Design and Implementation of Low-Cost Fog Computing Architecture for IoT-Based Applications

Ahmad Zainudin*, Cosmas Ifeanyi Nwakanma†, Dong-Seong Kim‡ and Jae-Min Lee§
*Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea
†‡§Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea
(*zai, †cosmas.ifeanyi, ‡dskim, §ljmpaul)@kumoh.ac.kr

2021 International Conference on Information and Communication Technology Convergence (ICTC) | 978-1-6654-2383-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICTC52510.2021.9620813

***Abstract*—Internet of Things (IoT), currently cloud IoT infrastructure is shifting toward fog computing through the task offloading process. Fog computing provides computing capabilities that bring closer to the IoT devices. By task offloading shifting to the fog, nodes can avoid high network congestion. Fog computing can improve the quality of service, especially for delay-sensitive applications by enabling low latency requirements. In the IoT infrastructure, low-cost devices are used for less expensive infrastructure deployments. Raspberry pi is used for computing infrastructure with cheaper development. A small single-board computer not only has high computing capacity but also allows high portability.**

***Index Terms*—fog computing, task offloading, low-cost, raspberry pi, smart home**

I. Introduction

Nowadays, the Internet of things (IoT) has become the spotlight for applying Industry 4.0. The benefits of this technology have overcome several problems in society. IoT is the interconnection through the internet network between smart devices and cloud servers consisting of sensor-driven, computing processes and data analytics applications and work together simultaneously. Smart factories, smart cities, smart cars, healthcare, etc., are ready for IoT applications.

In addition, in IoT infrastructure cloud computing is used for storage and data analysis from sensor devices that haven't storage and processing capabilities to decide. The transfer task from sensor devices that generate large data to the cloud is called task offloading [1]. Currently, the task offloading process is shifting toward fog computing instead of cloud computing. Fog computing provides computing capabilities that bring closer to the IoT devices. With task offloading shifting to the fog, nodes can avoid high network congestion. Fog computing can improve the quality of service, especially for delay-sensitive applications that allow low latency requirements.

Developing fog/edge layers requires high cost and extra maintenance such as server equipment, cold server rooms, etc. In IoT infrastructure, low-cost devices are used for less expensive infrastructure deployments. There are many types of these devices, such as Raspberry pi [2]. Low-cost single board computers such as TelosB motes [3] and raspberry pi are used for computing infrastructure with less expensive development. Small single board computer not only have high computing capacity but also allows a high portability. In [4] implement an edge computing for face recognition using NeuroEdge and smart device NM500 chip for Edge AI. The chip performance was evaluated with low power consumption and excellent accuracy. The contributions of this paper are summarized as follows:

- Explore different approaches for deploying a fog computing using a low-cost small single board computer with high computing capacity and high portability.
- Perform a low-cost fog computing infrastructure with a selector gateway as balancing to get low process delay for small scale smart home applications.
- Design a multi hop flow communication in task offloading using the MQTT protocol with a local MQTT broker to achieve low latency in delay-sensitive applications.

The outline of the paper is as follows. Section II describes previous related works on fog computing technology and fog computing design and implementation. Detail of the proposed system are present in Section III. Section IV presents performance and evaluation of the proposed system. Finally, Section V concludes this paper and provides information on future works.

II. Related Works

We found relevant references implementing a fog computing in IoT. In [1] a IoT task offloading to the dog was investigated. Reducing response time achieved by applying the load balancing the task on the fog nodes and conducted a scheduling algorithm using ant colony optimization (ACO). ACO task offloading can improve response time and effectively balance the task in the fog nodes. In [5] tasks offloading has been deployed using reinforcement learning in industrial IoT (IIoT) environment. Using this approach can minimize offloading computation costs and delay costs in IIoT scenarios.

Regarding fog computing infrastructure through low-cost devices, we found interesting implementation. In [2] a fog portable fog computing infrastructure is deployed through a raspberry pi cluster. This system offers low-cost development complemented by containerized application. FogPi is a low-power fog computing infrastructure that apply a Docker container with high scalability and allows vertical and horizontal migrations. The author mention that if the system has a higher number of replicas, FogPi can achieve a higher quality of services (QoS).

