A Bayesian Game Formulation of Power Dissipation and Response Time Minimization in a Mobile Cloud Computing System

Yanzhi Wang, Xue Lin, and Massoud Pedram
Department of Electrical Engineering
University of Southern California
Los Angeles, USA
{yanzhiwa, xuelin, pedram}@usc.edu

Abstract—**The rapidly developing cloud computing and virtualization techniques provide mobile devices with battery energy saving opportunities by allowing them to offload computation and execute applications remotely. A mobile device should judiciously determine whether to offload computation and which portion of application should be offloaded to the cloud. This paper considers a mobile cloud computing (MCC) interaction system consisting of multiple mobile devices and the cloud computing system. A Bayesian game formulation is proposed for the MCC interaction system. In this game, each mobile device determines the portion of its service requests for remote processing in the cloud computing system. All the mobile devices compete for the allocated resources in the data center. Each mobile device is aware of its own service request generating rate through effective prediction methods. It has only partial information about the other mobile devices. The objective of each mobile device is to minimize its power consumption as well as the service request response time. This paper proves that pure strategy Bayesian-Nash equilibrium in this game always exists and is unique. The optimal strategy for all the mobile devices achieving the Bayesian-Nash equilibrium is derived using convex optimization technique. Experimental results demonstrate the effectiveness of the proposed Bayesian game-based optimization framework. The mobile devices can achieve simultaneous reduction in average power consumption and average service request response time, by 27.3% and 63.7%, respectively, compared with baseline methods.**

Keywords-mobile cloud computing; mobile devices; game theory; Bayesian game; resource allocation

I. INTRODUCTION

Cloud computing has been envisioned as the next-generation computing paradigm for its advantages in on-demand service, ubiquitous network access, location independent resource pooling, and transference of risk [1]. Cloud computing shifts the computation and storage resources from the network edges to a "Cloud" from which businesses and users are able to access applications from anywhere in the world on demand [2][3][4]. In the cloud computing paradigm, the capabilities of business applications are exposed as sophisticated services that can be accessed over a network. Cloud service providers are incentivized by the profits by charging the remote clients for accessing these services. Clients are attracted by the opportunity for reducing or eliminating costs associated with "in-house" local provision of these services.

The underlying infrastructure of cloud computing consists of data centers and clusters of servers that are monitored and maintained by the cloud service providers [6]. These data centers have massive computation and storage capabilities. Service providers often end up over-allocating or over-provisioning their resources in these data centers in order to meet the clients' response time requirements or other service level agreements (SLAs) [5]. Such over-provisioning may increase the cost incurred on the data centers in terms of both the electrical energy cost and the carbon emission. Hence, optimal resource provisioning or allocation in the cloud computing system (or in the broader area of distributed computing system) is imperative in order to reduce the cost incurred on the data centers as well as the environmental impact, and has been investigated extensively in [7]-[13].

The emerging paradigm of mobile cloud computing (MCC) shifts the processing, memory, and storage requirements all together from the resource limited mobile devices to the resource unlimited cloud computing system [14]-[16]. MCC provides multiple potential advantages for the remote mobile devices [17][18], including improvement of the storage capacity for mobile users and reducing the risk of data and application lost on mobile devices by backing up users' data in the cloud. The potentially most important benefit for the mobile users is the extension of battery's operation time. The MCC system helps the mobile devices run the computation intensive applications, which typically consume a large amount of battery energy when running locally in the mobile devices. This is enabled by the recently developed virtualization techniques that allow the cloud to run mobile applications for the remote mobile devices [19]. This technique is referred to as *computation offloading* in the reference work [18][20].

*This research is sponsored in part by the Software and Hardware Foundations program of the NSF's Directorate for Computer & Information Science & Engineering.

