Detailed Analysis of the QoS-aware Service Provisioning Framework and its Algorithm in Fog Computing

Dhananjaya M K [#1], Dr. Kalpana Sharma [*2] and Dr. Amit Kumar Chaturvedi *[3]

[#1] *Ph.D. Scholar, Computer Sc Dept., Bhagwant University, Ajmer*
*[2] *Assistant Prof., Computer Sc Dept, Bhagwant University, Ajmer*
*[3]*Assistant Prof., MCA Deptt., Govt.* Engg. College, Ajmer
[#1] dhanunitte@gmail.com , [*2]kalpanasharma56@gmail.com, *[3] amit0581@gmail.com

Abstract

The backbone platform for providing computing power and storage services to IoT enabled environments is frequently cloud computing. The rapid processing of massive volumes of data is required by current Internet of Things (IoT) technology and services. Surveillance and object detection are currently the main goals, Nevertheless, network latency cannot be reduced by cloud computing to fulfil reaction time requirements. Cloud computing's issues are resolved by fog computing, which brings cloud services to the boundaries of the network. Fog computing reduces latency and network capacity utilization by moving calculations closer to the data-generating devices. Service provisioning is a promising stage for fog-cloud computing systems. Two fog-based frameworks, Fog Plan and Fog Offload, which can improve the QoS by reducing service delay, are provided in this paper. They make it possible for IoT application tasks to be scheduled and executed on a group of fog nodes. The impact of security concerns is also examined, along with potential fixes, in this study, which offers future security-related guidance to those in charge of creating and maintaining the Fog system. Asset provisioning is a promising stage for fog-cloud computing systems.

Keywords: Cloud Computing, Internet of Things (IoT), Surveillance and Object detection, Fog computing, Quality of Service (QoS).

1. Introduction

The previous years, the traditional methods of processing, data storage, and service supply have gained a new dimension to cloud computing. Deep neural networks are one of the rapidly emerging techniques in intelligent data processing (DNNs). They provide a variety of machine learning (ML) techniques that frequently outperform cutting-edge models. As a result of recent advancements in the Internet of Things (IoT), Big Data, and Machine Learning, more and more complex applications are appearing (ML). Real-time disease detection, autonomous vehicles, drone package delivery, and smart manufacturing are a few examples of such applications. Because many of these emerging applications require real-time data processing and have sensitivity to delays, it is challenging to ensure that they will receive a high quality of service (QoS) [1]. Since fog computing puts processing, storage, and networking resources closer to the user, it is seen to be one possible method for attaining QoS for these applications. Fog, however, provides some benefits, such as low latency, because processing can take place at the network edge, close to the end devices. The capacity to enable processing at specific locations is provided by these network edge devices, sometimes known as fog nodes, but they also provide a variety of security problems. In this dissertation, we demonstrate how fog computing may raise the quality of service in a number of cutting-edge ML and IoT applications. To enhance the quality of service (QoS) of deep learning applications distributed via network elements, such as fog nodes and edge devices, we present two novel techniques: ResiliNet and deepFogGuard. In the event of physical node failures, the two techniques are meant to increase the failure-resiliency of distributed neural network inference. Unlike the ResiliNet, whose design is based on a revolutionary method, deepFogGuard is based on the

idea of skip connections. Then, we provide two fog-based frameworks, Fog Plan and Fog offload, which can enhance QoS by reducing service delay [2]. Typical networking hardware, such as routers, switches, set-top boxes, proxy servers, Base Stations (BS), etc., is frequently present in a fog computing environment and can be positioned close to IoT devices and sensors. These parts offer a variety of processing, storage, networking, etc. capabilities that can help service-applications run more smoothly. Fog computing has made it possible for cloud-based services to be broadly distributed around the world because of the networking components. Fog computing also facilitates interoperability, scalability, location awareness, and real-time interactions [3].

