International Journal of Innovative
Computing, Information and Control
Volume **14**, Number **6**, December **2018**

 ISSN 1349-4198
pp. **2011–2024**

TRADE-OFF BETWEEN SERVICE DELAY AND POWER CONSUMPTION IN EDGE-CLOUD COMPUTING

Xu Wang[1,2], Hong Ni[1], Rui Han[1,*] and Xingwang Huang[1,2]

[1]National Network New Media Engineering Research Center
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
{ wangx; nih; huangxw }@dsp.ac.cn; *Corresponding author: hanr@dsp.ac.cn

[2]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

Received January 2018; revised May 2018

Abstract. *IoT (Internet-of-Things) applications tend to have strict service delay restrictions. Always connecting to the cloud to process data is inefficient for such applications. Edge-cloud computing is a new computing paradigm ready to back IoT applications. It shares the advantages of both edge computing and cloud computing. While service delay is key to QoE (Quality of Experience) of such service systems, power consumption is evenly crucial when it comes to operating costs in practice. Keeping down both service delay and power consumption is a challenging problem as they are negatively correlated via multiple factors. This paper focuses on the trade-off between service delay and power consumption of a typical hybrid edge-cloud system. We build the statistical mathematical service delay model as well as the power model and formulate an optimization problem which suggests optimal system configurations, task allocation destination and quantity. This optimization problem is solved using multi-objective evolutionary algorithm. Results show that the proposed system model is able to reflect characteristics of practical systems, and is worth testing in practice. Further decisions can be made by combining the Pareto optimal results with the diverse needs of each system or other additional subjective preferences to achieve ideal optimal system states respectively.*

Keywords: Cloud computing, Edge computing, Power consumption estimation, Quality of service, Modeling

1. **Introduction.** The booming IoT (Internet-of-Things) technology has made billions of resource constrained devices connected to each other via the Internet [1]. By 2020, up to 50 billion devices are expected to be connected to the Internet. Most of these edge devices are embedded devices with low-power processors and limited memory, typically a single-board computer or a comparable device. They possess significantly less computing power, less memory space and less storage capacity than servers do. As a result, no serious workload can be supported on such devices. One conventional solution to this problem is to introduce cloud to IoT devices.

However, accessing services from the cloud will bring high latency to the application, at the same time, congestion to the network [2]. Recent work has shown that hybrid edge-cloud system is a better solution to such IoT applications. In a hybrid edge-cloud system, edge computing can be used as a supplement to cloud computing [3]. By combining edge computing and cloud computing, we can make the best use of the advantages and avoid

DOI: 10.24507/ijicic.14.06.2011

the disadvantages [4]. Recently, edge-cloud architecture is widely used by IoT applications like smart cities [5, 6], smart homes [7, 8] and live data analytic [9].

The main service requirement of edge-cloud computing is to minimize service delay. Global optimal result cannot be achieved by lowering service delay of either individual sub-system; thus, joint optimizations concerning both edge sub-system and cloud sub-system have been put forward based on edge-cloud architecture [10, 11]. However, in real production environment, service delay is not the only factor that matters. For instance, servers in performance mode are able to process user requests faster and achieve better QoE (Quality of Experience) in terms of service delay. At the same time, these servers also drain electricity faster than others, and thus increase the operating cost [12]. Researches have been conducted to solve energy minimization problem of mobile edge devices [13]. Since edge-cloud systems are run by companies in pursuit of profit, operating cost should be taken into consideration when optimizing service delay, especially when server power, in addition to edge device power, amount to a significant fraction of a modern data center's recurring cost [14].

In this paper, we focus on the trade-offs between QoE in terms of service delay and management cost in terms of power consumption, including both edge sub-system and cloud sub-system. To get a clear picture of the trade-off process, we proposed mathematical models of the edge-cloud system, concerning system delays and power consumption of both edge sub-systems and cloud sub-system, based on system statistical information. In addition, we formed a multi-objective optimization problem using the proposed mathematical models. By solving this optimization problem, we provided Pareto optimal solutions, which could be a basis of further decision making.

The remainder of this paper is organized as follows. Section 2 goes through some background and related works. Section 3 gives out the system architecture as well as mathematical models of the edge-cloud system and formulates a multi object minimization problem. Section 4 shows the experimental evaluation with results and discussion. Section 5 summarizes this work and gives out some conclusions.

2. Background.

Cloud computing is constructed based on centralized architecture, where data centers play a crucial role. Cloud computing has many advantages such as on-demand and flexible resource allocation, fast deployment and free from hardware maintenance. In addition, services based on cloud computing can be accessed without temporal or spatial limitations [15]. Thus, cloud computing is wildly used in IoT scenario [16]. In spite of its advantages, there exist disadvantages. Data centers usually locate far from users, geographically and topologically. Data sets need to be transferred back and forth several times between user and data center to complete one service routine, which is time-consuming, especially when large data sets are encountered. In consideration of network latency and bandwidth, the total service response time can be further prolonged by data transmission delay [17]. Therefore, network performance has become the bottleneck of applications with strict latency requirements, e.g., IoT applications with cameras.

