide the workload across available fog devices, with task processing being minimized reaction time. The min-min method was applied to each cluster with success. A few of the many variables that the authors take into account include the distance between the cloud server and the closest cluster node and the capacity of each cluster to manage a waiting queue of work. The efficiency of the framework is higher and better for smaller cluster nodes. In the future, the proposed framework might be used in conjunction with an appropriate cloud computing environment.
- The priority-based request serving at fog computing centers was discussed by G. Chowdhary, et al. (Chowdhary and Rathod 2019), by focusing on the scenario in which a fog node in a fog computing center (FCC) is overloaded with the workload. the increased workload is transferred to nearer fog nodes instead

of the distant cloud. The originality of the suggested method is demonstrated by the ability to reduce the offloading of high-priority requests to other nodes by 11%.

- M. Maywood et al. (Mohd, Maswood, and Alharbi 2020) addressed load balancing strategies by proposing a novel Mixed-Integer Linear Programming (MILP) based optimization model in a three-tire cloud-fog computing system. The major goals of this work were to balance workloads (CPUs' processing capacity) and reduce bandwidth costs (network resources) in a "three-layer cloud -fog computing environment". Utilizing simulation tools, the suggested method's efficacy was assessed.
- F. Alqahtani (Alqahtani and Amoon 2021) suggests a dependable scheduling approach using the resources in cloud-fog environments, To better allocate requests to resources, the Load Balanced Service Scheduling Approach (LBSSA) classifies requests into three categories: real-time, important, and time-tolerant. The recommended approach also considers the resource failure rate when scheduling requests to ensure high reliability for requested services. To handle different requests, the approach provides several algorithms. cloudSim simulation tests are run to assess the LBSSA approach's performance in terms of computing resources available, how well they are being used, how load balance varies, and how long it takes to operate.
- M.Kaur, et al. (M.Kaur and Aron, R. 2021) proposed a load-balancing model for scientific workflow applications in the fog computing environment, the proposed algorithm Tabu-GWO-ACO combines the three algorithms Tabu search, Grey Wolf Optimization (GWO), and Ant Colony Optimization (ACO) into one hybrid form. The underloaded and overloaded fog nodes are identified using the Tabu search algorithm, and the fog nodes are subsequently optimized using GWO and ACO. This paper also suggested fog computing architecture of load balancing (FOCALB), a fog computing framework based on load balancing. This study takes into account scientific process applications to evaluate the effectiveness of the suggested strategy. The performance of Tabu-GWO-ACO is compared to that of other existing techniques to outperform them. By effectively balancing the load in the setting of fog computing, the suggested approach primarily seeks to improve resource usage.

Table 1: Provides a summary review of load balancing technique done by several researchers.

Author	Year	Algorithm	Main Focus
M. Verma et al.	2016	Real-time efficient scheduling "(RTES)"	_____
S. Hamrioui et al.	2017	"The Load-Balanced Service Scheduling Approach (LBSSA)"	Reduce latency and improve service quality.
D. Puthal, et al	2018	novel load balancing approach for "Edge Data Centers (EDCs)"	Improved quality of service and energy efficiency, but low scalability and specific to e-health.
Q. Fan et al	2018	Load Balancing (LAB)	Increasing security is More effective but with limited scalability.
M. Jimeno et al	2018	"(Tabu)" Search technique for the best load balancing across cloud and fog nodes	Reduce latency in computing and communications but don't take energy efficiency, cost, and bandwidth into account
H. Zhuang, et al	2018	"A self-similarity-based load balancing strategy for large-scale fog computing (SSLB)"	Reduce the amount of memory used and the cost of computing, however, complexity is great and scalability is not being considered.

