Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 191 (2021) 93–101

Procedia Computer Science

www.elsevier.com/locate/procedia

The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC)
August 9-12, 2021, Leuven, Belgium

Sustainability of Load Balancing Techniques in Fog Computing Environment: Review

Hamza Sulimani[a], Wael Y.Alghamdi[b], Tony Jan[c], Gnana Bharathy[d], Mukesh Prasad[a]

[a]*Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia*
[b]*Faculty of Computer Science and Information Technology, Ta'if University,Ta'if, Saudi Arabia*
[c]*School of IT and Engineering, Melbourne Institute of Technology, Sydney, Australia*
[d]*School of Information, Systems and Modelling, University of Technology Sydney, Sydney, Australia*

Abstract

The extreme workloads on the fog layer caused a misalignment in some fog nodes that affect its efficiency and degenerate fog technology's primary goal. Therefore, creating a balanced computing environment via the offloading process is the key. However, there are many obstacles to balance computing nodes in the fog environment, such as offloading strategy and its consequences due to the extreme offloading processes, and when need to offload and where. These obstacles are vital concerns among researchers. Thus, several studies have been conducted to enhance the fog system performance to increase the entire system's throughput. This paper explores the recent articles to determine the possible research gaps and opportunities to implement an efficient solution for load balancing in fog environments after analyzing and assessing the existing solutions. While most of the proposed solutions involve short-term solution, this literature review reveals the need to find out a sustainable resolution for load balancing to avoid listed obstacles by using the offloading technique with maintaining the bandwidth of the network.

Keywords: Load Balancing, Fog Computing, Offloading, Optimization, IoT, Time-latency, Sustainability.

1. Introduction

In recent years, cloud computing has emerged as an appealing computing paradigm. This computing system has acquired wide acceptance in the Internet of Things (IoT) technology as a suitable technique to manage and centrally

---

* Corresponding author. Tel.: +61-414 439 656
E-mail address: Hamza.h.sulimani@student.uts.edu.au

10.1016/j.procs.2021.07.015

process data. Cloud computing is typically composed of two layers: the cloud layer and the client IoT layer. Many modern IoT devices or objects existing in the client layer have sensors to measure the data's environmental metrics with a wide range of usages. The IoT layer prepares and sends this data to the cloud layer for execution. Over time, an immense amount of data is gathered from these devices by the cloud system for processing and analysis. However, some applications, such as healthcare, augmented reality, virtual reality, and smart grids, are affected by increasing the response time, which might call for cloud technology abandonment [1].

Later, fog computing was introduced as new technology to bridge some sensitive cloud/IoT technology gaps, such as time latency, location awareness, and Quality of Service (QoS). Fog computing acts as a complement system for the cloud environment by assigning it to an intermediate layer between the client and the cloud layers. Although the cloud layer has assigned heavy computational workloads to support bottomed layers, the fog layer has been designed to alleviate the cloud load by accomplishing tasks at the edge. In contrast, because of the massive IoT's generated data in the client layer, parts of fog devices, such as the central server in the high-density area, get overloaded. Furthermore, because of a rapid increase in the usage and implementation of IoT apps, fog computing's usage has increased dramatically [10]. These overloaded fog nodes have been struggled due to the heavy demands on services, which create unstable computing environments. Consequently, some nodes are extremely loaded while others remain idle. To address this situation, intelligent techniques must redistribute the streamed workloads among the computational devices in all the layers to create a balanced computing environment and elevate the demands on overloaded nodes.

On the other hand, and during the General Conference of the United Nations in 1984, the World Commission on Environmental and Development (WCED) introduced a sustainability term in a general context. However, many researchers define sustainability from a computing perspective; they introduce it around the same notion. Gonzallez et al. [6] defined sustainability as ”maintaining the functionality of a system without experiencing significant changes in its condition over time”. A fair amount of specialists have studied sustainability in many computing contexts to prevent available resources and maximize their usage. Unfortunately, many papers have proposed a wide range of appreciated solutions to enhance fog and cloud computing performance without considering sustainability.

In contrast, load balancing is an effective technique that enhances the entire computing environment and keeps all the computing nodes running equally. Without a doubt, by applying the latest load balancing concepts to any computational system, the entire system's performance will be enhanced, as well all nodes will be equally loaded. However, many researchers engaged, recently, in load balancing of fog computing. Since 2015, many papers have proposed and introduced a competitive computing system at the network's edge. Nevertheless, the load balancing techniques pose many challenges that might affect the impacts of such a solution. Efficiency, overhead, consequences of implementation, and improvement of the fault tolerance are some of the current challenges of load balancing.

