

Article

A Fog Computing Architecture with Multi-Layer for Computing-Intensive IoT Applications

Muhammad Muneeb¹, Kwang-Man Ko^{1,*} and Young-Hoon Park² ¹ Department of Computer Engineering, Sang-Ji University, Wonju 26339, Korea; muneeb@sangji.ac.kr² Division of Computer Science, Sookmyung Women's University, Seoul 04310, Korea; yh.park@sookmyung.ac.kr

* Correspondence: kkman@sangji.ac.kr; Tel.: +82-33-730-0486

Abstract: The emergence of new technologies and the era of IoT which will be based on compute-intensive applications. These applications will increase the traffic volume of today's network infrastructure and will impact more on emerging Fifth Generation (5G) system. Research is going in many details, such as how to provide automation in managing and configuring data analysis tasks over cloud and edges, and to achieve minimum latency and bandwidth consumption with optimizing task allocation. The major challenge for researchers is to push the artificial intelligence to the edge to fully discover the potential of the fog computing paradigm. There are existing intelligence-based fog computing frameworks for IoT based applications, but research on Edge-Artificial Intelligence (Edge-AI) is still in its initial stage. Therefore, we chose to focus on data analytics and offloading in our proposed architecture. To address these problems, we have proposed a prototype of our architecture, which is a multi-layered architecture for data analysis between cloud and fog computing layers to perform latency-sensitive analysis with low latency. The main goal of this research is to use this multi-layer fog computing platform for enhancement of data analysis system based on IoT devices in real-time. Our research based on the policy of the OpenFog Consortium which will offer the good outcomes, but also surveillance and data analysis functionalities. We presented through case studies that our proposed prototype architecture outperformed the cloud-only environment in delay-time, network usage, and energy consumption.

Keywords: IoT; data analysis; offloading; edge computing; fog computing

Citation: Muneeb, M.; Ko, K.-M.; Park, Y.-H. A Fog Computing Architecture with Multi-Layer for Computing-Intensive IoT Applications. *Appl. Sci.* **2021**, *11*, 11585. <https://doi.org/10.3390/app112411585>

Academic Editor: Byung-Seo Kim

Received: 24 September 2021

Accepted: 25 November 2021

Published: 7 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, with the advent of high-performance and high-spec IoT devices, SNS data, e-mail, Internet stream data, multimedia data, etc. are collected and large-capacity transmission is possible. Big data refers to the large amounts of un-formalized, semi-formalized or formalized data that flows continuously through and around organizations [1]. Big data analysis provides the advantage of reducing costs when storing huge data while performing various analyzes in various and efficient ways [2]. The emerging 5G networks intend to accomplish new opportunities by introducing very high carrier frequencies. The cloud as the only solution for addressing these problems is not applicable in this random nature of internet. The Big data may be generated from different resources such as e-Healthcare, smart city, etc. Data mining and data analysis are a core methodology that assists investigate huge data to explore patterns, meaningful information in the data set and this information can be used for another analysis to help answer complex questions. In the form of wireless wearable accessories which are attached to a patient for retrieving the information and then analyzing of data performed [3]. That is the challenge to tackle these issues through coordinating these resources which will provide effective utilization of handling massive amount of distributed data. The fog computing nodes usually do not have enough storage and computing resources for dealing with that huge IoT data. The design uses local computing nodes, between the end points (e.g., sensors, cameras, etc.)

and the cloud data center. Early big data applications were mostly deployed on-premises, whereas Hadoop was originally designed to work on clusters of small machines and edges. An increasing number of technologies such as CPI, storage, and networks facilitate processing data in the fog-cloud. For example, Cloudera [4] and Hortonworks [5] provide their distributions of the big data framework on the Amazon Web Services [6] and Microsoft Azure [7] clouds.

In [8] research, they presented a health-care system to support autonomic analysis of Medical IoT data. A Medical IoT-enabled health care system [9] in which smart gateways used to provide warehouse and data processing. A prototype of smart-city surveillance was developed [10], which pre-process the necessary data and make decision. The machine and deep learning have been widely used in various fields, the use of machine and deep learning in health-care is discussed in [11] to high accurate predicting and classifying the data. In [12] a fog based IoT health-care system is proposed for the integration of cloud-services in interoperable health-care solutions upon traditional health-care framework.

The main motivation of this paper is to use this multi-layer fog computing platform for enhancement of data analysis system based on IoT devices in real-time, as the current scenario of whole world due to COVID-19 has highlighted the importance of data and analysis system to avoid these circumstances in future with proper planning and management. We will suggest a multi-layered framework which many IoT devices will be used and connected to edge-fog gateways in collaborative manner to gather the optimal data needed. Through this study, we performed comparative experiments on delay time, network usage, and energy consumption when computing-intensive IoT applications are performed in cloud-only environments and in our framework, multi-layered fog architectures.

The remainder of this article is organized as follows. In the next Section 2 provides a background and related work, which follows by the Section 3 provide the system overview of our idea, a fog computing real-time data analytics platform in detail. The Section 4 provide the use case and evaluation based on the performance and then the last section provides the conclusion and future directions.

