# QoE and Power Efficiency Tradeoff for Fog Computing Networks with Fog Node Cooperation

Yong Xiao and Marwan Krunz
Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ
E-mails: yongxiao and krunz@email.arizona.edu

***Abstract*—This paper studies the workload offloading problem for fog computing networks in which a set of fog nodes can offload part or all the workload originally targeted to the cloud data centers to further improve the quality-of-experience (QoE) of users. We investigate two performance metrics for fog computing networks: users' QoE and fog nodes' power efficiency. We observe a fundamental tradeoff between these two metrics for fog computing networks. We then consider cooperative fog computing networks in which multiple fog nodes can help each other to jointly offload workload from cloud data centers. We propose a novel cooperation strategy referred to as *offload forwarding*, in which each fog node, instead of always relying on cloud data centers to process its unprocessed workload, can also forward part or all of its unprocessed workload to its neighboring fog nodes to further improve the QoE of its users. A distributed optimization algorithm based on distributed alternating direction method of multipliers (ADMM) via variable splitting is proposed to achieve the optimal workload allocation solution that maximizes users' QoE under the given power efficiency. We consider a fog computing platform that is supported by a wireless infrastructure as a case study to verify the performance of our proposed framework. Numerical results show that our proposed approach significantly improves the performance of fog computing networks.**

***Index Terms*—Fog computing, response-time analysis, power efficiency, offload forwarding, ADMM.**

## I. Introduction

Cloud computing has been proposed as a promising paradigm to meet the fast growing demand for computationally intensive applications and services. It provides users with versatile and on-demand services by effectively utilizing the hardware and software in cloud data centers. Large-scale data centers are massive and expensive, and therefore always built in the low-cost remote areas. Currently, how to provide high quality services for the widely geographical distributed users, especially the users at the edge of network, is still an open problem. This motivates a new framework referred to as the *fog computing*, which extends the cloud computing paradigm to the network edge. Formally, fog computing is defined as a visualized network architecture that "*uses one or a collaborative multitude of end-user clients or near-user edge devices to carry out a substantial amount of storage (instead of stored primarily in cloud data centers), communication (instead of routed over backbone networks), and control, configuration, measurement, and management*" [1]. Edge devices that provide services between end users and cloud data centers are commonly referred to as *fog nodes*. They can be resource-limited routers, gateways, and access points, and can also correspond to mobile devices with excessive computing resources that can be utilized to offer services for others [2]. Fog computing enables computational workload offloading through fog nodes which can further reduce the transmission latency and ease traffic congestions of the Internet. It also introduces many new services and applications that cannot fit well in the traditional cloud computing architecture. For example, large-scale environmental monitoring systems can deploy computational intensive applications at the sensors and utilize the fog computing architecture to achieve instantaneous environment monitoring and fast hazard detection [3], [4].

One of the main objectives of fog computing is to improve the service quality of users at the edge of the network. Most existing works focus on developing optimal resource allocation strategies that can maximize users' quality-of-service (QoS), a metric denoting the level of service performance that can be offered by the hardware platform or hosting infrastructure [5] such as processing capacity, resource utilization efficiency, processing delay of the cloud data centers. In particular, the resource provisioning problem for a cloud data center network has been modeled as an auction-based market in [6] in which users can develop bidding strategies to compete for the capacity of the cloud data centers with low costs. In [7], the authors introduced a service-oriented resource estimating and management framework for fog computing to maximize the resource utilization of the cloud data centers. A framework that supports computation offloading and data staging at the tactical edge was proposed in [8]. Motivated by the observation that the QoS cannot always reflect the actually service quality that is experienced by the users, the concept of quality-of-experience (QoE) was introduced recently as one of the main guiding paradigms for service quality of the cloud computing networks [9]. QoE can be regarded as an extension of the QoS by focusing more on the influence of

This research was supported in part by the National Science Foundation (grants # IIP-1265960, IIP-1535573, CNS-1563655, and CNS-1409172). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

978-1-5090-5336-0/17/$31.00 ©2017 IEEE

the interactivity of the cloud service experienced by the users. In this paper, we focus on the QoE of users measured by the average service response-time that can be influenced by the queueing delay and round-trip workload transmission time including that between users and fog nodes, that between fog nodes and cloud data centers as well as that between cooperative fog nodes.

Fog computing is not intended to replace cloud computing but to compliment it. In this paper, we study workload offloading for fog computing networks in which fog nodes can offload workload from cloud data centers to further improve the QoE of the users. We consider two performance metrics for this fog computing network: the users' QoE and the fog nodes' power efficiency, measured by the amount of power consumed by each fog node to offload each unit of workload from the cloud. We perform detailed response-time analysis under different scenarios and derive the optimal amount of workload to be offloaded by the fog nodes that can maximize the users' QoE under the given power efficiency. We observe a fundamental tradeoff between these two metrics. In addition, motivated by the observation that the users' QoE can be further improved if the workload offloading process of each fog node can be helped by other fog nodes, we consider a fog computing framework with fog node cooperation. We propose a novel cooperation strategy called *offload forwarding*, in which each fog node can forward part or all of its offloaded workload to other local fog nodes, instead of always forwarding all unprocessed workload to the cloud. We study the offload allocation problem in which all fog nodes jointly determine the optimal amount of offloaded workload to be forwarded and processed by each other to further improve the users' QoE. We analyze the QoE and power efficiency tradeoff under cooperative fog computing with offload forwarding and propose a novel distributed optimization framework based on distributed alternating direction method of multipliers (ADMM) via variable splitting to achieve this tradeoff. Our proposed algorithm does not require fog nodes to have back-and-forth negotiation or disclose their private information. Finally, as a case study, we consider a fog computing platform that is supported by a wireless network infrastructure. Numerical results show that our approach can significantly improve the performance of fog computing systems. To the best of our knowledge, this is the first work that studies the allocations of offloaded workload among cooperative fog nodes under the fog computing paradigm.

## II. Fog Computing Architecture and Problem Formulation

A generic fog computing architecture consists of a three-layer structure as illustrated in Figure 1:

1) *Cloud layer* comprises large-scale cloud data centers with high-performance computing units usually located in remote area that can be far from some users.
2) *Fog layer* contains a set of low-cost fog nodes that can be widely deployed in locations that are closer to the users. Each fog node has limited computing capability and power resource, and hence needs to carefully decide the amount of workload to be processed locally. If multiple neighboring fog nodes can communicate with each other through local communication infrastructure, they can help each other to jointly process the workload to further improve the QoE of users. Fog nodes can be deployed by the cloud data centers or third party service providers.
3) *User layer* consists of users that desire low-latency high QoE computing services. Since fog nodes are located in the vicinity of users, allowing each user to always submit its workload to nearby fog nodes can further expend the service coverage area and improve the QoE of the users.