The mobile devices should judiciously make decisions about whether to perform computation offloading and which portion of the running application should be offloaded to the cloud. Reference [18] provides an analysis and guideline on the conditions that computation offloading could help save the energy for mobile devices. It concludes that an application or task with high computation but limited data communication requirement could benefit from computation offloading. Reference [20] proposes MAUI to perform runtime dynamic control of computation offloading, by formulating the computation offloading problem as a linear programming optimization problem. Reference [21] provides a similar approach for the Android applications. Furthermore, the mobile devices should also be aware of other devices and the potential congestion level in the remote data center if all the mobile devices decide to offload their computations simultaneously. Reference [22] provides a congestion game-based optimization framework for the MCC system, in which each mobile device is a player and his strategy is to select one of the available servers in the cloud to offload computation. In the realistic cloud computing facilities, however, a centralized request dispatcher selects the target server for each service request (i.e., request for computation offloading) generated from the mobile devices [4]. The mobile devices do not select the target servers themselves.

In this paper, we consider an MCC interaction system consisting of multiple mobile devices and the cloud computing system. Each mobile device executes an application and generates service requests, which could either be processed locally in the mobile device or remotely in the cloud computing system through computation offloading. The cloud computing system consists of multiple servers dedicated for processing mobile service requests inside a data center. Service requests from the mobile devices are free to be dispatched to any server in the cloud computing system. The cloud computing controller allocates a portion of resources in each server for service request processing. The resource allocation results in the cloud computing system are pre-announced to the mobile devices.

In the MCC interaction system, each mobile device determines its optimal portion of service requests to be remotely processed in the cloud computing system. All the mobile devices compete for the allocated resources in the data center. The objective of each mobile device is to minimize a weighted combination of its average power consumption and average response time of service requests. Each mobile device is aware of its own service request generating rate through effective prediction methods. On the other hand, it is only aware of the probability distribution of the service request generating rates of the other mobile devices in the MCC system. Similarly, each mobile device is only aware of the probability distribution of the relative weight in the other devices between the average power consumption and average service request response time.

We provide a Bayesian game-based optimization framework [35] for the mobile devices in the MCC system since each mobile device has only partial information about the others. Each player in the Bayesian game is a mobile device and its strategy is the portion of service requests for remote processing. All the players in the game choose its strategy simultaneously. We prove that the pure strategy Bayesian-Nash equilibrium [34][35] of this game always exists and is unique. The Bayesian-Nash equilibrium is the optimal strategy profile in the Bayesian game in the sense that no player can find better strategy if he deviates from the current strategy unilaterally [34]. We derive the optimal strategy of each mobile device achieving the Bayesian-Nash equilibrium using convex optimization method [32]. Experimental results demonstrate the effectiveness of the proposed Bayesian game-based optimization framework of the MCC interaction system.

The rest of this paper is organized as follows. The MCC system model, including models for the mobile devices and the resource allocation framework in the cloud computing system, is presented in Section II. The Bayesian game-based formulation and optimization of the MCC interaction system are presented in Section III and Section IV, respectively. Experimental results are presented in Section V and the conclusion is in the last section.

II. Mobile Cloud Computing System Model

Consider an MCC interaction system comprised of N distributed mobile devices and a cloud computing system. These mobile devices such as smartphones, tablet PCs, are connected to the cloud through WiFi or 3G network. Each mobile device in the MCC system is identified by a unique ID, represented by index *i*. Figure 1 shows the *i*-th ($1 \leq i \leq N$) mobile device. Each *i*-th mobile device executes an application and generates service requests, which could either be processed locally in the mobile device or remotely in the cloud computing system through computation offloading.

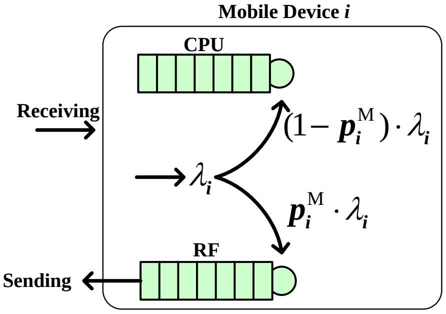
Mobile Device *i*
CPU
Receiving
$(1-\boldsymbol{p}_i^{\mathrm{M}})\cdot\lambda_i$
λ_i
$\boldsymbol{p}_i^{\mathrm{M}}\cdot\lambda_i$
RF
Sending

Figure 1. Conceptual structure for the mobile device: local or remote processing?

