A Review of Load Balancing in Fog Computing

Ashish Virendra Chandak
Department of Information Technology
Shri Ramdeobaba College of Engineering and Management
Nagpur, Maharashtra, India
Email: achandak33@gmail.com

Niranjan Kumar Ray
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, Odisha, India
Email: rayniranjan@gmail.com

***Abstract*—A number of IoT devices are continuously increasing day by day and generating a tremendous amount of data. This generated data passed to cloud for processing but some application requires a quick response. To overcome this situation, fog computing is used which is placed between IoT and Cloud computing. Fog computing is placed at the edge of the cloud data center. The allocation of resources and distributing workload is called load balancing. Load balancing ensures appropriate distribution of workload. Extensive research has been conducted to implement load balancing in fog computing. Having understood the importance of load balancing, this paper presents a survey of load balancing algorithms implemented in fog computing. The paper also presents an analysis based on different metrics of Fog computing and simulation tool used for implementing load balancing in fog computing.**

Keywords*—*load balancing, fog computing, IoT

I. Introduction

A number of smart phones, smart watches and IoT devices are continuously increasing day by day. All IoT devices contain sensors and generate a tremendous amount of data. All devices contain sensors to sense the data of the real-time environment. Tremendous amount of data is generated at the IoT layer. This data gets passed to cloud computing for processing. But some application viz. health, the military operation required a quick response where latency is critical. Delay issue has been discussed in [1]. To overcome this limitation, fog computing is placed between IoT and cloud. Fog computing does fast processing of data and provides quick response and only important data is pass to cloud computing layer. Fog computing uses edge devices to carry out a computation, communication, and storage [2], [3], [4]. In fog environment, data sensed by IoT devices is transmitted to fog nodes. As the data generation rate increases, some fog node gets overloaded. As a result, data processing time increases and it will affect delivery time. Delivery time is directly proportional to computing load on fog nodes. Heavy computing load on fog nodes translates into the long delivery time. To overcome this type of situation, proper coordination among fog nodes is required and some fog nodes should offload the task to less overloaded nodes. When some fog nodes are more overloaded, the computation time for processing task increases, and thus load should be pass to less overloaded nodes. Proper load balancing among fog nodes minimizes resource consumption, response time, increases resource utilization. Following are challenges associated with load balancing in fog computing.

- Efficiency: It is defined as how much load balancing algorithm is capable to transform load. It is indicated from resource utilization and response time.
- Overhead: It is defined as overhead involved in implementing load balancing algorithm in fog computing environment.
- Fault Tolerant: Even if one node fails, then load should be pass to other nodes and system should not stop working.

Load balancing in Fog Computing distributes the excess across available resources all the nodes. It is used to get both better user satisfaction and higher resource utilization, guaranteeing that no single node is overwhelmed, thus improving the system overall performance. The contribution of the paper is as follows:

- Fog computing architecture has been described.
- Various load balancing techniques used in fog computing has been identified.
- Common Simulation tool used for fog computing has been identified.

The rest of the paper is organized as follows. Section II describes fog computing architecture. Types of controller used for load balancing have been described in section III. Node uniformity in fog computing presented in section IV. Section V contain node allocation strategies. Study of load balancing strategies used in fog computing has been presented in section VI. Common simulation tool used in fog computing has been described in section VII. Finally, section VIII concludes the paper.

II. FOG COMPUTING ARCHITECTURE

Fog computing architecture is shown in Figure 1. It is divided into three layers viz. IoT device layer, Fog computing layer and Cloud computing layer.

- IoT Layer (Layer 1): This layer contains IoT devices viz. PDS, mobile, laptops etc. These devices are geographically distributed. These devices generate data by sensing it and pass data to the upper layer for processing and storage.
- Fog Computing Layer (Layer 2): This layer can contain nodes which can be router, gateway, switches, server etc. Fog is a framework having a number of

