Internet of Things 1–2 (2018) 14–26

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

An energy-efficient model for fog computing in the Internet of Things (IoT)

Ryuji Oma [a,*], Shigenari Nakamura [a], Dilawaer Duolikun [a], Tomoya Enokido [b], Makoto Takizawa [c]

[a] *Graduate School of Science and Engineering, Hosei University, Tokyo, Japan*
[b] *Faculty of Business Administration, Rissho University, Tokyo, Japan*
[c] *Department of Advanced Sciences, Faculty of Science and Engineering, Hosei University, Tokyo, Japan*

ARTICLE INFO

Article history:
Received 26 July 2018
Accepted 6 August 2018
Available online 13 August 2018

Keywords:
Energy-efficient fog computing
IoT(Internet of Things)
Energy-efficient IoT
Tree-based fog computing model

ABSTRACT

A huge number of devices like sensors in addition to computers are interconnected in the IoT (Internet of Things). In the cloud computing model, sensor data is transmitted to servers in networks and processed on the servers in a cloud. Here, networks are congested and servers are overloaded due to heavy traffic from sensors. In order to reduce the delay time and network traffic and increase the performance of the system, data and processes are distributed to not only servers in a cloud but also fog nodes in fog computing models. While the traffic of servers in a cloud can be reduced, the total electric energy consumed by fog nodes increases to process sensor data. In this paper, we newly propose a tree-based fog computing (TBFC) model to distribute processes and data to servers and fog nodes so that the total electric energy consumption of nodes can be reduced in the IoT. In the evaluation, we show the total electric energy consumption of nodes in the TBFC model is smaller than the cloud computing model.

1. Introduction

A huge number and various types of nodes including not only computers like servers and clients but also devices like sensors and actuators are interconnected in the Internet of Things (IoT) [1]. Sensor data is transmitted to servers in cloud [2] and processed to make a decision on actions to be done by actuators. Here, networks are congested and servers are overloaded due to heavy traffic from sensors. In order to realize the IoT, an intermediate layer named *fog* layer [1] is introduced between devices and clouds. The fog layer [1] is composed of fog nodes which are interconnected with other fog nodes, devices, and servers in networks. A fog node not only exchanges data with sensors and other fog nodes, i.e. does the routing functions but also processes the input data received from sensors and other fog nodes and sends the processed output data to other fog nodes. Fog nodes finally deliver processed data to servers in clouds. In addition to servers, a fog node makes a decision on actions to be done by actuators and sends the actions to actuators via other fog nodes. Thus, processes and data are distributed to not only servers but also fog nodes in the IoT while centralized to servers in the cloud computing model.

* Corresponding author.
E-mail addresses: ryuji.oma.6r@stu.hosei.ac.jp (R. Oma), eno@ris.ac.jp (T. Enokido), makoto.takizawa@computer.org (M. Takizawa).

https://doi.org/10.1016/j.iot.2018.08.003

We have to reduce the electric energy consumed in information systems in order to realize green societies [3]. Especially, the IoT is more scalable than the cloud computing systems [2] since a huge number and various types of nodes like sensors and actuators are interconnected in addition to computers like servers. By distributing data and processes to handle the data to fog nodes, the traffic of servers can be reduced and the electric energy consumed by servers can be accordingly reduced. Thus, sensor data is efficiently transmitted and processed. On the other hand, more electric energy is consumed by fog nodes since fog nodes do the computation on sensor data. We have to reduce the electric energy consumed by not only servers but also fog nodes. Power consumption models of a computer are proposed to show how much electric power [W] the computer consumes to perform application processes [4,5]. Computation models of a computer are also proposed, which give the expected execution time of each process on the computer [4,5]. By taking advantage of the power consumption and computation models, we can estimate the electric energy [J] to be consumed by each computer, i.e. fog node and server to process sensor data.

The linear fog computing (LFC) model is first proposed to reduce the electric energy consumption of the nodes in the IoT [6]. Here, fog nodes are linearly connected from sensors to servers. The linear model is simple and useful for fewer number of sensors. However, a large amount of sensor data cannot be efficiently handled by each fog node due to the limited computation capability of each fog node, especially, edge node. In this paper, we propose a more general model, a *tree-based fog computing* (*TBFC*) model where processes and data are distributed to a tree structure of fog nodes whose root node shows servers in clouds and leaf nodes are edge nodes which communicate with sensors and actuators [7]. We evaluate the TBFC model compared with the cloud computing model in terms of total electric energy consumption and total processing time of the nodes. We show the total electric energy consumption and total execution time of nodes can be reduced in the TBFC model compared with the cloud computing model.

In Section 2, we overview related studies. In Section 3, we present the system model. In Section 4, we propose the TBFC model. In Section 5, we evaluate the TBFC model.

2. Related studies

Sensors generate sensor data by sensing events in physical environment and actions to be done by actuators are decided by using the sensor data. Thus, sensor data has to be processed in the IoT (Internet of Things). The IoT is composed of not only servers and clients like PCs but also various types of things, e.g. watches, glasses, smart phones, and tablets, which are interconnected in networks [1]. Here, things are equipped with devices, i.e. sensors and actuators. More than 50 billion devices are expected to be interconnected in networks [8]. Cloud computing systems are composed of a cloud of servers which are interconnected in networks [8]. Sensor data and requests issued by sensors and clients are processed by servers in clouds to create responses and make a decision on actions. Then, responses are sent to the actuators and clients. If the IoT is realized in the cloud computing model [2], devices like sensors and actuators are connected to servers in a cloud through networks. Sensors collect the sensor data on physical environment and send the data to servers [1]. Servers in clouds receive the sensor data and process the sensor data to make a decision on actions. The sensor data are also stored in databases of servers. By processing and analyzing the sensor data, actions are decided and sent to actuators. On receipt of actions, actuators do the actions on physical environment. There are many types of IoT systems. For example, in Home Electric Management Systems (HEMS) [9], data on the total electric energy consumed by things in a home, e.g. air-conditioners, lights, refrigerators, and washing machines, is sent to servers in a cloud through networks. Then, users check the current electric power consumed by their house through smart phones and tablets. The Building Electric Management Systems (BEMS) [10] minimizes the total electric energy consumption of a building. Furthermore, to analyze the sensor data of the building, a manager of the building can receive the current situation of the building and find the faults of components. However, a huge number and various types of sensors recently generate the large volume of sensor data and send the sensor data in networks. Thus, servers in clouds receive the large volume of data from sensors. In traditional cloud computing systems [2], it is not easy to realize the IoT due to heavy network traffic and server overhead which imply long delay time to deliver sensor data to servers and actions to actuators. In order to realize the IoT, an intermediate layer named *fog* layer [11] is introduced between devices and clouds.