The limitation of network latency and bandwidth is not a problem that centralized architecture can easily solve. That is where edge computing comes in. Edge computing makes use of decentralized architecture, in which services are offloaded to network edge. By utilizing resources close at the network edge and minimizing the hops between service and use, lower service response time as well as more stable network link status is achieved [18]. Edge computing can remove the major network bottleneck and eliminate the potential point of failure, as well as achieve better QoE by bringing applications and user closer. Therefore, edge computing is particularly suitable for geographically distributed applications with strict latency requirements, such as IoT applications [2, 19]. However, edge

computing systems are usually built using light servers with lower performance standards as well as power profiles than cloud servers. Thus, more edge devices may be involved to accomplish the same workload. Attending to heavy workload may also cause SLA (Service-Level Agreement) violation in terms of service delay.

In a hybrid edge-cloud computing system, on the one hand, edge computing sub-system communicates with user directly via local area networks and handle user requests independently to provide services with low latency; on the other hand, edge computing sub-system can also be connected to cloud to leverage the considerable computing resource and functionality of a cloud computing sub-system to overcome its performance bottleneck.

3. System Model.

3.1. System architecture.

This paper focuses on a kind of hybrid edge-cloud computing system architecture as shown in Figure 1.

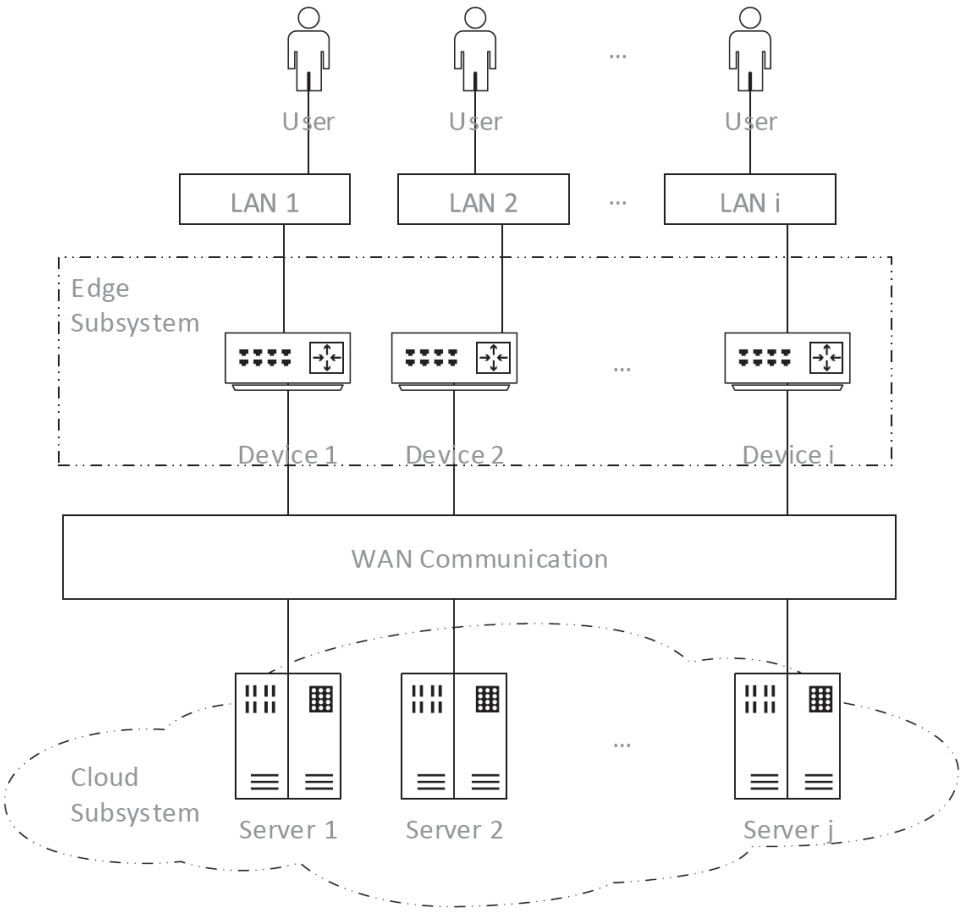

FIGURE 1. System architecture

The edge-cloud computing system consists of two parts, edge computing sub-system and cloud computing sub-system. $\mathcal{E}$ and $\mathcal{C}$ denote the sets of edge devices and cloud servers respectively. Edge computing sub-system and cloud computing sub-system are connected via public networks.

In a typical task execution scenario, shown in Figure 2, tasks are firstly sent separately from the user proxy to the nearest edge devices to process. Some less demanding tasks and tasks with strict latency requirements will be processed locally at the edge. The rest will be later on forwarded to and handled by the cloud platform via public networks.

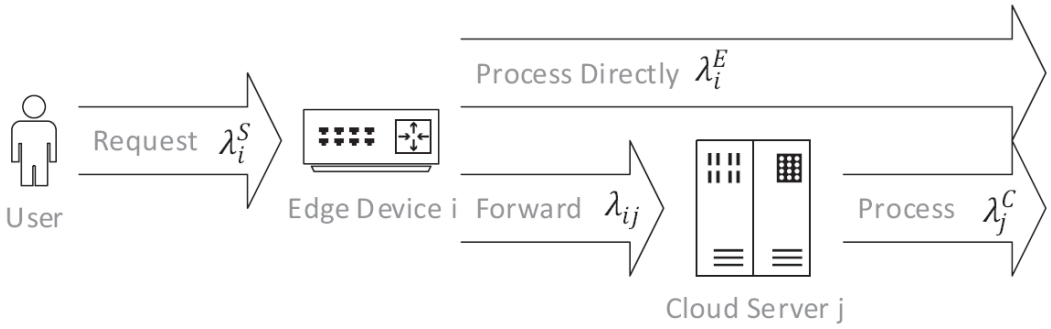

FIGURE 2. Workload flow

Edge devices are generally located within the vicinity of user proxy, so that they are connected via local area network, whose network latency can be neglected compared with subsequent network and processing delays. Edge devices are usually light servers, gateways, routers, etc.