This literature review explores the recent articles and knowledge about load balancing in the fog domain by answering some points, firstly, Exploring load balancing in fog system, definitions, types, purpose, proposed solutions, prose and its cons. Secondly, are there any proposed load balancing solution that introduced sustainability to tackle this issue? The study's overall structure takes two sections, which consider as the main contribution of this paper,

- It demonstrates the central concept for load balancing, its possible scenario and controller types.
- It explores and discusses the recent offloading algorithms and their relation with the sustainability concept (if any).

2. Load Balancing

An excessive computation or resource constraint, latency requirement, permanent or long-term storage, data management and organisation, privacy and security, accessibility, affordability, feasibility, and load balancing are criteria used when deciding to offload specific tasks [1]. In this section, we will expand writing about load balancing, especially in fog environments. In this field, various definitions of load balancing have been found. Load balancing aims to maintain the available resource in computational nodes by redistributing the arrival tasks among computing nodes to be equally loaded and avoid overload conditions of one device (or specific nodes) with many tasks [11, 15] also to avoid the idle condition for them. Accordingly, load balancing in fog computing refers to efficiently distributing arrival workload among a group of computing fog nodes, thus that the reliability of incoming tasks and the capacity of existing clients increase. Load balancing is beneficial for cloud service providers as well as fog providers to effectively distribute applications tasks to different fog nodes or cloud servers. Due to the fluctuation in the number of clients' requests, the load balancing technique is the most suitable technique for network computation, which works on a dynamic principle to avoid lagging in execution due to a long waiting list at one node. Among previous definitions we notice, it seems all have the same meaning.

Figure 1 represents a model of a load balancing in fog architecture, where the cloud server conducts the interaction with different layers. During load balancing process, the workloads exchanged among layers to filling the needs of clients regarding resources with less possible delay. Further, in another part, in case the client fails in receiving essential resources, the sensitive request gets sent to cloud.

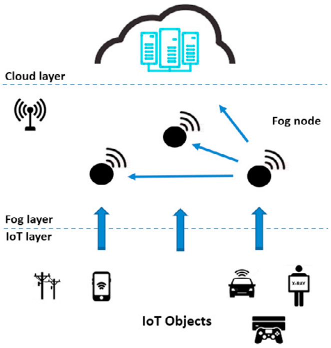


Fig. 1: Load Balancing in fog/cloud environment

Mainly, there are two main categories for load balancing in fog computing; static and dynamic load balancing techniques [19]. Static load balancing relies on distributing the tasks using the initial information on task requirements. These requirements determined at the beginning of the mission. Although this technique is easy to apply and configure, the static method has significant drawbacks. For example, they are changing the burden of one node during the system start-up to the maximum load. Therefore, task allocation is fixed and cannot modify during the execution of the process. While the network's initial design configures the distribution of the workloads to some nodes that seem not loaded, the nature of free movements in a fog environment will affect this strategy.

On the other hand, dynamic load balancing distributes tasks intelligently when one of the nodes becomes overloaded. Thus, selecting the destination node always depending upon the current information of traffic loads. Hence, the efficient real-time load estimation is vital for active dynamic load balancing. Moreover, Virtual Machine (VM) is used widely in most computing systems [7, 23], where it uses the same pool of resources on the same physical machine. Therefore, the load balancing function might be more complicated in the fog environment. Estimating the memory usage and CPU load for fog nodes and network load is the first stage in the algorithm; secondly, through analysing the available resources in fog devices and measure the synchronising level in fog zone. Third, be sure to notice the fault tolerance and avoid a single point of failure in case of failure [20].

2.1. Controller Types in Load Balancing

To create balanced computing environment, a dispatcher is essential to decide the required action to fulfil the balancing criteria. The main role for the dispatcher is to investigate the surrounding statuses of computing nodes and decide what the superior location to execute the arrival tasks in. There are mainly two strategies to implement load balancing in fog computing that are centralised and distributed controller.