A fog computing model is composed of devices, fog nodes, and clouds of servers [11,12]. Fog nodes support capability of not only data transmission like routers but also storages and computations, i.e. process the sensor data. Since sensor data is processed by each fog node in the fog computing model, a fog node generates smaller output data than input data received from sensors and other fog nodes. Then, the fog node forwards the output data to other fog nodes. Servers also can receive smaller data than sensor data collected by sensors. Furthermore, since the sensor data is processed near to devices, the actuators can act more promptly, i.e. in realtime manner from the cloud computing model. Recently, fog computing models are discussed by many researchers and current studies focus on implementation of fog computing systems [11]. In this paper, we newly consider an energy-efficient fog computing model to realize the green IoT. Two types of energy-efficient fog computing models, linear model [6] and tree-based model [7,13] are proposed in our previous studies. The linear model is composed of a sequence of fog nodes, where one parent fog node has one child fog node. Sensors generate sensor data from physical environment and send the sensor data to fog nodes named edge nodes. Fog nodes process the data and send the processed data to the other fog nodes. In the tree-based model, fog nodes are hierarchically structured. The edge node receives the sensor data from sensors and sends the data to a parent fog node. The fog node processes the input data and sends output data obtained by processing the input data to a parent fog node. Thus, data and process are distributed to

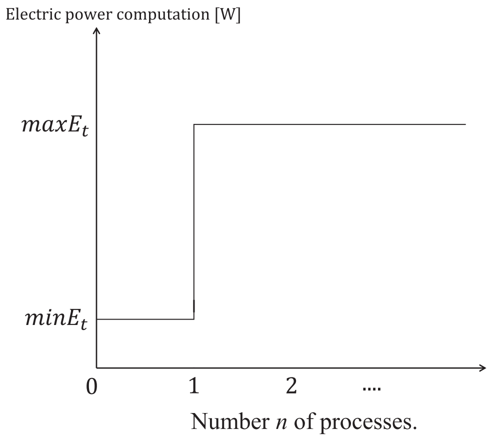

Fig. 1. Simple power consumption (SPC) model.

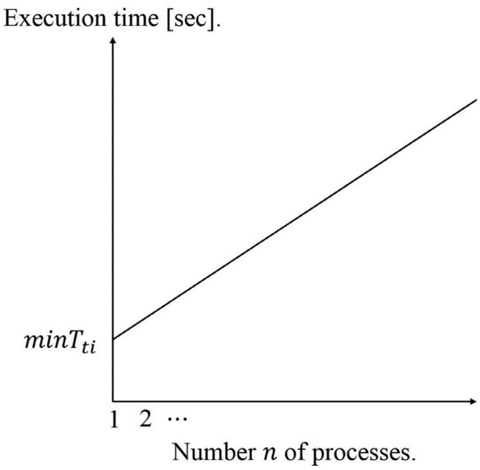

Fig. 2. Simple consumption (SC) model.

not only clouds but also fog nodes. In paper [7], computation complexity of a process performed by each fog node is not considered. In this paper, we propose a computation model of each fog node in a tree structure.

There are approaches to reducing the electric energy consumption of information systems. One is the hardware-oriented approach, where energy-efficient hardware components like CPUs and architectures are developed [14–16]. A computer consumes the electric power by performing processes. Thus, we have to consider not only the electric power of each hardware component but also software, especially application processes. Our approach is referred to as the *macro-level* approach [17,18], where we discuss how much electric power each computer totally consumes to perform application processes. Macro-level power consumption models of a computer are proposed to show how much electric power [W] the computer consumes to perform application processes [4,5]. In the simple power consumption (SPC) model [4], a computer s_t consumes maximum electric power $maxE_t$ if at least one application process is performed, otherwise minimum electric power $minE_t$ as shown in Fig. 1. The execution time [sec] of each process p_i linearly increases as the total number n of processes concurrently performed on server s_t increases as shown in Fig. 2. $minT_{ti}$ shows the minimum execution time of a process p_i on a server s_t where no other process is performed.

In the MLPCM (Multi-level Power Consumption with Multiple CPUs) model [19], the electric power consumption of a computer to perform application processes depends on the number of active CPUs, cores, and threads. Computation models of a computer are also proposed, which give the expected execution time of each process on the computer [4,5]. By using the models, we can estimate the electric energy to be consumed by a computation to perform application processes [19].

3. System model

In addition to computers, a huge number and various types of devices like sensors and actuators are interconnected in the IoT (Internet of Things) [1,20]. In addition to requests issued by clients, a large volume of sensor data are transmitted from sensors to servers in networks. Servers select actions by analyzing sensor data and send the action to actuators. On

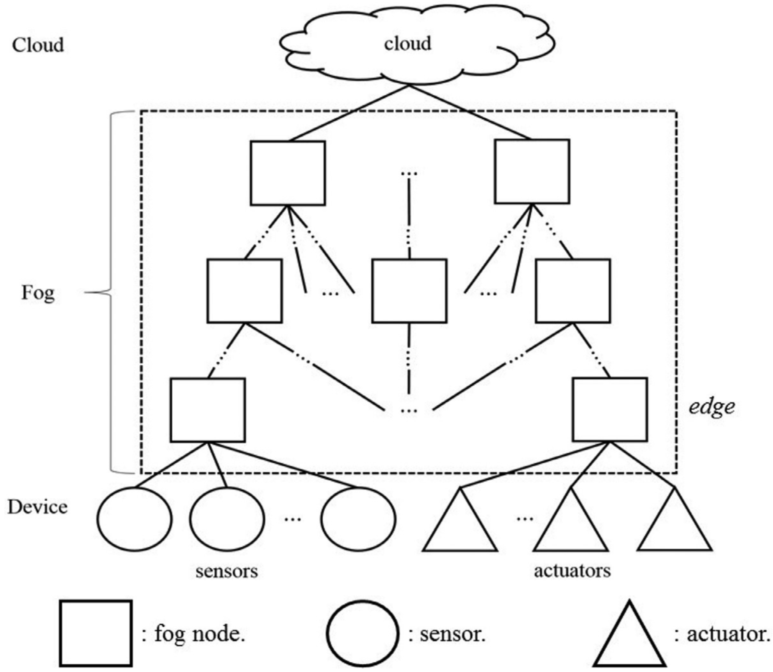

Fig. 3. Fog computing model.

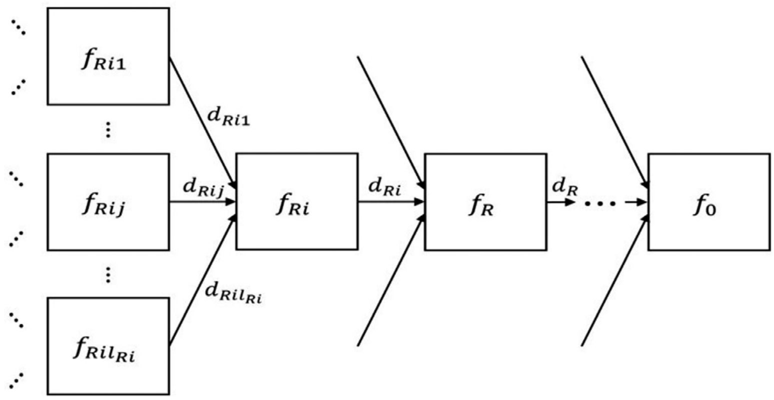

Fig. 4. A tree-based fog computing (TBFC) model.

receipt of an actions, each actuator performs the action on physical environment. In order to reduce the network traffic and satisfy the time constrains between sensors and actuators, the IoT is composed of three layers [21], cloud, fog, and device layers as shown in Fig. 3. Clouds are composed of servers [2], where each server supports applications with computation and storage services.

The device layer is composed of sensors and actuators. A sensor collects data obtained by sensing events occurring in physical environment. Sensor data collected by sensors is delivered to servers in networks. For example, sensor data is forwarded to neighbor sensor nodes in wireless networks as discussed in wireless sensor networks (WSNs) [20]. Sensor data is finally delivered to edge fog nodes at the fog layer. Based on the sensor data, actions to be done by actuators are decided in the IoT. Actuators receive actions from edge fog nodes and perform the actions on the physical environment.

