Design a D-Fog Scheme to Enhance Processing Performance of Real-Time IoT Applications in Fog Computing

Shin-Jer Yang[1], Wen-Hwa Liao[2], Wan-Lin Lu[1]

[1] Dept. of Computer Science and Information Management, Soochow University, Taipei, Taiwan
[2] Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, Taiwan
sjyang@csim.scu.edu.tw, whliao@ntub.edu.tw, gra231403@gmail.com

Abstract

The cloud computing has driven the fast development of the IoT services and applications. When a amount of real-time events in smart IoT applications are delivered to the cloud for processing, the loading on cloud computing becomes heavier, which can result in processing delays and exceeding their deadlines. The nodes in fog computing are closer to edge devices, which can process such these real-time events in this layer. With this fog process, the event processing efficiency can be enhanced and delays can also be solved in cloud computing.

This paper designs a set of fog computing framework called distributed fog computing scheme (D-FOG) that can improve the better processing performance of real-time IoT applications than conventional cloud computing. Through the simulation experiments, the final results indicate that the D-FOG yields the better performance in terms of Expired event rate, Event success rate, Average delay time, and Average processing time. Consequently, the proposed D-FOG is more effective and efficient in real-time events processing than conventional cloud computing.

Keywords: Cloud computing, Fog computing, D-FOG, Real-time IoT applications

1 Introduction

The maturation of cloud computing technology is driving rapid development of Internet of Things (IoT) services and applications. Cloud computing uses a concentrated processing architecture and virtual technology to produce a cloud environment that can provide large quantities of calculation [6]. This enables users to process a large amount of IoT events without the need to install high-performance computing equipment on the user-end. However, there are also disadvantages. One disadvantage in cloud computing is that to process each IoT event, the event must be uploaded to the cloud before it can be processed. Many factors in this process can increase the uploading time, which can cause the entire processing time to become longer [13]. IoT has many real-time applications, including product production monitoring in manufacturing and road monitoring in traffic and transportation, etc. [1, 11]. These services and applications have many events that require real-time processing. When a large number of events are uploaded to the cloud for processing at the same time, it can result in delayed processing and exceeding the processing deadline.

To solve delay problems derived from a large number of real-time events of smart IoT applications to be processed in cloud computing, the CISCO proposed an early fog computing concept in 2014 [7]. Fog computing technology uses a distributed processing framework to disperse calculation, transmission, control, and storage services on the user's equipment and system or on nearby systems. Examples are mobile phones and gateways, etc. Fog computing can be used to process urgent real-time events on edge devices near the user. This not only can reduce delays, but can also shorten process time and ensure that the calculation is completed within the deadline, which makes real-time event processing more efficient and lightens the load on cloud computing processing.

Although the fog computing concept was proposed in 2014, the OpenFog Consortium that specializes in the study of fog computing was not established until the end of 2015. The Consortium officially published its white paper on fog computing architecture in 2017 and fog computing gradually began to attract people's attention [3]. Currently, most studies are still focused on the precise definition of the fog computing concept. However, the purpose of this paper is to propose a fog computing scheme that can increase the efficiency of a real-time IoT application and solve processing delays. This scheme can also effectively shorten processing time to ensure that real-time events of smart IoT applications can be processed within the deadline. Thus, the main purposes of this paper are as follows:

- To solve processing delays and deadline problems that result from a large number of real-time IoT application events in cloud processing, we propose a

---

[*] Corresponding Author: Shin-Jer Yang; E-mail: sjyang@csim.scu.edu.tw

DOI: 10.53106/160792642021112206011

distributed fog framework into the conventional cloud computing processing architecture called the distributed fog computing scheme (D-FOG), to effectively improve processing efficiency for real-time events in existing IoT applications,

- In this paper, we will propose four key performance indicators (KPIs) such like Expired event rate, Event success rate, Average delay time, and Average processing time. Also, we will set up experimental environments for simulations.
- The simulations conducted in this paper used the proposed D-FOG scheme to simulate a real-time IoT application. The D-FOG can be used to effectively improve the processing efficiency of real-time IoT applications.

The rest of this paper is organized as follows. Section 1 describes the background of the D-FOG and the research scope and purpose. Section 2 surveys literature reviews about cloud computing, IoT and fog computing. In Section 3, this paper examines the operational flows of the D-FOG research processes and design of the D-FOG algorithm. Section 4 covers the simulated experiments setup and analyzes the results. Finally, we draw a conclusion and illustrate and analyze their simulation results, also we indicate the further research direction in Section 5.

2 Related Work

2.1 Cloud Computing

The Cloud computing is mainly to increase the capacity and capability of Information Technology (IT) networks by centralizing how to store and process data. It allows consumers to access information through the Internet without installing them in advance. In addition, cloud computing can also reduce the costs of building IT infrastructure and acquiring new resources. Cloud computing also achieves the benefits of multitenant architecture by maintaining one application. New services offerings can be created by integrating existing services of cloud computing and focusing on added value. Since it is possible to combine components of computing stack on demand, it is easier to turn ideas into real products with limited cost and focus on the product design [6, 22].