To find an analytical form of the average response time, service requests generated from the *i*-th mobile device are assumed to follow a Poisson process with an average generating rate of λ_i, which is predicted based on the behavior of the application. To be more realistic, each *i*-th mobile device knows its own λ_i value, whereas it is only aware of the probability distribution of the $\lambda_{i'}$ values of the other mobile devices in the MCC system.

The *i*-th mobile device chooses to offload each service request for remote processing in the cloud with probability $p_i^{\mathbf{M}}$, where the superscript **M** stands for 'mobile'. We name $p_i^{\mathbf{M}}$ the *offloading probability* of the *i*-th mobile device. These probability values for mobile devices are the optimization variables in the MCC optimization framework. According to the properties of the Poisson distribution [30], service requests that are generated from the *i*-th mobile device and processed remotely in the cloud follow a Poisson process with an average rate of $p_i^{\mathbf{M}} \cdot \lambda_i$, called the *offloading rate*. The service requests that are generated from the *i*-th mobile device and processed locally in the device follow a Poisson process with an average rate of $\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i$. When $p_i^{\mathbf{M}}$ becomes larger, the average response time for the locally processed service requests decreases while the average response time for remotely processed requests increases (due to the average delay increasing in sending/receiving a service request and request processing in the cloud.) In the perspective of power consumption of the *i*-th mobile device, the power consumption in the mobile CPU (for service requests for local processing) decreases while the power consumption in the radio frequency (RF) components for sending the service requests increases. Therefore, it is crucial for each mobile device to judiciously choose the optimal $p_i^{\mathbf{M}}$ considering the characteristics of service requests (i.e., computation and communication requirements), the anticipated probability distribution of offloading rate $p_{i'}^{\mathbf{M}} \cdot \lambda_{i'}$ of the other mobile devices, as well as the anticipated congestion level in the data center.

Let $\mu_i^{\mathbf{M}}$ denote the average service request processing rate in the *i*-th mobile device. Then the average response time of the locally processed service requests in the *i*-th mobile device is given by:

$$R_i^{\mathbf{M}}\left(p_i^{\mathbf{M}}; \lambda_i\right) = \frac{1}{\mu_i^{\mathbf{M}} - \left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i} \tag{1}$$

Let $\mu_i^{\mathbf{S}}$ denote the average speed in service request sending in the *i*-th mobile device, where the superscript **S** stands for 'sending'. We calculate as follows the average time for service request to wait in the mobile device before it is completely sent out:

$$R_i^{\mathbf{S}}\left(p_i^{\mathbf{M}}; \lambda_i\right) = \frac{1}{\mu_i^{\mathbf{S}} - p_i^{\mathbf{M}} \cdot \lambda_i} \tag{2}$$

$\mu_i^{\mathbf{S}}$ is proportional to the wireless channel capacity from the mobile device to the access point [25].

The power consumption in the *i*-th mobile device consists of two parts: (i) power consumption in the mobile CPU for local service request processing, and (ii) power consumption in the RF components (e.g., WiFi, 3G components) for sending the service requests to the cloud [28][29]. Both the CPU power consumption and the RF components power consumption can be further separated into a *dynamic power consumption* part when the CPU or RF components are active (i.e., when they are processing or sending service requests) and a *static power consumption* part. The average dynamic power consumption in the CPU of the *i*-th mobile device, denoted by $P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}; \lambda_i\right)$, is proportional to the portion of time that the CPU is active, given by $\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i / \mu_i^{\mathbf{M}}$. We calculate $P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}; \lambda_i\right)$ using:

$$P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}; \lambda_i\right) = \frac{\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i}{\mu_i^{\mathbf{M}}} \cdot P_{CPU,i}^{dyn,max} \tag{3}$$

where $P_{CPU,i}^{dyn,max}$ is the dynamic power consumption when the mobile CPU is active. Similarly, the average dynamic power consumption in the RF components of the *i*-th mobile device is given by:

$$P_{RF,i}^{dyn}\left(p_i^{\mathbf{M}}; \lambda_i\right) = \frac{p_i^{\mathbf{M}} \cdot \lambda_i}{\mu_i^{\mathbf{S}}} \cdot P_{RF,i}^{dyn,max} \tag{4}$$