Fog nodes are at a layer between the device and cloud layers [11]. Fog nodes are interconnected with other fog nodes in networks. A fog node supports the routing function where messages are routed to destination nodes, i.e. routing between servers and edge nodes like network routers [20]. Thus, fog nodes receive sensor data and forward the sensor data to servers in fog-to-fog communication. More importantly, a fog node does some computation on a collection of input data sent by sensors and other fog nodes. In addition, the input data is processed and new output data, i.e. processed data is generated by a fog node. For example, an average value is calculated by summing a collection of data obtained from sensor nodes. Thus, the output data is smaller than the input data. Data processed by a fog node is sent to neighbor fog nodes and servers

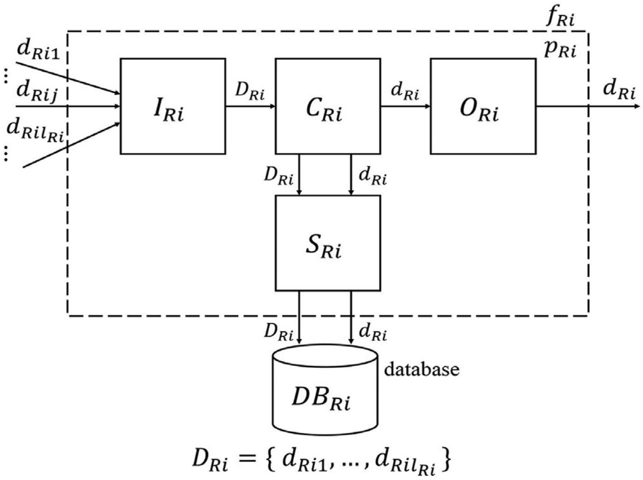

Fig. 5. Model of a process p_{Ri} on a fog node f_{Ri}.

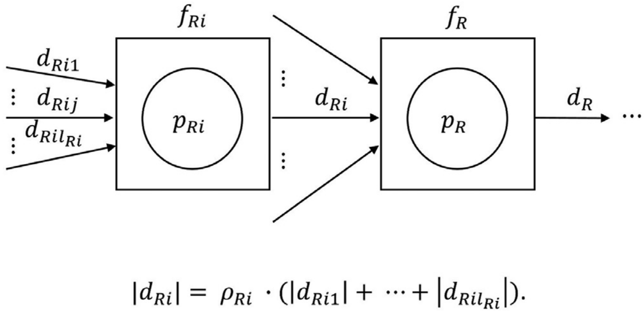

Fig. 6. Fog nodes.

finally receive data processed by fog nodes. In addition, a fog node makes a decision on what actions actuators have to do based on sensor data. Then, the edge nodes issue the actions to actuator nodes. A fog node is also equipped with storages to buffer data. Thus, data and processes are distributed to not only servers but also fog nodes in the fog computing model while centralized to servers of clouds in the cloud computing model.

4. A tree-based fog computing (TBFC) model in the IoT

4.1. Tree-structure of fog nodes

In the linear fog computing (LFC) model [6], the nearer to devices a fog node is, the larger data the fog node has to process. For example, an edge node receives sensor data from every sensor. In this paper, we newly consider a tree structure of fog nodes so that each fog node can process data of smaller size. Let f_0 be a root node which shows a cloud of servers. The root node f_0 has child fog nodes f_{01}, ..., f_{0l_0} ($l_0 \geq 0$). Here, each fog node f_{0i} also has child fog nodes f_{0i1}, ..., $f_{0il_{0i}}$ ($l_{0i} \geq 0$). Thus, each fog node has one parent fog node and child fog nodes as shown in Fig. 4. A notation f_R shows f_0, i.e. $R = 0$ if f_R is a root node. If f_R is an ith child of a fog node $f_{R'}$, f_R is $f_{R'i}$, i.e. R is a concatenation $R'i$ of R' and i. Suppose a fog node f_R is at level m of a tree and an ith child of a fog node $f_{R'}$. The label R of a fog node f_R shows a sequence of labels $0r_1r_2, ...r_{m-1}i$, where the label R' of the parent fog node $f_{R'}$ is $0r_1r_2,...,r_{m-1}$. Here, each $1 \leq r_i \leq l_{0r_1,...,r_{i-1}}$. Thus, the label R ($= 0r_1r_2,..., r_{m-1}i$) of a fog node f_R shows a path f_0, f_{0r_1}, $f_{0r_1r_2}$, ..., $f_{0r_1r_2,...,r_{m-1}}$ ($= f_R$) from a root f_0 to the fog node f_R. Here, the length $|R|$ of the label R is m. A fog node f_R is at level $|R|$ - 1 in the tree. Thus, each fog node f_R has child fog nodes f_{R1}, ..., f_{Rl_R} ($l_R \geq 0$) where f_{Ri} is an ith child fog node of the fog node f_R. In turn, f_R is a parent fog node of the fog node f_{Ri}. There is at most one parent fog node of each fog node. Each fog node f_{Ri} has also l_{Ri} (≥ 0) child fog nodes f_{Ri1}, ..., $f_{Ril_{Ri}}$. An edge node f_{Ri} has no child node ($l_{Ri} = 0$) while having one parent node. A root node f_0 has no parent node while having l_0 (≥ 0) child fog nodes.

A fog node f_{Ri} takes input data d_{Rij} sent by each child fog node f_{Rij} ($j = 1,..., l_{Ri}$). A process p_{Ri} in the fog node f_{Ri} does the computation on a collection D_{Ri} of input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ obtained from the child fog nodes f_{Ri1}, ..., $f_{Ril_{Ri}}$ and generates output data d_{Ri}. Then, the fog node f_{Ri} sends the output data d_{Ri} to the parent fog node f_R.

4.2. Fog nodes

In each fog node, not only routing functions but also computation on sensor data are performed. Each process p_{Ri} of a fog node f_{Ri} is modeled to be composed of four modules, an input I_{Ri}, computation C_{Ri}, output O_{Ri}, and storage S_{Ri} modules [6] as shown in Fig. 5. The input module I_{Ri} receives data d_{Rij} from each child fog node f_{Rij} ($j = 1,..., l_{Ri}$, $l_{Ri} \geq 0$). Then, the computation module C_{Ri} takes a collection D_{Ri} of the input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ and generates the output data d_{Ri} by doing the computation on the input data collection D_{Ri}. For example, d_{Ri} is an average value $(d_{Ri1} + ... + d_{Ril_{Ri}}) / l_{Ri}$ of the input data d_{Ri1}, ..., $d_{Ril_{Ri}}$. Then, the output module O_{Ri} sends the output data d_{Ri} to a parent fog node f_R in networks. The storage module S_{Ri} stores the input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ and output data d_{Ri} in the storage DB_{Ri}. For example, a collection of the output data d_{Ri} and input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ are buffered in the storage DB_{Ri}. If the fog node f_{Ri} fails to deliver the output data d_{Ri} to the parent f_R, the fog node f_{Ri} retransmits the data d_{Ri} which is stored in the database DB_{Ri}. Since the volume of the storage DB_{Ri} is limited, the most obsolete data is removed to make space to store new data if the storage DB_{Ri} is full.