978-1-7281-6052-8/19/$31.00 ©2019 IEEE
DOI 10.1109/ICIT48102.2019.00087

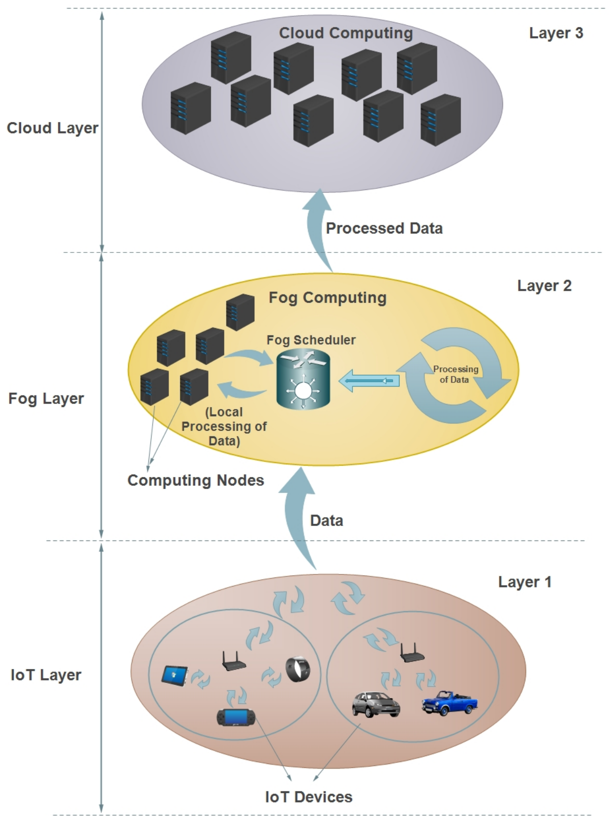

Fig. 1: Fog Computing Architecture

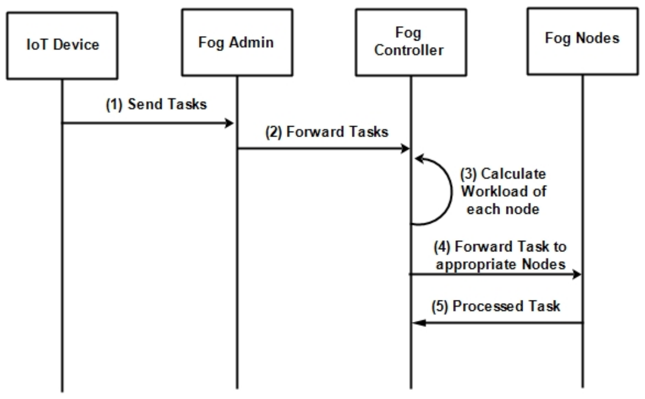

Fig. 2: Sequence Diagram of Fog Controller

free nodes. Fog may have a single computational node or a number of computational nodes connected in a dispersed manner. Data from layer 1 is processed and filtered in this layer. Nodes can participate in fog environment by offering resources. The communication between those resources during the execution of a task requires a scheduling layer that uses a distinct scheduling scenario. Fog scheduler allocates resources to tasks. Fog scheduler collects information from all modes, in order to allocate node to tasks. Based on this information, the task will be forwarded for execution to the most suitable cloud node. Fog scheduler uses a different heuristic for optimal task scheduling.

- Cloud Layer (Layer 3): This is top layer in architecture and this layer consist of the computing node, storage servers, database etc. It consists of powerful nodes which are used for extensive computation and store data coming lower layers. Cloud computing provides access to the server, storage, and database over the internet. The cloud computing layer is responsible for permanent, large-scale storage of data and in periodic manner

III. TYPES OF CONTROLLER IN LOAD BALANCING

Load balancing strategies are implemented using centralized and distributed controller. Sequence diagram of fog controller is shown in Figure 2.

A. Centralized Controller

Central load balancing in fog environment is supported by a centralized controller that balances load in fog nodes as shown in Figure 3. The advantages of a central controller for load balancing are that it is simpler to implement, easier to manage, and quicker to repair in case of a failure. Central controller has global state information in fog environment viz. resource utilization, load, connections. Controller for central strategies are implemented to monitor information globally. Centralized controller collect status of all nodes connect to it, then it is analyzed, and this analysis used for resource allocation.

B. Distributed Controller

A centralized controller can have communications bottleneck since all nodes must communicate with it. Central point of failure. If the centralized controller goes down, the entire system breaks. Lots of message passing mechanism is required since all nodes must keep the controller up to date with information. To overcome this problem, as shown in Figure 4, a distributed controller is used which coordinates with the local controller and the overhead of computation is distributed. The distributed controller improves the reliability and scalability of the network. While the drawback of the distributed controller is that it requires cooperation from the local controller.