2. Background and Related Work

The big-data analysis framework has capabilities to integrate, storage, pre-processing, manage, and apply sophisticated computational processing to the big data. There are platforms for big data in which Hadoop [13] is the basic and popular to provide a unified storage and processing environment for highly scalable data volumes. In IoT based edge computing environment, the volume and variety of data can cause consistency and processing issues, and data silos can result from the use of different platforms and data stored in a big data architecture. Big data analytics describes the process of performing sophisticated analytical tasks on data that typically includes grouping, aggregation, or iterative processes [2]. The raw data is generally unformalized that neither has a pre-defined data model nor is organized in a pre-defined manner. A ETL (Extract, Load and Transform) process extracts data from multiple sources, then transforms and loads it into a data storage for analysis [14].

2.1. Backgrounds

OpenFog Consortium (OpenFog) was released as an industrial alliance [15], which main goals was to accelerate the adoption and the deployment of fog computing, while addressing the main issues posed by this new paradigm. The consortium has published the architecture of an open fog computing system [16]. This provides a high-level framework of the fog node computing, their communications and management and will help drive standardization across the various layers and interfaces specified by the consortium. Ref. [17] was to implement a distributed and concurrent offloading framework, DisCO. A DisCO is to offload the computation-intensive or data-intensive division of the IoT application executed or collected on the edge device to the resourceful edge server with high computing resources and processing capacity in real time and then returned the results of executing.

A research contents to achieve profiling IoT applications in edge devices and migrating or transferring offload units with overhead to edge servers and concurrently executing them and returning results. Additionally, Ref. [17] designed a security mechanism technology to solve security issues that may occur when offloading a part of IoT application to edge server or exchanging data for simultaneous processing between mobile edge servers. A [18] introduced an intelligent edge-to-edge and edge-to-fog collaborative computing platform that enables collaborative processing in an edge-fog environment through elasticity and verify functionalities.

2.2. Related Works

The fog computing has special features which make it more appropriate for the applications necessary low latency, mobility, interactions, real-time analytics, and interplay with the cloud computing [19]. Less time-consume data analysis task can still go to the cloud for long-term storage and historical analysis. The various analysis techniques are addressed in [20]. In [21], the problem of minimizing time-delay was proposed as a delay-minimizing policy in which an analytical model was implemented to estimate delay time. Ref. [22] proposed an data analysis framework based on the fog computing for big data analytics in smart cities which is hierarchical, scalable, and distributed, and supports the integration of a massive number of things and services. The framework was consisting of four multi-layers. The FogGIS framework based on the fog computing was proposed for mining analytics from geospatial data. FogGIS had been used for preliminary analysis including compression and the transmission time to the cloud had been minimized by compression. Especially in healthcare it is becoming more attractive to deploy fog computing as structures introduce tightly connected to various medical devices into their health IT ecosystem.

3. Multi-Layered Architecture Overview

In our works we have presented a platform, which is a multi-layered architecture for data analytics of data come from smart IoT-based devices such as intelligent CCTVs. The tri-tiers consists of cloud computing, edge-fog computing and sensors which work together with one another. As we know that fog computing is mainly a virtualization technology that offers storage and computing between end devices and cloud layer. The schematic overview of our architecture is shown in Figure 1.

The first physical layer composes of a set of IoT devices and variable sensors. The core job of the sensors is to collect all data and send to edge-fog gateways via offloading. Then the second tier, which will be consists of edge gateways and edge-fog servers. When data arrives at edge gateways, the data needs to be filter-out, pre-processing, for further processing. In this process, 30–70% of meaningless data to data analytics is deleted. Through this, it is possible to reduce the data transmission burden and increase the data analysis processing speed. This tier work as a server. The amount of data arrives at edge servers then distributed among the various edge devices according to the computation requirements of the data for reducing the latency in real-time scenario. The offloading of requests works must be carried out using a proposed efficient task-scheduling algorithm.

3.1. Fog Layer Module

The Fog layer module components share the same concept with the OpenFog architecture, although the Edge devices behavior in the framework is a different way. Fog computing facilitates preprocessing data before they even reach the cloud, minimizing the communication time, as well as reducing the need for huge volume data storage via filtering. Generally, it is a suitable approach for the applications and services with the IoT. Our approach is concerned tightly with fog computing architecture. The edge devices are working in autonomous manner; therefore, every Fog Node (FN) which is comprise of edge servers and edge devices in the Fog layer is managing a set of computational tasks. In our architecture, we have introduced the data analytics and offloading module for the fol-

lowing tasks to perform such as; (i) the data generated from IoT devices gathered/collected for analysis if required and (ii) the task resources exceed from the given edge server then it will perform the offloading to other edge server in the premises or to the cloud. The task allocation and offloading are the important actions during the data analysis process because the completion of processes depends on it.