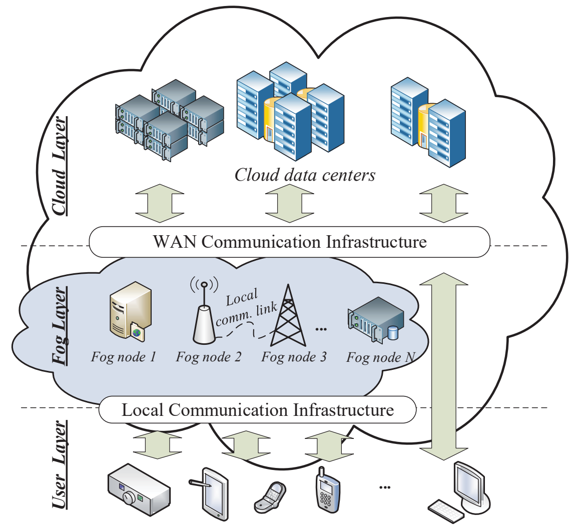


Fig. 1. A three-layer fog computing architecture.

Note that, in some systems, the fog and user layers can consist of the same type of devices. For example, in a wireless sensor network, sensors without sufficient computing resources can be regarded as elements in the user layer. These sensors can submit their excessive workload to other sensors in the fog layer that have surplus computing resources to process.

We consider a fog computing network with a set of $N$ fog nodes labeled as $\mathcal{F} = \{1, 2, \ldots, N\}$. We assume each user has been already associated with one or more fog nodes. The association between users and fog nodes can be decided by their physical locations and channel conditions (e.g., each user chooses its closest fog node to submit request) or decided by the cloud data centers. For example, users can send their service requests to cloud data centers following the same operation as the traditional cloud computing system and cloud data centers can then delegate one or more fog nodes to receive and process the workload submitted by the users. Each fog node $j$ can either process a portion $\alpha_j$ of its received workload using its local computing resources or forward all received workload to the cloud layer. Note that if $\alpha_j = 1$, it means that fog node $j$ will process all of its received workload. On the other hand, $\alpha_j = 0$ means fog node $j$ will not process any of its received workload but will directly forward all the workload to the cloud layer. The workload arrival rate of each fog node $j$, denoted by $\lambda_j$, is assumed to be fixed.

In this paper, we measure the QoE of users that require fog computing service by the average response time. We focus on the following two performance metrics;

1) *QoE (Response-time)* of users, which includes the round-trip workload transmission time and the queueing delay at the fog layer. In this paper, we follow a commonly adopted setting and consider an M/M/1 queueing system for each fog node to process the request of the users [10]. Let $R_j$ be the response-time of users served by fog node $j$. As mentioned before, one of the main objectives of fog computing is to improve the QoE of users. Since fog nodes are close to the users, allowing each fog node to offload nearby users' workload from the cloud can reduce the workload transmission time. However, each fog node has limited computing resources and processing a large amount of received workload locally will result in long queueing delay. Therefore, how to choose a balanced amount of workload to be offloaded by each fog node is an important problem.
2) *Power efficiency (Power consumption per unit of offloaded workload)* of each fog node is measured by the amount of power spent on offloading each unit of received workload. It is important for each fog node to maximize the power efficiency by minimizing its power consumed for processing the workload. It is known that the total amount of power consumed by a node depends on the power usage effectiveness (PUE) as well as the static and dynamic power consumption [11]. Specifically, the PUE is defined as the total power input from the power grid divided by the power consumption of the IT infrastructure. Static power consumption, also called leakage power, is mainly caused by the leakage currents and is unrelated to the usage of the computing resources of each fog node. Dynamic power consumption is the result of the circuit activity and is determined by the activity of computing resources. Let $e_j$ and $w_j^S$ be the PUE and static power consumption of fog node $j$. Let $w_j^D$ be the dynamic power consumed by fog node $j$ to offload each unit of workload. We can write the total power consumption of fog node $j$ per time unit as $w_j = e_j \left(w_j^S + w_j^D \alpha_j \lambda_j\right)$. The power efficiency of fog node $j$ can then be written as

$$\eta_j\left(\alpha_j\right) = \frac{w_j}{\alpha_j \lambda_j} = e_j\left(\frac{w_j^S}{\alpha_j \lambda_j} + w_j^D\right). \tag{1}$$

The main objective of this paper is to develop workload allocation strategies to maximize the QoE of users under a given power efficiency constraint. Formally, we try to solve the following optimization problem for a single-node fog computing network:

$$\min_{0 \le \alpha_j \le 1} R_j\left(\alpha_j\right) \quad \text{s.t. } \eta_j\left(\alpha_j\right) \le \bar{\eta}_j. \tag{2}$$

where $R_j\left(\alpha_j\right)$ is the response-time of fog node $j$ when it processes $\alpha_j$ portion of its received workload, and $\bar{\eta}_j$ is the maximum power efficiency supported by the hardware of fog node $j$. We will give a more detailed discussion on the response-time of fog node $j$ under different scenarios in Section III.

In this paper, we also consider cooperative fog computing networks in which fog nodes can help each other and share their offloaded workload. The main objective in this case is to minimize the average response time of all users. Note that different fog nodes can have different workload arrival rates. The optimization problem under cooperative fog computing can then be stated as follows:

$$\min_{\boldsymbol{\alpha}=[\alpha_1,\alpha_2\ldots,\alpha_N]} \sum_{j\in\mathcal{F}} R_j^C\left(\xi_j, \alpha_j\right) \quad \text{s.t. } \eta_j\left(\alpha_j\right) \le \bar{\eta}_j, 0 \le \alpha_j \le 1, \forall j \in \mathcal{F}, \tag{3}$$

where $R_j^C\left(\xi_j, \alpha_j\right)$ is the average response-time of all the users associated with fog node $j$ when fog node $j$ can be helped by others and $\xi_j$ is the weight factor for each fog node $j$ defined as $\xi_j = \frac{\lambda_j}{\sum_{k\in\mathcal{F}} \lambda_k}$. As will be discussed later in Section IV, we add a weight factor for the response time to ensure that users associated with different fog nodes of different workload arrival rates will have the same QoE. In contrast to the single-node fog computing model, the response time in cooperative fog computing also depends on the cooperation strategy among nodes. We will give a more detailed discussion about the strategies of the cooperative fog computing networks.

