Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption

Ruilong Deng, *Member, IEEE*, Rongxing Lu, *Senior Member, IEEE*, Chengzhe Lai, *Member, IEEE*, Tom H. Luan, *Member, IEEE*, and Hao Liang, *Member, IEEE*

***Abstract*—Mobile users typically have high demand on localized and location-based information services. To always retrieve the localized data from the remote cloud, however, tends to be inefficient, which motivates fog computing. The fog computing, also known as edge computing, extends cloud computing by deploying localized computing facilities at the premise of users, which prestores cloud data and distributes to mobile users with fast-rate local connections. As such, fog computing introduces an intermediate fog layer between mobile users and cloud, and complements cloud computing toward low-latency high-rate services to mobile users. In this fundamental framework, it is important to study the interplay and cooperation between the edge (fog) and the core (cloud). In this paper, the tradeoff between power consumption and transmission delay in the fog-cloud computing system is investigated. We formulate a workload allocation problem which suggests the optimal workload allocations between fog and cloud toward the minimal power consumption with the constrained service delay. The problem is then tackled using an approximate approach by decomposing the primal problem into three subproblems of corresponding subsystems, which can be, respectively, solved. Finally, based on simulations and numerical results, we show that by sacrificing modest computation resources to save communication bandwidth and reduce transmission latency, fog computing can significantly improve the performance of cloud computing.**

***Index Terms*—Cloud computing, fog computing, optimization, power consumption-delay tradeoff, workload allocation.**

I. Introduction

THE INTERNET has shifted to the cloud-based structure. As reported in Cisco Cloud Index (2013–2018), since 2008, most Internet traffic has originated or terminated in a data center. By 2016, it is predicted that nearly two-thirds of total workloads in traditional IT space will be processed in the cloud. However, with the surging mobile traffic generated in recent years, the transmission of the extraordinarily huge-volume data to the cloud has not only posed a heavy burden on communication bandwidth, but also resulted in unbearable transmission latency and degraded service to end users [2]–[4]. In addition to real-time interaction and low latency, with mobile users and traffic becoming dominant nowadays, the support of mobility and geo-distribution is also critical [5]–[7]. Therefore, with cloud becoming the overarching approach for centralized information storage, retrieval, and management, and mobile devices becoming the major destination of information, the successful integration of cloud computing and mobile applications therefore represents an important task.

To address the above challenges, Cisco has delivered the concept of fog computing in 2014, which aims to process in part workload and services locally on fog devices (such as hardened routers, switches, IP video cameras, etc.), rather than being transmitted to the cloud [8]. This is by introducing a new intermediate fog layer between mobile users and cloud as shown in Fig. 1. The fog layer is composed of geo-distributed fog servers which are deployed at the edge of networks, e.g., parks, bus terminals, shopping centers, etc. Each fog server is a highly virtualized computing system, similar to a light-weight cloud server, and is equipped with the on-board large-volume data storage, compute, and wireless communication facility. The fog servers bridges the mobile users and cloud. On one hand, fog servers directly communicate with the mobile users through single-hop wireless connections using the off-the-shelf wireless interfaces, such as WiFi, Bluetooth, etc. With the on-board compute facility and precached contents, they can independently provide predefined service applications to mobile users without assistances

Manuscript received January 17, 2016; revised May 4, 2016; accepted May 4, 2016. Date of publication May 10, 2016; date of current version January 10, 2017. This work was supported in part by the Cybersecurity Research Program at the School of Electrical and Electronic Engineering, Nanyang Technological University, in part by the Alberta Innovates Technology Futures Post-Doctoral Fellowship, in part by the International Science and Technology Cooperation and Exchange Plan in Shaanxi Province, China under Grant 2015KW-010, in part by the National Natural Science Foundation of China Research under Grant 61502386, and in part by the Natural Science and Engineering Research Council of Canada. The work of R. Lu was supported by the Nanyang Technological University's College of Engineering Proposal Preparatory Grant and MOE Tier 1 under Grant M4011450. An earlier version of this paper was presented at the IEEE ICC 2015 [1]. *(Corresponding author: Rongxing Lu.)*

R. Deng is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, and also with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: ruilong@ualberta.ca).

R. Lu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: rxlu@ntu.edu.sg).

C. Lai is with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (e-mail: lcz.xupt@gmail.com).

T. H. Luan is with the School of Information Technology, Deakin University, Burwood Melbourne, Vic. 3125, Australia (e-mail: tom.luan@deakin.edu.au).

H. Liang is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: hao2@ualberta.ca).

Digital Object Identifier 10.1109/JIOT.2016.2565516

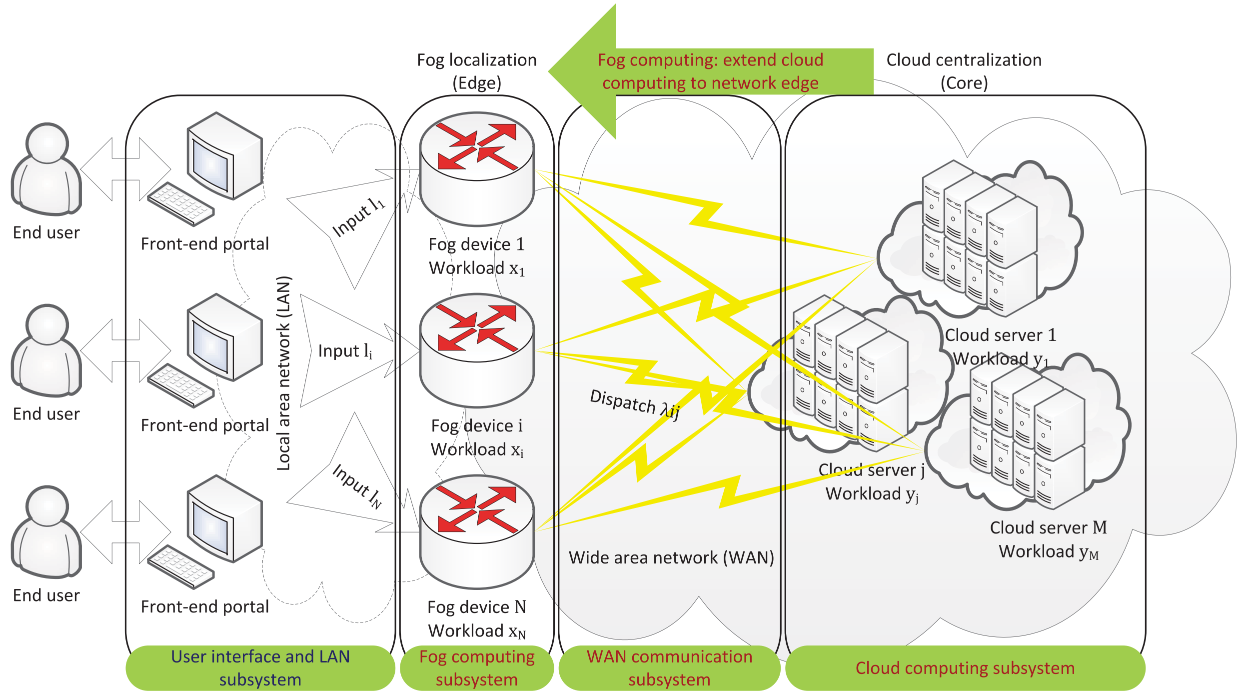


Fig. 1. Overall architecture of a fog-cloud computing system with four subsystems and their interconnections/interactions.

from cloud or Internet. On the other hand, the fog servers can be connected to the cloud so as to leverage the rich functions and application tools of the cloud. Therefore, "the fog is a cloud close to the ground." Fog computing is not to substitute but to complement cloud computing, in order to ease bandwidth burden and reduce transmission latency. In particular, the fog can support and facilitate applications that do not fit well with the cloud: 1) applications that require very low and predictable latency, such as online gaming and video conferencing; 2) geographically distributed applications such as pipeline monitoring and sensor networks; 3) fast mobile applications such as smart connected vehicles; and 4) large-scale distributed control systems such as smart energy distribution and smart traffic lights [9]–[12].