In each fog node f_{Ri}, input data is processed by the computation module C_{Ri} in addition to the routing function of the input I_{Ri} and output O_{Ri} modules. A notation $|d|$ shows the size [bit] of data d. Thus, the size $|d_{Ri}|$ of the output data d_{Ri} is smaller than the input data $D_{Ri} = \{d_{Ri1}, ..., d_{Ril_{Ri}}\}$, $|d_{Ri}| \leq |D_{Ri}|$ $(= |d_{Ri1}| + ... + |d_{Ril_{Ri}}|)$. The ratio $|d_{Ri}| / |D_{Ri}|$ is the *reduction ratio* ρ_{Ri} (≤ 1) of a fog node f_{Ri} (Fig. 6). For example, let D_{Ri} be a set $\{v_1, v_2, v_3, v_4\}$ of four numbers showing temperature obtained by child fog nodes f_{Ri1}, ..., f_{Ri4}, respectively. If the output data d_{Ri} is an average value v of the values v_1, ..., v_4, the reduction ratio ρ_{Ri} of the fog node f_{Ri} is $|d_{Ri}| / |D_{Ri}| = 1/4$.

On the other hand, servers and devices are interconnected with networks in the cloud computing model. Here, each fog node does just a routing function. Thus, each fog node f_{Ri} is only composed of input I_{Ri} and output O_{Ri} modules. In the root node f_0, every computation on the sensor data is performed.

4.3. Energy consumption in the fog computing model

We discuss the electric energy consumed by a fog node f_{Ri} to receive, do the computation on, store, and send data. In this paper, we assume the input I_{Ri}, computation C_{Ri}, storage S_{Ri}, and output O_{Ri} modules of each fog node f_{Ri} are serially performed for each collection of input data $D_{Ri} = \{d_{Ri}, ..., d_{Ril_{Ri}}\}$ as shown in Fig. 8.

Let $EI_{Ri}(x)$, $EC_{Ri}(x)$, $ES_{Ri}(x)$, and $EO_{Ri}(x)$ show the electric energy [J] consumed by the input I_{Ri}, computation C_{Ri}, storage S_{Ri}, and output O_{Ri} modules of a fog node f_{Ri} to input, do the computation on, store, and output data of size x, respectively. $TI_{Ri}(x)$, $TC_{Ri}(x)$, $TS_{Ri}(x)$, and $TO_{Ri}(x)$ denote time [sec] for a fog node f_{Ri} to input, do the computation on, store, and output data of size x, respectively. Here, the fog node f_0 stands for a server s in a cloud. The transmission time $TT_{R,\,Ri}(x)$ shows time to transmit data of size x [bit] between a child fog node f_{Ri} and a parent fog node f_R in networks. Here, we assume $TT_{R,\,Ri}(x) = TT_{Ri,\,R}(x)$. Let $b_{R,\,Ri}$ be the bandwidth [bps] of a network between a pair of nodes f_R and f_{Ri}. Here, $b_{R,\,Ri} = b_{Ri,\,R}$. The transmission time $TT_{R,\,Ri}(x)$ is $x/b_{R,\,Ri}$ [sec] between the parent fog node f_R and the child fog node f_{Ri}.

First, we consider the electric energy consumed by a fog node f_{Ri} to receive input data $D_{Ri} = \{d_{Ri1}, ..., d_{Ril_{Ri}}\}$ from child fog nodes f_{Ri1}, ..., $f_{Ril_{Ri}}$, generate output data d_{Ri} by processing the input data D_{Ri}, store the input data D_{Ri} and output data d_{Ri} in the storage DB_{Ri}, and send the output data d_{Ri} to the parent fog node f_R. The electric energy $TE_{Ri}(|D_{Ri}|)$ consumed by each fog node f_{Ri} is given as follows:

$$TE_{Ri}(|D_{Ri}|) = EI_{Ri}(|D_{Ri}|) + EC_{Ri}(|D_{Ri}|) + ES_{Ri}(|D_{Ri}| + |d_{Ri}|) + EO_{Ri}(|d_{Ri}|). \quad (1)$$

Here, $|D_{Ri}| = |d_{Ri1}| + ... + |d_{Ril_{Ri}}|$ and $|d_{Ri}| = \rho_{Ri} \cdot |D_{Ri}| = \rho_{Ri} \cdot (|d_{Ri1}| + ... + |d_{Ril_{Ri}}|)$. In the root node f_0, the electric energy $TE_0(|D_0|)$ is consumed as follows:

$$TE_0(|D_0|) = EI_0(|D_0|) + EC_0(|D_0|) + ES_0(|D_0| + |d_0|). \quad (2)$$

The execution time $ET_{Ri}(|D_{Ri}|)$ and $ET_0(|D_0|)$ of each fog node f_{Ri} and a root fog node f_0 to receive and compute input data D_{Ri}, send output data d_{Ri}, and store the data D_{Ri} and d_{Ri} in the storage DB_{Ri} are given as follows:

$$ET_{Ri}(|D_{Ri}|) = TI_{Ri}(|D_{Ri}|) + TC_{Ri}(|D_{Ri}|) + TS_{Ri}(|D_{Ri}| + |d_{Ri}|) + TO_{Ri}(|d_{Ri}|). \quad (3)$$

$$ET_0(|D_0|) = TI_0(|D_0|) + TC_0(|D_0|) + TS_0(|D_0| + |d_0|). \quad (4)$$

The execution time $TI_{Ri}(x)$, $TS_{Ri}(x)$, and $TO_{Ri}(x)$ [sec] of the input, storage, and output modules to handle data of size x are linearly proportional to the size x of data.

$$TI_{Ri}(x) = i_{Ri} \cdot x. \quad (5)$$

$$TS_{Ri}(x) = s_{Ri} \cdot x. \quad (6)$$

$$TO_{Ri}(x) = o_{Ri} \cdot x. \quad (7)$$

Here, i_{Ri}, s_{Ri}, and o_{Ri} are constants. The execution time $TC_{Ri}(x)$ of the computation module C_{Ri} to process input data of size x depends on an algorithm of the computation module C_{Ri}. In this paper, we consider two types of processes where computation complexities are $O(x)$ and $O(x^2)$. The execution time $TC_{Ri}(x)$ is $c_{Ri} \cdot x$ or $c_{Ri} \cdot x^2$, where c_{Ri} is a constant, depending on the computation complexity $O(x)$ and $O(x^2)$, respectively.

The power consumption models [4,22,23] are proposed to show how much electric power a computer consumes to perform processes. In this paper, we take the simple power consumption (SPC) model for simplicity [4]. Here, the power consumption PC_{Ri} [W] of a fog node f_{Ri} is maximum xE_{Ri} [W] if at least one process is performed, otherwise, PC_{Ri} is minimum mE_{Ri} [W] as shown in Fig, 1. The electric energy consumption $EC_{Ri}(x)$ [J] of a fog node f_{Ri} to process input data of size x is given as follows:

$$EC_{Ri}(x) = xE_{Ri}[W] \cdot TC_{Ri}(x)[sec]. \tag{8}$$

The electric power consumption PI_{Ri} and PO_{Ri} of the input I_{Ri} and output O_{Ri} modules are proportional to the receiving and transmission rates of a fog node f_{Ri}, respectively [5]. In this paper, we assume the input time $TI_{Ri}(x)$ is the transmission time of data of size x from a fog node f_{Rij} to the fog node $f_{Ri} = TT_{Rij,\, Ri}(x)$ and output time $TO_{Ri}(x) = TT_{Ri,\, R}(x)$. Hence, the electric energy consumption $EI_{Ri}(x)$ and $EO_{Ri}(x)$ [J] to receive and send data of size x, respectively, are given as follows:

$$EI_{Ri}(x) = PI_{Ri} \cdot TI_{Ri}(x). \tag{9}$$

$$EO_{Ri}(x) = PO_{Ri} \cdot TO_{Ri}(x). \tag{10}$$

PS_{Ri} [W] shows the electric power of a fog node f_{Ri} to store data in a database DB_{Ri} which depends on the access rate a_{Ri} [bps]. Hence, the electric energy consumption $ES_{Ri}(x)$ [J] of a fog node f_{Ri} to store input data of size x and output data of size $\rho_{Ri} \cdot x$ is given as follows:

$$ES_{Ri}(x) = PS_{Ri} \cdot TS_{Ri}((1 + \rho_{Ri}) \cdot x). \tag{11}$$

4.4. Energy consumption in the cloud computing model

In the tree-based cloud computing (TBCC) model, every sensor data is processed by the root node f_0, i.e. a server in a cloud. Similarly to the TBFC model, a root node f_0 stands for a cloud. Fog nodes are structured in a tree. Each fog node f_{Ri} just does the routing function, i.e. input and output modules are performed. The total electric energy $CE_{Ri}(|D_{Ri}|)$ and $CE_0(|D_0|)$ [J] are consumed by a fog node f_{Ri} and a root fog node f_0, respectively, as follows:

$$CE_{Ri}(|D_{Ri}|) = EI_{Ri}(|D_{Ri}|) + EO_{Ri}(|D_{Ri}|). \tag{12}$$

$$CE_0(|D_0|) = EI_0(|D_0|) + EC_0(|D_0|) + ES_0(|D_0| + |d_0|). \tag{13}$$

Here, the reduction ratio ρ_{Ri} is one ($\rho_{Ri} = 1$) for each fog node f_{Ri} since input data is not processed and is just forwarded by the fog node f_{Ri}. Hence, the size $|d_{Ri}|$ of the input data d_{Ri} is the same as the size $|D_{Ri}|$ (= $|d_{Ri}|$ + ... + $|d_{Ril_{Ri}}|$) of the input data.

5. Evaluation

5.1. Evaluation model

We evaluate the tree-based fog computing (TBFC) model compared with the tree-based cloud computing (TBCC) model. In this paper, we consider a height-balanced k-ary tree of fog nodes whose height is h, denoted by $\langle k, h\rangle$ tree. Here, each fog node has k child fog nodes and every edge fog node is at (h - 1) level.

In the TBFC model, each fog node does the computation in addition to the routing function. On the other hand, each fog node does only the routing function in the cloud computing model. In the TBFC model, the reduction ratio ρ_{Ri} of each process p_{Ri} on a fog node f_{Ri} is assumed to be the same, i.e. $\rho_{Ri} = \rho$. In the TBCC model, the reduction ratio ρ_{Ri} of each fog node f_{Ri} is one, $\rho = 1$.

We consider a server f_0 where the minimum electric power consumption mE_0 is 126.1 [W] and maximum electric power consumption xE_0 is 301.3 [W] with two Intel Xeon E5-2667 v2 CPUs [24]. Each fog node f_{Ri} is realized by a Raspberry Pi Model B [25]. Here, the minimum electric power mE_{Ri} is 2.1 [W] and the maximum electric power xE_{Ri} is 3.7 [W].

In order to make clear the computation rate of each fog node, we perform a same C program p which uses only CPU on the server f_0 and the fog node f_{Ri}. It takes $mT_0 = 0.879$ [sec] to perform the process p without any other process on the server f_0. The computation rate CR_0 of the server f_0 is assumed to be one. If the same process p is performed without any other process on a fog node f_{Ri}, it takes $mT_{Ri} = 4.75$ [sec]. Hence, the computation rate CR_{Ri} of each fog node f_{Ri} is $0.879/4.75 = 0.185$ for the server f_0. That is, the server f_0 is about 5.4 times faster than a fog node f_{Ri}.

Table 1 summarizes the parameters of the server and each fog node.

We consider a balanced $\langle k, h\rangle$ tree of fog nodes in the TBFC model and the TBCC model (Fig. 7). That is, each fog node f_R has k (≥ 1) child nodes f_{R1}, ..., f_{Rk} and every edge node is at the same level $h - 1$.

Let p be a process to handle sensor data. We assume a process p is realized as a sequence of subprocesses p_0, p_1, ..., p_m ($m \geq 1$). The process p_m takes sensor data from all the sensors and sends the output data to the process p_{m-1}. Thus,

Table 1
Parameters.

Parameters	DSLab. (Cloud)	Raspberry Pi (Fog)
CPU	Intel Xeon E5-2667 v2	Broadcom BCM2837
***maxE*[W]**	301.3	3.7
***minE*[W]**	126.1	2.1
CR	1	0.185

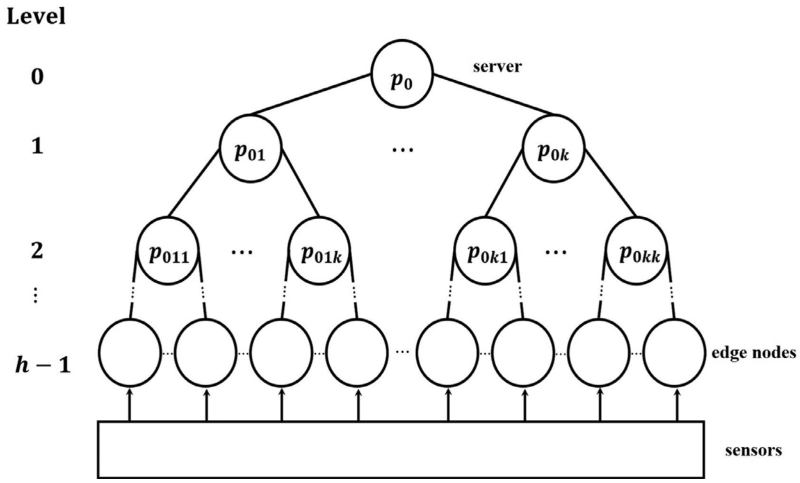

Fig. 7. $\langle k, h\rangle$ tree of fog nodes.

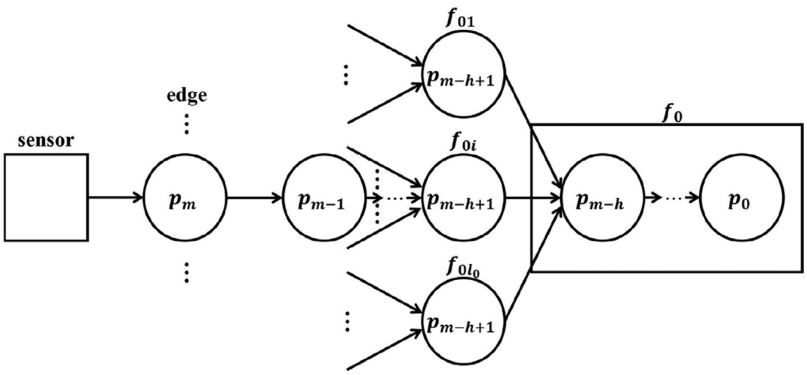

Fig. 8. Process.

each process p_i receives input data from a preceding process p_{i+1} and outputs data to a succeeding process p_{i-1}, which is obtained by processing the input data. The execution time of each process p_i to do the computation on input data of size x is assumed to be $O(x)$ or $O(x^2)$ in this evaluation. For example, if some item is selected in the input data of size x, the computation complexity is $O(x)$. If the input data is sorted, the complexity is $O(x^2)$. The process p_m is performed on k^{h-1}edge fog nodes of level h - 1. The process p_{m-1} is performed on k^{h-2} fog nodes of level h - 2. Thus, each fog node f_{Ri} of level l performs a process $p_{m-h+l+1}$ on k^l fog nodes. The process p_{m-h+2} is performed on k fog nodes of level 1, one level lower than the server f_0. A subsequence p_0, ..., p_{m-h+1} of processes are performed on the root node f_0 while each process p_l is performed on fog nodes at a level l - m + h ($l = m$ - h + 2,..., m) as shown in Fig. 8. In a tree of height h, there are totally (1 - k^h) / (1 - k) nodes.