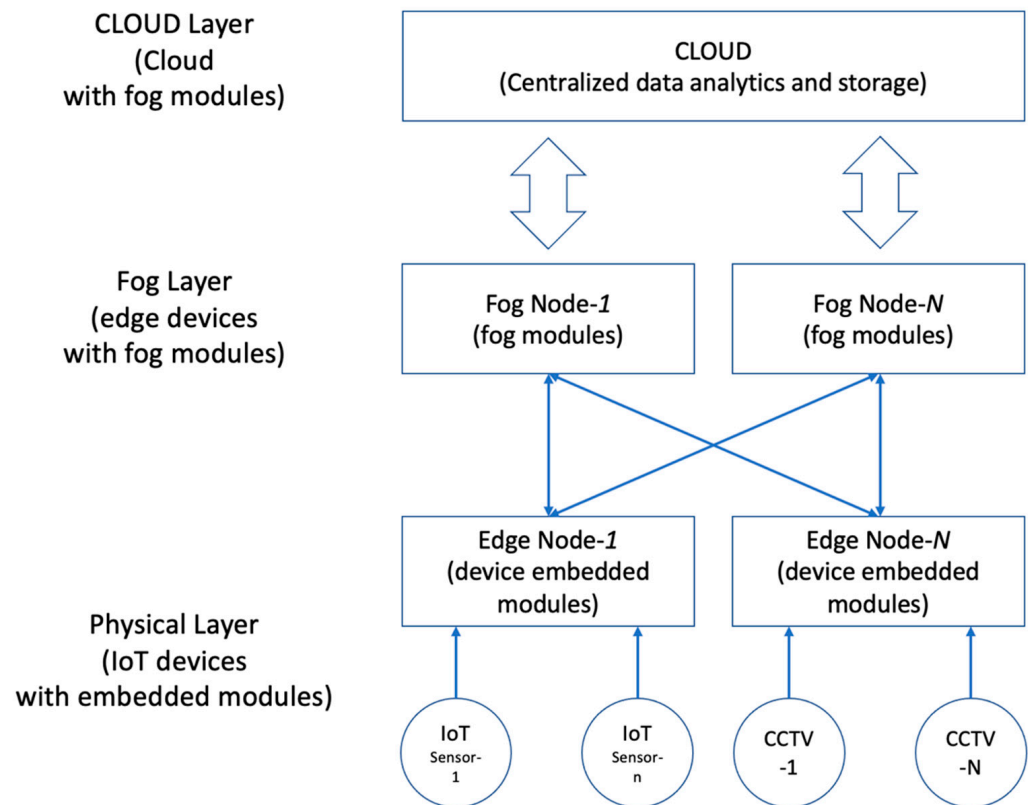


Figure 1. Schemes follow the same formatting. Overview of a typical Fog Module Architecture.

The fog module is based on [23] which will provide data analysis methods and the capabilities for IoT devices to communicate and collaborate with the cloud layer and the devices in the fog layer. It provides facilities for transferring fog side data to cloud and interacting with the cloud as well as a gateway. A gateway enables end-devices that are not directly connected to the internet to reach cloud services. Although the term gateway has a specific networking-oriented function, it is also used to describe a group of end-device that management and processes data on behalf of clustering of devices. Now, we will address our data analysis module in next sub-section for detail discussion.

3.2. Data Analysis in Fog Module

In Figure 2 shows data analytics processing close to the data source using Fog Module before the data volume and sizes are increase exponentially. In-stream data is locally analyzed in the Fog Module while data of the Fog Module is collected and transferred to the cloud side for offline data analytics and processes. The data analytics modules deployed in Fog Modules are periodically updated based on the policies decided and communicated by the cloud analytics. As the raw data are preprocessed, filtered, and cleaned in the Fog-module prior to offloading to the cloud modules, the amount and size of offloading data is smaller than the data generated by IoT devices. In addition, the analytics on Fog-module is real-time while the analytics on the cloud is offline. Fog-module has restricted computing and storage power compared with the cloud side, however, processing and management on the cloud side requires more latency time. The Fog module offers a high level of fault tolerance as the tasks can be transferred to the other Fog Modules in the vicinity in the

event of a failure. With the appearance of the resourceful IoT which enables real-time, high data-rate applications, moving analytics to the source of the data and enabling real-time processing seems a better approach. In future, Fog-module may adopt many different types of hardware components such as multi-core processor, GPU with fine granularity versus a cluster of similar nodes in the cloud.

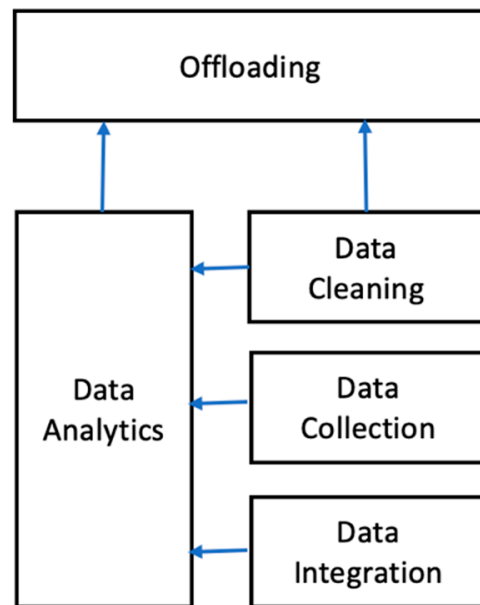


Figure 2. Data Analytics on the Fog Module before transferring to Cloud.