While the fog provides localization, i.e., enabling the real-time interaction and low latency at the network edge, the cloud provides centralization, the integration of which inspires applications that require the interplay and cooperation between the edge (fog) and the core (cloud), particularly for big data and Internet of Things [13]–[16]. From this perspective, we showcase some specific use cases of fog-cloud computing [17], [18]. For example, fog devices deployed inside a multifloor shopping center can deliver delay-sensitive services including indoor navigation and flyers distribution to mobile users through WiFi, and forward delay-tolerant requests such as feedback statistical analysis to cloud servers for centralized processing. Fog devices deployed at a park lot can provide the precached information including park maps and local accommodations, and, by connecting to cloud servers, send timely alerts and notifications to drivers. Fog devices deployed inside an interstate bus can deliver onboard video streaming and social networking services to passengers using WiFi. The onboard fog devices connect to cloud servers through cellular networks to refresh the precached contents and update application services, and also report users' data such as their feedbacks to cloud servers for centralized processing.

In this paper, we consider a fog-cloud computing system. On one hand, with the huge-volume and ever-increasing service requests, the power consumption on powering up (and cooling) cloud servers is soaring. It is thus important and desirable to consider the energy management in the fog-cloud computing system [19], [20]. On the other hand, it is equally crucial to guarantee the quality of service (e.g., latency requirements) of end users. The reason is that the unbearable response latency leads to revenue loss of service providers since end users will subscribe to other vendors with better service [21]. To this end, we systematically investigate the fundamental tradeoff between the power consumption and delay in the fog-cloud computing system.

In this paper, first, we model the power consumption function and delay function of each part of the fog-cloud computing system, and formulate the workload allocation problem. Then, we develop an approximate approach to solve the primal problem (PP) through decomposition, and formulate three subproblems (SPs) of three corresponding subsystems. These SPs can be, respectively, solved via existing optimization techniques. Finally, based on simulations and numerical results, we show that fog computing can significantly improve the performance of cloud computing in terms of reducing communication latency. To the best of our knowledge, this is an early effort toward providing a systematic framework of computation and communication co-design in the fog-cloud computing system. We hope that this pioneering work can throw light on how the fog can extend and complement the cloud. Specifically, the original contributions of this paper are summarized as follows.

1) We cast a mathematical framework to investigate the power consumption-delay tradeoff problem by workload allocation in the fog-cloud computing system.
2) We develop an approximate approach to decompose the PP into three SPs of corresponding subsystems, and solve them, respectively.
3) We conduct extensive simulations to demonstrate that the fog can significantly complement the cloud with much reduced communication latency.

Beyond the low latency characteristic as addressed in this paper, the possible advantages of a fog architecture include mobility support, geo-distribution, and location/context awareness [22], [23]. Not only can the geo-distributed fog device infer its own location, but also the fog device can track end users' devices to support mobility, which would be a game changing factor for location-based services and applications. Besides, the geo-distribution can also provide rich network context information, such as the local network condition, traffic statistics, and client status information, which can be used by fog applications to offer context-aware optimization.

The remainder of this paper is organized as follows. The related works are introduced in Section II. We describe the model of the fog-cloud computing system and formulate the power consumption-delay tradeoff problem in Section III. In Section IV, we approximately decompose the PP into three SPs of corresponding subsystems. Simulations are conducted in Section V with numerical results, and concluding remarks are drawn in Section VI with future work.

II. Related Works

Cloud computing, a kind of Internet-based paradigm, refers to both applications delivered as services over the Internet and the hardware and software in the data centers that provide these services [24], [25]. The research on cloud computing has attracted great attention with a large quantity of literatures. For example, Armbrust *et al.* [26] quantified comparisons between cloud and conventional computing, and identify the top technical and nontechnical obstacles and opportunities of cloud computing. The emergence of cloud computing has established a trend toward building massive, energy-hungry, and geographically distributed Internet data centers as cloud servers. Due to their enormous energy consumption, Rao *et al.* [19], [21] investigated how to coordinate the collection of data centers so as to minimize the electricity expense while maintaining the quality of the cloud computing service. This paper extends from the existing related papers on cloud computing to a newly emerged paradigm named fog computing. However, the transition is not trivial, since fog is quite different from cloud in terms of location, distribution, and computing capability.

On the other hand, fog computing, characterized by extending cloud computing to the network edge, has become a buzzword today [22], [23]. With similar frameworks such as cloudlet, follow me cloud, and edge computing, fog computing receives considerable attention recently. For example, Bonomi *et al.* [10] defined the characteristics of fog computing which make it an appropriate platform for a number of critical services and applications in Internet of Things and big data analytics. Stojmenovic and Wen [27] and Stojmenovic [28] reviewed a handful of literatures that expand the applications of fog computing to a series of real scenarios, such as smart grid, vehicular networks, cyber-physical systems, etc. Security and privacy issues are further disclosed according to current fog computing paradigm. Since the fog is not to substitute but to complement the cloud, it is worthy of studying the interaction and cooperation between them. However, existing methodologies need to be changed to accommodate the bi-layer fog-cloud model. To our knowledge, a systematic framework of computation and communication co-design does not seem to be studied so far in the context of fog-cloud. This paper serves as a starting point to address this issue, in which we study the tradeoff between power consumption and delay in the fog-cloud computing system.

III. System Model and Problem Formulation

We illustrate an overall architecture of the fog-cloud computing system in Fig. 1, which has been divided into four subsystems. The front-end portals act as user interfaces that receive service requests from end users. These requests are separately input to a set $\mathcal{N}$ of fog devices through a local area network (LAN). Since fog devices are generally located in the vicinity of end users, thus the LAN communication delay could be omitted [compared to wide area network (WAN)]. Fog computing can process some of the delay-sensitive requests and forward others to cloud computing [29]. There is a set $\mathcal{M}$ of cloud servers, each of which hosts a number of homogeneous computing machines. The unprocessed requests are dispatched from each fog device to each cloud server through a WAN. Since WAN covers a large geographical area from the edge throughout to the core, the communication delay and constrained bandwidth should be taken into account. In the following, we mainly consider the power consumption and computation/communication delay of the latter three subsystems (i.e., fog computing, WAN communication, and cloud computing). Some important notations used in this paper are summarized in Table I. In the rest of this paper, we also use the following mathematical notations from linear algebra: 1) $\boldsymbol{x}^T$ denotes the transpose of $\boldsymbol{x}$; 2) $\mathbf{1}$ denotes the all-ones vector; and 3) $\mathbf{0}$ denotes the all-zeros vector.

A. System Model

1) Power Consumption of Fog Device: For the fog device i, the computation power consumption can be modeled by a function of the computation amount x_i, which is a monotonic increasing and strictly convex function. The piece-wise linear function and quadratic function are two alternatives [30]. In fact, the fog computing devices can accommodate any form of power consumption functions as long as they satisfy the following two properties: 1) the computation power consumption always increases as the computation amount increases and 2) the marginal power consumption for each fog device is increasing. For simplicity but without loss of generality, we can express the power consumption P_i^{fog} of the fog device i

TABLE I
SUMMARY OF NOTATIONS

Symbol	Definition	Unit[a]
$i, N, \mathcal{N}$	index, number, set of fog devices	n/a
$j, M, \mathcal{M}$	index, number, set of cloud servers	n/a
l_i	traffic arrival rate to fog device i	$\sharp$(requests)$/s$
x_i	workload assigned to fog device i	$\sharp$(requests)$/s$
λ_{ij}	traffic rate dispatched from fog device i to cloud server j	$\sharp$(requests)$/s$
y_j	workload assigned to cloud server j	$\sharp$(requests)$/s$
L	total input from all front-end portals	$\sharp$(requests)$/s$
X	workload allocated for fog computing	$\sharp$(requests)$/s$
Y	workload allocated for cloud computing	$\sharp$(requests)$/s$
P	power consumption	unit power
D	delay	unit time
$\overline{D}$	system delay constraint	unit time
v_i	service rate at fog device i	$\sharp$(requests)$/s$
f_j	machine CPU frequency at cloud server j	$\sharp$(cycles)$/s$
σ_j	binary: on/off state of cloud server j	n/a
n_j	integer: machine number at cloud server j	n/a
d_{ij}	communication delay from fog device i to cloud server j	unit time
η_i	weighting factor at fog device i	n/a
$\overline{D}_j$	delay threshold at cloud server j	unit time

[a]The unit of a quantity may be omitted in the rest of the paper if it is specified here.

by the following function of the computation amount x_i:

$$P_i^{\text{fog}} \triangleq a_i x_i^2 + b_i x_i + c_i$$

where $a_i > 0$ and $b_i, c_i \geq 0$ are predetermined parameters.