K. Hassan, et al	2019	"(HCLB) Hill Climbing Load Balancing algorithm"	Low Overhead and High Scalability, but Energy consumption and Throughput research are needed.
S. Sumathy et al.	2019	Centralized load balancing algorithm	Reduced response and processing time and decreased network delay, however, security concerns are not taken into account.
G. Chowdhary, et al	2019	The priority-based request serving at fog computing centers	Reduce task processing response times and costs, however, centralized approaches use more energy.
M. Maswood et al.	2020	"Mixed-Integer Linear Programming (MILP)"	Minimize offloading but need to work on modeling techniques.
F. Alqahtani	2021	"The Load-Balanced Service Scheduling Approach (LBSSA)"	Reduce bandwidth costs but the energy consumption is not taken into account.
M.Kaur, et al.	2022	"The proposed algorithm Tabu-GWO-ACO"	efficient resource load balancing and scheduling but it is high Complexity.

5. Conclusions

Fog and the cloud have both been included in a computing environment. In this article, we look at the fog computing environment's traits, architecture, and services. fog computing is used as a system with a quick response time. Therefore, it should be used before cloud computing to get a quick response. The architecture of fog computing is described, which is three-layer: cloud computing, fog computing, and the Internet of Things. A fog computing environment is different from a cloud and according to this difference between both fog and cloud, to get the most out of fog architecture, it is crucial to intelligently distribute the workload by choosing the resources that are most appropriate for the task's characteristics. By dividing the workload between the two tiers (the cloud and the fog), a well-defined characterization and classification increase system performance and raise the attainable QoS. We also demonstrated the load-balancing strategies and methods currently employed in fog computing.

6. Acknowledgments

The authors are fully gratitude the University of Mosul – college of engineering - Computer Engineering department for their help in raising the standard of this paper.

Contribution of researchers

Authors have equal contribution in all the sections as shown:

- The fog environment has been described, including its characteristics, architecture, and services as described in paragraph 2.
- Classification of the Workload in cloud – fog computing as described in paragraph 3.
- Review and discuss some earlier research on load-balancing strategies in the fog environment as described in paragraph 4.

Conflict of interest

The authors declare that there is no conflict of interest.

References

Aazam, M., Zeadally, S., & Harras, K. A. (2018). Deploying Fog Computing in Industrial Internet of Things andIndustry 4.0. IEEE Transactions on Industrial Informatics, 14(10), 4674–4682.doi https://doi.org/10.1109/TII.2018.2855198.

Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J., & Tallent, N. R. (2017). A survey on load balancing algorithms for virtual machines placement in cloud computing. Concurrency and Computation: Practice and Experience, 22(6), 685–701.doi https://doi.org/10.1002/cpe.4123.

Alqahtani, F., & Amoon, M. (2021). Reliable scheduling and load balancing for requests in cloud-fog computing. Peer-to-Peer Networking and Applications, 14, No. 4. doi https://doi.org/10.1007/s12083-021-01125-2.

Baek, J., Kaddoum, G., Garg, S., Kaur, K., & Gravel, V. (2019). Managing Fog Networks using Reinforcement Learning Based Load Balancing Algorithm. April, 15–18. https://ieeexplore.ieee.org/document/8885745

Bigelow, S. J. (n.d.). Workload. Retrieved November 6, 2022, from https://www.techtarget.com/searchdatacenter/definition/workload

Calzarossa, Maria Carla, Luisa Massari, D. T. (2016). Workload Characterization: A Survey Revisited. ACM Computing Surveys (CSUR)., Vol. 48, N(3), pp 1–43. https://doi.org/10.1145/2856127.

Chiang, M., & Zhang, T. (2016). Fog and IoT: An Overview of Research Opportunities. IEEE Internet of Things Journal, 3(6), 854–864.doi https://doi.org/10.1109/JIOT.2016.2584538.

Chowdhary, G., & Rathod, D. (2019). Load Balancing of Fog Computing Centers : Minimizing Response Time of High Priority Requests. Int. J. Innov. Technol. Explor. Eng., 8,(October), no. 11, pp. 2713–2716. doi https://doi.org/10.35940/ijitee.K2171.0981119.

Cinzia Cappiello, P. P. and M. V. (2018). A Data Utility Model for Data-Intensive Applications in Fog Computing Environments. In Fog Computing: Concepts, Frameworks and Technologies. Springer International Publishing. doi https://doi.org/10.1007/978-3-319-94890-4.