Public network conditions are not so well as local area network that the network latency and capacity should be taken into account. In this paper, we consider the data transmission happens in a packet-switched network with symmetrical links.

One cloud server, usually consisting of multiple computing machines, is more powerful than edge devices. Machines on the same server share the same hardware configuration as well as the same power consumption profile. For simplicity, we assume that the machines are load-balanced and in steady state, such that each machine of one cloud server has the same resource utilization and the same traffic arrival rate, etc.

Consider a situation where a certain amount of workload is sent to edge devices by end users. On condition that workload is conducted by edge sub-system, due to negligible local area network delay, edge sub-system can achieve low overall service delay even with less powerful processing instruments. More devices in edge sub-system are involved to attend to the workload compared to cloud sub-system, especially when under high workload. The more devices involved, the more power consumed. On condition that workload is conducted by cloud sub-system, workload can be processed much faster than that is done by edge sub-system. The overall service delay can be high when network delays are added up. Moreover, one cloud server is more powerful, at the same time, more power consuming than an edge device. In spite of this, cloud sub-system utilizes significantly fewer machines under high workload. This is where the trade-off comes in. Decreasing service delay by utilizing edge sub-system more or lowering power consumption by utilization of cloud sub-system more, you cannot have it both ways.

By neglecting delays of local area network, processing delays at edge devices and cloud servers as well as transmitting delay across public network will be discussed. Moreover, power consumption models of both edge devices and cloud servers will be drawn.

3.2. **System delays.**

3.2.1. *Edge processing delay.* We assume that task arrivals at edge device E_i occur at rate λ_i^E according to a Poisson process, while task execution times have an exponential distribution with parameter μ_i^E. Given the average workload of each incoming task as L_0, by neglecting context switching overhead, essential CPU speed can be expressed as

$$v_i^E = L_0 \mu_i^E \tag{1}$$

The CPU utilization u_i^E can be expressed as

$$u_i^E = v_i^E \Big/ \hat{v}_i^E \tag{2}$$

where v_i^E and $\hat{v}_i^E$ denote the current and the maximum CPU processing speed of device E_i in MFLOPS (Mega FLoating-point Operations Per Second) respectively.

So

$$\mu_i^E = \frac{u_i^E \hat{v}_i^E}{L_0} \tag{3}$$

The processing delay at edge device can be modeled as an M/M/1 queue. According to queuing theory, the total response time, or in this case the processing delay, is the total amount of time a task spends in both the queue and in execution. The average processing delay of tasks on device E_i is

$$T_i^E = \frac{1}{\mu_i^E - \lambda_i^E} \tag{4}$$

in which $\lambda_i^E < \mu_i^E$ is required for the queue to be stable.

3.2.2. *Cloud processing delay.* We assume that task arrivals at one cloud server occur at rate λ_j^C according to a Poisson process, while task execution time has an exponential distribution with parameter μ_j^C. Tasks on the same server share the same task queue and are executed in first-come, first-served discipline. Moreover, we assume there are n_j machines running on a cloud server. For a cloud server, the task processing can be modeled as an M/M/c queue [20], where $c = n_j$.

Let the service occupancy be

$$\rho_j = \frac{\lambda_j^C}{n_j \mu_j^C} \tag{5}$$

and require $\rho_j < 1$ for the queue to be stable.

The probability that an arriving task needs to queue, aka the Erlang-C formula, is

$$E_c\left(n_j, \frac{\lambda_j^C}{\mu_j^C}\right) = \frac{\frac{1}{n_j!}\left(\frac{\lambda_j^C}{\mu_j^C}\right)^{n_j}}{\frac{1}{n_j!}\left(\frac{\lambda_j^C}{\mu_j^C}\right)^{n_j} + (1-\rho_j)\sum_{k=0}^{n_j-1}\frac{1}{k!}\left(\frac{\lambda_j^C}{\mu_j^C}\right)^{k}} \tag{6}$$

The average queuing time of tasks on server C_j is

$$W_j^{queue} = \frac{E_c\left(n_j, \frac{\lambda_j^C}{\mu_j^C}\right)}{\mu_j^C n_j (1-\rho_j)} \tag{7}$$

Similar to edge devices, μ_j^C can be converted to CPU utilization u_j^C

$$\mu_j^C = \frac{u_j^C \hat{v}_j^C}{L_0} \tag{8}$$

where $\hat{v}_j^C$ denotes the maximum CPU processing speed of server C_j in MFLOPS.