3.3. Offloading in Fog Module

In recent times, study shows that fog-edge computing technology provides an opportunity to overcome the limitation of hardware for the end user devices by offloading the computational-intensive tasks to the rich edge servers for execution. The execution in the edge servers follows the requirement of the task and provide the final results to the end devices. The fog computing paradigm brings both computational and network resources closer to the user. In this paper, we will keep a watchful eye on the fairness among the network nodes to which the offloading tasks are offloaded. A task offloading network architecture with multiple fog nodes is deployed in fog module. The performance factor will include the delay of each fog node. We will propose an offloading scheme with keeping in mind the selection of fog nodes according to the task scheduling metrics and then offload the task to the fog nodes which require minimum delay in performing the task.

Whenever a computing task is generated at the terminal node, a number of fog nodes are selected according to the performance requirements and the characteristics of these nearby fog nodes. Rather than computing the task locally, the task is divided into multiple subtasks and offloaded to these selected fog nodes for computing. The computing results are transmitted back to the terminal node afterwards. It is obvious that the nearest fog nodes with strongest computing capability will be selected when pursuing the minimum task delay and best performance.

In Figure 3, We show the process of offloading the information collected in IoT devices [17] based on our research.

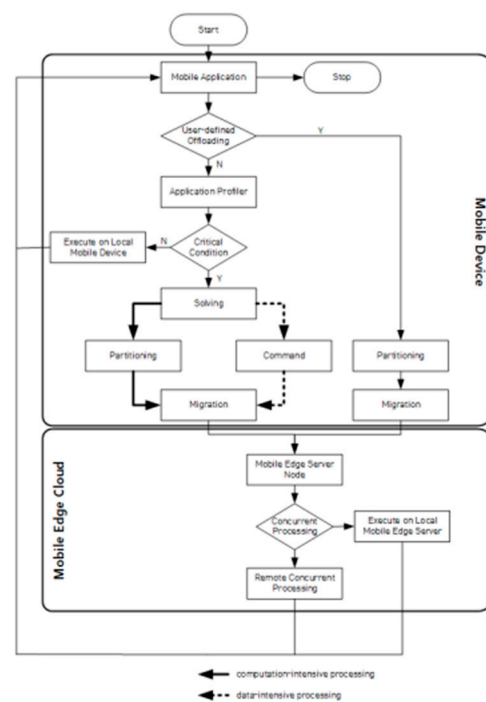


Figure 3. Data Analytics on the Fog Module before transferring to Cloud.

3.4. Cloud Layer

The cloud computing is the basic architecture of improving the computation of task in the IoT environment such as, Amazon AWS, Microsoft Azure IoT and so on. The fog computing is an extended form of cloud computing to provide the resources nearer the end user, therefore cloud cannot be excluded from the architecture. In this paper, the cloud plays the same role of providing the centralized data analytics and storage of data for future processing. However, our focus will be to less utilize the cloud in the computation of task, which can cause delay in performing the tasks. The tasks which require the more computation resources should be offloaded otherwise, the cloud act as a centralized storage. The cloud placement policy is based on the classical cloud-based implementation of applications where all units of an application execute in cloud data centers. The sensing, processing, and actuate loop in applications are executed by having sensors transmitting sensed data to the cloud side where it is processed, and actuators are notified whether action is required.

The cloud layer will be mainly responsible for complicated, resource consuming tasks and updates of rules for detection on the fog layer. Data collection phase is the prime aspects of these solutions, which sets the communication protocols between the components of an IoT software platform. The global data analysis and overall resource monitoring. Cloud gives an approved model of pay-as-you-go once Fog Module will be a attribute of the user. Depending on the IoT application, in the case of a limited access to power, fog core may be battery-powered and needs to be energy efficient while the cloud is supported with a constant source of power.

4. Application and Verification

We have discussed the application and how that can be validate with required performance metrics of our proposed model. Our proposed architecture is a multi-layered model in Figure 1. The physical layer belongs to the IoT based sensors and actuators or other devices. The gathered data from the end devices will be transmitted to fog layer for processing and analyzing of the tasks. The primary objective is to perform the data analysis without compromising the latency issue. For the purpose of data analytics in IoT based environment the fog computing is the best solution. The offloading of task can be

assigned within the edge devices and if it cannot be performed, then in that case it will be assigned to cloud layer. However, the main responsibility of cloud layer is to perform the management and storage of data for further analysis and monitoring of the system. For the purpose of comparing the results of our platform versus traditional fog computing platforms, we will use iFogSim toolkit to simulate the systems. The iFogSim [23] is a tool used for simulation of specified configurations and give the outputs, which makes convenient to observe end results. We will analyze the latency, computation overhead of our platform to show the impact of our platform. The proposed platform will benefit from both the cloud computing and fog computing. Our proposed ideas will be helpful in the field of IoT based applications and we will try to use the surveillance through camera networks as a case study for our project.