In [6], authors presented another side of fog computing that still challenges the heterogeneous fog systems. This system applies a distributed fog resource allocation algorithm, called TPRA (Task Priority based Resource Allocation) to address the long task of offloading queues. Each service provisioning IoT service is modeled as a workflow. The proposed system can address the conflicts of requesting resource and can reduce delays and balance the workloads in fog architecture.

Another approach for fog computing in IoT is shown in [7]. The author proposes a fog computing architecture based on Linux containers and orchestration platform to make a cluster of these nodes for soundscape monitoring case. In addition, the system applies a cooperate and schedule different task to achieve an efficient away. The overall computational power constraint can overcome by applying a cluster of interconnected nodes with equipped a Linux docker containers to add flexibility, adaptability, and responsiveness features.

III. PROPOSED SYSTEM MODEL

In this paper a fog computing infrastructure is deployed for smart home application scenario. This smart home application has three features, such as room comfort based on temperature and humidity data, room dimmer adjustment and face detection for home security. To run these features machine learning is applied. Room comfort feature is classified into 3 categories: quite comfortable, comfortable, and very comfortable. The room dimmer is adjusted based on the received light intensity data. Face detection feature is equipped with a mobile application that can be accessed by family members. Face detection results will be sent to the application. If the face detection indicates that they are not family member, then the family members who registered to the system will receive a notification from the system.

The system contains a sensor network layer, a fog computing layer, and a cloud layer. Between the sensor network layer and the fog computing layer there is a gateway as task offloading selector. The fog computing infrastructure of this system shown in Fig. 1. In the block diagram the fog nodes located between network sensor nodes and cloud computing. The fog nodes can be said as smart gateway which on duty to forward data from sensor node to the server.

A. Sensor Network Layer

There are three sensor nodes with different tasks. Temperature and humidity sensor nodes are used to measure the temperature and humidity of the room. The DHT11 sensor and MCU node be applied for this function. Temperature and humidity data send to the selector gateway using the MQTT protocol with runs the publisher MQTT. For face detection, a Logitech C310 HD Webcam is used with raspberry pi. The raspberry run a face detection program. If there is the face detect with this program, the camera will capture the face and sent the raw data as a task offloading message to the selector gateway. Another sensor node is LDR and MCU which be equipped Wi-Fi communication module and LED strips actuator.

B. Low-Cost Fog Computing Layer

Fog computing technology was first introduced by Cisco as virtualized technology environment with shifting the computing process closer to the sensor network devices [8]. The IoT-Fog-Cloud system consist of three layers: sensor network / sensor device layer, fog layer and cloud layer as illustrated in Fig. 1. The first layer includes all off sensor devices that generate the sensor data. The fog nodes (e.g., router, switch, and gateway) and servers are distributed in the fog layer that responsible for task offloading from sensor devices [9]. Fog nodes can't standalone unaccompanied cloud computing. The fog computing deployed specially to address the delay-sensitive applications, whereas cloud services can generate a higher latency caused the cloud far away from sensor devices.

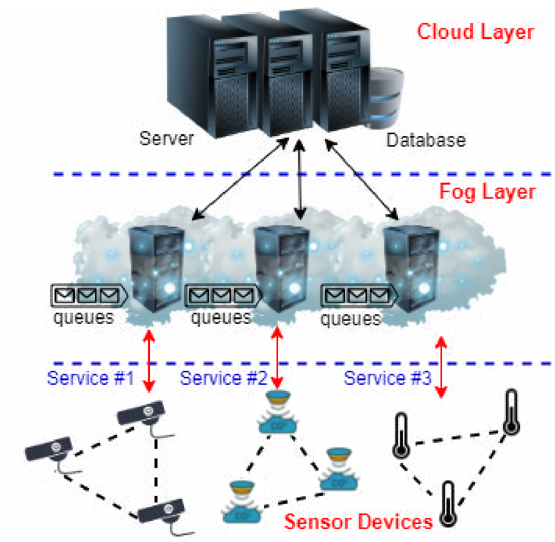

Fig. 1. Three-tier Architecture of IoT-Fog-Cloud System.