2) Computation Delay of Fog Device: Assuming a queueing system, for the fog device i with the traffic arrival rate x_i and service rate v_i, the computation delay (waiting time plus service time) D_i^{fog} is

$$D_i^{\text{fog}} \triangleq \frac{1}{v_i - x_i}.$$

3) Power Consumption of Cloud Server: Each cloud server hosts a number of homogeneous computing machines. The configurations (e.g., CPU frequency) are assumed to be equal for all machines at the same server. Thus, each machine at the same server has the same power consumption profile. We approximate the power consumption value of each machine at the cloud server j by a function of the machine CPU frequency $f_j : A_j f_j^p + B_j$, where A_j and B_j are positive constants, and p varies from 2.5 to 3 [21].

When the allocated workload increases, more cloud servers are powered on; while when it decreases, the excess servers are turned off for energy saving [31]. Let a binary variable σ_j denote the on/off state of the cloud server j, where 1 means that the server is on and 0 means off. Besides, let an integer variable n_j denote the number of turned-on machines at the cloud server j. Thus, the power consumption P_j^{cloud} of the cloud server j can be obtained by multiplying the on/off state, the on-state machine number, and each machine power consumption value [19]

$$P_j^{\text{cloud}} \triangleq \sigma_j n_j \left(A_j f_j^p + B_j\right).$$

4) Computation Delay of Cloud Server: The M/M/n queueing (or Erlang-C) model is employed to characterize each cloud server. In this model, the computation delay (waiting time plus service time) is $[((C(n, \lambda/\mu))/(n\mu - \lambda)) + (1/\mu)]$, where n is the number of machines, λ and μ are the traffic arrival rate and service rate, respectively, and $C(n, \lambda/\mu)$ is the Erlang's C formula [32, Ch. 2]. At the cloud server j, assume that each machine has the same service rate μ_j. We can generally convert μ_j to f_j by $\mu_j = f_j/K$, where K is in terms of $\sharp$(cycles)/request.

From the above, for the cloud server j with the on/off state σ_j and n_j turned-on machines, when each machine has the traffic arrival rate y_j and service rate f_j/K, respectively, the computation delay D_j^{cloud} is given by

$$D_j^{\text{cloud}} \triangleq \sigma_j \left[\frac{C\left(n_j, y_j K/f_j\right)}{n_j f_j/K - y_j} + \frac{K}{f_j}\right].$$

5) Communication Delay for Dispatch: Let d_{ij} denote the delay of the WAN transmission path from the fog device i to the cloud server j. Thus, when the traffic rate dispatched from the fog device i to the cloud server j is λ_{ij}, the corresponding communication delay D_{ij}^{comm} is

$$D_{ij}^{\text{comm}} \triangleq d_{ij}\lambda_{ij}.$$

B. Constraints

1) Workload Balance Constraint: Let L denote the total request input from all front-end portals. The traffic arrival rate from all front-end portals to the fog device i is denoted by l_i. Thus, we have

$$L \triangleq \sum_{i \in \mathcal{N}} l_i.$$

Besides, let X and Y denote the workload allocated for fog computing and cloud computing, respectively. Then, we have

$$\begin{cases} X \triangleq \sum_{i \in \mathcal{N}} x_i \\ Y \triangleq \sum_{j \in \mathcal{M}} y_j. \end{cases}$$

We describe the workload balance constraint on the traffic rate dispatched from each fog device to each cloud server. The end-user requests are either handled by a fog device, or forwarded to a cloud server to be processed. The corresponding relationships between the workload and traffic rate are listed as follows.

1) Workload balance constraint for each fog device

$$l_i - x_i = \sum_{j \in \mathcal{M}} \lambda_{ij} \qquad \forall i \in \mathcal{N}. \tag{1}$$

2) Workload balance constraint for each cloud server

$$\sum_{i \in \mathcal{N}} \lambda_{ij} = y_j \qquad \forall j \in \mathcal{M}. \tag{2}$$

from 1) and 2) we can easily obtain 3) workload balance constraint for the holistic fog-cloud computing system

$$L = X + Y.$$

2) Fog Device Constraint: For the fog device i, there exists a limit on the processing ability due to physical constraints. Let $x_i^{\max}$ denote the computation capacity of the fog device i. In addition, the workload x_i assigned to the fog device i should be no more than the traffic arrival rate l_i to that device. From the above, we have

$$0 \leq x_i \leq \min\left\{x_i^{\max}, l_i\right\} \qquad \forall i \in \mathcal{N}. \tag{3}$$

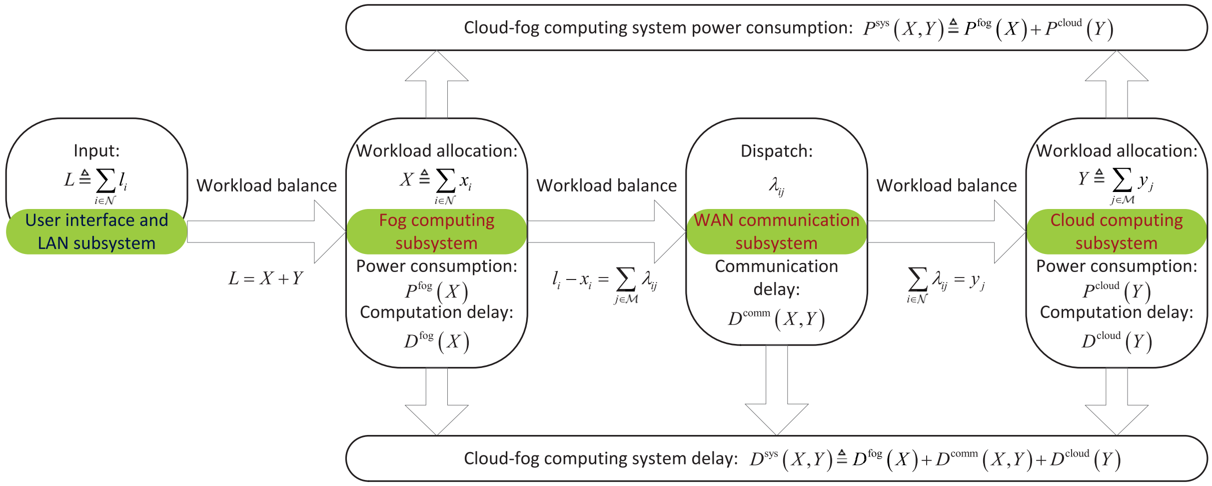


Fig. 2. Overall framework of power consumption-delay tradeoff by workload allocation in a fog-cloud computing system.