The American National Institute of Standards and Technology (NIST) defines cloud computing as “a type of computing resource access and sharing mode that can be adjusted at any time according to the user’s needs.” This type of computing can use minimal management work or service supplier interaction to achieve rapid configuration and release [15]. Cloud computing defined by NIST is composed of five essential characteristics, three service models, and four deployment types, as shown in Figure 1. The descriptions of cloud computing are as follows.

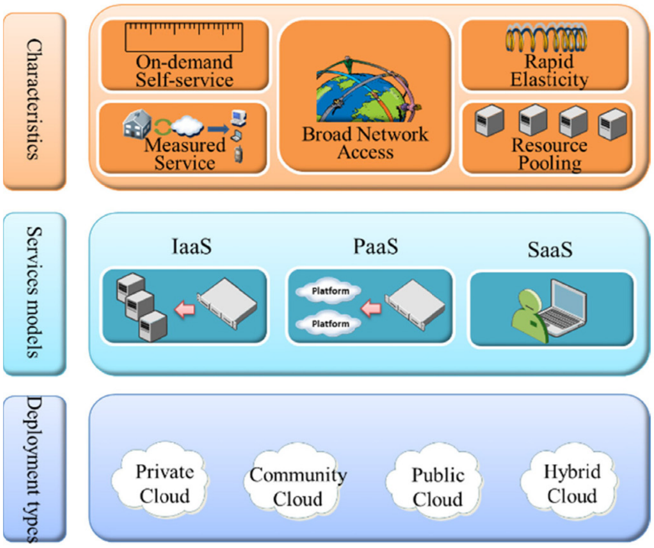

Figure 1. Overview of the cloud computing

The following are the five main essential characteristics:

(1) *On-demand Self-service:* The user can use the cloud service according to their own needs, and does not require using the cloud supplier to make settings. Users can use the webpage to adjust their own settings.

(2) *Broad Network Access:* Use the Internet connection to make service available everywhere so that the user can use the service just by connecting to the Internet.

(3) *Resource Pooling:* Cloud service providers can use a multi-rental model for users and allocate resources according to the user’s requirements.

(4) *Rapid Elasticity:* Users can rapidly adjust the resource scale according to needs.

(5) *Measured Service:* Resources on the cloud must be able to be measured. This means that the cloud provider can implement resource planning, control access, and tabulate fees.

Then, there are three main types of services models:

(1) *Software as a Service* (*SaaS*)*:* Users can directly implement programs provided on the cloud architecture and do not need to install any software system or worry about updating problems. Users can just pay according to their own use status.

(2) *Platform as a Service* (*PaaS*)*:* The cloud provider provides a development platform for users so users can use this platform’s development tools to install their own programs on the cloud for development and testing.

(3) *Infrastructure as a Service* (*IaaS*)*:* The cloud provider provides infrastructure facilities such as storage equipment and network. Users can flexibly control the required development environment in this environment. Users can freely choose the operating system or programs, and do not need to understand base level cloud architecture. They just directly use the services provided by the cloud provider.

In addition, there are four main deployment types:

(1) *Private Cloud:* Used by a single organization. This prevents security problems caused by different users. Generally, this cloud is managed by the organization itself or managed by a third-party.

(2) *Community Cloud:* A community cloud is mutually shared by multiple organizations. Generally, these organizations have similar objectives. Members in the organization can use the community cloud service to share important topics and use this to complete specific tasks or cooperate with policies.

(3) *Public Cloud:* Compared to a private cloud, a public cloud is provided for use by the public. It can be managed and operated by businesses or academic or government organizations.

(4) *Hybrid Cloud:* A hybrid cloud refers to a cloud formed by two or more of the aforementioned cloud types. Although different types of clouds are independent and separate, the two can be connected through a professional technology to transfer data and programs.

The development and application of cloud computing has not only improved efficiency, but also significantly reduced costs (building computer rooms, renting bandwidth, and management of information personnel). Also, it improves enterprises' mastery of production capacity, and allows enterprises to more focus on developing services. Therefore, cloud computing is widely used in fields such as AI, Internet of Things, machine learning, and big data [18].

2.2 Internet of Things (IoT) and Related Applications

A growing number of physical objects are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet. That realizes the idea of the Internet of Things (IoT). In modern society, the Internet has become one of the important communication channels between people. The rising of the IoT technology not only connects people together, but also communicates with other objects. In other words, the Internet is no longer just a communication channel between people, but also a bridge connecting global things and things, people and things. Nowadays, the use of IoT has grown rapidly. There are a lot of IoT applications being developed and deployed in various industries including environmental monitoring, healthcare service, inventory and production management, food supply chain, transportation, workplace and home support, security, and surveillance [4-5, 20].

The current IoT architecture is usually divided into three layers, the perception layer, the network layer, and the application layer, as shown in Figure 2 [12]. The following is a brief description of three conceptual layers.