As we will describe later, the users' QoE and fog nodes' power efficiency are closely related to each other. More specifically, we can observe a fundamental tradeoff between users' QoE and fog nodes' power efficiency in fog computing networks. In the next section, we will discuss this tradeoff for fog computing networks without fog node cooperation. We will study the QoE and power efficiency tradeoff for cooperative fog computing in Section IV.

## III. QoE and Power Efficiency Tradeoff

In this section, we first derive the response-time of users supported by fog computing. We then discuss the tradeoff between users' QoE and fog node's power efficiency under different scenarios. Since fog nodes cannot cooperate or communicate with each other, each fog node needs to make independent decision about the amount of workload to be processed by itself. In this section, without loss of generality, we focus on the workload offloading process of one fog node, labeled as fog node $j$.

### A. Response-time Analysis and Response-time Minimization Solution

Let $\tau_j^u$ be the average round trip time between fog node $j$ and its corresponding users. We assume the round trip time for workload transmission between each fog node $j$ and the cloud layer can be regarded as a constant denoted as $\tau^c$. Fog node $j$ can choose to serve its received workload using the following three ways:

*1) No Offloading:* Fog node $j$ can directly forward all the received workload to cloud data centers through the backbone IP network [1]. In this case, the fog computing network becomes equivalent to the traditional cloud computing network in which all the workload is processed by the cloud layer. As mentioned earlier, since cloud data centers are generally installed with high-performance workload processing units, the response time for data centers to process the workload forwarded by each fog node is much smaller than the time spent on workload transmission and, hence, in most existing applications, can be ignored [12]. In this paper, we follow the same line and assume the delay caused by cloud data centers to process the workload forwarded by each fog node is negligible. In this case, we can write the response time of fog node $j$ as

$$R_j^{W1} = \tau_j^u + \tau^c. \tag{4}$$

Since in this case fog node $j$ does not activate any computing resource to process its received workload, the power efficiency will not depend on users' response-time. As mentioned earlier, one of the main objectives of fog computing is to further improve users' QoE by offloading workload targeted to the cloud layer to the fog layer. In other words, (4) can be regarded as an upper bound of the response-time provided by fog node $j$ to its users.

*2) Full Offloading:* Fog node $j$ can also process all the received workload by itself using its local computing resources. In this case, the cloud layer will not process any workload for users associated with fog node $j$, i.e., $\alpha_j = 1$. Let $\mu_j$ be the maximum amount of workload that can be processed by the computing resources installed at the $j$th fog node at each time unit. We can write the response time of fog node $j$ in this case as

$$R_j^{W2}(\alpha_j) = \tau_j^u + \frac{1}{\mu_j - \lambda_j}. \tag{5}$$

*3) Partial Offloading:* Compared to cloud data centers, each fog node can only have limited computing resource. It is generally impossible to always allow each fog node to process all the received workload. We therefore consider the cases that each fog node $j$ only processes a portion $(1-\alpha_j)$ of its received workload using its own computing resources and forwards the rest of its received workload to the cloud, i.e., we have $0 \le \alpha_j < 1$ and $\alpha_j \lambda_j < \mu_j$. We can write the expected response time for fog node $j$ as

$$\begin{aligned} R_j^{W3}(\alpha_j) &= \tau_j^u + \alpha_j \left( \frac{1}{\mu_j - \alpha_j \lambda_j} \right) \\ &\quad + (1-\alpha_j)\tau^c. \end{aligned} \tag{6}$$

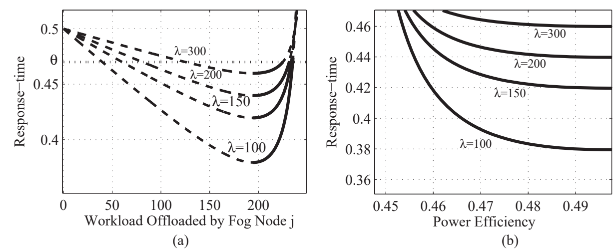


Fig. 2. Response-time under different amounts of offloaded workload and power efficiency.

Let us now consider the solution of problem (2) by substituting the response time equations in (4)–(6).

It can be easily observed that problem (2) is a convex optimization problem and, hence can be solved using the standard approach [13]. We omit the detailed derivation due to the limit of space.

### B. A Fundamental Tradeoff between Users' QoE and Fog Node's Power Efficiency

In Figure 2(a), we present users' response-time under different amounts of workload offloaded by the corresponding fog node. We can observe that there exists an optimal amount of offloaded workload to minimize the response-time of users. As observed in (1), the power consumption for each fog node to process each unit of workload decreases with the total amount of offloaded workload. In many practical applications, there is a maximum tolerable response-time for the users. We can therefore observe that the power efficiency maximization solution for fog computing in this case will be achieved when users' response-time becomes equivalent to the required maximum tolerable point. In Figure 2(a), we use solid line to highlight the segment between the response time minimization solution and the power efficiency maximization solution with maximum tolerable response-time $\theta$ for $R_j^{W3}(\alpha_j^*) < \theta < \infty$. We can observe from the highlighted segment that there is a fundamental tradeoff between the users' QoE measured by response time and the power efficiency of the fog node. This tradeoff can be specifically characterized by substituting (5) and (6) into (2) which is shown in Figure 2(b). We can observe that starting from the power consumption minimization point, the users' QoE decreases with the power consumption for the fog node to offload each unit of workload. With the power consumption of the fog node continues to grow, the decreasing speed of the response-time of the users reduces. In other words, in non-delay-sensitive applications such as voice/video call services, the fog node can choose a low power consumption solution as long as the resulting response time is tolerable for the users. For delay sensitive applications such as online gaming, it is ideal for the fog node to choose a high power consumption solution to satisfy users' QoE. We also present the tradeoff solutions with different workload arrival rates of fog node $j$ in Figure 2(b). We can observe that the users'

QoE increases with fog node $j$'s workload arrival rate under a fixed power efficiency. In addition, the higher the workload arrival rate, the smaller the changes of the response time under different power efficiency.

It can be observed that with the portion of workload to be offloaded by each fog node approaches 1, the resulting response-time will approach an infinite value. In other words, fog computing cannot always improve the response-time of the users without properly choosing the amount of workload to be offloaded by the fog node.