3) Cloud Server Constraint: For the cloud server j, first, we have

$$y_j \geq 0 \qquad \forall j \in \mathcal{M}. \tag{4}$$

Besides, there exists a limit on the computation rate of each machine due to physical constraints. Let $f_j^{\min}$ and $f_j^{\max}$ denote the lower and upper bound on the machine CPU frequency, respectively

$$f_j^{\min} \leq f_j \leq f_j^{\max} \qquad \forall j \in \mathcal{M}. \tag{5}$$

In addition, for the cloud server j, the number of machines n_j has an upper bound $n_j^{\max}$. Thus, for the integer variable n_j, we have

$$n_j \in \left\{0, 1, 2, \ldots, n_j^{\max}\right\} \qquad \forall j \in \mathcal{M}. \tag{6}$$

Finally, the binary variable σ_j denote the on/off state of the cloud server j. When σ_j equals 1, it means that the cloud server j is on; when σ_j equals 0, it means that the cloud server j is off, and meanwhile the number of on-state machines equals 0. Thus, we have

$$\sigma_j \in \{0, 1\} \qquad \forall j \in \mathcal{M}. \tag{7}$$

4) WAN Communication Bandwidth Constraint: For simplicity but without loss of generality, the traffic rate λ_{ij} is assumed to be dispatched from the fog device i to the cloud server j through one transmission path. Furthermore, these transmission paths do not overlap with each other. There is a limitation $\lambda_{ij}^{\max}$ on the bandwidth capacity of each path. Thus, the bandwidth constraint of the WAN communication is

$$0 \leq \lambda_{ij} \leq \lambda_{ij}^{\max} \qquad \forall i \in \mathcal{N}; \quad \forall j \in \mathcal{M}. \tag{8}$$

C. Problem Formulation

Towards the power consumption-delay tradeoff in fog-cloud computing, on one hand, it is important and desirable to minimize the aggregated power consumption of all fog devices and cloud servers. The power consumption function of the fog-cloud computing system is defined as

$$P^{sys} \triangleq \sum_{i\in\mathcal{N}} P_i^{fog} + \sum_{j\in\mathcal{M}} P_j^{cloud}.$$

On the other hand, it is equally crucial to guarantee the quality of service (e.g., latency requirements) of end users. The end-user experienced delay consists of the computation (including queueing) delay and communication delay. Therefore, the delay function of the fog-cloud computing system is defined as

$$D^{sys} \triangleq \sum_{i\in\mathcal{N}} D_i^{fog} + \sum_{j\in\mathcal{M}} D_j^{cloud} + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}} D_{ij}^{comm}.$$

We consider the problem of minimizing the power consumption of the fog-cloud computing system while guaranteeing the required delay constraint $\overline{D}$ for end users. That is, we have the PP

$$\min_{x_i, y_j, \lambda_{ij}, f_j, n_j, \sigma_j} \quad P^{sys}$$
$$\text{s.t.} \quad \begin{cases} D^{sys} \leq \overline{D} \\ (1)-(8). \end{cases}$$

The decision variables are the workload x_i assigned to the fog device i, the workload y_j assigned to the cloud server j, the traffic rate λ_{ij} dispatched from the fog device i to the cloud server j, as well as the machine CPU frequency f_j, the machine number n_j, and the on/off state σ_j at the cloud server j. The objective of workload allocation in the fog-cloud computing system is to tradeoff between: 1) the system power consumption and 2) the end-user experienced delay.

IV. Decomposition and Solution

Note that in PP, the decision variables come from different subsystems and are tightly coupled with each other, which makes the relationship between the workload allocation and the power consumption-delay tradeoff not clear. To address this issue, we develop an approximate approach to decompose PP into three SPs of corresponding subsystems, which can be, respectively, solved via existing optimization techniques. We illustrate the decomposition and each SP/subsystem interactions in Fig. 2, which provides an overall framework of power consumption-delay tradeoff by workload allocation in the fog-cloud computing system.

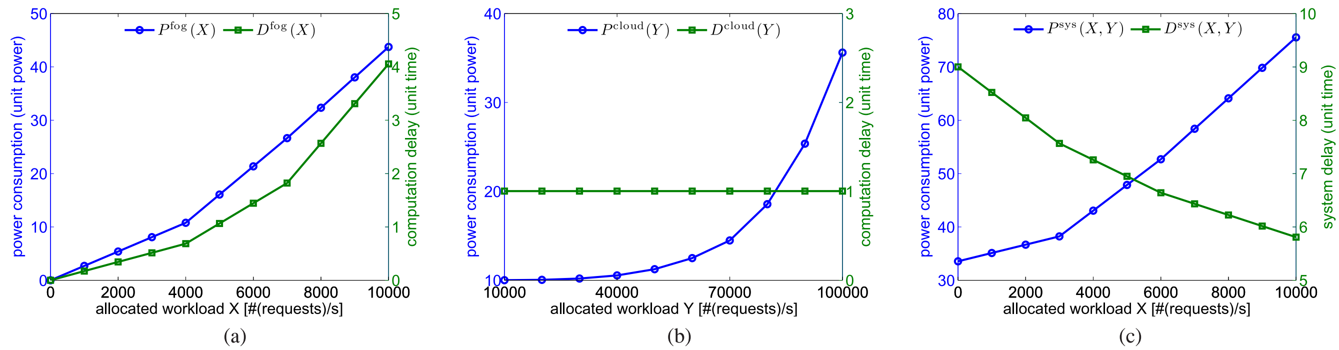


Fig. 3. Illustration of power consumption-delay tradeoff by workload allocation in a fog-cloud computing system. (a) Fog computing subsystem. (b) Cloud computing subsystem. (c) Fog-cloud computing system.

A. Power Consumption-Delay Tradeoff for Fog Computing

We consider to tradeoff between the power consumption and computation delay in the fog computing subsystem. That is, we have the the SP1

$$\min_{x_i} \sum_{i\in\mathcal{N}} \left(a_i x_i^2 + b_i x_i + c_i + \frac{\eta_i}{v_i - x_i}\right)$$
$$\text{s.t.} \begin{cases} \sum_{i\in\mathcal{N}} x_i = X \\ (3) \end{cases}$$

where the adjustable parameter η_i is a weighting factor to tradeoff between the power consumption and computation delay at the fog device i.

Given the workload X allocated for the fog computing subsystem, SP1 is a convex problem with linear constraints. This problem can be easily solved using convex optimization techniques such as interior-point methods [33]–[35, Ch. 11]. After we obtain the optimal workload x_i^* assigned to the fog device i, we can calculate the power consumption and computation delay in the fog computing subsystem, respectively, as

$$\begin{cases} P^{\rm fog}(X) = \sum_{i\in\mathcal{N}} \left[a_i\left(x_i^*\right)^2 + b_i x_i^* + c_i\right] \\ D^{\rm fog}(X) = \sum_{i\in\mathcal{N}} \frac{1}{v_i - x_i^*}. \end{cases}$$

B. Power Consumption-Delay Tradeoff for Cloud Computing

At the cloud server j, for the delay-sensitive requests, their response delay should be bounded by a certain threshold that is specified as the service level agreement, since the agreement violation would result in loss of business revenue. We assume that the response delay should be smaller than an adjustable parameter $\overline{D}_j$, which can be regarded as the delay threshold that identifies the revenue/penalty region at the cloud server j

$$D_j^{\rm cloud} \le \overline{D}_j.$$

We consider to tradeoff between the power consumption and computation delay in the cloud computing subsystem. That is, we have the the SP2

$$\min_{y_j, f_j, n_j, \sigma_j} \sum_{j\in\mathcal{M}} \sigma_j n_j \left(A_j f_j^p + B_j\right)$$
$$\text{s.t.} \begin{cases} \sum_{j\in\mathcal{M}} y_j = Y \\ D_j^{\rm cloud} \le \overline{D}_j \qquad \forall j \in \mathcal{M} \\ (4)-(7). \end{cases}$$