In this paper a fog computing infrastructure deployed by applying 3 Raspberry Pi 4 as fog servers and 1 Raspberry Pi 4 as gateway selector. The low-cost fog computing layer shown in Fig. 2. The gateway selector will select the fog server to process the offloading task from sensor devices.

In this smart home scenario there are three tasks, such as room comfortable decision task, room dimmer control task and face detection task. Sensor devices sent the sensor data using pub/sub MQTT protocol. The MQTT protocol is TCP/IP protocol for machine-to-machine communication without special address and make a communication topic. The low-cost fog layer architecture applies a multi-hop communication for task offloading process. Based on the Fig. 3 the publish service runs in the end devices which will send the sensor data to the server or another node. The nodes run a subscribe service to receive the sensor data. The gateway selector receives all off the sensor data from sensor nodes and runs queening process with FIFO algorithm. The first received data will forward first to the fog server.

Based on the Fig. 3 at the gateway selector the first data sensor will be forwarded to the fog server selected. The

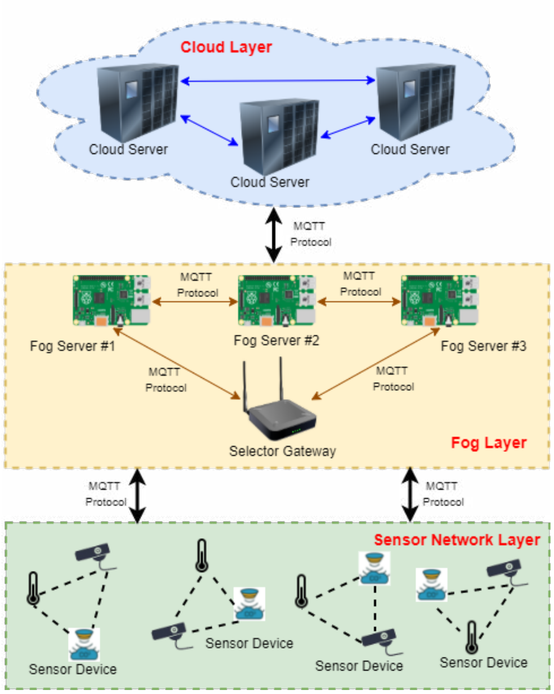

Fig. 2. Low-Cost Fog Layer Architecture.

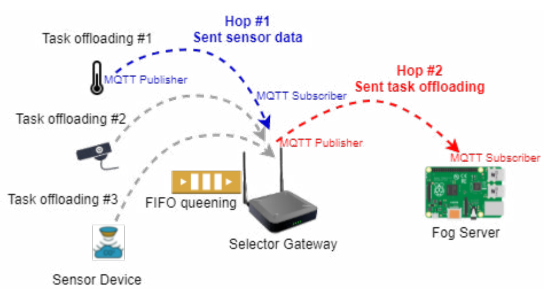

Fig. 3. Multi-hop Communication for Task Offloading Process.

selection of the for server for process the offloading task based the status (idle and occupied) of the fog server. In the gateway selector there is the fog server status list. The fog server can process all type of offloading task. Each fog server is embedded all the task offloading program. To simplify execution the task offloading message and task offloading ID send together. For running the task can call the name of the function that same with task offloading ID.

C. Cloud Computing Layer

The cloud is used to storage the sensor data as database server and execute the offloading task if all the fog server is running. The data value can be accessed by mobile application. The room comfort classification, room dimmer adjustment, face detection and family member classification result shown in mobile application. The comfort classification feature use K-means clustering to classify became 3 cluster (quite comfortable, comfortable, and very comfortable) based on near value.

IV. Performance Evaluation

In this paper, a delay process performance in low-cost fog computing has been evaluated. In this scenario all the sensor nodes are running and sent the offloading task to the selector gateway and forwarded to the selected fog server. The sensor data sent to the fog server use MQTT protocol through a local MQTT broker in Local Area Network (LAN). Data be saved in cloud database server via public internet network. The delay process measure result shown at Table I.