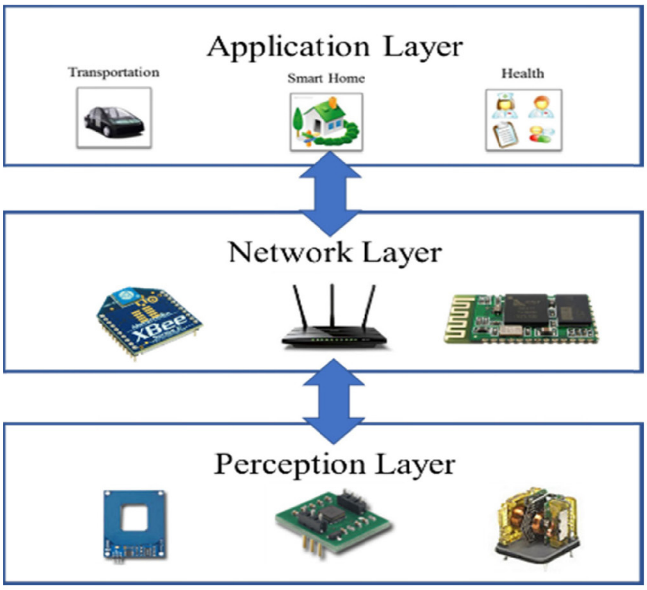

Figure 2. The conceptual architecture of IoT

(1) *Perception Layer:* It is also known as the sensor or device layer, the perception layer is implemented as the bottom layer in the IoT architecture. The perception layer uses smart devices (RFID, sensors, actuators, etc.) to interact with physical devices. The information measured, collected, and processed by objects is transmitted into the upper layer of the IoT architecture via smart devices.

(2) *Network Layer:* It is also known as the transmission layer, this layer is implemented as the middle layer in IoT architecture. The network layer is used to receive and process the information provided by the perception layer, and determine the routes to transmit the information to the hub, devices, and applications of the IoT via integrated networks. The network layer is the most important layer in the IoT architecture, because it combines various devices (switching, router, gateway, cloud computing, etc.) and various communication technologies (Bluetooth, Wi-Fi, Long-Term Evolution, etc.) is integrated in this layer. The network layer uses various communication technologies and protocols to transmit data between different things or applications among heterogeneous network.

(3) *Application Layer:* It is also known as the business layer and implemented as the top layer in IoT architecture. The application layer receives the data transmitted from the network layer and uses the data to provide the required services. For example, the application layer can provide storage services to back up the received data to the database, or provide analysis and evaluation services to predict the future state of physical devices. There are many applications in this layer, for example, smart grid, smart transportation, smart cities, etc., each application has different requirements.

Currently, The IoT has different applications and services in different fields. The following describes a few commonly seen IoT applications, including environmental monitoring; smart cities; smart business/inventory and product management; smart homes/building management; smart health-care; and smart security/ surveillance [1, 4, 11].

(1) *Transportation and Logistics Domain*: The advanced cars, trains, buses and bicycles are equipped with sensors, actuators and processing power. After the roads themselves and the transported goods are also equipped with tags and sensors, they can transmit important information to traffic control stations and transport vehicles, better arrange traffic routes, help warehouse management, provide the tourists with appropriate traffic information and monitor the status of transported goods.

(2) *Healthcare Domain:* The application of the Internet of Things technology provides many benefits for the healthcare domain, resulting in many applications, including tracking objects and personnel (staff and patients), identification and authentication of people, and automatic collection of sensing data.

(3) *Smart Environment Domain*: Smart environment means that whether it is an office, home, factory or leisure environment, the environment can be made relaxed and comfortable through smart objects.

(4) *Personal and social domain*: Applications in this domain allow users to interact with other people to maintain and establish social relationships. When we are doing something or have done something, it may be automatically triggered to send a message to a friend.

2.3 Fog Computing and Related Applications

The cloud computing can provide on-demand, scalable storage and processing service for IoT scalable requirements. However, for health monitoring, emergency response, and other delay-sensitive applications, the delay caused by transmitting data to the cloud and back to the application is unacceptable. In addition, transferring large amounts of data to the cloud for storage and processing is inefficient, because it takes up the bandwidth of the network and cannot be scalable [13, 19].

In order to solve the above problems, edge computing uses computing resources near the IoT sensor as local storage and preliminary data processing. This will reduce network congestion and speed up analysis and final decision-making. However, edge devices cannot handle multiple IoT applications competing for their limited resources, which will cause resource competition and increase processing delays [3, 14]. The fog computing seamlessly integrates edge devices and cloud resources to overcome these limitations. It avoids contention for edge resources by leveraging cloud resources and coordinating the use of distributed edge devices. Also, the fog computing is a distributed mechanism that uses cloud and edge resources and its own infrastructure to provide cloud-like services to the edge of the network, as shown in Figure 3 [8].

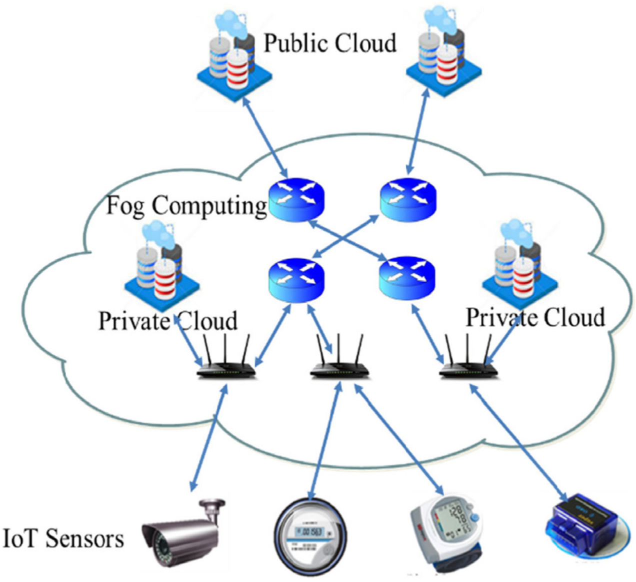

Figure 3. The fog computing framework

The fog computing uses edge devices near users to process IoT data locally for massive storage, communication, control, configuration, and management. The advantage of this method is the use of edge devices close to the sensor while taking advantage of the on-demand scalability of cloud resources. The fog computing includes the data processing or analysis applications in distributed clouds and edge devices. It can also assist in the management and programming of computing, network and storage services between the cloud data center and end devices. In addition, it supports user mobility, resource and interfaces heterogeneity, and distributed data analysis, and solves the needs of a wide range of distributed applications that need the requirement of low latency[10, 16-17].