Given the workload Y allocated for the cloud computing subsystem, SP2 is a mixed integer nonlinear programming (MINLP) problem, which is generally difficult to tackle. Since the generalized Benders decomposition (GBD) is an effective method to solve this problem with guaranteed optimality, we design the GBD algorithm in Appendix A [36]–[38, Ch. 13]. After we obtain the optimal workload y_j^* assigned to the cloud server j and the optimal solution f_j^*, n_j^*, and σ_j^*, we can calculate the power consumption and computation delay in the cloud computing subsystem, respectively, as

$$\begin{cases} P^{\rm cloud}(Y) = \sum_{j\in\mathcal{M}} \sigma_j^* n_j^* \left[A_j\left(f_j^*\right)^p + B_j\right] \\ D^{\rm cloud}(Y) = \sum_{j\in\mathcal{M}} D_j^{\rm cloud*} = \sum_{j\in\mathcal{M}} \sigma_j^* \overline{D}_j. \end{cases}$$

C. Communication Delay Minimization for Dispatch

We consider the traffic dispatch rate λ_{ij} to minimize the communication delay in the WAN subsystem. That is, we have the SP3

$$\min_{\lambda_{ij}} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{M}} d_{ij}\lambda_{ij}$$
$$\text{s.t.} \;\; (1),\; (2),\; \text{and } (8).$$

From Sections IV-A and IV-B, given the workload X allocated for fog computing and Y for cloud computing, we can obtain the optimal workload x_i^* assigned to the fog device i and y_j^* assigned to the cloud server j. Given x_i^* and y_j^*, SP3 is regarded as an assignment problem. Since this problem can be efficiently solved using the Hungarian method in polynomial time, we design the Hungarian algorithm in Appendix B [39]. After we obtain the optimal traffic rate λ_{ij}^* dispatched from the fog device i to the cloud server j, we can calculate the communication delay in the WAN subsystem as

$$D^{\rm comm}(X, Y) = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{M}} d_{ij}\lambda_{ij}^*.$$

TABLE II
PARAMETER SETUP

Parameter	Value	Parameter	Value
l_i	$[3\ 1.5\ 1.5\ 2\ 2]\times10^4$	$f_j^{\min}$	1.0
A_j	[3.206 4.485 2.370]	$f_j^{\max}$	[3.4 2.4 3.0]
B_j	[68 53 70]	$n_j^{\max}$	$[3\ 6\ 2.5]\times10^4$
p, K	3, 1	$\overline{D}_j$	unit time

D. Putting it All Together

Based on the above decomposition and the solution to three SPs, on one hand, the power consumption function of the fog-cloud computing system is rewritten as

$$P^{\text{sys}}(X, Y) \triangleq P^{\text{fog}}(X) + P^{\text{cloud}}(Y)$$

which means that the system power consumption comes from the fog devices and cloud servers. On the other hand, the delay function of the fog-cloud computing system is rewritten as

$$D^{\text{sys}}(X, Y) \triangleq D^{\text{fog}}(X) + D^{\text{cloud}}(Y) + D^{\text{comm}}(X, Y)$$

which means that the system delay comes from the computation delay of the fog devices and cloud servers, as well as the communication delay of the WAN.

After solving the above three SPs, we can approximately solve PP by considering the following approximate problem named PP-approx:

$$\min_{X,Y} \ P^{\text{sys}}(X, Y)$$
$$\text{s.t.} \ \begin{cases} D^{\text{sys}}(X, Y) \le \overline{D} \\ X + Y = L \end{cases}$$

which can be iteratively solved. The approximation ratio is dependent on the choice of two adjustable parameters η_i and $\overline{D}_j$. If these parameters could be chosen appropriately, then the solution to PP-approx would be the optimal solution to PP. How to evaluate the approximation ratio of the proposed decomposition is left as our future work.

V. NUMERICAL RESULTS

Simulation results are presented in this section to validate the power consumption-delay tradeoff by workload allocation to fog computing and cloud computing. For simplicity but without loss of generality, we consider the scenario with five fog devices and three cloud servers (Internet data centers) in the fog-cloud computing system. It can be extended to more fog devices and more cloud servers, with the similar results. Some important parameters used in the simulation are summarized in Table II, referring to [19] and [21]. The following results are obtained by MATLAB.

First, we vary the workload X allocated for fog computing from 0 to 10^4, to evaluate how they affect the power consumption $P^{\text{fog}}(X)$ and computation delay $D^{\text{fog}}(X)$ in the subsystem. Under different values of X, we solve SP1 and obtain the optimal workload x_i^* assigned to the fog device i. Based on this we calculate $P^{\text{fog}}(X)$ and $D^{\text{fog}}(X)$, and draw their curves in Fig. 3(a). It is seen that both power consumption and computation delay increase with the workload allocated for fog computing.

Then, we vary the workload Y allocated for cloud computing from 10^4 to 10^5, to evaluate how they affect the power consumption $P^{\text{cloud}}(Y)$ and computation delay $D^{\text{cloud}}(Y)$ in the subsystem. Under different values of Y, we solve SP2 and obtain the optimal workload y_j^* assigned to the cloud server j. Based on this we calculate $P^{\text{cloud}}(Y)$ and $D^{\text{cloud}}(Y)$, and draw their curves in Fig. 3(b). The result shows that the computation delay stays steady while the power consumption increases with the workload allocated for cloud computing.

Finally, based on the above x_i^* and y_j^*, we further solve SP3 and obtain the communication delay $D^{\text{comm}}(X, Y)$ in the WAN subsystem. Based on these we calculate the system power consumption $P^{\text{sys}}(X, Y)$ and delay $D^{\text{sys}}(X, Y)$, and draw their curves in Fig. 3(c). From the numerical results, we note that the power consumption of fog devices dominates the system power consumption, while the communication delay of the WAN dominates the system delay. Therefore, when the fog workload is low, the fog power consumption is low and so is the system power consumption, while the WAN communication delay is high and so is the system delay, and vice versa. The figure illustrates that, when some of workload is allocated for fog computing, the system delay decreases while the system power consumption increases. This is because in the fog-cloud computing system, cloud computing is more powerful and energy-efficient than fog computing; while the fog, with the advantage of physical proximity to end users, can sacrifice modest computation resources to save WAN bandwidth and reduce communication latency, in such a way to significantly improve the performance of the cloud.

VI. CONCLUSION

In this paper, we have introduced the vision of fog computing, a newly emerged paradigm that extends cloud computing to the edge of the network. Concretely, we develop a systematic framework to investigate the power consumption-delay tradeoff issue in the fog-cloud computing system. We formulate the workload allocation problem and approximately decompose the PP into three SPs, which can be, respectively, solved within corresponding subsystems. Simulation and numerical results are presented to show the fog's complement to the cloud. We hope that this pioneering work can provide guidance on studying the interaction and cooperation between the fog and cloud.

Note that in this paper the optimization is performed in a centralized manner. For the future work, we intend to further consider the case that the optimization is performed in a distributed manner. In that case, the required information exchange and communication overhead need to be carefully investigated.

APPENDIX A
SOLVE SP2 USING GBD ALGORITHM

Definition 1: Define $\boldsymbol{y}$, $\boldsymbol{f}$, $\boldsymbol{n}$, $\boldsymbol{\sigma}$ as the vectors of y_j, f_j, n_j, σ_j, and Y, F, N, Σ as the definition domains of y_j, f_j, n_j, σ_j, i.e., (4)–(7).

We now follow [38, Ch. 13] to solve SP2 using the GBD algorithm. For MINLP SP2, $\boldsymbol{y}$ and $\boldsymbol{f}$ are continuous, while $\boldsymbol{n}$ and $\boldsymbol{\sigma}$ are integer variables. Let $\boldsymbol{y}^*$, $\boldsymbol{f}^*$, $\boldsymbol{n}^*$, and $\boldsymbol{\sigma}^*$ denote the optimal solution. Clearly, finding the optimal integer variables $\boldsymbol{n}^*$ and $\boldsymbol{\sigma}^*$ is the critical part of solving MINLP. When the integer variables are determined, MINLP reduces to a linear programming (LP) problem, which is generally easy to tackle. In other words, once $\boldsymbol{n}^*$ and $\boldsymbol{\sigma}^*$ are determined, $\boldsymbol{y}^*$ and $\boldsymbol{f}^*$ can be easily solved.