The central controller utilises a core device that has a direct or indirect connection to other fog devices. It might be an independent device, or a fog device that has a load balancing feature [1]. This feature provides by modifying or configures the existing computing nodes as shown in Figure 2. For example, by configuring a router in the Fog environment, it can act as a router and controller of load balancing, at the same time. The pros of using a centralised controller technique for load balancing are that it is simpler to manage and implement. Moreover, the recovery time is minimal in case of failure. On the other hand, global state information in the network is a group of metric information of other devices. Thus, the device which represents the load balance has global state information of the current cluster. This data collects and analysis in load balance to use for redistribution process [4]. Where the cons in this technique are that there is a single failure point yet, we can minimise the consequences by configures another fog node to be hot stand-by.

(a) Central Controller

(b) Distributed Controller

Fig. 2: Controller in fog layer

Distributed Controller: Even though a centralised controller has many advantages in managing and controlling the load among devices in a network, it might have a communication bottleneck since it utilises the network bandwidth. Thus, the failure in this load balancer will cause failure to the entire system, more simply, a single point of failure. All fog devices must send some selective information periodically to the central controller to keep it updated. Therefore, cooperation is required between nodes to distribute the computation requests among them. Whereas the distributed controller, as shown in Figure 2, increases the scalability and reliability of the network, it relies on the cooperation of the local controller [4].

2.2. *Load Balancing in Fog Computing Application Scenarios*

Actually, load balance operation might take place at any computing level in fog system depending upon location of computing load. This location, which has high demand of computing, determent what type of algorithms suitable to be initialized and execute where every location has different technique to apply. There are many offloading scenarios in fog system. Figure 3 illustrates some of scenarios where offloading can be activated to make a load balancing. There are three locations to offload task in figure 3. For example, case A represents IoT/cloud model where the IoT device communicate directly with cloud platform without engage fog layer in communication such as some smart meter accessing direct a cloud service. In case B, the communication line represents offload from IoT objects to fog level, where fog is cooperating with the cloud in improving security or latency services or data may be stored and process. While case C shows IoT devices accessing the smart gateway, which locate in a middle-ware. It evaluates the task and decides on whether to executed locally or offload to the cloud instead of offloading to the fog depending on the requirements of the service. In case D, the operation represents offloading among fog devices (peer to peer) after receiving it from smart gateway or directly from IoT objects.

2.3. *Offloading Algorithms to Balance the Load*

In the this paper, the relative importance of load balancing is debated. Thereby, a considerable amount of literature has been published on load balancing. These studies show a noticeable improvement in time latency and QoS for the overall system. Therefore, this section examines recent methodologies for load balancing in the fog environment, like: Earlier, Janjic [8] proposes a work-stealing method for load balancing. The work-stealing algorithm is upon creating a pool for accumulated tasks. Thus, every node will own its pool of tasks. In the author words, the overloaded node can forward some tasks to other idle computing nodes. Despite that, the proposed method works to encourage the

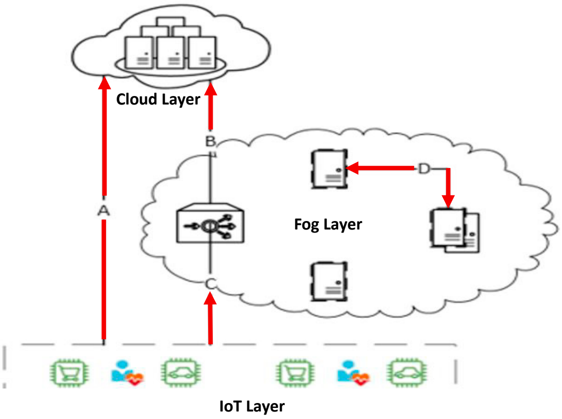


Fig. 3: Possible offloading scenarios in fog environment

idle node to look for overloaded nodes and steal a suitable number of tasks to execute. However, Janjic [8] does not attempt to give sufficient consideration to loaded nodes and cloud system to take decisions.

Oueis et al. [14] suggest a novel strategy for offloading multi-users requests in fog environments by forming dynamic fog clusters instead of forward the workloads to the cloud centre. The authors focus on improving a Quality of Users (QoE) by creating a locally Small Cell Cloud (SCC) through cooperating multi Single Cells (SC) to perform the forwarded tasks. The proposed algorithm posits that the pool of resources will improve the entire efficiency of the system. Therefore, the results revealed that the proposed algorithm significantly outperformed other selective algorithms, such as no clustering and static clustering scenarios. Although the paper proposes an innovation algorithm, it does not consider communication delay, which might have a crucial effect in some contexts.