## IV. Cooperative Fog Computing Networks

In fog computing systems, different fog nodes receive workloads with different rates. In this section, we introduce a novel cooperation strategy, referred to as the *offload forwarding*, for multiple fog nodes to help each other to jointly offload workload from the cloud layer. In this strategy, each fog node instead of always forwarding all its unprocessed workload to the cloud data center, can also forward part of its offloaded workload to other fog nodes with surplus computing resources to further reduce the response-time for its users. In other words, the fog nodes in the fog layer can be divided into two types: fog nodes of the first type, referred to as the *(workload processing service) requesters*, that will forward part of their workload to others to process, fog nodes of the other type, referred to as the *servers*, will help others to process their received workload. The amount of workload sent from each fog node to others will be based on the mutual agreement.

In this section, we present the response-time analysis for offload forwarding strategy and describe the delay and power efficiency tradeoff for the cooperative fog computing. We will present the detailed discussion about how to achieve the optimal solution presented in this section in a distributed fashion in the next section.

### A. Response-time Analysis of Workload Forwarding with One Server and One Requester

We first consider the case that fog node $j$ decides that the QoE of its users can be further improved by offloading part $(1-\beta_j)\alpha_j$ of its workload to fog node $i$ for $0 \le \beta_j < 1$, $i \ne j$ and $i, j \in \mathcal{F}$. Note that since fog node $i$ will always serve its own received workload first, the response time of the users associated with fog node $i$ does not change even fog node $i$ will process extra workload for fog node $j$. If fog node $j$ sends all the workload that is supposed to be forwarded to the cloud to fog node $i$, i.e., $\alpha_j = 1$ and $0 \le \beta_j < 1$, we can write the response-time of fog node $j$ as

$$R_j^{C1}(\alpha_j = 1, \beta_j) = \tau_j^u + \xi_j \left[\beta_j \left(\frac{1}{\mu_j - \beta_j \lambda_j}\right) (1-\beta_j)\left(\tau_{ij} + \frac{1}{\mu_i - \lambda_i - (1-\beta_j)\lambda_j}\right)\right]. \quad (7)$$

where $\xi_j = \frac{\lambda_j}{\lambda_i + \lambda_j}$ is the weighted factor. We add weighted factor $\xi_j$ for the queueing delay term of the response-time for each fog node $j$. This is because different fog nodes have different workload arrival rates and hence, to ensure users associated with different fog nodes to have the same queueing delay, we add a weighted factor that is proportional to the workload arrival rate at the queueing delay term for each fog node.

It is possible that the computing resources that are provided by fog node $i$ is insufficient to process all the rest of the workload received by fog node $j$. In this case, fog node $j$ can also forward a portion $(1-\alpha_j)$ of its workload to the cloud data center, i.e., $0 \le \alpha_j < 1$. In this case, fog node $j$ will only forward $(1-\beta_j)\alpha_j\lambda_j$ workload to fog node $i$, and the total amounts of workload to be processed by fog nodes $j$ and fog node $i$ are given by $\alpha_j\beta_j\lambda_j$ and $\lambda_i + (1-\beta_j)\alpha_j\lambda_j$, respectively. We can write the response-time of fog node $j$ in this case as

$$R_j^{C2}(\beta_j, \alpha_j) = \tau_j^u + \xi_j \left[\alpha_j\beta_j \left(\frac{1}{\mu_j - \alpha_j\beta_j\lambda_j}\right) + \alpha_j(1-\beta_j)\left(\tau_{ij} + \frac{1}{\mu_i - \lambda_i - (1-\beta_j)\alpha_j\lambda_j}\right)\right] + (1-\alpha_j)\tau^c. \quad (8)$$

By substituting the above equations into problem (3) to optimize the values of $\beta_j$ and $\alpha_j$, fog node $j$ can further improve the response-time of its users. We will give a more detailed discussion on how to achieve the optimal solution in Section V.

### B. Extending to General Cooperative Fog Computing Networks

We can further extend the above results into a general cooperative fog computing network with more than two fog nodes. In this network, each fog node can forward its offloaded workload to the other fog nodes in the fog layer and at the same time help other fog nodes to process their offloaded workload. In this case, each fog node $j$ can divide its offloaded workload into $N+1$ partitions including one partition of workload $\varphi_{jj}$ to be processed by itself, one partition $\varphi_{jc}$ to be forwarded to the cloud, and $N-1$ partitions, denoted as $\varphi_{jk}$ for the partition sent to fog node $k \in \mathcal{C}_j$, that will be send to other fog nodes in the fog layer. Note that it is not necessary for each fog node to always forward part of its offloaded workload to all the other fog nodes in the fog layer. If fog node $j$ does not forward any offloaded workload to fog node $i$, we have $\varphi_{ji} = 0$ for $i \ne j$ and $i, j \in \mathcal{F}$. We refer to $\boldsymbol{\varphi}_{j\bullet} = \langle \varphi_{ji} \rangle_{i \in \mathcal{F}\setminus\{j\}}$ as the request vector of fog node $j$. We also refer to $\boldsymbol{\phi}_{\bullet i} = \langle \varphi_{ji} \rangle_{j \in \mathcal{F}\setminus\{i\}}$ as the service vector of fog node $i$. Let $\boldsymbol{\phi} = \langle \varphi_{ji} \rangle_{i,j \in \mathcal{F}}$ be the workload processing matrix for the entire fog layer. We have $\sum_{k \in \mathcal{C}_j} \varphi_{jk} + \varphi_{jj} \le 1 \ \forall j \in \mathcal{F}$. The response-time of fog node $j \in \mathcal{F}$ can then be written as

$$R_j^{C3}(\xi_j, \boldsymbol{\varphi}_{j\bullet}) = \tau_j^u + \frac{1}{\sum_{i \in \mathcal{F}} \lambda_i}\left[\varphi_{jj}\left(\frac{1}{\mu_j - \varphi_{jj}}\right) + \sum_{i \in \mathcal{C}_j} \varphi_{ji}\left(\tau_{ji} + \frac{1}{\mu_i - \sum_{k \in \mathcal{F}} \varphi_{ki}}\right)\right] + \varphi_{ic}\tau^c, \quad (9)$$

where $\varphi_{ic} = 1 - \sum_{j\in\mathcal{F}\setminus\{i\}} \varphi_{ij}$, $\varphi_{kj} = \lambda_k \varphi_{kj}$ is the amount of workload processed by fog node $j$ for fog node $k$ and $\boldsymbol{\varphi}_{k\bullet} = \langle \varphi_{kj} \rangle_{j\in\mathcal{F}\setminus\{k\}}$ is the vector of workload to be forwarded by fog node $k$ to other fog nodes.