IV. NODE UNIFORMITY

In fog computing load balancing algorithm, the node used for scheduling is classified as homogeneous or heterogeneous.

A. Homogeneous Node

In this type of node category, all nodes have the same capacity and computation capability. All nodes have same characteristics. It is rarely adopted in fog environment since it cannot take advantage of the heterogeneous nature of resources.

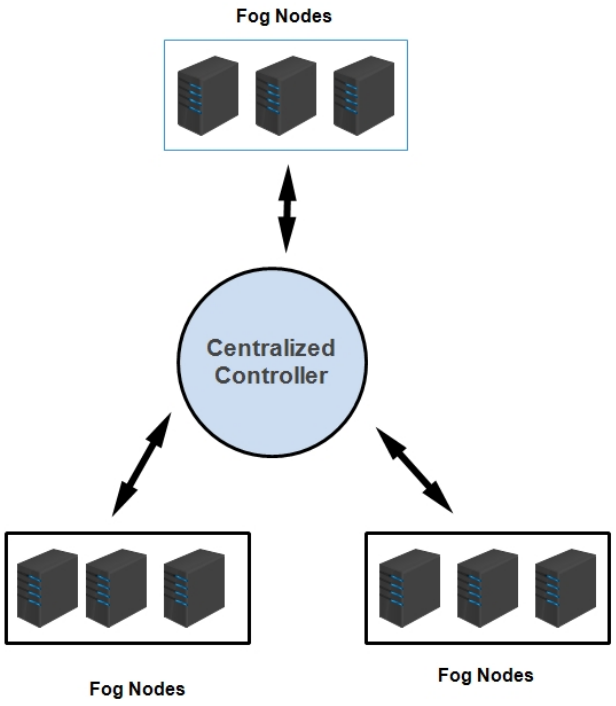

Fig. 3: Centralized Controller

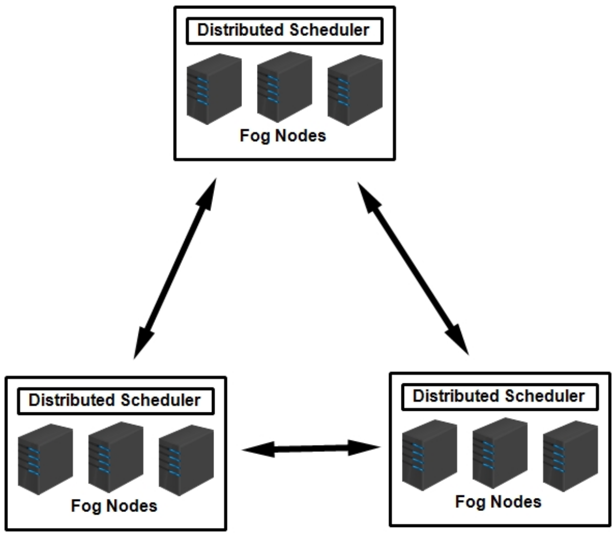

Fig. 4: Distributed Controller

B. *Heterogeneous Node*

In this category, all nodes have different capacity and computation capability. All fog nodes have different characteristics. The controller can select different nodes based on the requirement. In this category, based on task characteristics, nodes are selected.

V. NODE ALLOCATION

Node allocation in fog environment is classified as static and dynamic.

A. *Static Allocation*

In this category, node information is already known in advance and a specific number of nodes are already allocated irrespective of the number of tasks. However, it is not useful in a dynamic changing requirement [5].

B. *Dynamic Allocation*

In this case, nodes are dynamically allocated based on a number of tasks. This strategy gives better performance as compared to static allocation. Task requirement is obtained only when it comes to scheduling stage [6], [7].

VI. LOAD BALANCING STRATEGIES

Authors used several strategies to solve the load balancing problem in fog environment as shown in Fig 5. In this section, the study of existing load balancing strategies in fog computing has been carried out.