Fog computing can be utilized in many real-time IoT applications such as smart grid, smart factory, industrial automation, smart traffic management, smart medical care, and real-time analysis application systems, which are separately introduced below [2, 9, 21]:

(1) *Smart Grid:* This uses smart meters to monitor real-time electricity supply in areas and uses fog computing platform for analysis. If there are any special changes, the grid will respond and use the fastest speed to stabilize the grid.

(2) *Smart Factory and Industrial Automation:* This uses factory environment sensors to collect temperature or gas content values inside factories. The information is transmitted to fog nodes for real-time data analysis. If an abnormal value is discovered, the system can automatically notify factory personnel in real-time for emergency processing. This increases the factory's operational safety. Production line sensors can also be used to detect the line's work status and transmit the data to fog nodes for analysis. This can be used to check in real-time whether the products on the automated production line conform to product

specifications. If an abnormality is discovered, the fog nodes will transmit a correction command to the automated production line and adjust the production line work in real-time. Combining production line monitoring and fog nodes' real-time data analysis can increase the factory's production efficiency.

(3) *Smart Traffic Management:* This uses roadside deployment of fog nodes to collect vehicle traffic information and conduct real-time data analysis to respond to traffic in real-time. This can be used to automatically adjust traffic lights and alleviate traffic congestion. If a car accident happens, the system can alert rescue units in real-time, suggest to other drivers to use alternative routes, and achieve real-time sharing of traffic information.

(4) *Smart Medical Care:* This uses wearable devices to monitor physiological values and upload the monitored data to fog nodes for real-time data analysis. When there is an abnormal physiological value, the system can send a help signal in real-time to the family or directly to a rescue unit for emergency processing to make the best use of the crucial treatment period.

(5) *Real-time Analysis Application:* Weather monitoring is a real-time analysis application. Various types of sensors are used to collect temperature, wind speed, and rainfall quantity data, which are uploaded to the fog nodes for real-time data analysis. This provides the public with accurate weather data and can also send out real-time severe weather warnings so that the public can have more time to make disaster prevention preparations.

We discussed the various features of cloud computing, edge computing, and fog computing in detail. Table 1 gives the overall comparisons of their tasks, response time, computing power, and network bandwidth.

Table 1. Features comparison among cloud computing, edge computing, and fog computing

Features	Cloud	Edge	Fog
Task	Large amount of remote computing	High-performance real-time computing	Near-end computing
Response time	Long	Short	Medium
Computing ability	High	Low	Medium
Network bandwidth	High	Low	Medium

3 Operations Issue and Algorithm Design in D-FOG

The D-FOG scheme proposed in this paper introduces a distributed fog computing framework into the conventional cloud computing processing architecture to effectively process real-time events in smart IoT applications. Each event's priority sequence is used to conduct the process queuing. The operational flow procedures of D-FOG are as described below and shown in Figure 4.

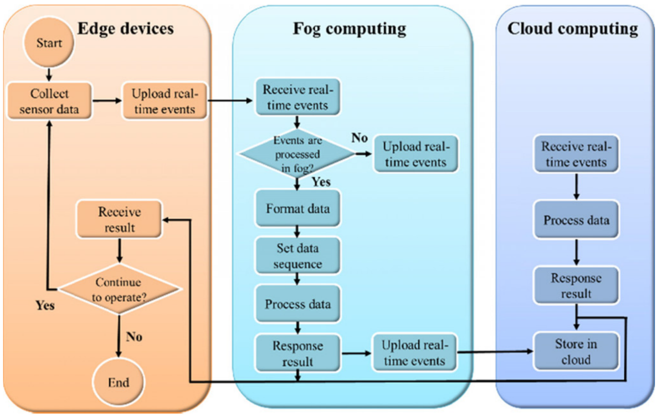

Figure 4. D-FOG operational flow procedures

(1) The fog nodes can receive events produced by edge devices and first format the data.

(2) The D-FOG scheme can determine which events will be processed in the fog computing layer based on priority level, data quantity, and processing time for each event. Each event's priority can be determined by the IoT applications. Hence, events that do not require real-time processing will be uploaded to cloud computing for processing. Higher priority events that require real-time processing are immediately placed into the processing queue.

(3) After the events are processed, the results are immediately sent back to the edge devices. After the event data that is to be stored in formatted type, they will be uploaded to the cloud storage in lots.

(4) After all the processing procedures have been completed, the fog nodes will release memory and wait for new events to be uploaded.