On the other hand, the (average) static power consumptions in the CPU and the RF components of the *i*-th mobile device are constant values denoted by $P_{CPU,i}^{sta}$ and $P_{RF,i}^{sta}$, respectively. The overall power consumption in the *i*-th mobile device is given by

$$\begin{aligned} P_{Mobile,i}\left(p_i^{\mathbf{M}}; \lambda_i\right) = P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}; \lambda_i\right) + P_{RF,i}^{dyn}\left(p_i^{\mathbf{M}}; \lambda_i\right) \\ + P_{CPU,i}^{sta} + P_{RF,i}^{sta} \end{aligned} \tag{5}$$

Figure 2 shows the structure of the target resource allocation system in cloud computing with a service request pool, a data center as the service provider as well as a central resource management node. The data center consists of *M* potentially heterogeneous servers that are dedicated for service request processing from the mobile devices (clients). We use *j* as the index of the servers in the data center.

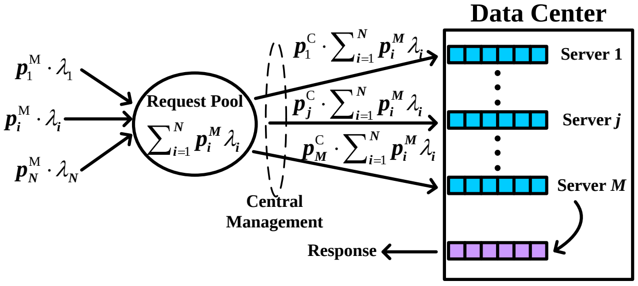
Data Center
$p_1^{\mathrm{M}} \cdot \lambda_1$
$p_i^{\mathrm{M}} \cdot \lambda_i$
$p_N^{\mathrm{M}} \cdot \lambda_N$
Request Pool
$\sum_{i=1}^{N} p_i^{M} \lambda_i$
$p_1^{\mathrm{C}} \cdot \sum_{i=1}^{N} p_i^{M} \lambda_i$
$p_j^{\mathrm{C}} \cdot \sum_{i=1}^{N} p_i^{M} \lambda_i$
$p_M^{\mathrm{C}} \cdot \sum_{i=1}^{N} p_i^{M} \lambda_i$
Server 1
Server *j*
Server *M*
Central Management
Response

Figure 2. Conceptual structure of the resource allocation system in cloud computing.

The service request pool contains the service requests generated from all the remote mobile devices. According to the properties of the Poisson distribution [30], the total service request generating rate of the request pool is given by $\sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i$. A service request can be dispatched to any server in the data center. The request dispatcher assigns a request to the j-th server with probability $p_j^{\mathbf{C}}$, where the superscript $\mathbf{C}$ stands for 'cloud'. According to the properties of the Poisson distribution [30], the service requests arriving at the j-th server follow a Poisson process with an average arrival rate of $p_j^{\mathbf{C}} \cdot \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i$. As long as a service request is dispatched to a server, the server creates a dedicated virtual machine (VM) for that service request, loads the application executable and starts execution.

Each j-th server in the cloud computing system allocates a portion of its total resources, denoted by $\phi_j^{\mathbf{C}}$ $(0 \le \phi_j^{\mathbf{C}} \le 1)$, for servicing the service requests generated from mobile devices. By using the well-known formula in M/M/1 queues [31], the average processing time of the service requests dispatched to that server is calculated as

$$R_j^{\mathbf{C}}\left(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda}\right) = \frac{1}{\phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}} - p_j^{\mathbf{C}} \cdot \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i} \qquad (6)$$

where $\mu_j^{\mathbf{C}}$ denotes the average service request processing speed when all the resources in the server are allocated for request processing.