The GBD algorithm is an iterative approach for solving MINLP and the underlying intuition is described as follows. MINLP is decomposed into a master problem (MP) and an SP. MP is an integer programming problem, which aims to determine the integer variables by considering only the integer constraints (with lower bound solution LB). When the integer variables are determined, MINLP reduces to SP (an LP problem) to determine the continuous variables (with upper bound solution UB). In general cases the determined integer variables are not optimal, but they can be improved by adding new integer constraints into MP, such that the search/feasible space shrinks and the newly determined integer variables gradually approach the optimum. For example, then SP has the feasible solution but UB $>$ LB (i.e., $\boldsymbol{n}$ and $\boldsymbol{\sigma}$ are not optimal), in order to improve $\boldsymbol{n}$ and $\boldsymbol{\sigma}$, the new LB should be large than previous LBs, by adding the feasibility constraint (9b) into MP. When SP is infeasible to solve, in order to avoid obtaining the improper $\boldsymbol{n}$ and $\boldsymbol{\sigma}$ again, the infeasibility constraint (9b) is added into MP. The optimal solution is converged when $|\text{UB} - \text{LB}| \le \epsilon$, where ϵ is error tolerance (stopping criterion). The iterative approach is summarized in Algorithm 1, which involves the following definitions.

Definition 2: Objective function $\mathcal{F}(\boldsymbol{f}, \boldsymbol{n}, \boldsymbol{\sigma})$ and constraint functions $\mathcal{G}(\boldsymbol{y})$, $\mathcal{H}(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{n}, \boldsymbol{\sigma})$

$$\begin{cases} \mathcal{F}(\boldsymbol{f}, \boldsymbol{n}, \boldsymbol{\sigma}) \triangleq \sum_{j\in\mathcal{M}} \sigma_j n_j \left(A_j f_j^p + B_j\right) \\ \mathcal{G}(\boldsymbol{y}) \triangleq \sum_{j\in\mathcal{M}} y_j - Y \\ \mathcal{H}(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{n}, \boldsymbol{\sigma}) \triangleq \left[h_1, \dots, h_j, \dots, h_M\right]^T \\ h_j \triangleq \sigma_j \left[\dfrac{C\left(n_j, y_j K/f_j\right)}{n_j f_j/K - y_j} + \dfrac{K}{f_j}\right] - \overline{D}_j. \end{cases}$$

Thus, MINLP SP2 is

$$\min_{\boldsymbol{y}\in Y, \boldsymbol{f}\in F, \boldsymbol{n}\in N, \boldsymbol{\sigma}\in\Sigma} \ \mathcal{F}(\boldsymbol{f}, \boldsymbol{n}, \boldsymbol{\sigma}) \quad \text{s.t.} \begin{cases} \mathcal{G}(\boldsymbol{y}) = 0 \\ \mathcal{H}(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{n}, \boldsymbol{\sigma}) \le \boldsymbol{0}. \end{cases}$$

Definition 3: MP^k

$$\min_{\boldsymbol{n}\in N, \boldsymbol{\sigma}\in\Sigma, \text{LB}} \ \text{LB} \tag{9a}$$

$$\text{s.t.} \begin{cases} \text{LB} \ge \mathcal{F}\left(\boldsymbol{f}^i, \boldsymbol{n}, \boldsymbol{\sigma}\right) + \lambda^i \mathcal{G}\left(\boldsymbol{y}^i\right) \\ \quad + \left(\boldsymbol{\mu}^i\right)^T \mathcal{H}\left(\boldsymbol{y}^i, \boldsymbol{f}^i, \boldsymbol{n}, \boldsymbol{\sigma}\right) \forall i \in \mathcal{I}^k \\ 0 \ge \lambda^j \mathcal{G}\left(\boldsymbol{y}^j\right) + \left(\boldsymbol{\mu}^j\right)^T \mathcal{H}\left(\boldsymbol{y}^j, \boldsymbol{f}^j, \boldsymbol{n}, \boldsymbol{\sigma}\right) \forall j \in \mathcal{J}^k. \end{cases} \tag{9b}$$

Algorithm 1: GBD Algorithm for Solving SP2

/* Initialization */
Set $k \leftarrow 1$, $\mathcal{I}^1 \leftarrow \emptyset$, $\mathcal{J}^1 \leftarrow \emptyset$, $UB^0 \leftarrow +\infty$;
while do
  Solve MP^k by, e.g., branch and bound;
  if *feasible solution* **then**
    Obtain solution $\left(\boldsymbol{n}^k, \boldsymbol{\sigma}^k, LB^k\right)$;
  else if *unbounded solution* **then**
    Choose arbitrary $\boldsymbol{n}^k \in N$ and $\boldsymbol{\sigma}^k \in \Sigma$;
    Set $LB^k \leftarrow -\infty$;
  end if
  Solve $\text{SP}\left(\boldsymbol{n}^k, \boldsymbol{\sigma}^k\right)$ by, e.g., dual decomposition;
  if *feasible solution* **then**
    Obtain solution $\left(\boldsymbol{y}^k, \boldsymbol{f}^k\right)$ and Lagrangian multiplier $\left(\lambda^k, \boldsymbol{\mu}^k\right)$;
    Set $UB^k \leftarrow \min\left\{UB^{k-1}, \mathcal{F}\left(\boldsymbol{f}^k, \boldsymbol{n}^k, \boldsymbol{\sigma}^k\right)\right\}$;
    if $\left|UB^k - LB^k\right| \le \epsilon$ **then** /* Converged */
      return $\left(\boldsymbol{y}^k, \boldsymbol{f}^k, \boldsymbol{n}^k, \boldsymbol{\sigma}^k\right)$;
    else /* Add feasible constraint */
      Set $\mathcal{I}^{k+1} \leftarrow \mathcal{I}^k \cup \{k\}$, $\mathcal{J}^{k+1} \leftarrow \mathcal{J}^k$;
    end if
  else if *infeasible solution* **then**
    Solve $\text{SPF}\left(\boldsymbol{n}^k, \boldsymbol{\sigma}^k\right)$ by, e.g., dual decomposition;
    Obtain solution $\left(\boldsymbol{y}^k, \boldsymbol{f}^k\right)$ and Lagrangian multiplier $\left(\lambda^k, \boldsymbol{\mu}^k\right)$;
    Set $UB^k \leftarrow UB^{k-1}$;
    /* Add infeasible constraint */
    Set $\mathcal{I}^{k+1} \leftarrow \mathcal{I}^k$, $\mathcal{J}^{k+1} \leftarrow \mathcal{J}^k \cup \{k\}$;
  end if
  Set $k \leftarrow k + 1$;
end while

Definition 4: $\text{SP}(\boldsymbol{n}^k, \boldsymbol{\sigma}^k)$

$$\min_{\boldsymbol{y}\in Y, \boldsymbol{f}\in F} \ \mathcal{F}\left(\boldsymbol{f}, \boldsymbol{n}^k, \boldsymbol{\sigma}^k\right) \quad \text{s.t.} \begin{cases} \mathcal{G}(\boldsymbol{y}) = 0 \\ \mathcal{H}\left(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{n}^k, \boldsymbol{\sigma}^k\right) \le \boldsymbol{0}. \end{cases}$$

Definition 5: SP feasibility-check $\text{SPF}(\boldsymbol{n}^k, \boldsymbol{\sigma}^k)$

$$\min_{\boldsymbol{y}\in Y, \boldsymbol{f}\in F, \boldsymbol{s}} \ \boldsymbol{1}^T \boldsymbol{s} \quad \text{s.t.} \begin{cases} \mathcal{G}(\boldsymbol{y}) = 0 \\ \boldsymbol{s} \ge \mathcal{H}\left(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{n}^k, \boldsymbol{\sigma}^k\right). \end{cases}$$

APPENDIX B
SOLVE SP3 USING HUNGARIAN ALGORITHM

The Hungarian algorithm is a combinatorial optimization approach that solves the assignment problem in polynomial time. We define

$$C_{ij} \triangleq \min\left\{l_i - x_i, \lambda_{ij}^{\max}, y_j\right\} \qquad \forall i \in \mathcal{N}, j \in \mathcal{M}.$$

Thus, SP3 can be equivalently transformed into a standard form of the assignment problem

$$\min_{z_{ij}} \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}} d_{ij}C_{ij}z_{ij}$$
$$\text{s.t.} \begin{cases} \sum_{j\in\mathcal{M}} z_{ij} = 1 & \forall i \in \mathcal{N} \\ \sum_{i\in\mathcal{N}} z_{ij} = 1 & \forall j \in \mathcal{M} \\ 0 \le z_{ij} \le 1 & \forall i \in \mathcal{N};\quad j \in \mathcal{M} \end{cases} \tag{10}$$

where z_{ij} represents the assignment of the fog device i to the cloud server j, taking value 1 if the assignment is done and 0 otherwise. This formulation allows also fractional values, but there is always an optimal solution where the variables take integer values. This is because the constraint matrix is totally unimodular.