Let S be the total size of sensor data [B] collected by sensor nodes. Each edge node receives sensor data of size S / k^{h-1} since there are k^{h-1} edge nodes. Thus, the higher tree, i.e. the larger h, the small sensor data each fog node receives. In this evaluation, we assume $S = 1$ [MB].

The network N supports every pair of nodes with the same bandwidth b. The bandwidth b is assumed to be 200 [Kbps].

In the TBCC model, we consider a $\langle k, h\rangle$ tree of fog nodes from sensors to the root node f_0, i.e. a server s in the same way as the TBFC model. Each fog node f_{Ri} just forwards messages to a parent fog node f_R. Hence, each fog node f_{Ri} supports

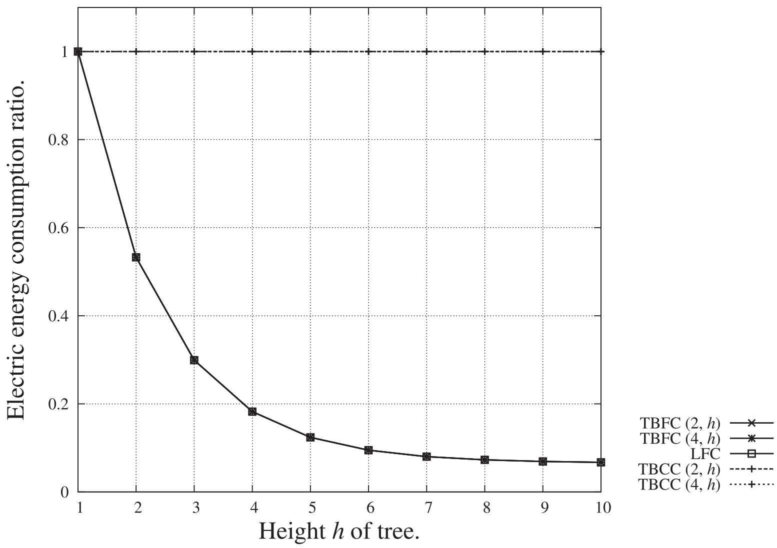

Fig. 9. Total electric energy consumption ratio with computation complexity $O(x)$ for height h.

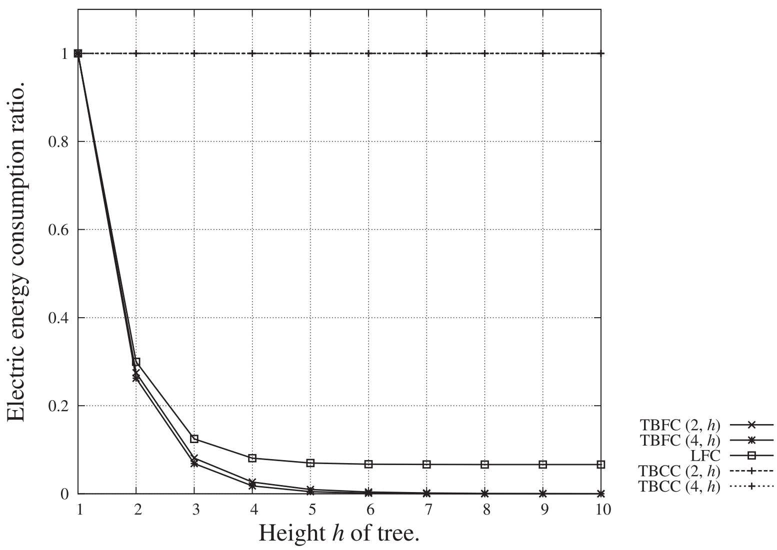

Fig. 10. Total electric energy consumption ratio with computation complexity $O(x^2)$ for height h.

only the input module I_{Ri} and output module O_{Ri}. The data obtained from sensors is just forwarded from a fog node f_{Ri} to another fog node f_R.

TBFC(k, h) and TBCC(k, h) stand for the tree-based fog computing (TBFC) and tree-based cloud computing (TBCC) models for a $\langle k, h \rangle$ tree. The linear fog computing (LFC) model of height h means the TBFC(1, h) model.

5.2. Evaluation results

Fig. 9 shows the ratio of the total electric energy consumed by nodes in the TBFC(2, h), TBFC(4, h), TBCC(2, h), TBCC(4, h), and linear fog computing (LFC) ($k = 1$) models to the TBCC(2, 1) for the tree height h. Here, the computation complexity of each process is $O(x)$ for size x of input data. As shown in Fig. 9, the total electric energy consumption of fog nodes is the same in the TBFC models and TBCC models, respectively, and can be reduced in the TBFC and LFC models compared with the cloud computing (TBCC) model. The total electric energy consumption of the TBCC(2, h) and TBCC(4, h) models are TBCC(1, 1) independently of the height h. The total electric energy consumption of fog nodes exponentially decreases as the number n of fog nodes increases in the TBFC(4, h), TBFC(2, h), LFC models.

Fig. 10 shows the ratio of the total electric energy consumption where the computation complexity of each process is $O(x^2)$. The total electric energy consumption ratios of the TBFC (2, h) and TBFC(4, h) models are smaller than the LFC model.

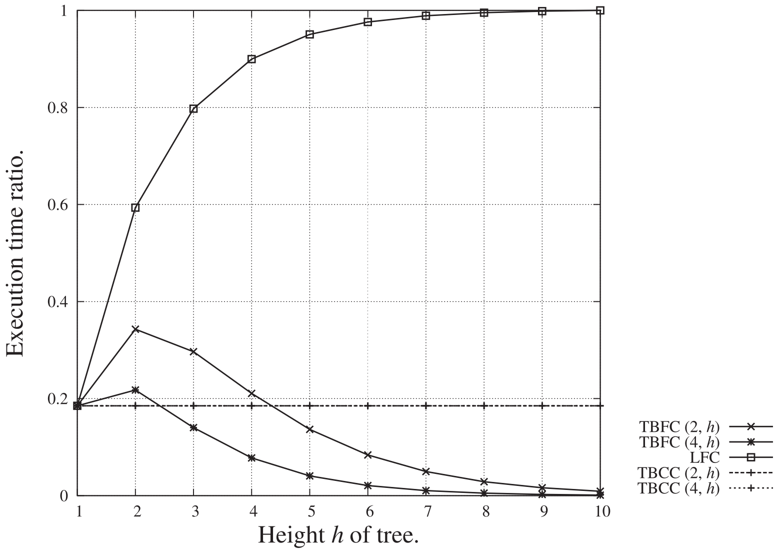

Fig. 11. Total execution time ratio with computation complexity $O(x)$ for height h.