4.1. Camera Surveillance

The camera surveillance system has acquired much attention in recent years. The interdisciplinary applications in areas of safety, security, transportation, IoT, smart farm, smart factory, and health-care domain. However, monitoring video streams from the system of cameras manually is not practical and time-consuming works. Therefore, the tools and applications are needed that analyze data collecting from cameras and analyze the results in a way that is beneficial to the client. The requirements and functions of such a system has many factors according to the environment provided. The most common requirements are as follows: (i) low-latency communication, (ii) processing huge amount of data, and (iii) reduce complex and heavy computing processing.

The use case studied in this paper is providing the surveillance through camera and real-time analytics with minimum latency. The centralize tools are not desirable for analyzing camera-generated data due to the complex and huge volume of data, which requires the central processing system. This is the reason which leads to high latency in the system and the bandwidth requirement increases exponentially. The surveilling system aims at coordinate multiple cameras with different place to monitoring and surveil a given area. The intelligent camera which detects motion and transmit the video to surveillance application.

As shown in Figure 4, the surveillance will consist of motion detector, object tracker, PTZ Control, and user interface in the fog module. The surveillance is fed live streams by a lot of CCTV cameras and the PTZ control in each camera constantly control the PTZ attributes. In the sensor layer, a motion detection is implemented in the intelligent cameras used in the use-case. It will continuously read the raw video streams captures by the camera to find motion of an object. When the detection of motion event occurs in the camera's FOV, then it transmitted to next step processing for the object tracker. In the object tracking mode, the evaluated coordinates of the tracked objects and evaluated an optimal PTZ configuration of all the cameras covering the area so that the objects tracked can be captured in the most effective manner. The PTZ control will be enable on each camera and adjusts the physical camera to confirm to the PTZ parameters sent by the object tracker. The methodology presents a user interface by sending a fraction of the video streams including each tracked object to the end device.

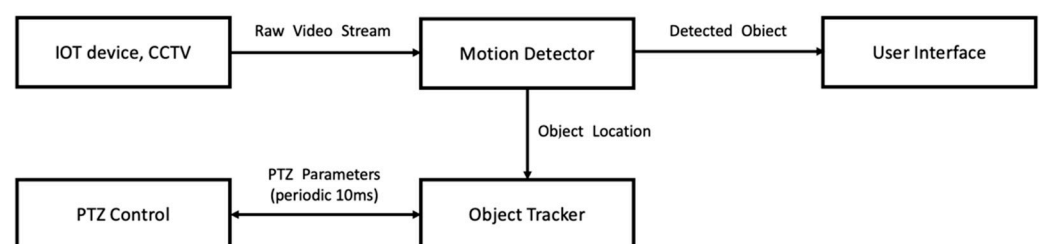


Figure 4. The Methodology of the surveillance case study.

4.2. Verification and Estimation

In this section, we have simulated a fog computing environment for the application use-case study using the topology of the use-case study in the Figure 5. Then, we evaluated efficiencies, latency, and network use.

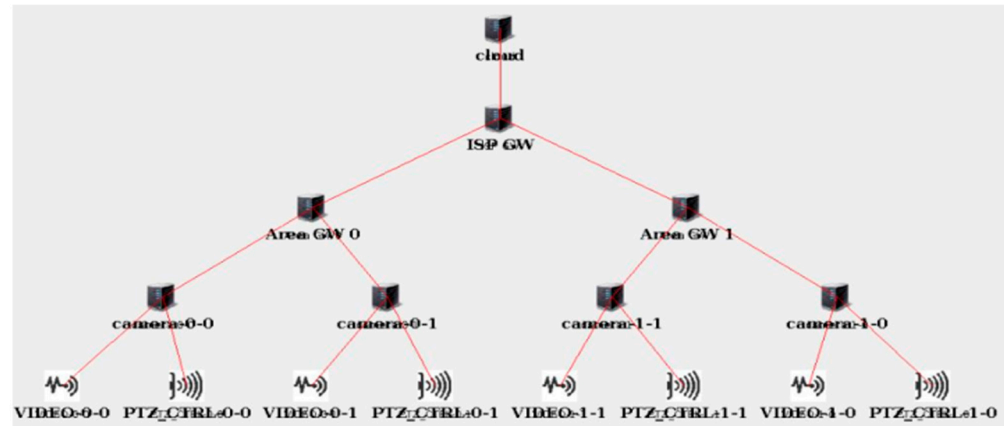


Figure 5. The topology of Fog Module use-case study.