Based on the operational flow of the D-FOG as shown in Figure 4, the pseudo code of the D-FOG algorithm can be designed as follows:

4 Simulations Environments and Results Analysis

This section describes the simulations environment setup including software and hardware settings, list and illustrate KPIs and their calculation formulas. Also, we make the analysis of each KPI and summarize the simulations results.

```
Algorithm D-FOG(){
INPUT:
  RTED = []
  FRTE = []
  Seqlist = []
  count = 0
  dcount = RTE * initial_ratio
  Uploadlist = []
  DPARAMETER = []
  RTE = 0

OUTPUT:
  To complete real-time IoT events and response messages.

Method:
BEGIN{
  // Receive the real-time events
  RTED = sensor.recv (1024)

  // Format the data into a format
  FRTE = DATAFOMAT (RTED)

  SEQ (FRTE) //Set event processing sequence
  IF (count > dcount):
      PROCESSINGHELP()
    PROCESSINGDATA()
  ELSE
    PROCESSINGDATA()
  UPLOADDATA()      // Upload data to cloud storage
  PARAMETERUPDATE() // Update the default parameter
  RELEASEMEM()      // Release memory
} END

PROCEDURE DATAFOMAT(data){
  x = data.split (",")
  RETURN x
} END DATAFOMAT

PROCEDURE SEQ(data){
  y = []
  IF (data[0] == "data type"):
      y.append(sequence)
      FOR index in range(len(data)):
      y.append(data[index])
  Seqlist.append(y)
  count = count + 1
} END SEQ

PROCEDURE PROCESSINGHELP(){
  remsg=""
  fog2.connect((host, port))
  FOR index in range(len(Seqlist)):
      IF (index != 0):
      remsg = remsg + a[index] + ","
  fog2.send(remsg.encode('utf-8'))
  a=fog2.recv(1024)
  RETURN a
} END PROCESSINGHELP

PROCEDURE PROCESSINGDATA(){
  remsg = ""
```

```
  bamsg = []
  PD = Seqlist[0]
  RTEd = DPARAMETER[0]
  TT = PD [3]
  IF (NOW-TT > RTEd):
     RETURN EE
  ELSE:
     IF (PD[1] == " data type "):
         IF (float(PD[3]) > 40):
             PD.append("error")
         ELSE:
             PD.append("cloud")
     FOR index in range(len(a)):
         IF (index != 0):
             remsg=remsg + PD[index] + ","
     bamsg.append(PD[5])
     bamsg.append(remsg)
     PD.append(get_time_stamp())
     RETURN bamsg
} END PROCESSINGDATA

PROCEDURE UPLOADDATA(){
  uploadlist.append (data)
} END UPLOADDATA

PROCEDURE PARAMETERUPDATE (){
  DPARAMETER [0] = deadline
  IF (RTE == 100):
     deadline[0] = 100APT * initial_ratio
  ELSE IF (RTE == 200):
     deadline[1] = 200APT * initial_ratio
  ELSE IF (RTE == 500):
     deadline[2] = 500APT * initial_ratio
  ELSE IF (RTE == 1500):
     deadline[2] = 1500APT * initial_ratio
} END PARAMETERUPDATE

PROCEDURE RELEASEMEM(){
global count
del Seqlist
count = count - 1
Uploadlist = []
FRTE = []
RTED = []
} END RELEASEMEM

END D-FOG
```

4.1 Experiments Configuration Setup

For this experiment, we can utilize the VirtualBox to set up a fog computing simulated environment and also use Web Services to set up a cloud computing virtual environment on the Windows. Also, we take the environmental monitoring of smart factory as a practical case for simulations. The data types of real time events include temperature and humidity. The simulations data is generated by the random function in the sensor emulation of the smart factory system. The simulated environment is divided into three main parts. The first part is the edge devices, which are responsible for collecting data and transmitting the events regularly to the fog computing in the second part. The fog selects the real-time event that it can process and the remaining events are uploaded to the third part: cloud computing, for processing. The simulated environment configuration architecture is shown in Figure 5. The related hardware/software (HW/SW) configuration specifications are as shown in Table 2.

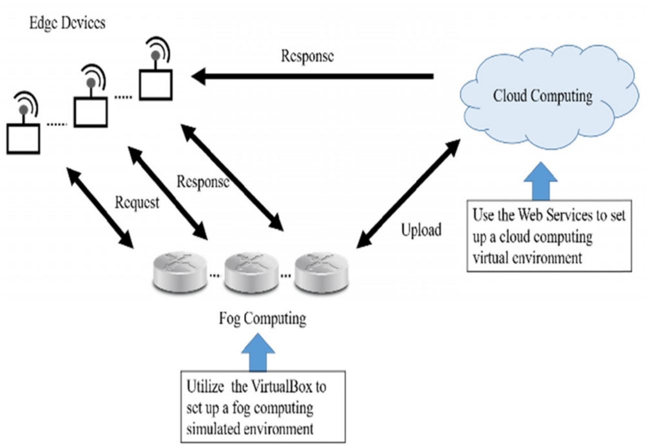

Figure 5. Simulated environment configuration architecture

Table 2. Simulations environment specifications

HW/SW	Physical Machine	VM
O.S.	Windows 10	Ubuntu
CPU	Intel® Core™ i7-4790 CPU @ 3.60GHz	1 Cores with 2.6 GHz
Memory	8 GB	2 GB
Disk	250 GB	50 GB

4.2 Descriptions of KPIs

In this paper, we propose four key performance indicators (KPIs) in terms of Expired event rate, Event success rate, Average delay time, and Average processing time which are used to analyze the simulation results to be compare between the D-FOG and conventional cloud schemes. The descriptions of the KPIs are shown in Table 3.