Costa, B., Bachiega, J., De Carvalho, L. R., & Araujo, A. P. F. (2022). Orchestration in Fog Computing: A Comprehensive Survey. ACM Computing Surveys, 55(2).doi https://doi.org/10.1145/3486221 .

Dastjerdi, A. V., & Buyya, R. (2016). Fog Computing: Helping the Internet of Things Realize Its Potential. Computer, 49(8), 112–116.doi https://doi.org/10.1109/MC.2016.245 .

Datta, S. K., Bonnet, C., & Haerri, J. (2015). Fog Computing architecture to enable consumer centric Internet of Things services. Proceedings of the International Symposium on Consumer Electronics, ISCE, 1–2. doi https://doi.org/10.1109/ISCE.2015.7177778

Fan, Q., Member, S., & Ansari, N. (2018). Towards Workload Balancing in Fog Computing Empowered IoT. IEEE Transactions on Network Science and Engineering, PP(X), 1. doi https://doi.org/10.1109/TNSE.2018.2852762.

Habibi, P., Member, S., Farhoudi, M., Leon-garcia, A., & Fellow, L. (2020). Fog Computing : A Comprehensive Architectural Survey. 69105–69133. https://ieeexplore.ieee.org/abstract/document/9046806.

Hamrioui, S., Lorenz, P., & Grtc, M. (2017). Load Balancing Algorithm for Efficient and Reliable IoT Communications within E-health Environment. IEEE Global Communications Conference, 1–6. https://doi.org/10.1109/GLOCOM.2017.8254435.

Hao, Z., Novak, E., Yi, S., & Li, Q. (2017). Challenges and Software Architecture for Fog Computing. IEEE Internet Computing, 21(2), 44–53.doi%20https:/doi.org/10.1109/MIC.2017.26.

Hassan, K., B, N. J., Zahid, M., & Ansar, K. (2019). Hill Climbing Load Balancing Algorithm on Fog Computing: Vol. vol 24. Springer International Publishing. doi https://doi.org/10.1007/978-3-030-02607-3.

He, S., Cheng, B., Wang, H., Xiao, X., Cao, Y., & Chen, J. (2018). Data security storage model for fog computing in large-scale IoT application. INFOCOM 2018 - IEEE Conference on Computer Communications Workshops,

39–44.doi https://doi.org/10.1109/INFCOMW.2018.8406927.

Hu, P., Member, S., Ning, H., Member, S., Qiu, T., & Member, S. (2016). Fog Computing-Based Face Identification and Resolution Scheme in Internet of Things. 3203(c), 1–11. doi https://doi.org/10.1109/TII.2016.2607178.

Jimeno, M., Téllez, N., Salazar, A., & Nino-ruiz, E. D. (2018). A Tabu Search Method for Load Balancing in Fog Computing. Int. J. Artif. Intell., 16,(September), no. 2, pp. 106–135. https://www.researchgate.net/publication/327752530_A_Tabu_Search_Method_for_Load_Balancing_in_Fog_Computing.

Kaur, M., & Aron, R. (2021). A systematic study of load balancing approaches in the fog computing environment. In Journal of Supercomputing (Vol. 77, Issue 8). Springer US.doi https://doi.org/10.1007/s11227-020-03600-8.

Kumari, S., Singh, S., & April, M. (2017). Fog Computing : Characteristics and Challenges. Vol.6(2), 113–117. https://www.researchgate.net/publication/340272352_Fog_Computing_Characteristics_and_challenges.

Lin, C. C., & Yang, J. W. (2018). Cost-Efficient Deployment of Fog Computing Systems at Logistics Centers in Industry 4.0. IEEE Transactions on Industrial Informatics, 14(10), 4603–4611. doi https://doi.org/10.1109/TII.2018.2827920.

Lyu, X., Ren, C., Ni, W., Tian, H., & Liu, R. P. (2018). Distributed Optimization of Collaborative Regions in Large-Scale Inhomogeneous Fog Computing. IEEE Journal on Selected Areas in Communications, 36(3), 574–586.doi https://doi.org/10.1109/JSAC.2018.2815359 .