We can rewrite the optimization problem in (3) with (9) as follows.

$$\max_{\boldsymbol{\varphi}_{1\bullet},\ldots,\boldsymbol{\varphi}_{N\bullet}} \sum_{j=1}^{N} R_j^{C3}\left(\xi_j, \boldsymbol{\varphi}_{j\bullet}\right) \tag{10}$$

$$\text{s.t.} \sum_{k\in\mathcal{C}_j} \varphi_{jk} + \varphi_{jj} + \varphi_{jc} = \lambda_j, \tag{11}$$

$$\sum_{k\in\mathcal{F}} \varphi_{kj} \leq \min\{\mu_j, \chi_j\}, 0 \leq \varphi_{kj} \leq \lambda_k, \forall k, j \in \mathcal{F}. \tag{12}$$

It can be observed that, in order for each fog node $j$ to calculate the portions of workload to be forwarded to other fog nodes, fog node $j$ needs to know the workload processing capabilities and the workload arrival rates of all the other fog nodes which can be private information and impossible to be known by fog node $j$. In the next section, we will propose a distributed optimization framework based on distributed ADMM via variable splitting which allows all the fog nodes to jointly optimize the average response-time of the fog layer without disclosing their private information.

### *C. QoE and Power Efficiency Tradeoff for Cooperative Fog Computing Networks*

In Figure 3(a), we present the minimized response-time of the fog layer in a cooperative fog computing network calculated by solving problem (10). Note that the workload processed by each fog node can consist of both its own received workload and the workload sent from other fog node. We can observe that the response-time of the fog layer is closely related to the amount of workload processed by each fog node. We use black grid to highlight the area between the response-time minimization solution and the power efficiency maximization solution with a fixed maximum tolerable response-time. By substituting the power efficiency definition in (1), we can also present the relationship between the fog layer's response-time and each fog node's power efficiency for a two-node cooperative fog computing network with offload forwarding in Figure 3(b). Similar to the single-node fog computing, we can observe a fundamental tradeoff between the response-time of all the users served by the fog layer and the power efficiency of each fog node. In addition, even if the power consumption for each fog node to offload each unit of workload has been limited to a very small value, it is still possible to achieve the response-time constraint of each fog node using the workload forwarding.

## V. A Distributed Optimization Framework for Cooperative Fog Computing

As mentioned previously, deciding the optimal amount of workload to be processed by each fog node is important to achieve the optimal QoE and power efficiency tradeoff in cooperative fog computing networks. Unfortunately, the optimization problem in (10) is non-smooth and therefore cannot be solved by using the traditional optimization approaches that can only handle smooth objective function [13]. One popular tool to solve the non-smooth optimization problem is the ADMM-based approaches [14]. In this section, we propose a novel distributed optimization framework based on distributed ADMM via variable splitting to maximize the QoE of users with a given power efficiency constraint of fog nodes. Our proposed framework utilizes the structures of our optimization problem in (10) to decompose the original problem into $N$ subproblems each of which can be solved by each fog node using its private information. The subproblem optimization of all the fog nodes will be coordinated through a workload forwarding coordinator (WFC) which can be established by the cloud data centers or is one of the components in the cloud layer. Note that, as will be discussed later, the WFC does not need to know the maximum workload processing capability or workload arrival rate of each fog node and our algorithm only requires very limited information exchanged between each fog node and WFC.

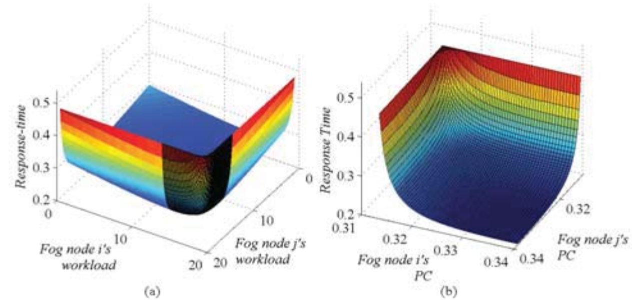


Fig. 3. Response-time under different amounts of processed workload and power consumptions (PC) for each fog node to offload each unit of workload.

Unfortunately, the traditional ADMM approach in [14] cannot be directly applied to solve our problem because of the following reasons:

1) Traditional ADMM can only be utilized to solve the optimization problem without inequality constraints [14]. However, our optimization problem in (10) includes inequality constraints and therefore cannot be directly solved by the ADMM approach.
2) Traditional ADMM approach can only solve problems with two blocks of random variables. However, the optimization problem in (10) consists of more than two variables to optimize.
3) Traditional ADMM approach is a centralized optimization approach which requires the private information of each agent to be shared with others. In addition, most existing distributed ADMM approaches are proposed to solve the consensus optimization problem in which a local copy of the model parameter has to be shared and updated by all the agents [15]. In fog computing networks, fog nodes may not always

want to share their private information with each other and hence cannot utilize these existing approaches to optimize their performance.

To solve issue 1), we introduce a set of $N+1$ indicator functions that include each of separable inequality constraints in (12) and incorporate these indicator functions into the objective function of our problem to convert the constraint optimization problem into the unconstrained one. More specifically, we define $\mathcal{G}_i = \{\boldsymbol{\varphi}_{\bullet i} : \sum_{k\in\mathcal{F}} \varphi_{ki} \le \min\{\mu_i, \bar{\xi}_i\}, \xi_{ki} > 0,\ \forall k \in \mathcal{F}\}$ as the polyhedra of each constraint of fog node $i$ in problem (3) where $\boldsymbol{\varphi}_{\bullet i} = \langle \varphi_{ki} \rangle_{k\in\mathcal{F}\setminus\{i\}}$ is the vector of amounts of workload to be processed by fog node $i$ for other fog nodes. We then can define an indicator function $\mathbf{I}_{\mathcal{G}_i}(\boldsymbol{\psi}_i)$ as follows:

$$\mathbf{I}_{\mathcal{G}_i}(\boldsymbol{\varphi}_{\bullet i}) = \begin{cases} 0, & \boldsymbol{\varphi}_{\bullet i} \in \mathcal{G}_i, \\ +\infty, & \boldsymbol{\varphi}_{\bullet i} \notin \mathcal{G}_i. \end{cases} \tag{13}$$

We also introduce an indicator function to characterize the inseparable constraint in (11) which is defined as

$$\mathbf{I}_{\mathcal{G}_c}(\boldsymbol{\psi}) = \begin{cases} 0, & \boldsymbol{\psi} \in \mathcal{G}_c, \\ +\infty, & \boldsymbol{\psi} \notin \mathcal{G}_c, \end{cases} \tag{14}$$

where $\boldsymbol{\psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_N]$, $\mathcal{G}_c = \{\boldsymbol{\psi} : \sum_{i=1}^{N} I_N \boldsymbol{\psi}_i \le 1\}$, $I_N$ is an identity matrix with size $N$, $\boldsymbol{\psi} \in \mathbf{R}^{N\times N}$, $\boldsymbol{\psi}_i \in \mathbf{R}^N$.