- (1) Evaluation of the case study: For presenting the flexibility of iFogSim, the surveillance application has been used to estimated. The special surveillance sector has 4 intelligent cameras monitoring the area. Each of the four intelligent cameras can recognize and track objects within a 180-degree range. In addition, using 4 cameras, shooting data collected from 8 areas within the range is used. The number of special surveillance sector has been varied from 1 to 8. Four intelligent cameras are located at the vertex positions of the square. One camera can recognize and track an object within a 180-degree range, and by combining four cameras, capture data can be obtained from 8 locations within the shooting range. The number of surveilled areas is varied across physical topology configurations Config 1, Config 2, Config 3, and Config 4, having 1, 2, 4, and 8 surveilled areas, respectively. The network latencies between devices are listed in Table 1. On the basis of the aforementioned configuration of entities, a physical topology is designed. The topology has the cloud data center at the apex and smart cameras at the edge of the network. Smart cameras are fed live video streams in the form of tuples for performing motion detection and the PTZ control of the camera has been modeled as an actuator. Similarly, the analytics performs at cloud and edge are adopted for offloading application modules on the physical network. In case of cloud, all operators in the application are placed on the cloud data center except the motion detector module, which is bound to the intelligent cameras. However, in the edge-gateway, the object detector and object tracker modules are pushed to Wi-Fi gateways connecting the cameras in a surveilled area to the internet. The simulation of this use-case was performed for a period of 1000 s.

Table 1. Description of Network Link for Surveillance through Camera.

Source	Destination	Latenc (ms)
Camera	area switch	2
Area GW	ISP gateway	2
ISP Gateway	CLOUD DC	100

- (2) Average latency of control loop: Figure 6 shows the average processing latency time of sensing actuation control loop. In the case of cloud strategy, as shown in Figure 6, cloud data centers turned to a bottleneck in execution of the modules, which caused a notably important increase in latency. On the other hand, the Edge-GW succeeds in

maintaining low latency, as it places the modules critical to the control loop close to the network edge.

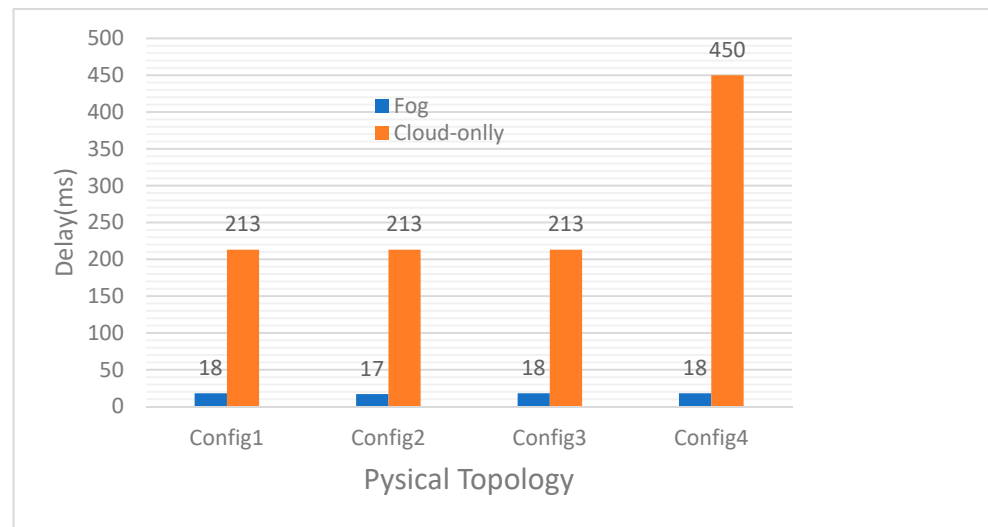


Figure 6. Average latency of control loop.

- (3) Network Usage: Figure 7 shows the network use of the intelligent surveillance application for the placement strategies. As number of devices connected to the application increases, the load on the network increases significantly in the case of cloud only deployment in contrast with edge-GW deployment. This observation can be attributed to the fact that in the Fog-based execution, most of the data-intensive communication takes place through low-latency links. Hence, Object tracker is placed on the edge devices, which substantially decrease the volume of data sent to a centralized cloud data center.