Veram et al. [21] present a data replication technique to adopt the load balancing algorithm in the fog environment. It reveals that maintain the necessary data, which essential to execute the tasks, has improved the load balancing in the proposed model. It proves that the data replication technique in the fog paradigm has improved the total execution time after a specific time from a startup of the system. This study has introduced a novelty in load balancing, which ignores a suitable technique to tackle the dramatic increase of data in the fog layer. Verma et al. [20] proposed efficient real-time scheduling (RTES) for load balancing in the fog paradigm that categorised the tasks to soft or real tasks with the execution deadline to accomplish real task requirements. Thus, the local master server distributes the received workloads among fog nodes according to their categories. It shows that RTES get minimum response time, among other techniques such as Multi-Objective Task Scheduling (MOTS), First Come First Serve (FCFS), and PRIORITY. Even though the RTES architecture policy has got unexceptional results, it does not show any long-term solution for fog environments.

By combining graph partitioning theory with fog computing characteristic, Ningning [13] proposed another framework for load balancing, which is based on graph reporting by atomising a high computational power to multi virtual machines while limited nodes to a single virtual machine. This framework allows reducing the number of immigration processes in the whole system. However, the paper fails to show any sustainable solution to the focused obstacle.

Jijin [9] suggested a service load balancing algorithm for virtual Fog Access Points (v-FAPs) to addressing a task assignment issue that creates a task graph to lists all tasks with the required resources and service graph contain all available nodes with available resources. Thus, the mapping process adopts the relation among them by assigning a suitable task to a suitable node(s). Although it has shown efforts to reduce consuming time, it focuses on 5G networks, not on fog environment.

Similarly, Xiao et al. [22] utilized users' QoE as a performance metric to evaluate the proposed offloading algorithm in fog computing that aim to decrease the degree of cloud dependency by share the tasks among the surrounded fog nodes. The algorithm tends to partition the tasks into sub-tasks and then forward them equally to fog and cloud. Even though it suggests pushing the tasks to other computing nodes, it mainly focuses on the offloading technique, not on the load balancing algorithm. Also, Xiao et al. [2] introduced Follow Me Fog (FMF), a framework enforcing a new flexible handover timing scheme among a wide range of computation access points. FMF enhances the offloading by supporting a pre-migration mechanism, which pre-migrates tasks when the handover is expected to occur. Although the results show to improve the latency by 36.5 % in their experiment, the proposed framework focused on mobile IoT devices.

Multi-users or multitenant aspects has studied [12], which suggest the Multitenant Load Distribution Algorithm for fog Environments (MtLDF). The main objective for MtLDF is to enhance the load balancing with more than one user for every fog node. It uses carbon emission and energy consumption as metrics to evaluate the system. It proposes Fog Management Layer (FML) as a central node in the fog layer, redirecting the tasks to free nodes or not loaded. If all Fog nodes are overload, the FML sorts the tasks in a queue to execute in the cloud layer. However, MtLDF consumes additional time by forwarding all tasks to FML for the consultant instant execute them immediately at the received fog.

Tellez et al. [18] proposed a Tabu method to enhance the load balancing operation between fog and cloud servers, which consider resource constraints during a decision. The forwarding load balancing decision in the Tabu method relies on the computational cost of computing tasks in the Cloud system and computational cost for computing tasks in fog nodes. Tabu method technique relies on taking advantage of the storage and computational power in fog layers to perform the tasks, scheduling tasks and selecting appropriate nodes in the cloud or Fog layer for execution.