Furthermore, when workload varies, server management systems will power up or hibernate machines accordingly to achieve power savings [14]. Let δ_j be the status indicator of server C_j and

$$\delta_j = \begin{cases} 0, & \text{if server } C_j \text{ is hibernated} \\ 1, & \text{if server } C_j \text{ is active} \end{cases} \tag{9}$$

The total response time, or in this case the processing delay, is the total amount of time a task spends in both the queue and in execution. The average processing delay of tasks on server C_j is

$$T_j^C = \delta_j \left(W_j^{queue} + \frac{1}{\mu_j^C} \right) = \delta_j \left(\frac{E_c\left(n_j, \frac{\lambda_j^C}{\mu_j^C}\right)}{\mu_j^C n_j (1-\rho_j)} + \frac{1}{\mu_j^C} \right) \tag{10}$$

3.2.3. *Transmission delay.* Based on previous network assumption, links between edge devices and cloud servers are stable and symmetrical. The transmission delay between edge device E_i and cloud server C_j is

$$T_{ij} = \lambda_{ij} \tau_{ij} \tag{11}$$

where λ_{ij} denotes the task arrival rate from edge device E_i to cloud server C_j; τ_{ij} denotes the transmission delay between them.

Given the capacity of transmission route between edge device E_i and cloud server C_j as $\lambda_{ij}^{\max}$. Then

$$0 \le \lambda_{ij} \le \lambda_{ij}^{\max} \tag{12}$$

3.2.4. *System delay summary.* The total delay of entire edge-cloud system, which consists of processing delays of both edge and cloud sub-systems along with the transmission delay in between, can be drawn as follows

$$T = \sum_{i \in \mathcal{E}} T_i^E + \sum_{j \in \mathcal{C}} T_j^C + \sum_{i \in \mathcal{E}} \sum_{j \in \mathcal{C}} T_{ij} \tag{13}$$

3.3. **Energy profiles.**

3.3.1. *Edge energy profile.* Edge devices come in various forms and architectures, so that their power consumption profiles vary accordingly. However, a general model can be drawn similar to server power consumption profile. See next section for more details. Generally, edge devices consume more power as the workload increases, while still drains significant power to maintain basic system functionalities even when idle. Thus, the power consumption of edge device E_i can be expressed as

$$P_i^E = A_i^E + k_i^E u_i^E \tag{14}$$

where A_i^E denotes the power consumption when a device is idle; k_i^E denotes a linear growth factor; u_i^E denotes the CPU utilization and lies within interval of $[0, 1]$ inclusively.

3.3.2. *Cloud energy profile.* Researchers of [21] studied the power consumption of connection servers, whose tasks are CPU, network, and memory intensive with minimal disk operations. As application memory is usually pre-allocated to prevent run-time performance degradation, CPU utilization is the key factor of server power consumption. According to data presented in [21], sleeping or hibernated servers consume minimal power (about 3% of the peak power), while the power consumption of active servers increased quasi-linearly with CPU utilization. Moreover, idle servers consume up to 66% of the peak power to maintain basic system functionalities.

So the power consumption of a certain machine can be expressed as function of CPU utilization as follows

$$P_0 = \begin{cases} S_j^C, & \text{when hibernated} \\ A_j^C + k_j^C u_j^C, & \text{when active} \end{cases} \tag{15}$$

where S_j^C and A_j^C denote the power consumption when a machine is sleeping and idle, respectively; k_j^C denotes a positive linear growth factor; u_j^C denotes the CPU utilization and lies within interval of $[0, 1]$ inclusively.

By neglecting the power consumption of sleeping servers, the total power consumption of server C_j can be drawn as follows

$$P_j^C = \delta_j n_j \left(A_j^C + k_j^C u_j^C\right) \tag{16}$$

where n_j denotes the number of active machines of cloud server C_j and is subject to availability. Assuming that the total number of machines of server C_j is $n_j^{\max}$, then

$$n_j \leq n_j^{\max}, \quad n_j \in \mathbb{N} \tag{17}$$

3.3.3. *System energy summary.* The total power consumption of entire edge-cloud system, which consists of power consumption of both edge and cloud sub-systems, can be drawn as follows

$$P = \sum_{i \in \mathcal{E}} P_i^E + \sum_{j \in \mathcal{C}} P_j^C \tag{18}$$

3.4. **Miscellaneous.** Given the total task arrival rate at edge device E_i as λ_i^S. As each task should either be handled by an edge device onsite or be forwarded to a cloud server, thus

$$\lambda_i^S = \lambda_i^E + \sum_{j \in \mathcal{C}} \lambda_{ij} \tag{19}$$

$$\lambda_j^C = \sum_{i \in \mathcal{E}} \lambda_{ij} \tag{20}$$

Moreover, each edge device has limited computation resource, which leads to limited processing capacity of incoming tasks. Let the task arrival rate limit, or processing capacity, of edge device E_i be $\hat{\lambda}_i^E$ tasks per second. Then

$$0 \leq \lambda_i^E \leq \min\left(\lambda_i^S, \hat{\lambda}_i^E\right) \tag{21}$$

3.5. **Problem formulation.** Consider the problem of achieving the least energy consumption as well as the shortest system delay, which can be expressed as a multi-objective optimization problem as follows

$$\begin{aligned}
&\underset{\lambda_i^E, \lambda_j^C, \lambda_{ij}, u_i^E, u_j^C, \delta_j, n_j}{\textbf{minimize}} && (T, P) && \\
&\textbf{subject to} && \lambda_i^S = \lambda_i^E + \sum_{j \in \mathcal{C}} \lambda_{ij}, && \forall i \in \mathcal{E} \\
& && \lambda_j^C = \sum_{i \in \mathcal{E}} \lambda_{ij}, && \forall j \in \mathcal{C} \\
& && 0 \leq \lambda_i^E \leq \min\left(\lambda_i^S, \hat{\lambda}_i^E\right), && \forall i \in \mathcal{E} \\
& && 0 \leq \lambda_{ij} \leq \lambda_{ij}^{\max}, && \forall i \in \mathcal{E},\ \forall j \in \mathcal{C} \\
& && 0 \leq u_i^E,\ u_j^C \leq 1, && \forall i \in \mathcal{E},\ \forall j \in \mathcal{C} \\
& && \delta_j \in \{0, 1\}, && \forall j \in \mathcal{C} \\
& && n_j \leq n_j^{\max},\ n_j \in \mathbb{N}, && \forall j \in \mathcal{C}
\end{aligned} \tag{22}$$