To illustrate the Hungarian algorithm for solving the above problem, without loss of generality, we consider a simple case with $|\mathcal{N}| = 4$ and $|\mathcal{M}| = 3$. Since the two sets $\mathcal{N}$ and $\mathcal{M}$ should be of equal size, we add an additional dummy cloud server CS_3. The above problem can be viewed graphically: three fog devices FD_1, FD_2, FD_3, and FD_4 as well as three cloud servers CS_1, CS_2, CS_3, and CS_4 (including the dummy one). The lines from FD_i to CS_j represent the values of cost $d_{ij}C_{ij}$, with all $d_{i4}C_{i4}$ setting to 0. For generality, we define the cost matrix to be the $n \times n$ matrix

$$C \triangleq \begin{bmatrix} d_{11}C_{11} & \dots & d_{1n}C_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1}C_{n1} & \cdots & d_{nn}C_{nn} \end{bmatrix}.$$

An assignment is a set of n entry positions in the cost matrix, no two of which lie in the same row or column. The sum of the n entries of an assignment is its cost. An assignment with the smallest possible cost is called an optimal assignment.

Theorem 1: If a number is added to or subtracted from all of the entries of any one row or column of a cost matrix, then one optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix.

Algorithm 2 applies Theorem 1 to a given $n \times n$ cost matrix to find an optimal assignment. To illustrate Algorithm 2 for solving (10), without loss of generality, we consider a simple case as shown in transformation (11), shown at the top of the next page. Step ① is to subtract 0 from each row. Step ② is to subtract 35 from column 1, 75 from column 2, 55 from column 3, and 0 from column 4. Step ③ is to cover all zeros with the minimum number of horizontal or vertical lines. Since the minimum number of covering lines is less than 4, we find that 10 is the smallest entry not covered by any line, and then subtract 10 from each uncovered row. Step ④ is to add 10 to each covered column. Since the minimum number of covering lines is 4, an optimal assignment of zeros is obtained. Step ⑤ is to make the same assignment for the original cost matrix. Thus, the optimal assignment for this case is $z^*_{12} = z^*_{23} = z^*_{34} = z^*_{41} = 1$ with the smallest cost of 175.

Algorithm 2: Hungarian Algorithm for Solving (10)

Subtract the smallest entry in each row from all the entries of its row;
Subtract the smallest entry in each column from all the entries of its column;
while do
    Draw lines through appropriate rows and columns so that all the zero entries of the cost matrix are covered and the minimum number of such lines is used;
    `/* Test for optimality */`
    if *the minimum number of covering lines is n* **then**
        return an optimal assignment of zeros;
    else if *the minimum number of covering lines is less than n* **then** `/* An optimal assignment of zeros is not yet possible */`
        Determine the smallest entry not covered by any line;
        Subtract this entry from each uncovered row;
        Add this entry to each covered column;
    end if
end while

Algorithm 3: Update Parameters in SP3

for $i \in \mathcal{N}, j \in \mathcal{M}$ **do**
    if $z^*_{ij} == 1$ **then**
        if $C_{ij} == l_i - x_i$ **then** `/* Remove` i `*/`
            $\lambda^*_{ij} \leftarrow l_i - x_i$;
            $\mathcal{N} \leftarrow \mathcal{N}\backslash\{i\}$;
            $y_j \leftarrow y_j - \lambda^*_{ij}$;
        else if $C_{ij} == \lambda^{\max}_{ij}$ **then** `/* Remove` $i{\sim}j$ `*/`
            $\lambda^*_{ij} \leftarrow \lambda^{\max}_{ij}$;
            $l_i - x_i \leftarrow l_i - x_i - \lambda^*_{ij}$;
            $y_j \leftarrow y_j - \lambda^*_{ij}$;
            $C_{ij} \leftarrow \infty$;
        else if $C_{ij} == y_j$ **then** `/* Remove` j `*/`
            $\lambda^*_{ij} \leftarrow y_j$;
            $l_i - x_i \leftarrow l_i - x_i - \lambda^*_{ij}$;
            $\mathcal{M} \leftarrow \mathcal{M}\backslash\{j\}$;
        end if
    end if
end for

Based on the optimal assignment to problem (10), we update the corresponding parameters in SP3 according to Algorithm 3. Then we get a new assignment problem (10). In the same way, by adding additional dummy cloud servers, we have two sets of nodes with equal size, together with the corresponding cost matrix. Again, we apply Hungarian Algorithm 2 to solve (10), obtain the optimal assignment, and update parameters in SP3 according to Algorithm 3. This process goes so on and so forth until all the unprocessed requests have been dispatched from fog devices to cloud servers.

$$
C = \begin{bmatrix} 90 & 75 & 75 & \boxed{0} \\ 35 & 85 & 55 & \boxed{0} \\ 125 & 95 & 90 & \boxed{0} \\ 45 & 110 & 95 & \boxed{0} \end{bmatrix} \xrightarrow{①} \begin{bmatrix} 90 & \boxed{75} & 75 & \boxed{0} \\ \boxed{35} & 85 & \boxed{55} & 0 \\ 125 & 95 & 90 & 0 \\ 45 & 110 & 95 & 0 \end{bmatrix} \xrightarrow{②} \begin{bmatrix} \cancel{55} & \cancel{0} & \cancel{20} & \cancel{0} \\ \cancel{0} & \cancel{10} & \cancel{0} & \cancel{0} \\ 90 & 20 & 35 & \cancel{0} \\ \boxed{10} & 35 & 40 & \cancel{0} \end{bmatrix}
$$

$$
\xrightarrow{③} \begin{bmatrix} 55 & 0 & 20 & 0 \\ 0 & 10 & 0 & 0 \\ 80 & 10 & 25 & -10 \\ 0 & 25 & 30 & -10 \end{bmatrix} \xrightarrow{④} \begin{bmatrix} \cancel{55} & \boxed{\cancel{0}} & \cancel{20} & \cancel{10} \\ \cancel{0} & \cancel{10} & \boxed{\cancel{0}} & \cancel{10} \\ \cancel{80} & 10 & 25 & \boxed{\cancel{0}} \\ \boxed{\cancel{0}} & 25 & 30 & \cancel{0} \end{bmatrix} \xrightarrow{⑤} \begin{bmatrix} 90 & \boxed{75} & 75 & 0 \\ 35 & 85 & \boxed{55} & 0 \\ 125 & 95 & 90 & \boxed{0} \\ \boxed{45} & 110 & 95 & 0 \end{bmatrix} \tag{11}
$$

References

[1] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proc. IEEE ICC*, London, U.K., 2015, pp. 3909–3914.

[2] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Netw.*, vol. 28, no. 4, pp. 46–50, Jul./Aug. 2014.

[3] N. Kumar, S. Misra, J. J. P. C. Rodrigues, and M. S. Obaidat, "Coalition games for spatio-temporal big data in Internet of Vehicles environment: A comparative analysis," *IEEE Internet Things J.*, vol. 2, no. 4, pp. 310–320, Aug. 2015.