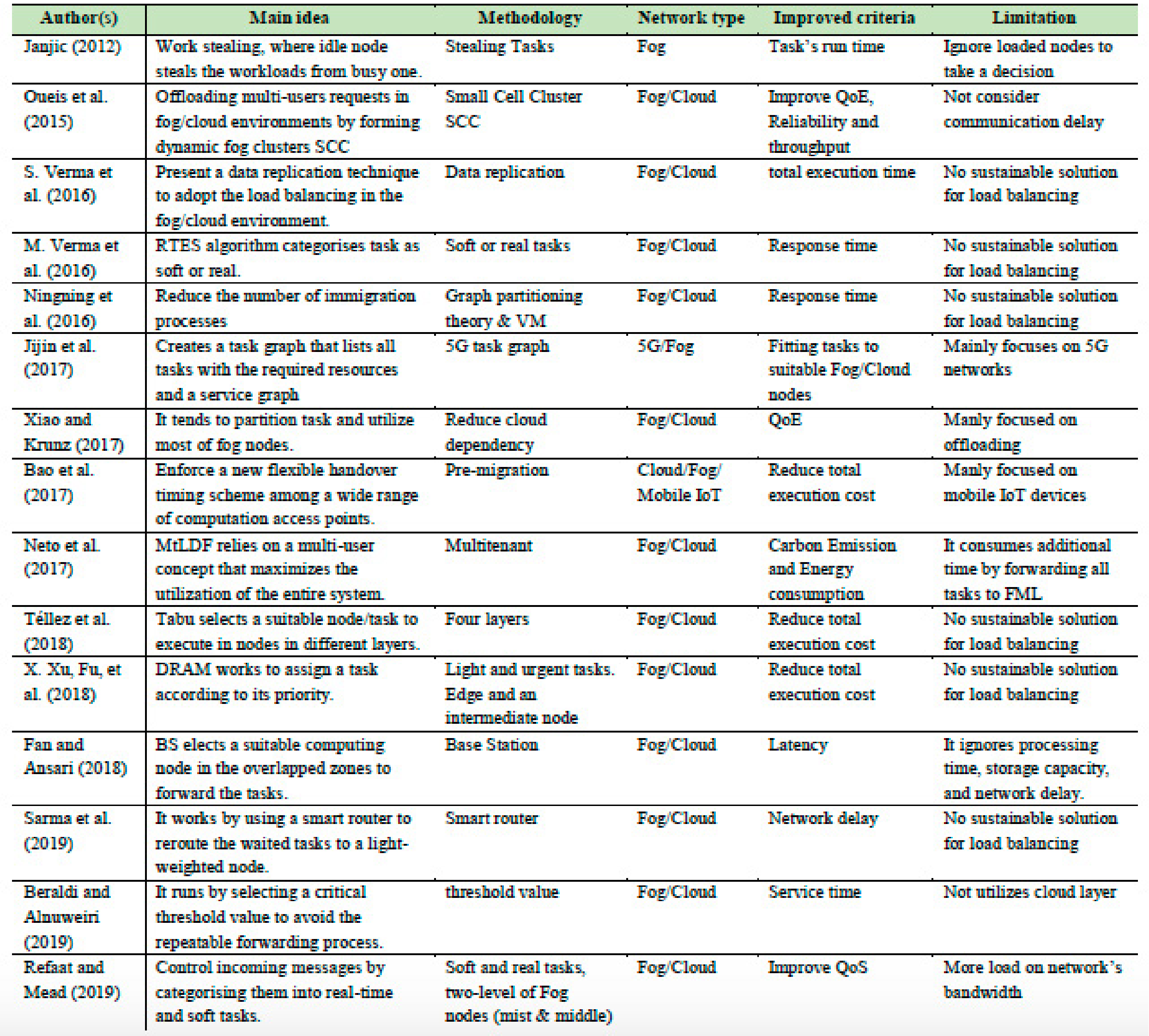
A Dynamic Resource Allocation Method (DRAM), another load balancing algorithm, is presented by Xu et al. [23]. The loose-fitting tasks with high volume computation and storage select to the cloud layer. This method works to improve the achievements of each computational parts in nodes to avoid overload or low utilisation of the participant's nodes. However, it fails to initiate a sustainable theme to avoid the continuous traffic on the network bandwidth caused by migration processes.

Fan et al. [5] proposed the LoAd Balancing (LAB) scheme to enhance the computing level in the fog environment, which associating with cellular base stations, where the IoT objects connected to base stations. It focuses on IoT clients who locate in the overlapped area of some base stations. While LAB shows competitive results comparing to other algorithms such as SINR and alpha-distributed algorithm, it focuses on improving IoT devices' efficiency in overlapped areas among cellular base stations and ignoring other cases. The researchers have proposed many techniques to reduce unnecessary data to send via networks. Serma et al. [17] presented a smart gateway as a load balancer in a fog environment. In this model, the data is pre-processed before transmitting over the network, which may minimise the traffic on the net and improve the whole system performance. However, it fails to consider factors such as processing, storage capacity, and network delay.