Table 3. The descriptions of three keys performance indicators

Key Performance Indicators (KPIs)	Purposes
Expired Event Rate (EER, Unit: %)	The EER represents that the percentage of processing time of each real-time event is exceeded the deadline time (EE) for all real-time event (RTE), as shown in Formula (1).
Event Success Rate (ESR, Unit: %)	The ESR is to confirm that the percentage of real-time event is successfully completed within the deadline (CE), as shown in Formula (2).
Average Delay Time (ADT, Unit: *ms*)	This KPI is to measure the average delay time (ADT) of each real-time event, as shown in Formula (3).
Average Processing Time (APT, Unit: *ms*)	This KPI is to measure the average processing time (APT) of each real-time event, as shown in Formula (4).

The correctly identified for Expired event rate, Event success rate, Average delay time, and Average processing time are expressed in Formulas (1), (2), (3), and (4).

$$\text{EER}\ (\%) = (\text{EE} / \text{RTE}) \times 100 \quad \textbf{(1)}$$

$$\text{ESR}\ (\%) = (\text{CE} / \text{RTE}) \times 100 \quad \textbf{(2)}$$

$$\text{ADT}\ (ms) = \text{DT} / \text{RTE} \quad \textbf{(3)}$$

$$\text{APT}\ (ms) = \text{PT} / \text{RTE} \quad \textbf{(4)}$$

4.3 Results Analysis

We did perform five time experiments for averaging the simulation results under D-FOG and conventional cloud processing architectures (i.e. Cloud platform). Also, the simulations were carried out using 100, 200, 500, and 1500 real-time events (Assume: constant packet size) in smart factory system, respectively. The simulation results in terms of KPIs: Expired event rate, Event success rate, Average delay time, and Average processing time are as shown in Figure 6, Figure 7, Figure 8, and Figure 9, respectively. Hence, the total simulation results are summarized as shown in Table 4.