A. *SDN Appraoch*

Number of IoT devices are growing at alarming rate and data generation is tremendously increased. This phenomenon become more and more serious day by day. To solve this type of situation, better hardware should be built. But it will increase configuration cost. To overcome this issue, software defined network is used in traditional network, controller software and forwarding element, i.e. switches, controller software is integrated in single box. When we want implement new policies, networking operator has to configure each and every individual device. Software defined networking is one of the new revolutions in the networking devices (viz. router/switches) programmable and allow them to be controlled by central element. Thus, network can be customizable. This new paradigm has created the interest from both industry and academia since last couple of years. It is approach to computer networking in which network control is decoupled from the hardware and given software application. This architecture decouples the control function and forwarding function and enables the network control to become directly programmable. The control and management of network devices is performed by controller. The controller interacts with the network resources via a standardize interface which is used to collect network state information and distribute control commands to be enforced in the network devices. SDN provides efficient network design, management and control by providing vendor independent control interfaces [8], [9]. He at. al. [10] proposed software defined cloud/ fog networking (SDCFN) architecture in the internet of vehicles for workload balancing. Author proposed SDN based modified particle swarm optimization for load balancing which minimizes latency.

B. Graph Partitioning

Ningning et.al. [10] proposed a graph partitioning algorithm for load balancing in fog computing. As node in fog computing can leave and join anytime, dynamic changes in network should be considered. Physical node graph model is considered as virtual machine graph model. Then according to resource distance and task load balancing, virtual machine node provide service to user using graph partitioning.

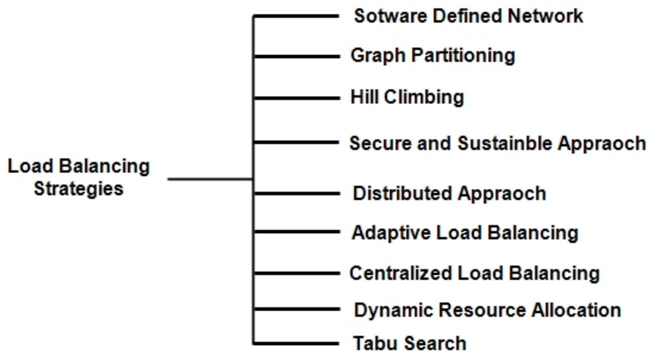
Load Balancing Strategies
Sotware Defined Network
Graph Partitioning
Hill Climbing
Secure and Sustainble Appraoch
Distributed Appraoch
Adaptive Load Balancing
Centralized Load Balancing
Dynamic Resource Allocation
Tabu Search

Fig. 5: Load Balancing Strategies Classification

C. Hill Climbing Appraoch

Zahid et. al. [11] use a Hill Climbing Approach for load balancing in fog computing. This technique uses mathematical optimization for searching. This algorithm is iterated until some best solution is found. In this algorithm, the loop is incremented until a closest available node finds out. Then it picks the best possible node and assigns a request to it for processing.

D. Secure and Sustainable load balancing Approach

Puthal et. al. [12], [13], [14] proposed a secure and sustainable load balancing in edge data center fog computing. Edge data center placed between cloud data centers and data sources they are positioned to decrease the latency and network congestion by processing data streams and user requests in near real-time. The author compares the proposed solution with static, random and proportional task allocation strategy. Simulation results show that the proposed solution outperforms another strategy in response time and in successful edge data center finding for the task.

E. Distributed Approach

Fog nodes are located to close to IoT devices. Therefore, data generated from IoT devices can move to fog nodes for processing. Some fog nodes are lightly loaded while others may be overloaded. Thus, load allocation among fog nodes may affect by communications latency and computing latency. To solve this problem, Fan et. al. [15] proposed a distributed workload balancing approach in a fog network to minimize the latency by associating IoT devices to suitable fog node.

F. Adaptive Load Balancing

Hamrioui et.al. [16] proposed an adaptive load balancing algorithm which take into account changes occur in network. Author compared proposed solution with NewReno, JTCP and COCM. Simulation results shows that proposed approach outperforms in comparison with NewReno, JTCP and COCM. The proposed algorithm is Load Balancing Algorithm for IoT communications within e-health environment and it is based on the integration of IoT communication parameters in the flow control process supported by Transmission Control Protocol.