TABLE I
Delay Process Test Results for Low-Cost Fog Computing Architecture

Sample	Room comfort classifications (s)	Room dimmer adjustment (s)	Face detection (s)
1	0.152	0.302	6.512
2	0.143	0.313	6.562
3	0.145	0.315	6.653
4	0.170	0.323	6,617
5	0.140	0.328	6.646
6	0.160	0.374	6.583
7	0.147	0.341	6.588
8	0.148	0.379	6.544
9	0.163	0.354	6.665
10	0.152	0.361	6.648
Average	**0.152**	**0.339**	**6.602**

Based on the Table I, the process delay of room comfortable classification task is 0,152 seconds, the room light dimmer adjustment is 0,339 seconds, and face detection application with a process delay is 6,602 seconds. All the data were measured 10 times and got the average value. Data from the sensor devices can be sent to the fog server through selector gateway. The fog server manages to execute all the task offloading properly and the decision data can be accessed by the mobile applications.

V. Conclusion and Future Work

In this paper, a low-cost fog computing infrastructure has been deployed using raspberry pi 4 as fog server. Furthermore, in this system a selector gateway is applied to achieve a balance of task offloading. Based on the test results the process delay of room comfortable classification task is 0,152 seconds, the room light dimmer adjustment is 0,339 seconds, and face detection application with a process delay is 6,602 seconds. Applying a local MQTT broker can reduce the process delays. Thus, for the next works, enhancements of selector gateway by applying a machine learning (ML) algorithm to address the long queening in the selector gateway which imply from the high rate of task offloading request from sensor nodes. Another ML algorithm are certainly needed and can be compared with each other to get higher quality of service performance and achieve low latency in delay-sensitive applications.

Acknowledgment

This research work was supported by Priority Research Centers Program through NRF funded by MEST

(2018R1A6A1A03024003) and the Grand Information Technology Research Center support program (IITP-2021-2020-0-01612) supervised by the IITP by MSIT, Korea.

REFERENCES

[1] M. K. Hussein and M. H. Mousa, "Efficient Task Offloading for IoT-Based Applications in Fog Computing Using Ant Colony Optimization," *IEEE Access*, vol. 8, pp. 37 191–37 201, 2020.

[2] C. Martín, D. R. Torres, M. Díaz, and B. Rubio, "FogPi: A Portable Fog Infrastructure through Raspberry Pis," in *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, 2020, pp. 1–3.

[3] M. Lavassani, S. Forsström, U. Jennehag, and T. Zhang, "Combining Fog Computing with Sensor Mote Machine Learning for Industrial IoT," *Sensors*, vol. 18, no. 5, p. 1532, 2018.

[4] C. I. Nwakanma, J.-W. Kim, J.-M. Lee, and D.-S. Kim, "Edge AI prospect using the NeuroEdge Computing System: Introducing a Novel Neuromorphic Technology," *ICT Express*, vol. 7, no. 2, pp. 152–157, 2021.

[5] M. S. Hossain, C. I. Nwakanma, , J.-M. Lee, and D.-S. Kim, "Edge Computational Task Offloading Scheme using Reinforcement Learning for IIoT Scenario," *ICT Express*, vol. 6, no. 4, pp. 291–299, 2020.

[6] H. Tran-Dang and D.-S. Kim, "Task Priority-based Resource Allocation Algorithm for Task Offloading in Fog-enabled IoT Systems," in *2021 International Conference on Information Networking (ICOIN)*, 2021, pp. 674–679.

[7] R. Fayos-Jordan, S. Felici-Castell, J. Segura-Garcia, A. Pastor-Aparicio, and J. Lopez-Ballester, "Elastic Computing in the Fog on Internet of Things to Improve the Performance of Low-Cost Nodes," *Electronics*, vol. 8, no. 12, p. 1489, 2019.

[8] T. Qayyum, A. W. Malik, M. A. Khan Khattak, O. Khalid, and S. U. Khan, "FogNetSim++: A Toolkit for Modeling and Simulation of Distributed Fog Environment," *IEEE Access*, vol. 6, pp. 63 570–63 583, 2018.

[9] H. Tran-Dang and D.-S. Kim, "FRATO: Fog Resource Based Adaptive Task Offloading for Delay-Minimizing IoT Service Provisioning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 10, pp. 2491–2508, 2021.