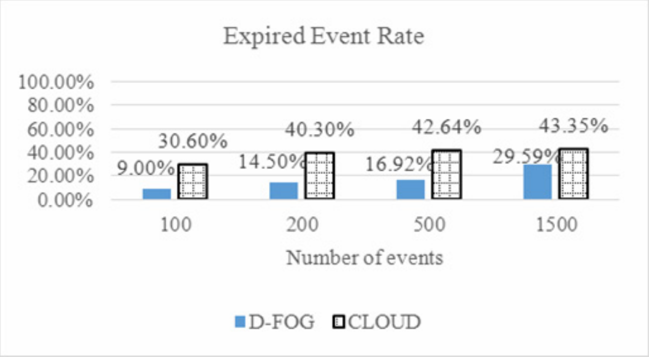

Figure 6. Expired event rate

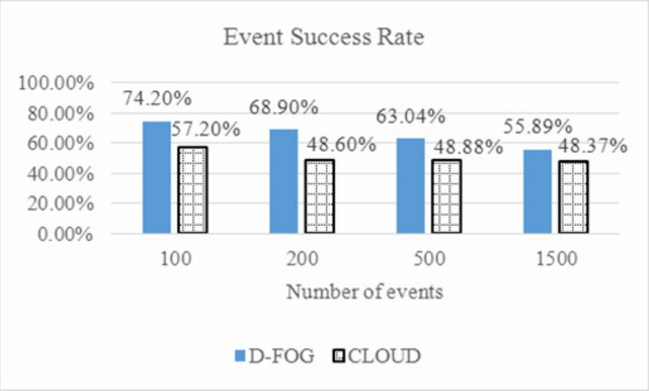

Figure 7. Event success rate

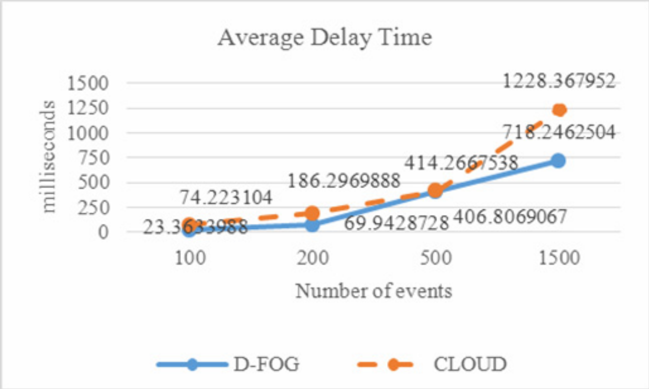

Figure 8. Average delay time

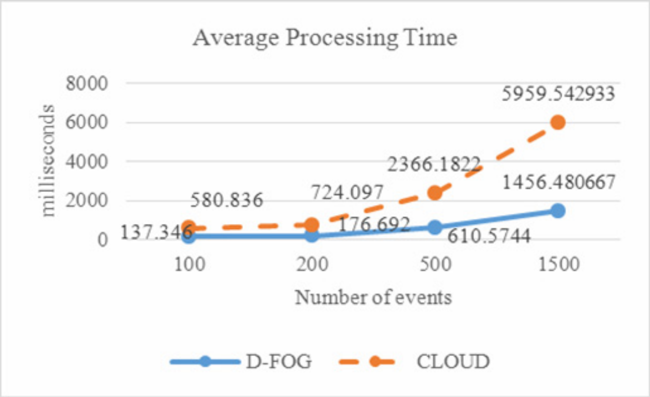

Figure 9. Average processing time

Table 4. The summarized experimental results

		D-FOG		
Events KPIs	100	200	500	1500
Expired Event Rate	9.00%	14.50%	16.92%	29.59%
Event Success Rate	74.20%	68.90%	63.04%	55.89%
Average Delay Time(*ms*)	23.36	69.94	406.81	718.25
Average Processing Time(*ms*)	137.35	176.69	610.57	1456.48
		CLOUD		
Events KPIs	100	200	500	1500
Expired Event Rate	30.60%	40.30%	42.64%	43.35%
Event Success Rate	57.20%	48.60%	48.88%	48.37%
Average Delay Time(*ms*)	74.22	186.30	414.27	1228.37
Average Processing Time(*ms*)	580.84	724.10	2366.18	5959.54

The experimental results indicate that when using the D-FOG scheme **i**n the case of 100 events, the Expired event rate can be reduced by 21.6%, the Event success rate can be increased by 17%, Average delay time can be reduced by 50.86 *ms* to achieve 68.52% improvement ratios, and the Average processing time can be reduced by 443.49 *ms* to achieve 76.35% improvement ratios. In the case of 200 events, the Expired event rate can be reduced by 25.8%, the Event success rate can be increased by 20.3%, Average delay time can be reduced by 116.35 *ms* to achieve 62.46% improvement ratios, and the Average processing time can be reduced by 547.41 *ms* to achieve 75.6% improvement ratios. In the case of 500 events, the Expired event rate can be reduced by 25.72%, the Event success rate can be increased by 14.16%, Average delay time can be reduced by 7.46 *ms* to achieve 1.8% improvement ratios, and the Average processing time can be reduced by 1755.61 *ms* to achieve 74.2% improvement ratios. In the case of 1500 events, the Expired event rate can be reduced by 13.76%, the Event success rate can be increased by 7.52%, Average delay time can be reduced by 510.12 *ms* to achieve 41.53% improvement ratios, and the Average processing time can be reduced by 4503.06 *ms* to achieve 75.56% improvement ratios. The comparative simulations are illustrated in Table 5. Hence, this proves that use of the D-FOG scheme can significantly improve the better events processing efficiency of real-time IoT applications than conventional cloud architecture.

Table 5. Comparative improving ratios for simulations

		Improving Ratios		
Events KPIs	100	200	500	1500
Expired Event Rate	21.60%	25.80%	25.72%	13.76%
Event Success Rate	-17.00%	-20.30%	-14.16%	-7.52%
Average Delay Time	68.52% (-50.86 *ms*)	62.46% (-116.35 *ms*)	1.80% (-7.46 *ms*)	41.53% (-510.12 *ms*)
Average Processing Time	76.35% (-443.49 *ms*)	75.60% (-547.41 *ms*)	74.20% (-1755.61 *ms*)	75.56% (-4503.06 *ms*)

5 Conclusion

As the quantity of real-time IoT applications increase, the number of real-time events that require real-time processing will increase. When a large number of these events are simultaneously uploaded to cloud computing for processing, it can cause processing delays and make the processing time pass the processing deadline. To solve this problem, some people have proposed the fog computing concept. Current studies on fog computing are focused on defining its concept and application. In this paper, we propose a novel scheme that actually utilizes the fog computing framework, which we called the D-FOG.

The proposed D-FOG utilizes the fog computing as processing scheme framework in this paper. Also, we set up the experimental environments for simulations using four KPIs in terms of: Expired event rate, event success rate, Average delay time, and average processing time. The experimental results of KPIs indicate that in Expired event rate can be lower by 21.6%, 25.8%, 25.72% and 13.76%; Event success rate can be increased of 17%, 20.3%, 14.16% and 7.52%; Average delay time can be reduced by 50.86 *m*s, 116.35 *m*s, 7.46 *m*s and 510.12 *m*s, and also Average processing time can be reduced by 443.49 *m*s, 547.41 *m*s, 1755.61 *m*s and 4503.06 *m*s in 100, 200, 500 and 1500 real-time events of smart factory system, respectively. In summary, the results of this D-FOG scheme are compared to that of conventional cloud computing processing to prove that the proposed D-FOG scheme is more efficient and has better results than a conventional cloud computing processing architecture. Also, the D-FOG can provide a more efficient distributed processing architecture for all real-time IoT applications. Consequently, the D-FOG scheme not only can solve processing delays in real-time event processing, but can also provide superior service quality in real-time IoT applications to achieve better processing efficiency and effective outcomes.

In the future research, we will expand the execution of simulation experiments, incorporate more fog computing nodes and process larger amounts of data to obtain more accurate data results. In the information security, we will enhance the secure communication with neighboring nodes in the D-FOG scheme, establish a trust mechanism for communication between fog nodes, and make the transmission, processing, and storage of data more secure.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 4, 2347-2376, Fourth Quarter, 2015.