2 Related work

According to Badidi Elarbi [4], the extensive use of IoT technologies and sensors has led to the delivery of many services in new and faster ways. Yet, time-sensitive IoT applications are often unable to cope with the substantial latency that IoT data streams may encounter when transmitted to the cloud. For this kind of application, fog computing-based solutions are gaining popularity because to the low latency they can guarantee and give. Abdu Manis Haruna's [5] Fog computing framework for distributed computing aims to facilitate analytics and IoT gadgets. Due to the increasing growth in the number of ubiquitous network devices, also known as IoT devices, new kinds of mobility, resource allocation, and management are impending in Fog infrastructures because different applications have varying requirements, particularly in terms of response time.

Bayer Timo and Moedel Lothar [6] suggested that Combining cloud computing and fog computing presents unique difficulties in terms of data integration, service execution, and communication capabilities. One area of particular interest is the decision of whether a service is better performed on a fog node or in the cloud. The most important distribution criterion was provided in this work, along with an architecture for measuring and gathering the necessary data for service distribution. A appropriate distribution method was then applied, and the services were finally deployed on the best node. The architecture also offers a great setting for creating new distribution algorithms or refining old ones to meet specific needs brought on by mixing cloud and fog computing.
Kumar According to Mohit and Sharma [7], an increase in requests over the past 10 years has led to an increase in burden in cloud systems. Inefficient scheduling techniques struggle with resource imbalances, which worsen service quality. The primary idea behind scheduling is to allocate complex and varied tasks among cloud resources in order for the scheduling algorithm to avoid the imbalance problem. According to M. Wazid and A. K. Das [8], many fog systems will be much smaller than clouds (such as a fog node on a car, in a factory, or on an oil rig), and as a result, they might not have as many resources as the clouds to defend themselves. Additionally, it's possible that not all fog systems have the worldwide intelligence required for threat detection. However, Fog's proximity to end users and location on the edge also make it possible for it to handle particular emerging IoT security challenges. In the actual world, resources are generally distributed unevenly due to a variety of factors, such as the network infrastructure, incoming data amount, and setup time. Yet, when all fog nodes perform at their maximum levels in accordance with each resource that is available, the pressures of the Reduced cloud and network traffic allows an FC environment to host more DL applications [9].

3. Role of Mobility in Fog computing

Weak mobility and strong mobility are the two different types of node mobility. Hardware problems or battery drain are the main causes of inactivity. If weak mobility is not immediately identified, there will be limited communication between sensor nodes.. High mobility happens when a node intentionally switches from one gateway in the same network to another or moves as a result of environmental causes like wind, water, or rain. From a different angle, node mobility can be divided into two groups: micro and macro. A sensor node moving from one gateway to another inside the same network is what causes micro mobility. As a node switches from one network to another, macro mobility takes place. There is just one point of view, which has been acknowledged.

The Internet of Things (IoT) nodes will be more mobile and geographically dispersed to an emerging computing architecture called as "The Fog," which will also give end users context-aware applications with low latency [10]. It serves as a barrier separating IoT devices from the Cloud. However, a lot of the requirements created by fog computing increase the cost of security administration. The confidentiality, integrity, and availability of data and applications are at risk due to the cloud's inherent security and trust flaws and some exposed IoT characteristics. For large data rate applications, fog-based systems have problems that can't be fixed with present methods. For instance, mobility management cannot be guaranteed if the connection between Fog and the Cloud is lost.

We do not consider nodes with fluctuating behaviour that momentarily flip between gateways. Due to its ability

to avoid gateway overload, node oscillation is essential for the handover process. As a result, there may occasionally be a brief loss of connectivity between gateways and other sensor nodes.

Many real-time IoT systems must support mobility due to the possible repercussions of missing or delayed data when travelling. In order to enable mobility, an IoT system must feature a handover or hand-off mechanism in charge of easily de-registering a sensor node from a source access-point and registering it to a new access-point. The high requirements for security, latency, network coverage, and dependability make it difficult to implement a better handover system for complete mobility assistance in crucial areas like healthcare.