The decision variables are the task arrival rate λ_i^E at edge device E_i, the task arrival rate λ_j^C at cloud server C_j, the task arrival rate λ_{ij} from edge device E_i to cloud server

C_j, the CPU utilization u_i^E and u_j^C of edge device E_i and cloud server C_j, along with the active state δ_j of cloud server C_j and the number of machines n_j on cloud server C_j. This is a multi-objective minimization optimization. The decision variables are coupled with each other, making edge sub-system and cloud sub-system tangled.

4. Experimental Evaluation.

4.1. Parameter configurations. For simplicity and without loss of generality, we consider an edge-cloud computing system with 5 edge devices and 2 cloud servers. For heterogeneity considerations, each edge device as well as cloud server is set to a different specification. Essential parameters of such a system are given in Table 1 and Table 2. This system setup can be altered or extended to match target system.

Table 1. Simulation parameters

key	value
L_0	$[5, 10]$ MFLO
λ_i^S	$[40, 80, 120, 160, 200]$ s^{-1}
v_i^E	$[540, 640, 800, 840, 960]$ MFLO $\cdot\, \mathrm{s}^{-1}$
A_i^E	$[50, 55, 60, 65, 70]$ W
k_i^E	$[20, 22.55, 25, 27.5, 30]$
v_j^C	$[4000, 5000]$ MFLO $\cdot\, \mathrm{s}^{-1}$
A_j^C	$[60, 70]$ W
k_j^C	$[35, 40]$
$n_j^{\max}$	$[3, 2]$

Table 2. WAN transaction delays (ms)

τ_{ij}	E_1	E_2	E_3	E_4	E_5
C_1	20	25	30	35	40
C_2	40	35	30	25	20

4.2. Individual inspection. In this section, we focus on the differences between edge devices and cloud machines. For instance, by feeding one edge device E_3 and one cloud server C_1 with tasks at the same arrival rate of 40 s^{-1} respectively, we varied the CPU utilization of each individual and observed the service delay as well as the power consumption. All machines of C_1 are activated. Both cloud progressing delay and transmission delay are calculated as cloud delay. The result is shown in Figure 3.

An edge device generally consumed much less energy than a cloud machine did. When it comes to system delay, the utilization of an edge device can effectively affect the system delay, while on the contrary, a cloud server with more than 20% utilization introduced an almost steady system delay. Furthermore, an edge device had to utilize a much more portion of its computing resource to achieve the same service delay as a cloud machine. However, an edge can achieve a lower service delay than a cloud server can get.

It should be noted that an edge device required at least 50% utilization to maintain task queue stable, while a cloud machine only required about 3%. The reason for this is that a cloud machine possesses much more computing power than an edge device, thus can

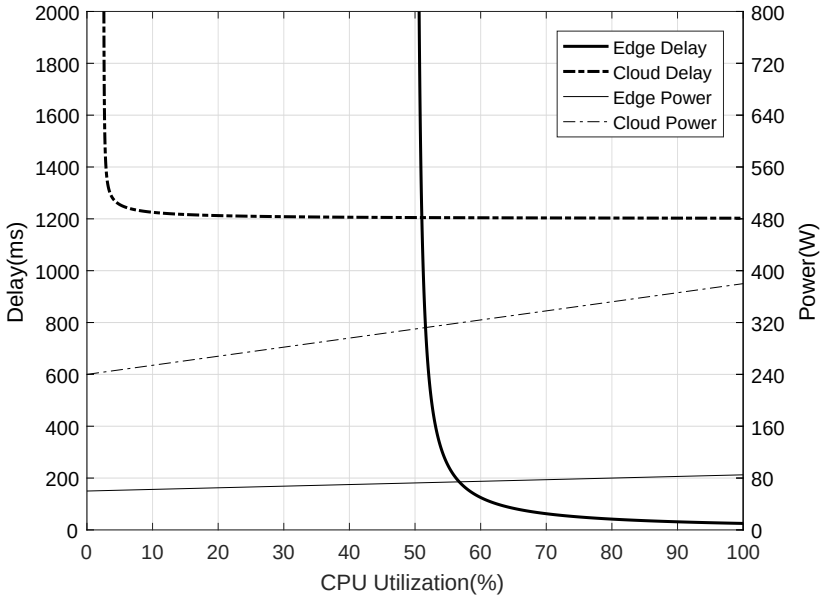

FIGURE 3. Delay and power against CPU utilization

process the same number of tasks by utilizing less computing resource. In other words, multiple edge devices have to be involved to process tasks that a single cloud server can accomplish. In this case, E_3 can process up to 80 tasks per second, while that number is 1600 to C_1. Obviously, 20 E_3s consume much more power than one single C_1 does.