[2] H. Atlam, R. Walters, G. Wills, Fog Computing and the Internet of Things: A Review, *Big Data and Cognitive Computing*, Vol. 2, No. 2, pp. 1-18, June, 2018.

[3] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog Computing and its Role in the Internet of Things, *Proceedings of the 1st Edition of the MCC workshop on Mobile Cloud Computing*, Helsinki, Finland, 2012, pp. 13-16.

[4] E. Borgia, The Internet of Things Vision: Key Features, Applications and Open Issues, *Computer Communications*, Vol. 54, pp. 1-31, December, 2014.

[5] A. Botta, W. de Donato, V. Persico, A. Pescapé, Integration of Cloud Computing and Internet of Things: A Survey, *Future Generation Computer Systems*, Vol. 56, pp. 684-700, March, 2016.

[6] R. Buyya, J. Broberg, A. M. Goscinski, *Cloud Computing: Principles and Paradigms*, John Wiley & Sons, 2011.

[7] CISCO, Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are, 2015. Retrieved from https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf.

[8] A. V. Dastjerdi, R. Buyya, Fog Computing: Helping the Internet of Things Realize Its Potential, *IEEE Computer*, Vol. 49, No. 8, pp. 112-116, August, 2016.

[9] M. S. de Brito, S. Hoque, R. Steinke, A. Willner, T. Magedanz, Application of the Fog Computing Paradigm to Smart Factories and Cyber-Physical Systems, *Transactions on Emerging Telecommunications Technologies*, Vol. 29, No. 4, Article No. e3184, pp. 1-14, April, 2018.

[10] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on Fog Computing: Architecture, Key Technologies, Applications and Open Issues, *Journal of Network and Computer Applications*, Vol. 98, pp. 27-42, November, 2017.

[11] H. D. Kotha, V. M. Gupta, IoT Application, A Survey, *International Journal of Engineering & Technology*, Vol. 7, No. 2.7, pp. 891-896, March, 2018.

[12] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications, *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp. 1125-1142, October, 2017.

[13] W. Lumpkins, The Internet of Things Meets Cloud Computing, *IEEE Consumer Electronics Magazine*, Vol. 2, No. 2, pp. 47-51, April, 2013.

[14] I. Martinez, A. Jarray, A. S. Hafid, Scalable Design and Dimensioning of Fog-Computing Infrastructure to Support Latency Sensitive IoT Applications, *IEEE Internet of Things Journal*, Vol. 7, No. 6, pp. 5504-5520, June, 2020.

[15] P. Mell, T. Grance, *The NIST Definition of Cloud Computing*, SP 800-145, September, 2011. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[16] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, P. A. Polakos, A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges, *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 1, pp. 416-464, First Quarter, 2018.

[17] M. Mukherjee, L. Shu, D. Wang, Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges, *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 3, pp. 1826-1857, Third Quarter, 2018.

[18] T. H. Noor, S. Zeadally, A. Alfazi, Q. Z. Sheng, Mobile Cloud Computing: Challenges and Future Research Directions, *Journal of Network and Computer Applications*, Vol. 115, pp. 70-85, August, 2018.

[19] P. Singh, A. Nayyar, A. Kaur, U. Ghosh, Blockchain and Fog Based Architecture for Internet of Everything in Smart Cities, *Future Internet*, Vol. 12, No. 4, Article No. 61, April, 2020.

[20] L. D. Xu, W. He, S. Li, Internet of Things in Industries: A Survey, *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 4, pp. 2233-2243, November, 2014.

[21] S. Yi, Z. Hao, Z. Qin, Q. Li, Fog Computing: Platform and Applications, *Proceedings of 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, Washington, DC, USA, 2015, pp. 73-78.

[22] M. Yigit, V. C. Gungor, S. Baktir, Cloud Computing for Smart Grid Applications, *Computer Networks*, Vol. 70, pp. 312-329, September, 2014.

Biographies

Shin-Jer Yang is currently a Distinguished Professor in the Department of Computer Science and Information Management, Soochow University, Taipei, Taiwan. Professor Yang is the author/coauthor of more than 129 refereed technical papers (Journals and Conferences) on Wired / Wireless Networking and Applications, Cloud/Internet Computing Applications and Services, and Network Management and Security. Also, he takes in charge of more than 30 research projects. His research interests include Wired/Wireless Networking Technologies and Applications, Cloud Computing and Applications, AIoT Applications, Network Management and Security, and Information Management.

Wen-Hwa Liao received the Ph.D. degree in computer science and information engineering from National Central University, Taiwan, in 2002. He is currently a Full Professor with the Institute of Information and Decision Sciences, National Taipei University of Business, Taiwan. His current research interests include Internet of Things, Cloud Computing, Wireless Sensor Networks, and Artificial Intelligence. He has served as the Associate Editor of the International Journal of Distributed Sensor Networks (IJDSN) and International Journal of Vehicle Information and Communication Systems (IJVICS). He also has served as an Associate Guest Editor for SCI-indexed journal International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC).

Wan-Lin Lu Currently, she is a Research Assistant in the Department of Computer Science and Information Management, Soochow University, Taipei, Taiwan. Her research interests include Cloud Computing and Service, IoT and Fog Applications, and Web Applications Design.