The data center sends back the response to a service request after finishing processing it. We calculate as follows the average time for the response to wait in the data center before it is completely sent out:

$$R^{\mathbf{R}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda}) = \frac{1}{\mu^{\mathbf{R}} - \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i} \qquad (7)$$

where the superscript **R** stands for 'receiving' (i.e., the mobile device receives the response from the data center.)

Therefore, the average response time of a service request generated from the i-th mobile device (either processed locally or remotely) is given by:

$$R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}}) = \left(1 - p_i^{\mathbf{M}}\right) \cdot R_i^{\mathbf{M}}\left(p_i^{\mathbf{M}}; \lambda_i\right) + p_i^{\mathbf{M}} \cdot \left(R_i^{\mathbf{S}}(p_i^{\mathbf{M}}; \lambda_i) + \sum_{j=1}^{M} p_j^{\mathbf{C}} \cdot R_j^{\mathbf{C}}\left(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda}\right) + R^{\mathbf{R}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda}) \right) \qquad (8)$$

where $\boldsymbol{p}^{\mathbf{C}} = \{p_1^{\mathbf{C}}, p_2^{\mathbf{C}}, \dots, p_M^{\mathbf{C}}\}$ and $\boldsymbol{\phi}^{\mathbf{C}} = \{\phi_1^{\mathbf{C}}, \phi_2^{\mathbf{C}}, \dots, \phi_M^{\mathbf{C}}\}$.

III. Game Theoretic Problem Formulation

We consider the MCC interaction system comprised of the N mobile devices and the cloud computing system. We assume that the request dispatching and resource allocation results in the cloud computing system, i.e., the $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ vectors, are pre-announced to all the mobile devices. Then each of the mobile devices will determine the optimal portion $p_i^{\mathbf{M}}$ of its total service requests for remote processing in the cloud, and compete for the allocated resources in the data center. We provide a Bayesian game-based optimization framework for the mobile devices in the MCC system in the rest of this section. Each player in the Bayesian game is a mobile device and his strategy is the portion $p_i^{\mathbf{M}}$ for remote processing. All the players in the game choose strategy simultaneously. The objective of each i-th mobile device is to minimize the following objective function:

$$w_i \cdot Power_i + (1 - w_i) \cdot Latency_i \qquad (9)$$

where $Power_i$ and $Latency_i$ are the expectation values of the average power consumption and average service request response time of the i-th mobile device, respectively, since each i-th mobile device has only partial information about the other devices. The weight coefficient w_i does not have to be the same for a mobile device at all times. For example, when a mobile device's battery is full, it could reduce the value of w_i because the battery energy is not a bottleneck at this time; when its battery energy drops below a critical level, it could increase the weight on the power consumption and perhaps offload more computation. For the sake of reality, each i-th mobile device is only aware of the probability distribution of the relative weight $w_{i'}$ values of the other mobile devices in the MCC system.

Let $\theta_i = \{\lambda_i, w_i\}$ denote the *type* of the i-th mobile device. Type of players is an important standard term in the context of Bayesian game [34]. Each player in a Bayesian game has only partial information about the types of the other players. In our case, the type of a player in the Bayesian game is in a continuous space. Let $\theta_{-i} = \{\theta_1, \theta_2, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_N\}$ denote the types of all the mobile devices in the MCC system other than the i-th one. In a similar way, we define λ_{-i}, w_{-i}, and $p_{-i}^{\mathbf{M}}$. We use $Prob(\theta_{-i})$ to denote the probability distribution of θ_{-i}, which is the joint probability distribution of λ_{-i} and w_{-i}. The i-th mobile device is aware of such probability distribution $Prob(\theta_{-i})$ of other devices.