By including the above indicator functions into the objective function of our optimization problem, we can convert the original problem (3) with inequality constraints into the following optimization problem without inequality constraints.

$$\begin{aligned} \min_{\boldsymbol{\varphi}_{\bullet 1},\ldots,\boldsymbol{\varphi}_{\bullet N},\boldsymbol{\psi}} & \sum_{i\in\mathcal{F}} \left( R_i^{C3}(\xi_i, \boldsymbol{\varphi}_{\bullet i}) + \mathbf{I}_{\mathcal{G}_i}(\boldsymbol{\varphi}_{\bullet i}) \right) \\ & + \mathbf{I}_{\mathcal{G}_c}(\boldsymbol{\psi}) \\ \text{s.t.} \quad & \boldsymbol{\varphi}_{\bullet i} - \boldsymbol{\psi}_i = 0, \forall i \in \mathcal{F}. \end{aligned} \tag{15}$$

To solve issue 2), we need to first convert the original optimization problem with multiple random variables in (15) into the form with two blocks of random variables. Following the same line as [16], we can show that the solution of the optimization problem in (15) is equivalent to solving the optimization problem with the following augmented Lagrangian form with two blocks of random variables. We can write the $\boldsymbol{\varphi}$-optimization subproblem as

$$\begin{aligned} \boldsymbol{\varphi}^{t+1} &= \arg\min_{\boldsymbol{\varphi}} \mathcal{L}_\rho \left( \boldsymbol{\varphi}_{\bullet 1}, \boldsymbol{\varphi}_{\bullet 2}, \ldots, \boldsymbol{\varphi}_{\bullet N}, \boldsymbol{\psi}^t, \boldsymbol{\Lambda}^t \right) \\ &= \arg\min_{\boldsymbol{\varphi}} \sum_{i\in\mathcal{F}} \Big\{ R_i^{C3}(\xi_i, \boldsymbol{\varphi}_{i\bullet}) + \mathbf{I}_{\mathcal{G}_i}(\boldsymbol{\varphi}_{\bullet i}) \\ &\quad - \boldsymbol{\Lambda}_i^t \left( \boldsymbol{\varphi}_{\bullet i} - \boldsymbol{\psi}_i^t \right) + \frac{\rho}{2} \|\boldsymbol{\varphi}_{\bullet i} - \boldsymbol{\psi}_i^t\|_2^2 \Big\}, \end{aligned} \tag{16}$$

where $\rho$ is the augmented Lagrangian parameter and $\boldsymbol{\Lambda}$ is the matrix of the dual variables [16].

We can write the $\boldsymbol{\psi}$-updating problem as

$$\boldsymbol{\psi}^{t+1} = \arg\min_{\boldsymbol{\psi}} \frac{\rho}{2} \|\boldsymbol{\varphi}^{t+1} - \boldsymbol{\psi}^t + \frac{1}{\rho} \boldsymbol{\Lambda}^t\|_2^2 + \mathbf{I}_{\mathcal{G}_c}(\boldsymbol{\psi}). \tag{17}$$

The dual variable update sub-problem can then be written as follows

$$\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^t - \rho \left( \boldsymbol{\varphi}^{t+1} - \boldsymbol{\psi}^{t+1} \right). \tag{18}$$

We can observe that the subproblem optimization in (16)–(18) is equivalent to the form of the traditional ADMM with two random variables: $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$.

Finally, to solve issue 3), we need to prove that the augmented Lagrangian form of our optimization problem in (16) can be decomposed into $N$ subproblems each of which can be individually solved by each fog node using its private information. More specifically, we can prove that the augmented Lagrangian form of our optimization problem has the following features.

**Theorem 1:** The augmented Lagrangian form of the objective function of our optimization problem in (10) is separable and convex.

*Proof:* First, we prove the augmented Lagrangian form of the objective function of our optimization problem is separable. We can rewrite (10) as follows:

$$\mathcal{L}_\rho \left( \boldsymbol{\varphi}_{\bullet 1}, \boldsymbol{\varphi}_{\bullet 2}, \ldots, \boldsymbol{\varphi}_{\bullet N}, \boldsymbol{\psi}, \boldsymbol{\Lambda} \right) = \sum_{i=1}^{N} \mathcal{L}_{S_i} \left( \boldsymbol{\varphi}_{\bullet i}, \boldsymbol{\psi}_i, \boldsymbol{\Lambda}_i \right), \tag{19}$$

where

$$\begin{aligned} \mathcal{L}_{S_i}(\boldsymbol{\varphi}_{\bullet i}) &= S_i(\boldsymbol{\varphi}_{\bullet i}) + \mathbf{I}_{\mathcal{G}_i}(\boldsymbol{\varphi}_{\bullet i}) + \Lambda_i^T (\boldsymbol{\varphi}_{\bullet i} - \boldsymbol{\psi}_i) \\ &\quad + \frac{\rho}{2} \|\boldsymbol{\varphi}_{\bullet i} - \boldsymbol{\psi}_i\|_2^2, \end{aligned} \tag{20}$$

and $S_i(\boldsymbol{\varphi}_{\bullet i})$ is defined as

$$\begin{aligned} S_i(\boldsymbol{\varphi}_{\bullet i}) &= \tau_i^u + \boldsymbol{\varphi}_{ii} \left( \frac{1}{u_i - \boldsymbol{\varphi}_{ii}} \right) \\ &\quad + \sum_{j\in\mathcal{F}\setminus\{i\}} \boldsymbol{\varphi}_{ji} \left( \tau_{ji} + \frac{1}{u_i - \boldsymbol{\varphi}_{ii} - \sum\limits_{k\in\mathcal{F}\setminus\{i\}} \boldsymbol{\varphi}_{ki}} \right) \\ &\quad - \sum_{j\in\mathcal{F}} \boldsymbol{\varphi}_{ji} \tau^c + \tau^c. \end{aligned} \tag{21}$$

It can be observed that the variables in $\mathcal{L}_{S_i}$ can be calculated by fog node $i$ and are independent with the variables associated with other fog nodes. This proves that augmented Lagrangian form of our optimization problem (10) can be separated into $N$ subproblems each of which can be solved by each fog node using its private information.