Serma et al. [17] considered network delay and computing time as metrics to evaluate their proposed technique. They use a smart router as a central point to redistribute the arrival tasks among computing nodes. The paper evaluates the proposed system in different scenarios. The results show that the cooperative between fog nodes and cloud layer in computing areas get appealing results. However, It does not offer a sustainable solution for continuous load in the target system. In Berald et al. [3] paper, a Power-of-Random choices algorithm proposed. It designs by limits the threshold value which responsible for forwarding tasks to other node or performs locally. By selecting a critical threshold value, the number of forwarding tasks decreases. This technique aims to reduce the number of iterations to explore a free or least loaded Fog node. Even though the proposed scheme shows some positive results, it still not showing any role for the cloud.

Refaat et al. [16] proposed a novel scheduling model to control incoming messages in the fog system, which categorise them into real-time and soft requests. A Decentralize Load-Balance Scheduling (DLBS) is a model that mainly relies on task priority and the current load status for every fog node. The authors divided fog nodes into mist and middle edge nodes. However, both have the exact power specification, mist assigned to sensitive tasks that improve QoS of the entire system and the middle fogs for soft tasks or low priorities. This policy can cause an overload of the network, where almost all soft tasks forward to intermediate nodes. Therefore, this algorithm causes a massive load on network bandwidth, increasing the transmission time lag and, consequently, total time. Table 1 summarizes the related literature by lists the central concept, methodology, platform, enhanced metric, and the limitation of each study.

Table 1: Related Works Summary



Author(s)	Main idea	Methodology	Network type	Improved criteria	Limitation
Janjic (2012)	Work stealing, where idle node steals the workloads from busy one.	Stealing Tasks	Fog	Task's run time	Ignore loaded nodes to take a decision
Oueis et al. (2015)	Offloading multi-users requests in fog/cloud environments by forming dynamic fog clusters SCC	Small Cell Cluster SCC	Fog/Cloud	Improve QoE, Reliability and throughput	Not consider communication delay
S. Verma et al. (2016)	Present a data replication technique to adopt the load balancing in the fog/cloud environment.	Data replication	Fog/Cloud	total execution time	No sustainable solution for load balancing
M. Verma et al. (2016)	RTES algorithm categorises task as soft or real.	Soft or real tasks	Fog/Cloud	Response time	No sustainable solution for load balancing
Ningning et al. (2016)	Reduce the number of immigration processes	Graph partitioning theory & VM	Fog/Cloud	Response time	No sustainable solution for load balancing
Jijin et al. (2017)	Creates a task graph that lists all tasks with the required resources and a service graph	5G task graph	5G/Fog	Fitting tasks to suitable Fog/Cloud nodes	Mainly focuses on 5G networks
Xiao and Krunz (2017)	It tends to partition task and utilize most of fog nodes.	Reduce cloud dependency	Fog/Cloud	QoE	Manly focused on offloading
Bao et al. (2017)	Enforce a new flexible handover timing scheme among a wide range of computation access points.	Pre-migration	Cloud/Fog/ Mobile IoT	Reduce total execution cost	Manly focused on mobile IoT devices
Neto et al. (2017)	MtLDF relies on a multi-user concept that maximizes the utilization of the entire system.	Multitenant	Fog/Cloud	Carbon Emission and Energy consumption	It consumes additional time by forwarding all tasks to FML
Téllez et al. (2018)	Tabu selects a suitable node/task to execute in nodes in different layers.	Four layers	Fog/Cloud	Reduce total execution cost	No sustainable solution for load balancing
X. Xu, Fu, et al. (2018)	DRAM works to assign a task according to its priority.	Light and urgent tasks. Edge and an intermediate node	Fog/Cloud	Reduce total execution cost	No sustainable solution for load balancing
Fan and Ansari (2018)	BS elects a suitable computing node in the overlapped zones to forward the tasks.	Base Station	Fog/Cloud	Latency	It ignores processing time, storage capacity, and network delay.
Sarma et al. (2019)	It works by using a smart router to reroute the waited tasks to a light-weighted node.	Smart router	Fog/Cloud	Network delay	No sustainable solution for load balancing
Beraldi and Alnuweiri (2019)	It runs by selecting a critical threshold value to avoid the repeatable forwarding process.	threshold value	Fog/Cloud	Service time	Not utilizes cloud layer
Refaat and Mead (2019)	Control incoming messages by categorising them into real-time and soft tasks.	Soft and real tasks, two-level of Fog nodes (mist & middle)	Fog/Cloud	Improve QoS	More load on network's bandwidth

3. Challenges in Fog Computing

while fog computing classified as the evolved extension of the cloud computing system to handle IoT related problems and shortcomings at the network edge. However, in fog computing, processing nodes are distributed and heterogeneous. Furthermore, the services based on fog technology have to work with various aspects of the restricted environment. Moreover, assurance of security is dominant in fog computing. Therefore, Discovering the challenges of fog computing from service-oriented, structural, security perspectives in this technology can be listed as follows [10].