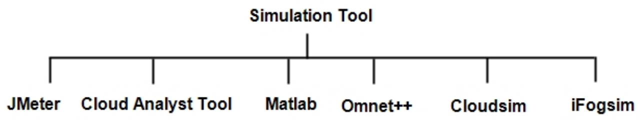

Fig. 6: Simulation Tool

G. Centralized Load Balancing Policy

Manju et. al. [17] proposed central load balancing policy for the fog computing environment. The author designed a policy such that load is evenly distributed among nodes and reduce task processing response time. Resources are divided into reliable and unreliable category at fog layer. Laptops, mobile phones, tablets, connected vehicle are considered as unreliable fog nodes while routers, switches, and base stations are considered as a reliable nodes. Reliable networking resources act as a controller node. Controller node receives updates about the number of available idle and busy nodes and is responsible for forwarding the task to appropriate nodes.

H. Dynamic Resource Allocation

Load balancing is key issue in fog environment. A resource is selected according to requirement. To tackle this situation, the dynamic resource allocation strategy can be used. This is achieved using static resource allocation and dynamic service migration [18].

I. Tabu Search

Tabu search find out an optimal solution using neighborhood search which iteratively moves to better solution every time. This search will stop until some stopping condition found. Local search fails to explore poor score areas. But tabu search explores search space which is unexplored and carefully scrutinizes the neighborhood of each solution. Optimal load balancing is major concern in fog computing. To achieve optimal load balancing, Tellez et. al. [19] use tabu search fog computing for load balancing.

VII. COMMON SIMULATION TOOL AND PERFORMANCE METRICS

Concerning unpredictable behavior of fog nodes, a simulation environment can be created using a simulation tool to understand the performance of load balancing algorithms. Table I shows summary of experimental configuration while Table II shows various performance metrics identified in load balancing strategies. Fig 6 shows the simulation tool identified for fog environment. They are described as follows:

- JMeter: The Apache JMeter is desktop-based application designed in Java language to load test functional behavior and measure performance. It is designed for testing of web applications. It can be used to test performance both on static and dynamic resources. It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types [20].
- Cloud Analyst Tool: Visual modeling and simulation of large scale applications can be done using cloud analyst tool. This tool is kept on the top of CloudSim. It can generate information about response time of the request, allows description of application workloads, including information of the geographic location of users generating traffic and location of data centers, number of users and data centers, and a number of resources in each data center [21].
- iFogSim: iFogSim enables the simulation of resource management and application scheduling policies across edge and cloud resources under different scenarios and conditions. iFogSim is designed in a way that makes it capable of evaluation of resource management policies applicable to fog environments with respect to their impact on latency, energy consumption, network congestion, and operational costs. iFogSim also allows application designers to test the design of their application against metrics like cost, network use, and perceived latency. It simulates edge devices, cloud data centers, and network links to measure performance metrics [22].
- Omnet++: Omnet++ is C++ based simulator used for simulation of communication network, processor and distributed systems. It provide basic machine and tools to write such simulation. Omnet++ module communicate with each other using message passing mechanism [23].
- Cloudsim: Cloudsim is java based simulator used for modeling and simulation of cloud and fog computing environment [24]. It helps to explore solution on load balancing algorithms.
- IoTSim-Edge: IoTSim-Edge is java based simulator used for modeling and simulation of IoT and Edge computing environment [25]. It provides all the IoT and edge/fog computing features to experiment the ideas.

VIII. Conclusions

In this paper, we survey load balancing techniques used for computing. Fog computing is used as a fast response time system. Thus, to get a quick response, it should be placed before cloud computing. We first describe fog computing architecture which is three-layer viz. IoT layer, fog computing layer, cloud computing layer. We surveyed existing load balancing algorithm used for fog computing. We have identified the optimization parameter and simulation tool used for fog computing. We have summarized our finding in Table I and II.

Algorithm	**Experimental Configuration**	**Simulation Tool**
Graph Partitioning [10]	Ten computers with i5 processor with Linux System and 4 GB memory	JMeter
Hill Climbing approach [11]	Simulation with six cluster	Cloud Analyst Tool
Secure load balancing [12]	Intel Core i7 processor and 8 GB RAM	Matlab
Adaptive Approach [16]	Nodes are deployed in 200m X 200m	Omnet++
Centralized load balancing approach [17]	Datacenter and Fog node	Cloud Analyst Tool
Dynamic Resource Allocation [18]	Fog service and Fog nodes are used	Cloudsim

TABLE I: A Summary of Experimental Configuration and Simulation Tool

Metric	**Optimization Behavior**	**Algorithm**
Latency	Minimize	SDN [10]
Run time	Minimize	Graph Partitioning [10]
Average Processing Time, Response Time	Minimize	Hill Climbing approach [11]
Response Time	Minimize	Secure load balancing [12]
Average throughput rate	Minimize	Adaptive Approach [16]
Latency	Minimize	Distributed Approach [15]
Latency	Minimize	Centralized Load Balancing [17]

TABLE II: Identified Performance Metrics

References

[1] M. Aazam and E.-n. Huh, "Fog computing and smart gateway based communication for cloud of things," 08 2014, pp. 464–470.