The fact that distributed storage and fog services are provided by edge gateways for Internet of Things (IoT) devices makes this problem substantially more challenging [11]. To update and synchronise the distributed storage when controlling mobility, a handover mechanism must connect seamlessly with Fog services.

4. Methodology

The programming model for foglets that Suarez et al. suggested enables distributed programming among fog nodes. For storing and retrieving application-generated data on local nodes, Foglets offers APIs for spatiotemporal data abstraction. Foglets maintains the application components on the Fog nodes and configures its activities for a specific region using the Foglets API [12]. Container-based visualisation is used to implement foglets. The study's model accounts for the fog nodes' constrained capacity, the unpredictable nature of request arrival, the cost of sending requests to distant clouds, and the cost of bringing new services to fog nodes.

A. Proposed Structure for QoS-Aware Fog ServiceProvisioning

Our suggested four-layer method for offering fog services with QoS awareness is shown in Figure 1. IoT Gateways (IoTG), Fog Broker, clusters of Fog nodes, and higher level applications make up the framework's core parts. The Fog reduces network traffic by processing data streams, which also enables time-sensitive applications to become dormant.The Fog Broker, a crucial part of the system, is in charge of planning task execution on appropriate Fog Nodes utilising particular orchestration techniques [13].

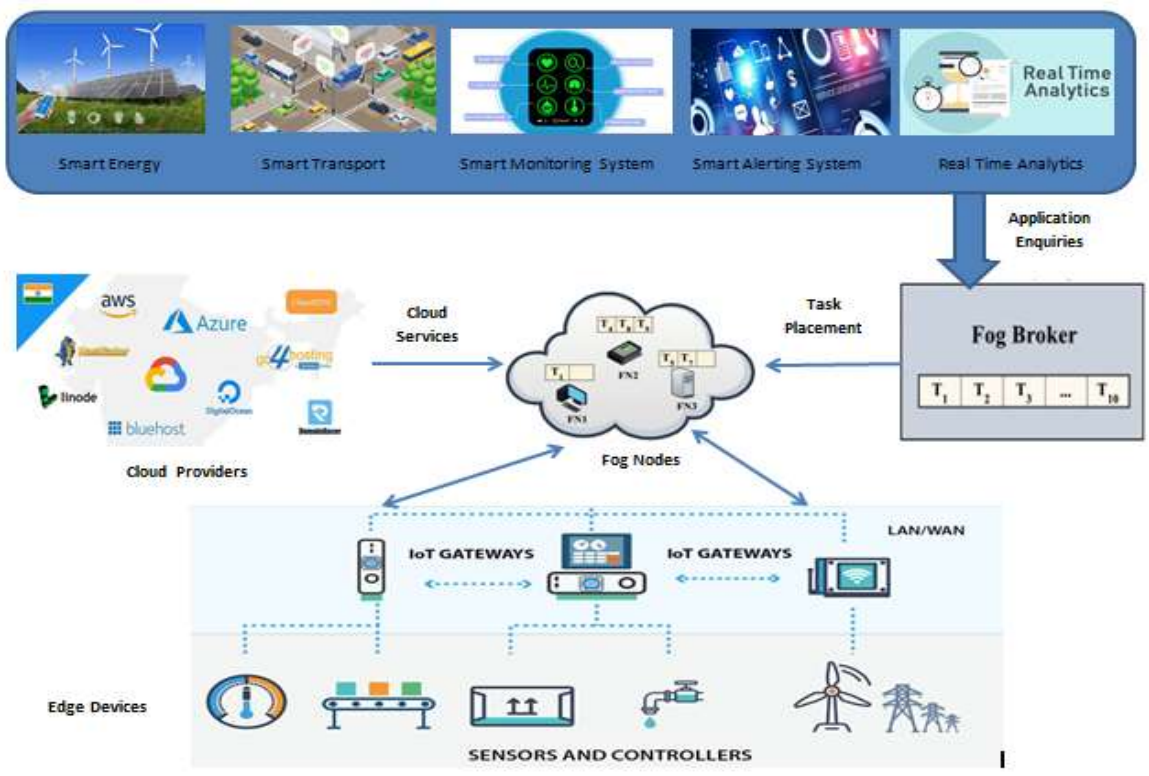

Fig. 1 Architecture of Fog-based Service Provisioning

B. *Service Placement Algorithm*

Every fog control node's controller component executes the suggested methodology. To solve the issue of service placement in the fog-cloud computing environment, we present the Most Delay-sensitive Application First in this paper (MDAF). The main tenet of MDAF is to host delay-sensitive application services as nearly as possible on resources that are close to the customers [14]. The initial computational devices in the fog are used because they are more accessible to customers than cloud resources. To assess each application's delay sensitivity, we use its deadline. There, the MDAF method is used to allocate each application's processing services to the best computing resources [15]. The method also aims to distribute services as widely as feasible over fog computing resources to increase QoS while lowering cloud execution costs.