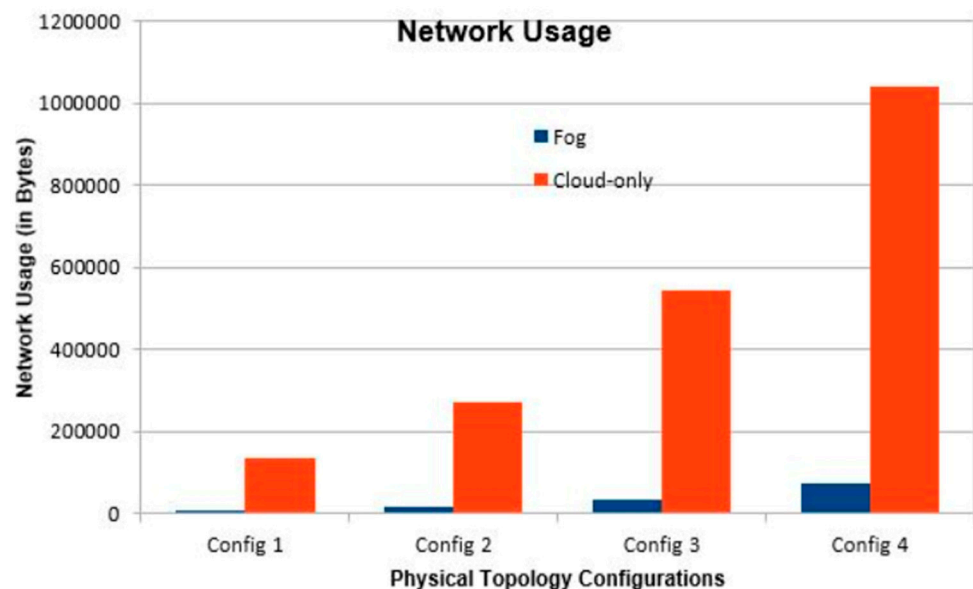


Figure 7. Average latency of control loop.

- (4) Energy Consumption: The energy consumption is also the important aspect while performing the offloading task. The Figure 8 demonstrates the energy consumption by grouping of devices in the simulation. The deployment of applications on fog devices has been compared to the deployment on the cloud data centers. The motion detection executed by camera in the video frames, which draw out a large amount of power. Hence, as shown in the Figure 8, when surveillance spots increase, the

consumption of energy in these devices increases, too. The cloud data center energy consumption decreases when the tasks perform at fog devices.

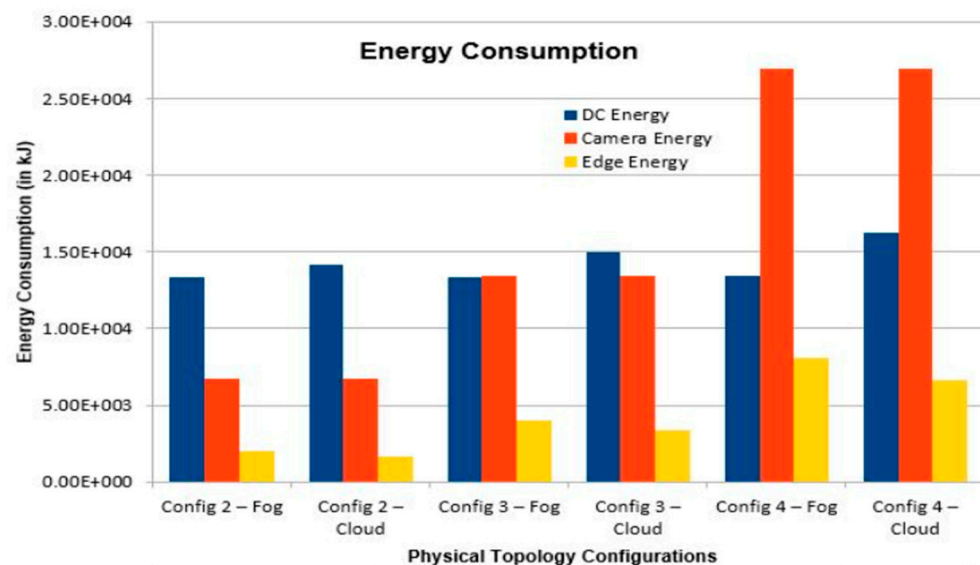


Figure 8. Energy consumption in Fog and Cloud placement.

5. Conclusions

In this article, we addressed iFogSim to model and simulate Fog computing environments. Specially, iFogSim allows investigation and comparison of resource management techniques on the basis of QoS criteria such as latency under different workloads and network usage. We have showed the case study of surveillance through camera networks with the simulation results. The results related to latency, network usage and energy consumption presented are showing the importance of the architecture proposed on the fog computing-based methodology. In the future, we plan to supplement and implement the collaboration method between fog computing nodes, which has not been completed in this paper. In this process, we plan to increase the possibility of collaboration between west-east computing nodes based on deep learning. In addition, there are many issues that may occur in the process of information transfer between offloading and fog computing nodes. The current experiment was conducted with laboratory-level simulations, but in the future, experiments to overcome unpredictable situations by conducting experiments in the real field where various variables exist are planned.

Author Contributions: Investigation, M.M. and K.-M.K.; Methodology, K.-M.K. and Y.-H.P.; Supervision, K.-M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Sang-Ji University Graduate School. This work was supported by the National Research Foundation of Korea(NRF)323grant funded by the Korea government(MSIT) (No. 2021R1F1A1048903).

Acknowledgments: We are thank to Sangji University Graduate School Supportings.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mehdipour, F.; Noori, H.; Javadi, B. Energy-Efficient Big Data Analytics in Datacenters. *Adv. Comput.* **2016**, *100*, 59–101.
2. Mehdipour, F.; Javadi, B.; Mahanti, A.; Ramirez-Prado, G. Fog computing realization for big data analytics. *Fog Edge Comput. Princ. Paradig.* **2019**, *30*, 259–290.
3. Chen, N.; Chen, Y.; Ye, X.; Ling, H.; Song, S.; Huang, C.-T. Smart city surveillance in fog computing. In *Advances in Mobile Cloud Computing and Big Data in the 5G Era*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 203–226.
4. Chawda, R.K.; Thakur, G. Big data and advanced analytics tools. In Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking (CDAN'16), Indore, India, 18–19 March 2016; pp. 1–8.