[4] C. Lai, R. Lu, D. Zheng, H. Li, and X. Shen, "Toward secure large-scale machine-to-machine communications in 3GPP networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 12–19, Dec. 2015.

[5] T. H. Luan, L. X. Cai, J. Chen, X. S. Shen, and F. Bai, "Engineering a distributed infrastructure for large-scale cost-effective content dissemination over urban vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 3, pp. 1419–1435, Mar. 2014.

[6] S. He, J. Chen, X. Li, X. S. Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1268–1282, Jun. 2014.

[7] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[8] The Network. *Cisco Delivers Vision of Fog Computing to Accelerate Value From Billions of Connected Devices*. Accessed on Sep. 14, 2014. [Online]. Available: http://newsroom.cisco.com/press-release-content?articleId=1334100

[9] S. He *et al.*, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1931–1942, Oct. 2013.

[10] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, 2014, pp. 169–186.

[11] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 570–582, Jun. 2015.

[12] J. Chen *et al.*, "Dynamic channel assignment for wireless sensor networks: A regret matching based approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 95–106, Jan. 2015.

[13] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[14] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[15] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[16] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric Internet of Things services," in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Madrid, Spain, 2015, pp. 1–2.

[17] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint arXiv:1502.01815*, 2015.

[18] R. Suryawansh and G. Mandlik, "Focusing on mobile users at edge and Internet of Things using fog computing," *Int. J. Sci. Eng. Technol. Res.*, vol. 4, no. 17, pp. 3225–3231, 2015.

[19] L. Rao, X. Liu, L. Xie, and W. Liu, "Coordinated energy cost management of distributed Internet data centers in smart grid," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 50–58, Mar. 2012.

[20] L. Yu, T. Jiang, Y. Cao, and Q. Qi, "Carbon-aware energy cost minimization for distributed Internet data centers in smart microgrids," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 255–264, Jun. 2014.

[21] L. Rao, X. Liu, M. D. Ilic, and J. Liu, "Distributed coordination of Internet data centers under multiregional electricity markets," *Proc. IEEE*, vol. 100, no. 1, pp. 269–282, Jan. 2012.

[22] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. ACM Workshop Mobile Big Data (Mobidata)*, Hangzhou, China, 2015, pp. 37–42.

[23] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Wireless Algorithms, Systems, and Applications*. Cham, Switzerland: Springer, 2015, pp. 685–695.

[24] X. Wang, X. Chen, C. Yuen, W. Wu, and W. Wang, "To migrate or to wait: Delay-cost tradeoff for cloud data centers," in *Proc. IEEE Globecom*, Austin, TX, USA, 2014, pp. 2314–2319.

[25] X. Wang, C. Yuen, N. Ul Hassan, W. Wang, and T. Chen, "Migration-aware virtual machine placement for cloud data centers," in *Proc. IEEE ICC Workshop*, London, U.K., 2015, pp. 1940–1945.

[26] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[27] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Warsaw, Poland, 2014, pp. 1–8.

[28] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proc. Aust. Telecommun. Netw. Appl. Conf. (ATNAC)*, Southbank, Vic., Australia, 2014, pp. 117–122.

[29] X. Wang, C. Yuen, X. Chen, N. U. Hassan, and Y. Ouyang, "Cost-aware demand scheduling for delay tolerant applications," *J. Netw. Comput. Appl.*, vol. 53, pp. 173–182, Jul. 2015.

[30] M. Maleki, K. Dantu, and M. Pedram, "Power-aware source routing protocol for mobile ad hoc networks," in *Proc. ACM Int. Symp. Low Power Electron. Design*, Monterey, CA, USA, 2002, pp. 72–75.

[31] F. Ahmad and T. N. Vijaykumar, "Joint optimization of idle and cooling power in data centers while maintaining response time," *ACM SIGPLAN Not.*, vol. 45, no. 3, pp. 243–256, 2010.

[32] N. Gautam, *Analysis of Queues: Methods and Applications*. Boca Raton, FL, USA: CRC Press, 2012.

[33] R. Deng, Y. Zhang, S. He, J. Chen, and X. S. Shen, "Maximizing network utility of rechargeable sensor networks with spatiotemporally-coupled constraints," *IEEE J. Sel. Areas Commun.*, to be published, doi: 10.1109/JSAC.2016.2520181.

[34] J. Ren *et al.*, "Joint channel access and sampling rate control in energy harvesting cognitive radio sensor networks," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2016.2555806.

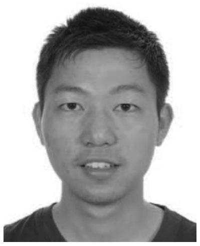
[35] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ., 2004.

[36] Z. Yang, K. Long, P. You, and M.-Y. Chow, "Joint scheduling of large-scale appliances and batteries via distributed mixed optimization," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2031–2040, Jul. 2015.

[37] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2015.2470218.

[38] D. Li and X. Sun, *Nonlinear Integer Programming*. New York, NY, USA: Springer, 2006.

[39] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, nos. 1–2, pp. 83–97, 1955.



Ruilong Deng (S'11–M'14) received the B.Sc. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2009 and 2014, respectively.

He was a Visiting Scholar with Simula Research Laboratory, Fornebu, Norway, in 2011, and the University of Waterloo, Waterloo, ON, Canada, from 2012 to 2013. He was a Research Fellow with Nanyang Technological University, Singapore, from 2014 to 2015. He is currently an AITF Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His current research interests include smart grid, cognitive radio, and wireless sensor network.

Dr. Deng currently serves as an Editor for the IEEE/KICS JOURNAL OF COMMUNICATIONS AND NETWORKS, and a Guest Editor for the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING and the *Journal of Computer Networks and Communications*. He also serves/served as a Technical Program Committee Member for IEEE Globecom, IEEE ICC, IEEE SmartGridComm, and EAI SGSC.

Rongxing Lu (S'09–M'11–SM'15) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2012.

From 2012 to 2013, he was a Post-Doctoral Fellow with the University of Waterloo. Since 2013, he has been an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His current research interests include computer network security, mobile and wireless communication security, and applied cryptography.

Dr. Lu was a recipient of the Canada Governor General Gold Medal.

Chengzhe Lai (M'14) received the B.S. degree in information security from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2008, and the Ph.D. degree from Xidian University, Xi'an, in 2014.

He is currently with the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, and the National Engineering Laboratory for Wireless Security, Xi'an, China. His current research interests include wireless network security, privacy preservation, and machine-to-machine (M2M) communications security.

Tom H. Luan (M'13) received the B.Sc. degree from Xi'an Jiaotong University, Xi'an, China, in 2004, the M.Phil. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2007, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2012.

Since 2013, he has been the Lecturer of Mobile and Apps with the School of Information Technology, Deakin University, Burwood, Vic., Australia. His current research interests include vehicular networking, mobile content distribution, fog computing, and mobile cloud computing.

Hao Liang (S'09–M'14) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2013.

He has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, since 2014. From 2013 to 2014, he was a Post-Doctoral Research Fellow with the Broadband Communications Research Laboratory and Electricity Market Simulation and Optimization Laboratory, University of Waterloo, Waterloo, ON, Canada. His current research interests include smart grids, wireless communications, and wireless networking.

Dr. Liang serves/served as a Guest Editor for the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING and the *Journal of Computer Networks and Communications*. He was the System Administrator of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2009 to 2013. He has been a Technical Program Committee Member for major international conferences in both information/communication system discipline and power/energy system discipline, including the IEEE International Conference on Communications, the IEEE Global Communications Conference, IEEE VTC, the IEEE Innovative Smart Grid Technologies Conference, and the IEEE International Conference on Smart Grid Communications. He was a recipient of the Best Student Paper Award from the IEEE 72nd Vehicular Technology Conference, Ottawa, ON, Canada, in 2010.