Let us prove that the objective function of problem (10) is also convex. It can be directly shown that the domain of variables in the objective function of (10) is a polyhedra which is a convex set. We can also show that the second derivative of each individual item in $S_i(\boldsymbol{\varphi}_{\bullet i})$ is always positive which means that it is a convex function with respect to each individual variable. Following the property that a nonnegative weighted sum of convex function $f = \sum_{i=1}^{N} c_i f_i$, $f : \mathbf{R}^N \to \mathbf{R}$ is convex if and only if $f_i$ is convex and $c_i$ is a constant for all $i \in \{1, 2, \ldots N\}$, we can prove that the objective function of problem (3) is convex. ■

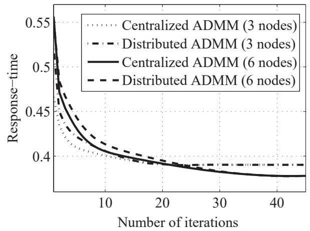


Fig. 4. Convergence rate of Algorithm 1.

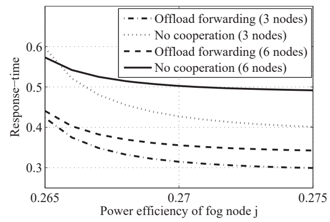


Fig. 5. Response time under different power efficiencies.

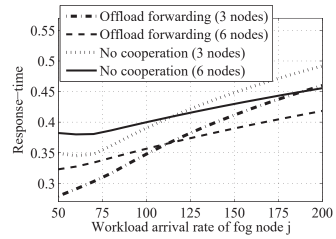


Fig. 6. Response time under different workload arrival rates.

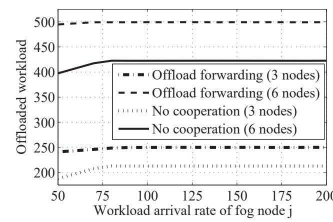


Fig. 7. The amount of workload offloaded by fog nodes under different workload arrival rates.

The first part of the above theorem proves that each fog node $i$ only need to calculate its own separable sub-problem individually. More specifically, each fog node $i$ will calculate the optimal service vector $\boldsymbol{\varphi}^*_{\bullet i}$ by solving the following sub-problem:

$$\boldsymbol{\varphi}^{t+1}_{\bullet i} = \arg\min_{\varphi_{\bullet i}} \mathcal{L}_{S_i}\left(\boldsymbol{\varphi}_{\bullet i}, \boldsymbol{\psi}^t_i, \Lambda^t_i\right) \tag{22}$$

The second part of Theorem 1 proves that if there exists an algorithm that can converge to an optimal solution, this optimal solution will be unique and stable.

Let us present the detailed description of our algorithm below.

**Algorithm 1: Distributed Optimization for Workload Forwarding**

*Initialization*: Each fog node $i$ chooses an initial service vector $\boldsymbol{\varphi}^0_{\bullet i}$ and WFC chooses an initial dual variable $\boldsymbol{\Lambda}^0$.

WHILE t=0, 1, ...

i) *Fog node updating*: Each fog node $i$ calculates $\boldsymbol{\varphi}^{t+1}_{\bullet i}$ by solving (22) and then sends the resulting $\boldsymbol{\varphi}^{t+1}_{\bullet i}$ and $\lambda_k$ to the WFC,

ii) *WFC Updating*: WFC calculates $\boldsymbol{\psi}^{t+1}$ by solving $\psi$-updating problem in (17).

iii) *Dual Variable Updating*: WFC updates dual variables $\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^k - \rho\left(\boldsymbol{\varphi}^{t+1} - \boldsymbol{\psi}^{t+1}\right)$ and sends $\boldsymbol{\varphi}^{t+1}_i$ and $\Lambda^{t+1}_i$ to fog node $i$.

ENDWHILE

In the above algorithm, each fog node $i$ needs to solve problem (22) with its own private information of fog node $i$. Each fog node $i$ sends its optimized solution $\boldsymbol{\varphi}_{\bullet i}$ to the WFC and WFC will then update the dual variable for all the fog nodes and send each fog node $j$ with its individual dual variable related to the sub-problem in (22). In other words, in Algorithm 1, each fog node does not need to disclose its private information and still can achieve the global optimal solution of (10).

We have the following result for the above algorithm.

**Theorem 2:** Algorithm 1 converges to the global optimal solution of Problem (3) with convergence rate of $O\left(1/t\right)$[1].

*Proof:* The convergence of Algorithm 1 follows directly from the standard ADMM approach. We omit the detailed description due to limit of space. ∎

## VI. Numerical Results

In this section, we consider fog computing network supported by a wireless network system to evaluate the workload offloading performance of our proposed approaches. In this systems, users are mobile devices with computational intensive applications, and the fog nodes are the access points (e.g., cellular base stations in cellular networks or Wi-Fi access points for a Wi-Fi network.) managed by a network operator. Note that the network operator cannot control the cloud layer but can only manage the fog nodes in the fog layer. Users and fog nodes are connected with the wireless communication channels and the round trip workload submission time from each user to its closest fog node is given by $\tau^u_j = 1/\left(\log_2\left(1 + \frac{h}{\sqrt{d^3}}w\right)\right)$ where $w$ is the transmit power, $d$ is the distance between each user and its closest fog node, and $h$ is the ratio of channel coefficient to the additive noise level. Fog nodes can communicate with cloud data center through the Internet. We assume the round trip workload submission time between any fog node and the cloud data center is set to $\tau^c = 50ms$. We assume there exist local communication links among fog nodes and the round trip workload forwarding time between any two fog nodes is the same given by $\tau_{ij} = 20ms$. Each fog node serves 10 users within the coverage area and the distance between each fog node and its associated fog nodes follows a uniformly randomly distribution between 20 and 200 meters, a range that covers the small cell and Wi-Fi networks. We assume each user will only submit its workload to the closest fog nodes with a constant rate.