5. Experimentation Results

Results of experimental work performed for the two service deployment strategies, overcast and foggy, are recorded. The execution plan was created with consideration for concerns including server usage, network congestion, and application response time. Every Fog server in this execution plan has a defined computing capacity and is computed in MIPS. Similar to how incoming jobs have varied processing loads, they are also counted in MIPS[16]. The size, mean-inter-arrival time, and change in the amount of tuples all have a direct impact on how well the Fog device works.

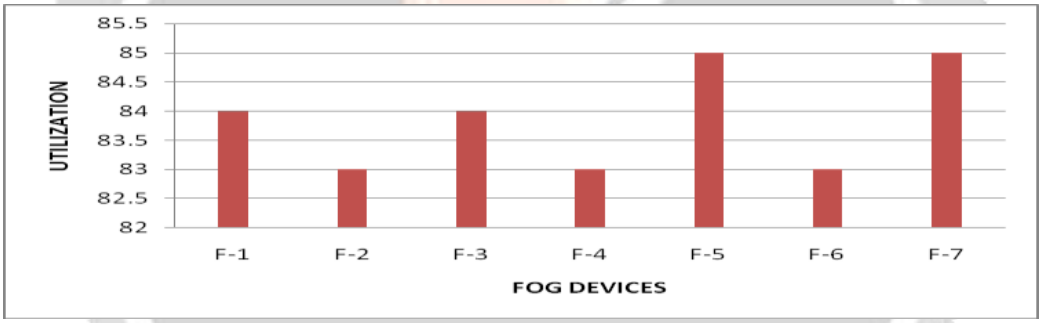

Fig.**2** Server utilization

Fig. 2 displays how the seven Fog servers are being used. The Y-axis shows the percentage of utilisation for fog devices, and the X-axis shows the fog devices themselves [17].

6. Conclusion

Allocating IoT applications to the appropriate Fog node in the cluster of Fog Nodes that is now accessible through the services of the Fog Broker is one of the key components of the architecture for provisioning Fog Services. Network traffic is decreased and the latency of time-sensitive applications is decreased as a result of the fog's processing of data streams. The primary element of the architecture is the fog broker, which is in charge of allocating task execution to the appropriate fog nodes using a variety of scheduling methods. In fog computing, edge devices—which have more processing power and are situated nearer the data sources than cloud resources—handle user requests. This research proposes an effective resource management technique that takes into account network utilisation, execution time, and energy usage as QoS criteria. Finding a workable solution to the service placement problem in such systems is the difficult part. To address this issue, fog-cloud computing systems have implemented a QoS-aware service placement policy that places the most delay-sensitive application services as near to the clients as is practical. The efficacy of the strategies has been assessed in a fog computing environment using the iFogSim toolkit, and experimental findings indicate that the suggested technique performs better in terms of QoS measures. We looked at services as a type of computation in an application. A service is regarded by us as one processing unit. That means that deploying a service just requires the use of one device. Occasionally, the output of one service will act as the input for another service.