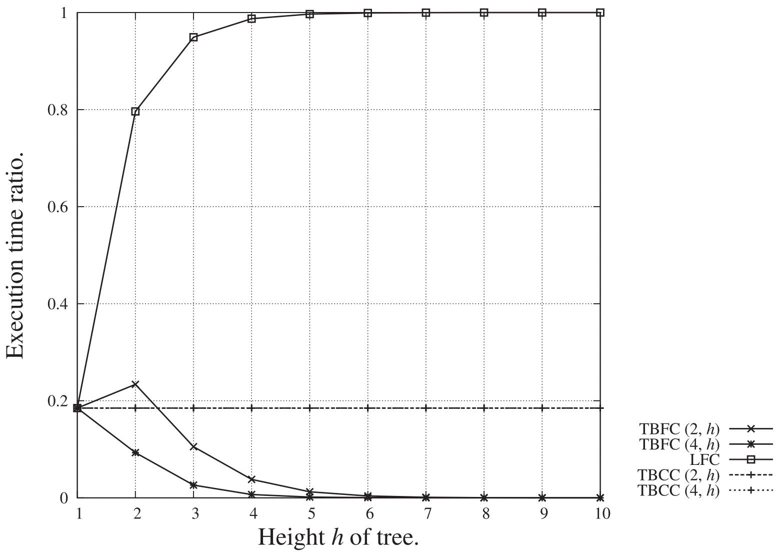

Fig. 12. Total execution time ratio with computation complexity $O(x^2)$ for height h.

Figs. 11 and 12 show the ratios of the total execution time of nodes in the TBFC(k, h) model to the TBCC(2, 1) model for tree height h where the computation complexity of each process is $O(x)$ and $O(x^2)$, respectively. The total execution time ratio can be reduced in the TBFC (4, h) and TBFC(2, h) models compared with the TBCC models for $h \geq 3$. For $h \leq 2$ and $h \leq 4$, it takes longer time to perform the process in the TBFC(2, h) and (4, h) models than the TBCC model, respectively. As shown in Fig. 11, the total execution time ratio of the LFC model monotonically increases as the tree height h increases for $O(x)$ processes. The total execution time of the TBCC model is constant and shorter than the LFC model. On the other hand, Fig. 12 shows the total execution time where the execution time of the process is $O(x^2)$ for size x of input data. The total execution time monotonically increases in the LFC model and is invariant in the TBCC models. The total execution time ratio of the TBFC (4, h) model monotonically decreases and shortest for computation complexity $O(x^2)$ of the processes. The total execution time ratio of the TBFC(2, h) model increases for $h \leq 2$ and decreases for $h > 2$.

Next, the electric energy consumption and execution time are measured for reduction ratio ρ where $h = 5$. Here, the canonical electric energy and execution time is defined to be ones of the TBCC(k, 5) with $\rho = 1.0$. We consider the ratios of electric energy consumption and execution time of each algorithm to the canonic ones. Figs. 13 and 14 show the electric energy consumption ratio of nodes for computation complexity $O(x)$ and $O(x^2)$ for reduction ratio ρ, respectively. As shown in Fig. 13, the electric energy consumption ratios of the TBFC(2, 5), TBFC(4, 5), and LFC models are the same and smaller

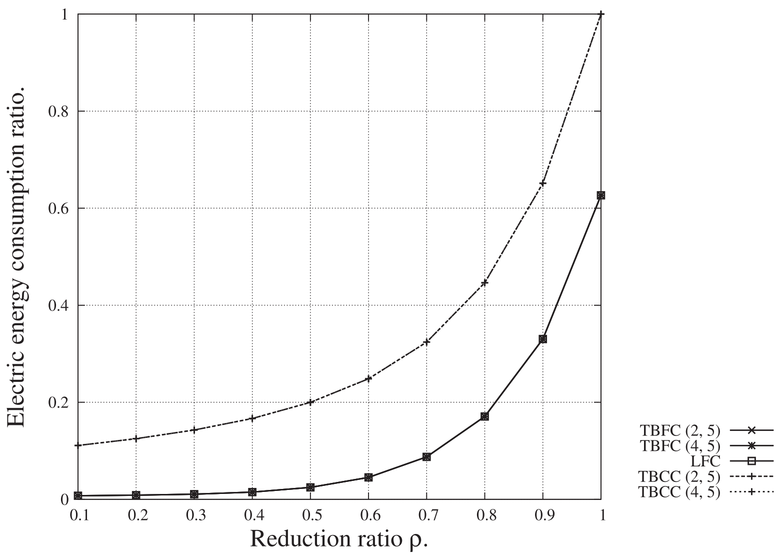

Fig. 13. Total electric energy consumption ratio with computation complexity $O(x)$ for reduction ratio ρ ($h = 5$).

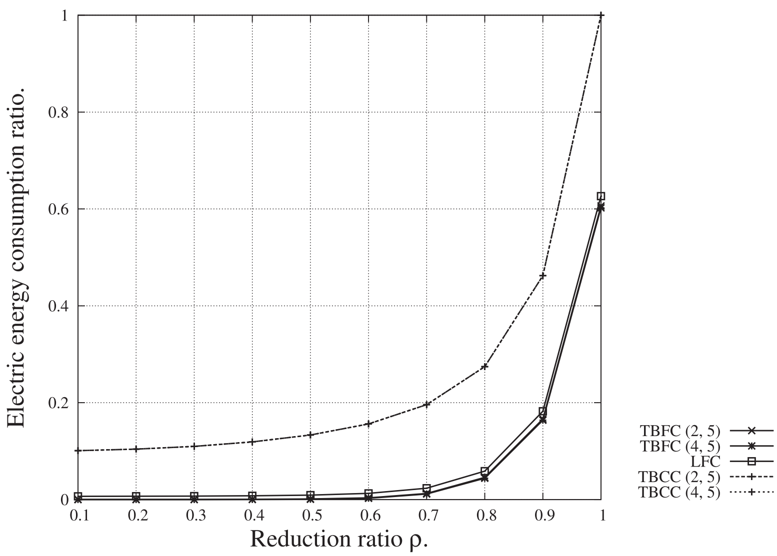

Fig. 14. Total electric energy consumption ratio with computation complexity $O(x^2)$ for reduction ratio ρ ($h = 5$).

than the TBCC models for computation complexity $O(x)$. In Fig. 14, the ratio of the TBFC(4, 5) model is a little bit smaller than the TBFC(2, 5) model.

Figs. 15 and 16 show the execution time ratio of nodes with computation complexity $O(x)$ and $O(x^2)$ for reduction ratio ρ, respectively. The execution time ratio monotonically increases as the reduction ratio ρ increases. The TBCC models imply larger ratios than the TBFC(4, 5) model for reduction ratio ρ and the TBFC(2, 5) model for $\rho \leq 0.8$ where the computation complexity is $O(x)$. For the computation complexity $O(x^2)$, the TBFC(4, 5) model supports the shortest ratio and the TBFC(2, 5) model supports smaller ratio than the TBCC models.