In summary, the edge sub-system can achieve lower system delay, consuming less energy under a low workload and more energy under high workload compared to the cloud sub-system; the cloud sub-system, on the contrary, introduces longer service delay due to network conditions, and consumes more energy under a low workload and less energy under high workload compared to the edge sub-system.

4.3. **Pareto efficiency.** In this multi-objective optimization problem, these two objectives are conflicting, and no single solution exists that simultaneously optimizes each objective. There exist a number of Pareto optimal solutions. A solution is called Pareto optimal, if none of the objectives can be improved without degrading the other objectives. Without additional subjective preferences, all Pareto optimal solutions are considered equally good.

In this paper, we solved the problem with a multi-objective evolutionary algorithm, the implementation of which is publicly accessible as function 'gamultiobj' in the Genetic Algorithm Direct Search Toolbox (GADST) introduced in MATLAB R2007b. It uses a controlled elitist genetic algorithm (a variant of NSGA-II [22]). An elitist GA always favors individuals with better fitness value (rank). A controlled elitist GA also favors individuals that can help increase the diversity of the population even if they have a lower fitness value. It is important to maintain the diversity of population for convergence to an optimal Pareto front. Diversity is maintained by controlling the elite members of the population as the algorithm progresses. The Pareto front result is shown in Figure 5. Results are subject to specifications of edge devices, specifications of cloud servers, network conditions, and workload intensity.

Each circle marks a pair of service delay and power consumption decided by a set of decision variables in Figure 4 and Figure 5. Results marked by circles are shown in Tables 3, 4, 5, and 6.

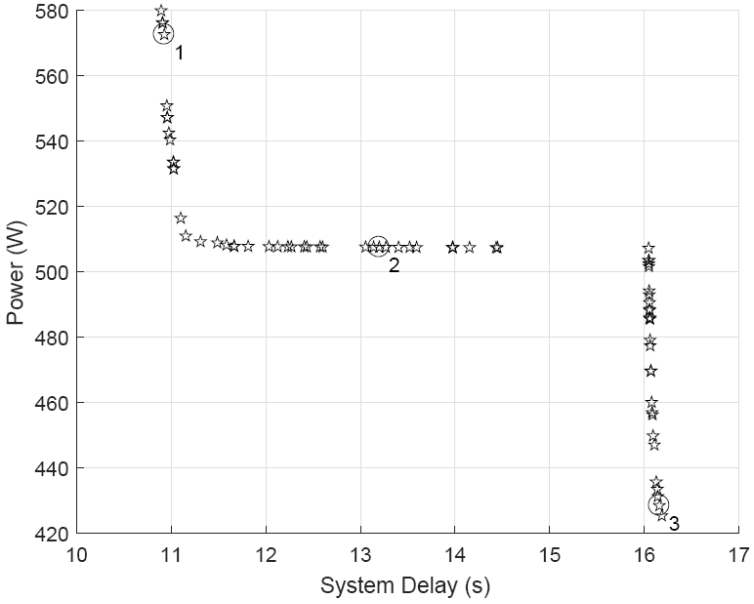

FIGURE 4. Pareto front ($L_0 = 5$)

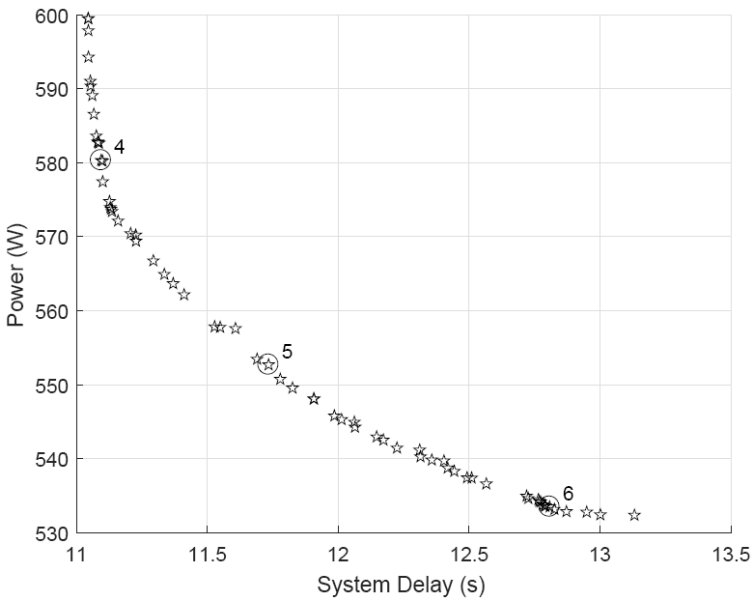

FIGURE 5. Pareto front ($L_0 = 10$)

In Figure 4, lower workload was sent to the system, where average workload size was 5 MFLO. Numerical results show that the workload attended by edge sub-system decreased from Result 1 to Result 3, at the same time, power consumption decreased, and service delay increased. Pareto front shown in Figure 4 possesses segmented characteristic. Result 1 represents a situation in which more work was done by edge sub-system; Result 3 represents a situation in which more work was done by cloud sub-system; Result 2 represents a situation in between. Different measures were taken to decrease power consumption, e.g., slowing down edge devices (significant around Result 1), utilizing more cloud resource (significant around Result 2), and slowing down cloud servers (significant around Result 3), all at the cost of increasing service delay.