5. Hortonworks Institute. *Hortonworks*; Hortonworks Institute: Santa Clara, CA, USA, 2017.
6. Jackson, K.R.; Muriki, K.; Ramakrishnan, L.; Runge, K.J.; Thomas, R.C. Performance and cost analysis of the Supernova factory on the Amazon AWS cloud. *Sci. Program.* **2012**, *19*, 107–119. [[CrossRef](#)]
7. Gia, T.N.; Jiang, M.; Sarker, V.K.; Rahmani, A.M.; Westerlund, T.; Liljeberg, P.; Tenhunen, H. Low-cost fog-assisted health-care iot system with energy-efficient sensor node. In Proceedings of the 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 1765–1770.
8. Rahmani, A.M.; Gia, T.N.; Negash, B.; Anzanpour, A.; Azimi, I.; Jiang, M.; Liljeberg, P. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. *Future Gener. Comput. Syst.* **2018**, *78*, 641–658. [[CrossRef](#)]
9. Yi, H.; Cho, Y.; Paek, Y.; Ko, K. DADE: A Fast Data Anomaly Detection Engine for Kernel Integrity Monitoring. *J. Supercomput.* **2019**, *75*, 4575–4600. [[CrossRef](#)]
10. Faust, O.; Hagiwara, Y.; Hong, T.J.; Lih, O.S.; Acharya, U.R. Deep learning for healthcare applications based on physiological signals: A review. *Comput. Methods Programs Biomed.* **2018**, *161*, 1–13. [[CrossRef](#)] [[PubMed](#)]
11. Mahmud, R.; Koch, F.L.; Buyya, R. Cloud-fog interoperability in IoT-enabled healthcare solutions. In Proceedings of the 19th International Conference on Distributed Computing and Networking, Varanasi, India, 4–7 January 2018; pp. 1–10.
12. Hadoop. Available online: <http://hadoop.apache.org/> (accessed on 9 October 2020).
13. Intel Big Data Analytics White Paper, Extract. In *Transform and Load Big Data with Apache Hadoop*; Intel: Santa Clara, CA, USA, 2013.
14. Open Fog Consortium. Available online: <https://www.openfogconsortium.org/> (accessed on 15 June 2017).
15. OpenFog Reference Architecture for Fog Computing. Available online: <https://www.openfogconsortium.org/wp-content/uploads/OpenFog> (accessed on 19 June 2017).
16. Dastjerdi, A.V.; Buyya, R. Fog Computing: Helping the Internet of Things Realize Its Potential. *J. Comput.* **2016**, *49*, 112–116. [[CrossRef](#)]
17. Ko, K.; Son, Y.; Kim, S.; Lee, Y. DisCO: A distributed and concurrent offloading framework for mobile edge cloud computing. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; pp. 763–766.
18. Muneeb, M.; Joo, S.; Ham, G.; Ko, K. An Elastic Blockchain IoT-based Intelligent Edge-Fog Collaboration Computing Platform. In Proceedings of the 2021 International Symposium on Electrical, Electronics and Information Engineering, Seoul, Korea, 19–21 February 2021; pp. 447–451.
19. Alturki, B.; Reiff-Marganiec, S.; Perera, C. A hybrid approach for data analytics for internet of things. In Proceedings of the Seventh International Conference on the Internet of Things (IoT'17), Linz, Austria, 22–25 October 2017; p. 108.
20. Yousefpour, A.; Ishigaki, G.; Jue, J.P. Fog Computing: Towards Minimizing Delay in the Internet of Things. In Proceedings of the IEEE International Conference on Edge Computing (EDGE), Online and Satellite Sessions, 10–14 December 2021; pp. 17–24.
21. Tang, B.; Chen, Z.; Hefferman, G.; Wei, T.; He, H.; Yang, Q. A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities. In Proceedings of the ASE BD&SI '15 Proceedings of the ASE Big Data and Social Informatics, Kaohsiung, Taiwan, 7–9 October 2015.
22. Barik, R.K.; Dubey, H.; Samaddar, A.B.; Gupta, R.D.; Ray, P.K. FogGIS: Fog Computing for geospatial big data analytics. In Proceedings of the IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON), Varanasi, India, 9–11 December 2016; pp. 613–618.
23. Mehdi-pour, F.; Javadi, B.; Mahanti, A. FOG-Engine: Towards Big Data Analytics in the Fog. In Proceedings of the Dependable, Autonomic and Secure Computing, 14th International Conference on Pervasive Intelligence and Computing, Auckland, New Zealand, 8–12 August 2016; pp. 640–646.