[2] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2603–2616, June 2018.

[3] X. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, "Optimal schedule of mobile edge computing for internet of things using partial information," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2606–2615, Nov 2017.

[4] X. Lyu, C. Ren, W. Ni, H. Tian, and R. P. Liu, "Distributed optimization of collaborative regions in large-scale inhomogeneous fog computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 574–586, March 2018.

[5] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C. Lin, "Edge of things: The big picture on the integration of edge, iot and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.

[6] S. K. Mishra, D. Puthal, B. Sahoo, S. Sharma, Z. Xue, and A. Y. Zomaya, "Energy-efficient deployment of edge dataenters for mobile clouds in sustainable iot," *IEEE Access*, vol. 6, pp. 56 587–56 597, 2018.

[7] M. Tiwary, D. Puthal, K. S. Sahoo, B. Sahoo, and L. T. Yang, "Response time optimization for cloudlets in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 119, pp. 81 – 91, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731518302430

[8] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 493–512, 2014.

[9] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Threats to networking cloud and edge datacenters in the internet of things," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 64–71, 2016.

[10] S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Communications*, vol. 13, no. 3, pp. 156–164, 2016.

[11] M. Zahid, N. Javaid, K. Ansar, K. Hassan, M. KaleemUllah Khan, and M. Waqas, "Hill climbing load balancing algorithm on fog computing," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, F. Xhafa, F.-Y. Leu, M. Ficco, and C.-T. Yang, Eds. Cham: Springer International Publishing, 2019, pp. 238–251.

[12] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and sustainable load balancing of edge data centers in fog computing," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 60–65, 2018.

[13] D. Puthal, R. Ranjan, A. Nanda, P. Nanda, P. P. Jayaraman, and A. Y. Zomaya, "Secure authentication and load balancing of distributed edge datacenters," *Journal of Parallel and Distributed Computing*, vol. 124, pp. 60 – 69, 2019.

[14] D. Puthal, S. P. Mohanty, S. A. Bhavake, G. Morgan, and R. Ranjan, "Fog computing security challenges and future directions [energy and security]," *IEEE Consumer Electronics Magazine*, vol. 8, no. 3, pp. 92–96, 2019.

[15] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered iot," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2018.

[16] S. Hamrioui and P. Lorenz, "Load balancing algorithm for efficient and reliable iot communications within e-health environment," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[17] A. B. Manju and S. Sumathy, "Efficient load balancing algorithm for task preprocessing in fog computing environment," in *Smart Intelligent Computing and Applications*, S. C. Satapathy, V. Bhateja, and S. Das, Eds. Singapore: Springer Singapore, 2019, pp. 291–298.

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

[19] N. Tellez, M. Jimeno, A. Salazar, and E. Nino-Ruiz, "A tabu search method for load balancing in fog computing," *International Journal of Artificial Intelligence*, vol. 16, pp. 78–105, 09 2018.

[20] R. Abbas, Z. Sultan, and S. N. Bhatti, "Comparative analysis of automated load testing tools: Apache jmeter, microsoft visual studio (tfs), loadrunner, siege," in *2017 International Conference on Communication Technologies (ComTech)*, 2017, pp. 39–44.

[21] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 446–452.

[22] H. Gupta, A. Dastjerdi, S. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments," *Software: Practice and Experience*, 06 2016.

[23] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," 01 2008, p. 60.

[24] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software Practice and Experience*, vol. 41, pp. 23–50, 01 2011.

[25] D. N. Jha, K. Alwasel, A. Alshoshan, X. Huang, R. Naha, S. Battula, S. Garg, D. Puthal, P. James, A. Zomaya, and S. Dustdar, "Iotsim-edge: A simulation framework for modeling the behaviour of iot and edge computing environments," 10 2019.