TABLE 3. Simulation result – edge subsystem

Group	Result 1		Result 2		Result 3	
i	$\lambda_i^E\,(s^{-1})$	u_i^E	$\lambda_i^E\,(s^{-1})$	u_i^E	$\lambda_i^E\,(s^{-1})$	u_i^E
1	34.12	0.78	20.17	0.43	0.00	0.00
2	34.12	0.69	20.17	0.35	0.00	0.00
3	34.12	0.71	20.17	0.46	0.00	0.00
4	35.00	0.64	20.17	0.28	0.00	0.00
5	34.50	0.80	20.17	0.18	0.00	0.00
Group	Result 4		Result 5		Result 6	
i	$\lambda_i^E\,(s^{-1})$	u_i^E	$\lambda_i^E\,(s^{-1})$	u_i^E	$\lambda_i^E\,(s^{-1})$	u_i^E
1	34.13	0.92	30.89	0.78	25.52	0.71
2	34.68	0.83	31.10	0.69	25.28	0.60
3	34.96	0.85	31.11	0.57	25.58	0.46
4	33.97	0.85	30.84	0.57	25.33	0.46
5	34.13	0.82	31.03	0.50	25.58	0.34

TABLE 4. Simulation result – transmission

	λ_{ij}	E_1	E_2	E_3	E_4	E_5
Result 1	C_1	5.75	5.75	5.75	5.58	5.52
	C_2	0.13	40.13	80.13	119.41	159.91
Result 2	C_1	3.36	3.36	3.36	3.36	3.36
	C_2	16.47	56.47	96.47	136.47	176.47
Result 3	C_1	0.00	0.00	0.00	0.00	0.00
	C_2	40.00	80.00	120.00	160.00	200.00
Result 4	C_1	5.79	5.54	5.37	5.86	5.79
	C_2	0.09	39.78	79.67	120.17	160.09
Result 5	C_1	5.11	5.20	5.05	5.32	5.09
	C_2	3.99	43.70	83.84	123.84	163.88
Result 6	C_1	4.03	4.69	4.09	4.58	4.09
	C_2	10.45	50.03	90.33	130.08	170.33

TABLE 5. Simulation result – cloud subsystem

Group	Result 1			Result 2			Result 3		
j	$\lambda_j^C\,(s^{-1})$	u_j^C	n_j	$\lambda_j^C\,(s^{-1})$	u_j^C	n_j	$\lambda_j^C\,(s^{-1})$	u_j^C	n_j
1	28.36	0.53	1	16.81	0.39	1	0.00	0.00	0
2	399.71	0.83	1	482.35	0.57	1	600.00	0.65	1
Group	Result 4			Result 5			Result 6		
j	$\lambda_j^C\,(s^{-1})$	u_j^C	n_j	$\lambda_j^C\,(s^{-1})$	u_j^C	n_j	$\lambda_j^C\,(s^{-1})$	u_j^C	n_j
1	28.34	0.30	3	25.77	0.20	3	21.49	0.09	3
2	399.78	0.87	2	419.26	0.91	2	451.22	0.93	1

TABLE 6. Simulation result – service delay and power consumption details

Group	Result 1		Result 2		Result 3	
	Delay (s)	Power (W)	Delay (s)	Power (W)	Delay (s)	Power (W)
Edge Subsystem	0.072	390.62	0.203	341.00	0.142	332.42
Transmission	10.851	0	12.974	0	15.994	0
Cloud Subsystem	0.010	181.89	0.023	166.42	0.024	95.95
Total	10.933	572.51	13.200	507.42	16.160	428.37
Group	Result 4		Result 5		Result 6	
	Delay (s)	Power (W)	Delay (s)	Power (W)	Delay (s)	Power (W)
Edge Subsystem	0.203	404.86	0.322	379.28	0.445	363.25
Transmission	10.843	0	11.352	0	12.182	0
Cloud Subsystem	0.048	175.45	0.061	173.40	0.180	170.22
Total	11.091	580.31	11.735	552.68	12.808	533.47

In Figure 5, higher workload was sent to the system, where average workload size was 10 MFLO. Similarly, from Result 4 to Result 6, the workload attended by edge sub-system decreased with decreased power consumption and increased service delay. However, due to high workload input, edge sub-system was no longer capable of handling all the workload. Furthermore, forwarding most workload to cloud sub-system and leaving minimal workload to edge sub-system were not an option, as workload allocation of this kind is not able to decrease power consumption without compromising service delay, or vice versa. As a result, Pareto front shown in Figure 5 possesses no segmented characteristic. The major measure to trade-off between power consumption and service delay was scheduling workload, along with other measures such as adjusting the performance of edge devices and/or cloud servers, powering up or hibernating cloud servers.

According to Table 6, most service time is spent on the transmission route instead of actual task processing procedure. This is exactly the reason for the high latency of a cloud system. By employing more resource from edge sub-system, we got a system with shorter service delay and more power consumption; on the contrary, by employing more cloud resource, power consumption dropped at the cost of the rise of service delay.

A decision maker can choose between these results at his own discretion. For example, if less power consumption is preferred, the upper left part of the results may be selected; if shorter service delay is preferred, the lower right part of the results may be selected. Detailed result selection is subject to additional subjective preferences and thus beyond the scope of this paper.