References:

[1]. Design of secure key management and user authentication approach for fog computing services by M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos is published in Next Generation Computer Systems, vol. 91, 2019, pp. 475–492.

[2]. The authors of "Fog Computing: Helping the Internet of Things Realize Its Potential," Dastjerdi, Amir Vahid, and RajkumarBuyya, appeared in Computer 49, no. 8 (2016): 112-116.

[3]. "On the Effect of Fog Colonies Partitioning on Fog Application Makespan," in IEEE, 6th International Conference on Future Internet of Things and Cloud (FiCloud), editors: C. Guerrero, I. Lera, and C. Juiz, pp. 377–384.

[4]. Advances in Intelligent Systems and Computing, Cham, January 2019, pages 330–341; "A Fog Node Architecture for Real-Time Processing of Urban IoT Data Streams" University of the United Arab Emirates, Springer International Publishing, Badidi Elarbi, Al Ain, and the United Arab Emirates.

[5]. Eighth International Conference on Innovative Computing Technologies (INTECH), pages 191–196. User mobility and resource scheduling and management in fog computing to enable IoT devices IEEE, 2017, Francisca Oladipo, Ariwa Ezendu, and Abdu Manis Haruna.

[6]. A Fog-Cloud Computing Infrastructure for Condition Monitoring and Providing Industry 4.0 Services, in CLOSER, pp. 233–240. articles in science and technology from SCITEPRESS, 2019. Christoph Reich, Lothar Moedel, and Timo Bayer,

[7]. "A complete investigation 143:1-33 (October 2019), Journal of Network and Computer Applications, for scheduling methods in cloud computing. Goel Anubhav, Singh S. P., Kumar Mohit, and Sharma S. C.

[8]. IEEE Internet Things J 3(6):854-864, 2020, "Fog and IoT: An Overview of Research Opportunities," Tao Z., Mung C.

[9]. Internet of Things, Elsevier, "Fog computing: principles, architectures, and applications," pp. 61–75 Sun X. and N. Ansari (2018) Internet of things is made possible by mobile edge computing. (Paper Invited) (2016) IEICE Trans Commun E101-B(3):60 Vahid Dastjerdi, A. Gupta, R. Buyya, N. Calheiros, S. K. Ghosh (2016).

[10]. International Journal of Cloud Computing, "Docker Container Deployment in Fog Computing Infrastructures," pages 1–20, 2018. Arif Ahmed and Guillaume Pierre.

[11]. Service Oriented Computing and Applications, vol. 11, no. 4, pp. 427-443, 2017. "Optimized IoT service placement in the fog." M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner are among the authors.

[12]. Task Placement on Fog Computing More Efficient for Provision of IoT Applications, Wireless Communications and Mobile Computing, vol. D. H. Nguyen, V. A. Le, M.-Q. Tran, D. T. Nguyen, and T. V. Pham.

[13]. An Efficient Fog Computing Architecture and Algorithm for Resource Provisioning, International Journal of Information Engineering and Electronic Commerce, vol. 8, no. 1, (2016), p. 48. S. Agarwal, S. Yadav, and A. K. Yadav

[14]. "QoS-aware Autonomic Resource Management in Cloud Computing: A System Analysis," ACM Computing Surveys, vol. 48, no. 3, (2015), pp. 1-46.

[15]. A gateway based fog computing architecture for wireless sensors and actuator network, 18th International Conference on Advanced Communication Technologies (ICACT), IEEE, 2016, pp. 210–213, by W. Lee, K. Nam, H.-G. Roh, and S.-H. Kim.

[16]. A. Vahid Dastjerdi, S. K. Ghosh, H. Gupta, and R. Buyya's (2016) preprint, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge, and Fog Computing Environments," is available online.