We first consider the convergence performance of Algorithm 1 for a fog computing network with different number of fog nodes in Figure 4. We can observe that our proposed algorithm can converge to the global optimal solution within the first few iterations (less than 30 iterations in both cases). In addition, the network size (number of fog nodes in the network) does not have any noticeable affect on

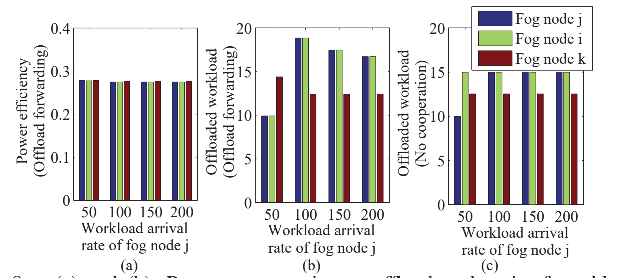


Fig. 8. (a) and (b): Power consumption to offload each unit of workload and amount of offloaded workload of each fog node with different workload arrival rates of fog node $j$, and (c) amount of offloaded workload without fog node cooperation.

[1]We follows Bachmann-Landau notations: $f = O(g)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} < +\infty$.

the convergence rate of our algorithm. We also present the convergence rate when the centralized ADMM in [14] is applied to solve our optimization problem in (10). We can observe that our proposed algorithm achieves similar convergence performance as the centralized one.

In Figure 5, we present the response-time and power efficiency tradeoff curves for a fog computing network with and without fog node cooperation when only one fog node (fog node $j$) changes its power efficiency while the power efficiency of others are fixed. We observe that offload forwarding significantly reduces the response-time of the users especially when the power consumption of fog node $j$ is low. We can also observe that, for the given power efficiency of all the fog nodes, the response-time can be significantly reduced when the number of fog nodes increases. In other words, our proposed offload forwarding strategy can further improve the QoE and power efficiency tradeoff for cooperative fog computing networks.

In Figures 6 and 7, we investigate the effect of fog nodes' workload arrival rates on the users' response time and the workload to be offloaded by the fog layer, respectively. We again fix the workload arrival rates of all the fog nodes except fog node $j$. We can observe in Figures 6 that, compared to the case without fog node cooperation, offload forwarding reduces the growing speed of the response time when the workload arrival rate of fog node $j$ increases. We can observe in Figure 7 that the amount of workload to be offload by the fog nodes increases almost linearly when the workload arrival rate for fog node $j$ is small. However, with the workload arrival rate continues to grow, the total amount of offloaded workload that can be offloaded by the fog layer approaches to a fixed value.

In Figure 8, we compare the power consumption and offloaded workload of each individual fog node with and without fog node cooperation under various workload arrival rates of fog node $j$. We can observe that our proposed offload forwarding strategy can balance the power efficiency and workload offloading performance of different fog nodes. More specifically, fog node $j$ will process the workload for other when its workload arrival rate is small and will forward part of the offloaded workload to others when its received workload approaches the maximum workload processing capability.

## VII. Conclusion and Future Work

In this paper, the workload offloading problem has been studied for fog computing networks. We have investigated the relationship between two performance metrics for fog computing networks: users' QoE and fog nodes' power efficiency. We have discussed the tradeoff between these two metrics for a single-node fog computing network. We then extend our result into the fog computing network with fog node cooperation. In this network, fog nodes can help each other to jointly offload workload from the cloud layer. We have introduced a novel fog node cooperation strategy referred to as the offload forwarding. In this strategy, each fog node can forward part or all of its offloaded workload to other local fog nodes to further improve the QoE of users. We have studied the workload allocation problem for offload forwarding-enabled cooperative fog computing networks in which each fog node can decide the optimal partitions of workload to be forwarded to other fog nodes as well as those to be processed for other fog nodes under a given power efficiency constraint. We have investigated the QoE and power efficiency tradeoff for cooperative fog computing networks and propose a distributed ADMM via variable splitting algorithm to approach the global optimal workload allocation that maximizes users' QoE under a given power efficiency of fog nodes. Finally, we have considered a wireless network-supported fog computing system as a case study to verify the performance of our proposed approach. Numerical results have been presented to verify the performance of our approach.

## References


[1] M. Chiang, "Fog networking: An overview on research opportunities," *arXiv preprint arXiv:1601.00835*. [Online]. Available: http://arxiv.org/pdf/1601.00835

[2] D. Hoang, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[3] A. Vahid Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog Computing: Principals, Architectures, and Applications," *ArXiv preprints*, Jan. 2016.

[4] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. of the IEEE HotWeb*, Washington DC, Nov. 2015, pp. 73–78.

[5] D. Ardagna, G. Casale, M. Ciavotta, J. F. Pérez, and W. Wang, "Quality-of-service in cloud computing: modeling techniques and their applications," *Journal of Internet Services and Applications*, vol. 5, no. 1, p. 11, 2014.

[6] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang, "How to bid the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 71–84, 2015.

[7] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. of the IEEE PerCom Workshops*, St. Louis, MO, Mar. 2015, pp. 105–110.

[8] G. Lewis, S. Echeverria, S. Simanta, B. Bradshaw, and J. Root, "Tactical cloudlets: Moving cloud computing to the edge," in *Proc. of the IEEE MILCOM*, Baltimore, MD, Oct. 2014, pp. 1440–1446.

[9] P. Casas and R. Schatz, "Quality of experience in cloud services: Survey and measurements," *Computer Networks*, vol. 68, pp. 149 – 165, 2014, communications and Networking in the Cloud.

[10] N. Tran, C. Do, S. Ren, Z. Han, and C. S. Hong, "Incentive mechanisms for economic and emergency demand responses of colocation datacenters," *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 12, pp. 2892–2905, Dec 2015.

[11] A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya *et al.*, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in computers*, vol. 82, no. 2, pp. 47–111, 2011.

[12] M. Keller and H. Karl, "Response-time-optimized distributed cloud resource allocation," *arXiv preprint arXiv:1601.06262*, 2016. [Online]. Available: http://arxiv.org/abs/1601.06262

[13] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[15] R. Zhang and J. T. Kwok, "Asynchronous distributed admm for consensus optimization." in *Proc. of the ICML*, Beijing, China, Jun. 2014, pp. 1701–1709.

[16] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block admm with o (1/k) convergence," *arXiv preprint arXiv:1312.3040*, 2013.