In general, on the one hand, edge sub-system in such edge-cloud computing system contains considerable edge devices with low individual power consumption, low computing performance and low network latency, which can provide service with low service delay at close range in terms of network topology; cloud sub-system, on the other hand, contains limited number of cloud servers with high individual power consumption, high computing performance and high network latency, which can provide service with higher service delay because of the high network latency. It is worth mentioning that edge sub-system had to utilize a lot more devices than cloud sub-system did to attend to the same amount of workload. As a result, in spite of the low individual power consumption, edge sub-system was less power efficient than cloud sub-system; in spite of the higher computing performance, cloud sub-system was less latency friendly than edge sub-system.

5. **Conclusions.** By establishing a statistical system model and solving the proposed optimization problem, we made it clear that even though service delay and power consumption are conflicting, but can be controlled by setting proper system configuration. Solving the proposed minimization problem will give a better picture of how service delay and power consumption vary against each other. The service provider can select among results according to other details, including SLA, unit electricity price and other expenses, to achieve final designated optimal state.

However, the proposed system model can still be perfected. The proposed system model is built based on quite a few assumptions, including the stable state assumption. As with other statistical models, it takes time for statistical variables to reflect real-time system state. So the proposed model works well with system with stable states, but may produce unexpected results when encountering major workload or network fluctuations. Decision making methodology aggregation along with real-time adjustment functionality should be our step.

Acknowledgment. This work is partially supported by the IACAS Young Elite Researcher Project (Project No. QNYC201715). And the authors would like to thank the anonymous referees for their comments and suggestions. Furthermore, the authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] L. Atzori, A. Iera and G. Morabito, The Internet of Things: A survey, *Computer Networks*, vol.54, no.15, pp.2787-2805, 2010.

[2] F. Bonomi, R. Milito, P. Natarajan and J. Zhu, Fog computing: A platform for Internet of Things and analytics, *Big Data and Internet of Things: A Roadmap for Smart Environments*, vol.546, pp.169-186, 2014.

[3] J. Pan and J. Mcelhannon, Future edge cloud and edge computing for Internet of Things applications, *IEEE Internet of Things Journal*, vol.5, no.1, 2018.

[4] J. A. Stankovic, Research directions for the Internet of Things, *IEEE Internet of Things Journal*, vol.1, no.1, pp.3-9, 2014.

[5] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, Internet of Things for smart cities, *IEEE Internet of Things Journal*, vol.1, no.1, pp.22-32, 2014.

[6] T. Higashino, H. Yamaguchi, A. Hiromori, A. Uchiyama and K. Yasumoto, Edge computing and IoT based research for building safe smart cities resistant to disasters, *IEEE International Conference on Distributed Computing Systems*, pp.1729-1737, 2017.

[7] J. Pan, L. Ma, R. Ravindran and P. Talebifard, Homecloud: An edge cloud framework and testbed for new application delivery, *International Conference on Telecommunications*, pp.1-6, 2016.

[8] H. Wang, J. Gong, Y. Zhuang, H. Shen and J. Lach, Healthedge: Task scheduling for edge computing with health emergency and human behavior consideration in smart homes, *International Conference on Networking, Architecture, and Storage*, pp.1-2, 2017.

[9] S. K. Sharma and X. Wang, Live data analytics with collaborative edge and cloud processing in wireless IoT networks, *IEEE Access*, vol.5, pp.4621-4635, 2017.

[10] S. H. Park, O. Simeone and S. S. Shitz, Joint optimization of cloud and edge processing for fog radio access networks, *IEEE Trans. Wireless Communications*, vol.15, no.11, pp.7621-7632, 2016.

[11] T. G. Rodrigues, K. Suto, H. Nishiyama and N. Kato, Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control, *IEEE Trans. Computers*, vol.66, no.5, pp.810-819, 2017.

[12] L. I. Hongjia, C. I. Song, T. Lin, H. Wang and Z. Wang, Research on green index based energy consumption modeling method for cloud computing system, *Journal of Network New Media*, 2013.

[13] X. Tao, K. Ota, M. Dong, H. Qi and K. Li, Performance guaranteed computation offloading for mobile-edge cloud computing, *IEEE Wireless Communications Letters*, 2017.

[14] F. Ahmad and T. N. Vijaykumar, Joint optimization of idle and cooling power in data centers while maintaining response time, *ACM Sigplan Notices*, vol.45, pp.243-256, 2010.

[15] S. Srinivasan and V. Getov, Navigating the cloud computing landscape – Technologies, services, and adopters, *Computer*, vol.44, no.3, pp.22-23, 2011.
[16] C. Stergiou, K. E. Psannis, B. G. Kim and B. Gupta, Secure integration of IoT and cloud computing, *Future Generation Computer Systems*, 2016.
[17] D. C. Marinescu, *Cloud Computing: Theory and Practice*, Morgan Kaufmann, 2017.
[18] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, Edge computing: Vision and challenges, *IEEE Internet of Things Journal*, vol.3, no.5, pp.637-646, 2016.
[19] S. K. Datta, C. Bonnet and J. Haerri, Fog computing architecture to enable consumer centric Internet of Things services, *IEEE International Symposium on Consumer Electronics (ISCE)*, pp.1-2, 2015.
[20] P. G. Harrison and N. M. Patel, *Performance Modelling of Communication Networks and Computer Architectures*, 1993.
[21] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao and F. Zhao, Energy-aware server provisioning and load dispatching for connection-intensive Internet services, *NSDI*, vol.8, pp.337-350, 2008.
[22] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA, 2001.