M.Kaur and Aron, R. (2021). FOCALB : Fog Computing Architecture of Load Balancing for Scientific FOCALB : Fog Computing Architecture of Load Balancing for Scientific Workflow Applications. Journal of Grid Computing, Vol 19, No(January 2022), 1-22.doi https://doi.org/10.1007/s10723-021-09584-w.

Mahmud, R., Srirama, S. N., Ramamohanarao, K., & Buyya, R. (2019). Quality of Experience (QoE)-aware placement of applications in Fog computing environments. Journal of Parallel and Distributed Computing, 132(August 2019), 190–203.doi https://doi.org/10.1016/j.jpdc.2018.03.004.

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A Survey on Mobile Edge Computing: The Communication Perspective. IEEE Communications Surveys and Tutorials, 19(4), 2322–2358. doi https://doi.org/10.1109/COMST.2017.2745201.

Milani, A. S., & Navimipour, N. J. (2016). Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends. Journal of Network and Computer Applications, 71, 86–98. doi https:/doi.org/10.1016/j.jnca.2016.06.003.

Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: A big picture. Journal of King Saud University - Computer and Information Sciences, 32(2), 149–158. doi https://doi.org/10.1016/j.jksuci.2018.01.003.

Mohd, M., Maswood, S., & Alharbi, A. G. (2020). A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in A Cooperative Three-Layer Fog-Cloud Computing Environment. IEEE Access, 8, 113737–113750.doi https://doi.org/10.1109/ACCESS.2020.3003263.

Negash, B., Rahmani, A. M., Liljeberg, P., & Jantsch, A. (2018). Fog Computing Fundamentals in the Internet-of-Things. In Fog Computing in the Internet of Things (pp. 3–13). Springer International Publishing. doi https://doi.org/10.1007/978-3-319-57639-8_1.

Neghabi, A. A., Navimipour, N. J., Hosseinzadeh, M., & Rezaee, A. (2018). Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature. IEEE Access, 6, 14159–14178.doi https://doi.org/10.1109/ACCESS.2018.2805842.

Puthal, D., Obaidat, M. S., Nanda, P., Prasad, M., Mohanty, S. P., & Zomaya, A. Y. (2018). Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing. May, 60–65. https://ieeexplore.ieee.org/abstract/document/8360851.

Qiao, G., Leng, S., Zhang, K., & He, Y. (2018). Collaborative task offloading in vehicular edge multi-access networks. IEEE Communications Magazine, 56(8), 48–54. doi https://doi.org/10.1109/MCOM.2018.1701130.

Rahimi, M., Songhorabadi, M., & Kashani, M. H. (2020). Fog-based smart homes: A systematic review. Journal of Network and Computer Applications, 153(March).doi https://doi.org/10.1016/j.jnca.2020.102531

Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., & Liljeberg, P. (2017). Exploiting Smart E-Health Gateways at the Edge of Healthcare Internet-of-Things : A Fog Computing Approach. Future Generation Computer Systems.doi https://doi.org/10.1016/j.future.2017.02.014.

Rahul, S., & Aron, R. (2021). Fog computing architecture, application and resource allocation: a review. In CEUR Workshops (Vol. 4638, pp. 0-2).

Raza, Z., & Jangu, N. (2022). Workload Classification For Better Resource Management in Fog-Cloud Environments. International Journal of Systems and Service-Oriented Engineering, 12(1), 1–14. doi https://doi.org/10.4018/ijssoe.297135.

Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., McCann, J., & Leung, K. (2013). A Survey on The Ietf Protocol Suite for The Internet of Things: Standards, challenges, and Opportunities. IEEE Wireless Communications, 20(6), 91–98.doi https://doi.org/10.1109/MWC.2013.6704479.

Shi, C., Ren, Z., Yang, K., Chen, C., Zhang, H., Xiao, Y., & Hou, X. (2018). Ultra-low latency cloud-fog computing for industrial Internet of Things. IEEE Wireless Communications and Networking Conference, WCNC, 2018-April, 1–6.doi https://doi.org/10.1109/WCNC.2018.8377192.