6. Concluding remarks

The IoT is composed of various types of devices like sensors and actuators in addition to computers like servers and clients. The IoT is more scalable than traditional networks like the Internet. Furthermore, huge amount of sensor data are transmitted and processed in networks. In order to realize the IoT, the fog computing model [1] is proposed where processes and data are distributed to not only servers but also fog nodes in order to reduce the delay time and processing overhead of servers. On the other hand, a huge amount of electric energy is consumed by a large number of nodes. Hence, it is critical to reduce the electric energy consumption of nodes in the IoT. In this paper, we proposed the tree-based fog computing (TBFC)

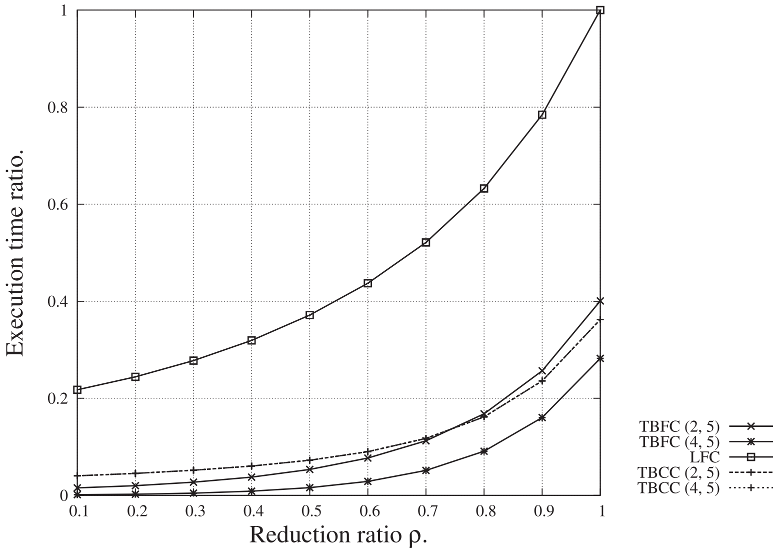

Fig. 15. Total execution time ratio with computation complexity $O(x)$ for reduction ratio ρ ($h = 5$).

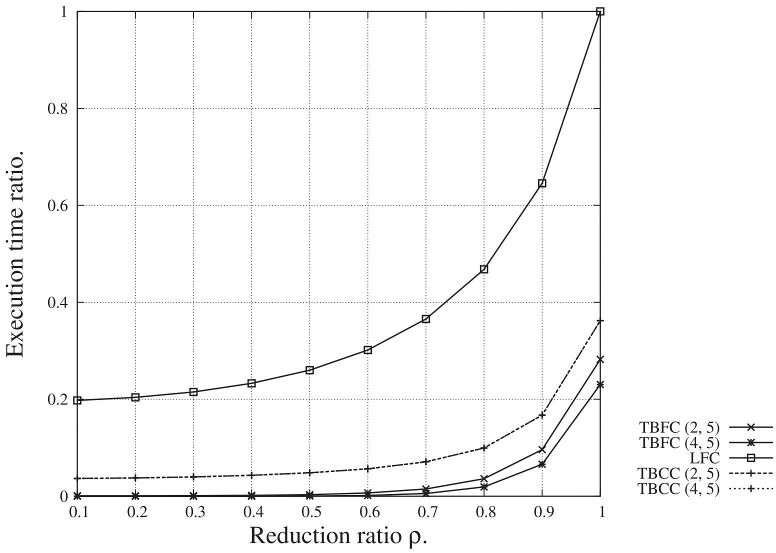

Fig. 16. Total execution time ratio with computation complexity $O(x^2)$ for reduction ratio ρ ($h = 5$).

model to reduce the total electric energy consumption and total execution time of fog nodes in the IoT. We evaluated the TBFC model compared with the cloud computing model. We showed the total electric energy consumption of nodes and the execution time of nodes can be reduced in the TBFC model compared with the cloud computing model.

Acknowledgment

This work was supported by JSPS KAKENHI grant number 15H0295.

References

[1] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, CISCO Press, 2018.
[2] M. Creeger, Cloud computing: an overview, Queue 7 (5) (2009) 3–4.
[3] Google green, http://www.google.com/green/2015.
[4] T. Enokido, A. Ailixier, M. Takizawa, An extended simple power consumption model for selecting a server to perform computation type processes in digital ecosystems, IEEE Trans. Ind. Inform. 10 (2014) 1627–1636.
[5] T. Enokido, A. Ailixier, M. Takizawa, An integrated power consumption model for communication and transaction based applications, in: Proceedings of the 25th IEEE International Conference on Advance Information Networking and Applications (AINA-2011), 2017, pp. 98–109.

[6] R. Oma, S. Nakamura, T. Enokido, M. Takizawa, An energy-efficient model of fog and device nodes in IoT, in: Proceedings of the 32nd IEEE International Conference on Advanced Information Networking and Applications (AINA-2018), 2018, pp. 301–306.
[7] R. Oma, S. Nakamura, T. Enokido, M. Takizawa, A tree-based model of energy-efficient fog computing systems in IoT, in: Proceedings of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2018), 2018, pp. 991–1001.
[8] S. Ray, Y. Jin, A. Raychowdhury, The changing computing paradigm with internet of things: a tutorial introduction, IEEE Design Test 33 (2016) 76–96.
[9] Y. sung Son, T. Pulkkinen, K. deok Moon, C. Kim, Home energy management system based on power line communication, IEEE Trans. Consum. Electron. 56 (2010) 1380–1386.
[10] Z. Shen, K. Yokota, J. Jin, A. Tagami, T. Higashino, In-networks self-learning algorithms for BEMS through a collaborative fog platform, in: Proceedings of the IEEE the 32nd International Conference on Advanced Information Networking and Applications (AINA-2018), 2018, pp. 1162–1169.
[11] A.M. Rahmani, J.-S.P. P. Liljeberg, A. Jantsch, Fog Computing in the Internet of Things, Springer, 2018.
[12] M. Chiang, B. Balasubramanian, F. Bonomi, Fog for 5G and IoT, WILEY, 2017.
[13] R. Oma, S. Nakamura, D. Duolikun, T. Enokido, M. Takizawa, Evaluation of an energy-efficient tree-based model of fog computing, in: Proceedings of the 21st International Conference on Network-Based Information Systems (NBiS-2018) (accepted), 2018.
[14] Amd ryzen, 2017, https://www.amd.com/en/ryzen.
[15] Intel xeon processor 5600 series. the next generation of intelligent server processors, 2010, http://www.intel.com/content/www/us/en/processors/xeon/xeon-5600-brief.html.
[16] Ssd, 2014, https://searchstorage.techtarget.com/definition/SSD-solid-state-drive.
[17] T. Enokido, A. Ailixier, S.M. Deen, M. Takizawa, Power consumption-based server selection algorithms for communication-based systems, in: Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS-2010), 2010, pp. 201–208.
[18] T. Enokido, A. Ailixier, M. Takizawa, A model for reducing power consumption in peer-to-peer systems, IEEE Syst. J. 4 (2010) 221–229.
[19] H. Kataoka, S. Nakamura, D. Duolikun, T. Enokido, M. Takizawa, Multi-level power consumption (MLPC) model and energy-aware server selection (EA) algorithm, Int. J. Grid Utility Comput. (IJGUC) 8 (2017) 201–210.
[20] F. Zhao, L. Guibas, Wireless Sensor Networks: An Information Processing Approach, Morgan Kaufmann Publishers, 2004.
[21] X. Yao, L. Wang, Design and implementation of IOT gateway based on embedded μtenux operating system, Int. J.Grid Utility Comput. 8 (1) (2017) 22–28.
[22] T. Enokido, A. Ailixier, M. Takizawa, Process allocation algorithms for saving power consumption in peer-to-peer systems, IEEE Trans. Ind. Electron. 58 (6) (2011) 2097–2105.
[23] D. Duolikun, H. Kataoka, T. Enokido, M. Takizawa, Simple algorithms for selecting an energy-efficient server in a cluster of servers, Int.J. Commun. Netw. Distrib. Syst. (IJCNDS) (Accepted) (2017) (2017).
[24] Dl360p gen8, https://www8.hp.com/h20195/v2/getpdf.aspx/c04128242.pdf?ver=2.
[25] Raspberry pi 3 model b, 2016, https://www.raspberrypi.org/products/raspberry-pi-3-model-b.