3.1. Service-Oriented

Resources enrich not all fog nodes. Therefore, comprehensive scale application enhancement in resource-restricted nodes is not natural compared to traditional data centres. Therefore, distributed application development needs for potential programming platforms in Fog are required to implement. Moreover, a fog administrator is required to clarify the policies to distribute required tasks among sensors/IoT devices, fog infrastructure.

3.2. Structural Issues

The infrastructure of fog computing consists of various components from both core and verge of networks. These types of components are equipped with a different computation but not designed for general computing. Therefore, redesign or modified the computation unit for the component is an extremely challenging part of the system setup. Additionally, Based on execution operations and operational requirements, the selection of the suitable device, places of deployment, and corresponding resource configuration are crucial in fog computing as well. Computational devices are spread across network boundaries in fog computing and can be shared or virtualised. In this case, it is necessary to define suitable metrics, strategies for inter-nodal cooperation, and efficient resource provisioning.

3.3. Security Aspects

Fog computing rely on conventional networking components, it is highly defenceless to security attacks. Maintenance of privacy and authenticated access to computing and storage services in a widely distributed model, such as fog computing, is challenging to ensure. Therefore, Maintaining QoS is difficult during the implementation of security, where the data-centre integrity adequate and makes security topic in fog computing challenging.

4. Discussion

Nowadays, a plethora of useful IoT apps relies on the quality of time latency to run effectively. Fog computing might be the appropriate technology to enhance the new-generation real-time apps. Because of the dramatically increasing number of IoT users, some of the computational nodes in a fog system become loaded, while the others remain idle. Thus, it has become essential to reinforce the cooperation and integration among computing nodes through offloading technique to run most of the processing nodes equally and reduce the computational and storage power wastage (in the idle nodes) [5, 15, 21]. This paper discusses the previous literature on load balancing and the different techniques used to improve a fog model. Collectively, these studies outline the critical role and the need for offloading to improve system performance, QoS, time latency, and power efficiency in fog computing [1, 3, 5, 11, 12, 13, 15]. Therefore, the load balancing through offloading processes promises tremendous benefits in a fog environment.

However, the strong relationship between enhancing the performance of computing nodes and offloading process has been reported in the literature. In effect, the offloading technique performs by reinforce the current algorithms and inventing a new one (software). This paper set out with the aim of assessing the drawbacks of offloading processes in the aimed system. Moreover, an initial objective of this paper is to identify if any previous study introduced a sustainable concept to balance the load in a fog system. To the best of our knowledge, the outcomes showed no load balancing solution based on a long-term concept was present in literature reviews. This finding highlights adding an extra computing node to an affected area or moving the idle node to another location (hardware modification) with high demands to overcome the paper issue. This modification, in nodes position, must be done through specific software, which must have historical load data for all computing points. However, creating a hybrid load balancing system considers as a markable research gap. It can thus be suggested that even though the software solution has a noticeable improvement for load balancing, hardware/software resolution will enhance the computing system in a better way. There is abundant room for further progress in determining the feasibility of including hardware modification in the position of fog nodes to enhance the load balancing.

5. Conclusion

This paper discovers and investigates the concept of sustainability in fog systems lacking for most proposed algorithms that aim to relieve the burden of affected nodes by redistributing the workloads to the neighbour's nodes. However, this resolution philosophy might cause jumps in a number of offloading processes that affect the network bandwidth by default. To avoid that, designing a monitoring system to propose recommendations to empower the coverage service is the solution. It would suggest adding, moving, or removing the affected node after analysis and evaluating the current situation. The finding reported in this article shed new light on merging software/hardware techniques to balance the computing load. It lays the groundwork for future research into hybrid load balancing by using a suitable machine learning method. The revealed solution introduces the load balancing in sustainability concept, which opens a new era to control and manage distributed computing nodes worldwide.

References

[1] Aazam, M., Zeadally, S., Harras, K.A., 2018. Offloading in fog computing for iot: Review, enabling technologies, and research opportunities. Future Generation Computer Systems 87, 278–289.