Singh, S., & Chana, I. (2016). Cloud resource provisioning: survey, status and future research directions. Knowledge and Information Systems, 49(3), 1005–1069. doi https:/doi.org/10.1007/s10115-016-0922-3.

Singh, S. P., Kumar, R., Sharma, A., & Nayyar, A. (2020). Leveraging energy-efficient load balancing algorithms in fog computing. March, 1–16.doi https:/doi.org/10.1002/cpe.5913.

Singh, S. P., Nayyar, A., Kumar, R., & Sharma, A. (2019). Fog computing: from architecture to edge computing and big data processing. Journal of Supercomputing, 75(4), 2070–2105.doi https://doi.org/10.1007/s11227-018-2701-2.

Stephanie Vozza. (2022). Redefining Workloads in Cloud Environments. https://www.nutanix.com/theforecastbynutanix/technology/rethinking-cloud-workloads.

Sultan, O. H., & Khaleel, T. (2022). Challenges of Load Balancing Techniques in Cloud Environment: A Review. Al-Rafidain Engineering Journal (AREJ), 27(2), 227–235. doi https://doi.org/10.33899/rengj.2022.134056.1179

Sumathy, S., & Manju, A. B. (2019). Efficient load balancing algorithm for task preprocessing in fog computing environment. In Smart Innovation, Systems and Technologies (Vol. 105). Springer Singapore. doi https://doi.org/10.1007/978-981-13-1927-3_31.

Talaat, F. M., Saraya, M. S., Saleh, A. I., Ali, H. A., & Ali, S. H. (2020). A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment. Journal of Ambient Intelligence and Humanized Computing, 11(11), 4951–4966.doi https:/doi.org/10.1007/s12652-020-01768-8.

Téllez, N., Jimeno, M., Salazar, A., & Nino-Ruiz, E. D. (2018). A Tabu search method for load balancing in fog computing. International Journal of Artificial Intelligence, 16(2), 106–135. https://www.researchgate.net/publication/327752530_A_Tabu_Search_Method_for_Load_Balancing_in_Fog_Computing.

Tim Mell, P. G. (2009). Draft NIST Working Definition of Cloud Computing. National Institute of Standards and Technology, 53(March), 50. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Vailshery, L. S. (2022). Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030. Statista. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

Velde, V., & Rama, B. (2017). An advanced algorithm for load balancing in cloud computing using fuzzy technique. Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017, 2018-Janua, 1042–1047. doi https://doi.org/10.1109/ICCONS.2017.8250624.

Verma, M. (2016). Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment. .." Int. J. Inf. Technol. Comput. Sci, Vol 8, No.(April), 1–10. doi https://doi.org/10.5815/ijitcs.2016.04.01.

Verma, M., Bhardawaj, N., & Yadav, A. K. (2015). An architecture for Load Balancing Techniques for Fog Computing Environment. 269–274.doi https://doi.org/10.090592/IJCSC.2015.627.

Yi, S., Hao, Z., Qin, Z., & Li, Q. (2016). Fog computing: Platform and applications. Proceedings - 3rd Workshop

on Hot Topics in Web Systems and Technologies, HotWeb 2015, November 2015, 73–78. doi https://doi.org/10.1109/HotWeb.2015.22.

Zhang, G., Shen, F., Yang, Y., Qian, H., & Yao, W. (2018). Fair task offloading among fog nodes in fog computing networks. IEEE International Conference on Communications, 2018-May, 1–6. doi https://doi.org/10.1109/ICC.2018.8422316.

Zhang, P., Liu, J. K., Richard Yu, F., Sookhak, M., Au, M. H., & Luo, X. (2018). A Survey on Access Control in Fog Computing. IEEE Communications Magazine, 56(2), 144–149. doi https://doi.org/10.1109/MCOM.2018.1700333.

Zhuang, H., Li, C., Wang, Q., & Zhou, X. (2018). SSLB Self-Similarity-Based Load Balancing for Large-Scale Fog Computing. Arabian Journal for Science and Engineering, 43, no. 12, pp. 7487–7498. doi https://doi.org/10.1007/s13369-018-3169-3.