[2] Bao, W., Yuan, D., Yang, Z., Wang, S., Li, W., Zhou, B.B., Zomaya, A.Y., 2017. Follow me fog: Toward seamless handover timing schemes in a fog computing environment. IEEE Communications Magazine 55, 72–78.

[3] Beraldi, R., Alnuweiri, H., 2019. Exploiting power-of-choices for load balancing in fog computing, in: ICFC2019, IEEE. pp. 80–86.

[4] Chandak, A., Ray, N.K., 2019. A review of load balancing in fog computing, in: 2019 International Conference on Information Technology (ICIT), IEEE. pp. 460–465.

[5] Fan, Q., Ansari, N., 2018. Towards workload balancing in fog computing empowered iot. IEEE Transactions on Network Science and Engineering 7, 253–262.

[6] González-Mejía, A.M., Eason, T.N., Cabezas, H., Suidan, M.T., 2014. Social and economic sustainability of urban systems: comparative analysis of metropolitan statistical areas in ohio, usa. Sustainability science 9, 217–228.

[7] Ibrahim, S., Jin, H., Cheng, B., Cao, H., Wu, S., Qi, L., 2009. Cloudlet: towards mapreduce implementation on virtual machines, in: Proceedings of the 18th ACM international symposium on High performance distributed computing, pp. 65–66.

[8] Janjic, V., 2012. Load balancing of irregular parallel applications on heterogeneous computing environments. Ph.D. thesis. University of St Andrews.

[9] Jijin, J., Seet, B.C., Chong, P.H.J., Jarrah, H., 2017. Service load balancing in fog-based 5g radio access networks, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), IEEE. pp. 1–5.

[10] Mahmud, R., Kotagiri, R., Buyya, R., 2018. Fog computing: A taxonomy, survey and future directions, in: Internet of everything. Springer, pp. 103–130.

[11] Mon, M.M., Khine, M.A., 2019. Scheduling and load balancing in cloud-fog computing using swarm optimization techniques: A survey, Seventeenth International Conference on Computer Applications (ICCA 2019).

[12] Neto, E.C.P., Callou, G., Aires, F., 2017. An algorithm to optimise the load distribution of fog environments, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE. pp. 1292–1297.

[13] Ningning, S., Chao, G., Xingshuo, A., Qiang, Z., 2016. Fog computing dynamic load balancing mechanism based on graph repartitioning. China Communications 13, 156–164.

[14] Oueis, J., Strinati, E.C., Barbarossa, S., 2015. The fog balancing: Load distribution for small cell cloud computing, in: 2015 IEEE 81st vehicular technology conference (VTC spring), IEEE. pp. 1–6.

[15] 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. IEEE Communications Magazine 56, 60–65.

[16] Refaat, H.E., Mead, M.A., 2019. Dlbs: Decentralize load-balance scheduling algorithm for real-time iot services in mist computing. Editorial Preface From the Desk of Managing Editor? 10.

[17] Sarma, B., Kumar, G., Kumar, R., Tuithung, T., 2019. Fog computing: An enhanced performance analysis emulation framework for iot with load balancing smart gateway architecture, in: 2019 International Conference on Communication and Electronics Systems (ICCES), IEEE. pp. 1–5.

[18] Téllez, N., Jimeno, M., Salazar, A., Nino-Ruiz, E., 2018. A tabu search method for load balancing in fog computing. Int. J. Artif. Intell 16.

[19] Verma, M., Bhardawaj, N., Yadav, A.K., 2015. An architecture for load balancing techniques for fog computing environment. International Journal of Computer Science and Communication 8, 43–49.

[20] Verma, M., Bhardwaj, N., Yadav, A.K., 2016a. Real time efficient scheduling algorithm for load balancing in fog computing environment. Int. J. Inf. Technol. Comput. Sci 8, 1–10.

[21] Verma, S., Yadav, A.K., Motwani, D., Raw, R., Singh, H.K., 2016b. An efficient data replication and load balancing technique for fog computing environment, in: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE. pp. 2888–2895.

[22] Xiao, Y., Krunz, M., 2017. Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE. pp. 1–9.

[23] Xu, X., Fu, S., Cai, Q., Tian, W., Liu, W., Dou, W., Sun, X., Liu, A.X., 2018. Dynamic resource allocation for load balancing in fog environment. Wireless Communications and Mobile Computing 2018.