In the Bayesian game formulation, each player chooses potentially different strategies when its type is different. Hence, we use $p_{-i}^{\mathbf{M}}(\theta_{-i})$ to denote $p_{-i}^{\mathbf{M}}$ when the (joint) type of all the mobile devices other than the i-th one is given by θ_{-i}. Please note that we implicitly assume pure strategy here for each mobile device in the Bayesian game. Then based on (9), we derive the cost function[1] of the i-th mobile device, denoted by $Cost_i\left(\theta_i, \theta_{-i}, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\theta_{-i})\right)$ when θ_i, θ_{-i}, $p_i^{\mathbf{M}}$, and $p_{-i}^{\mathbf{M}}(\theta_{-i})$ are all given:

[1] which is the inverse value of the payoff function or utility function in the standard game theory context.

$$
\begin{aligned}
&Cost_i\left(\theta_i, \theta_{-i}, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\theta_{-i})\right) = \\
&\qquad w_i \cdot P_{Mobile,i}\left(p_i^{\mathbf{M}}; \lambda_i\right) + (1 - w_i) \cdot R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda})
\end{aligned} \tag{10}
$$

where $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda})$ is equivalent to $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}})$ defined in (8) since $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ are given in prior to the mobile devices. $Cost_i\left(\theta_i, \theta_{-i}, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\theta_{-i})\right)$ is a linear combination of the mobile device's power consumption $P_{Mobile,i}\left(p_i^{\mathbf{M}}; \lambda_i\right)$ and the average request response time $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda})$. The first term of (10), $w_i \cdot P_{Mobile,i}\left(p_i^{\mathbf{M}}; \lambda_i\right)$, depends only on θ_i and $p_i^{\mathbf{M}}$; while the second term of (10), $(1 - w_i) \cdot R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda})$, depends on all of the θ_i, θ_{-i}, $p_i^{\mathbf{M}}$, and $p_{-i}^{\mathbf{M}}(\theta_{-i})$ parameters[2].

In the Bayesian game formulation, each *i*-th mobile device minimizes an expectation value of $Cost_i\left(\theta_i, \theta_{-i}, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\theta_{-i})\right)$ over θ_{-i}, given by:

$$
\begin{aligned}
&COST_i\left(\theta_i, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\cdot)\right) = \\
&\int_{\theta_{-i}} Cost_i\left(\theta_i, \theta_{-i}, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\theta_{-i})\right) \cdot Prob(\theta_{-i}) \cdot d\theta_{-i}
\end{aligned} \tag{11}
$$

In Bayesian games, player (mobile device) *i* has knowledge of its own type θ_i, chooses the best-suited strategy $p_i^{\mathbf{M}}$ based on the anticipation of the strategy profile $p_{-i}^{\mathbf{M}}(\cdot)$ of the other players [34][35]. Please note that $p_{-i}^{\mathbf{M}}(\cdot)$ denotes the mapping from any θ_{-i} to $p_{-i}^{\mathbf{M}}$ of the other players than player *i*. We provide the optimization procedure for each *i*-th mobile device in the Bayesian game in Section IV.

For each mobile device in the MCC interaction system, decision making intervals can be defined based on the behavior of the dynamic parameters in the system. This is because the solution found by the presented algorithm is acceptable only when the parameters used to find the solution are approximately valid. Although some small changes in the parameters can be effectively tracked and responded to by proper reactions of the computation offloading manager in the mobile devices, large changes cannot be handled in this way. In the remainder of this paper, the computation offloading optimization problem at each decision epoch is presented and a solution is provided, but we do not discuss the estimation, prediction, and dynamic changes in the system because these issues are out of the scope of this paper.

IV. Game Theoretic Optimization

As the mobile devices are considered to be non-cooperative among each other in the Bayesian game derived in Section III for the MCC system, we are interested in the existence and uniqueness of the pure strategy Bayesian-Nash equilibrium [34][35]. Bayesian-Nash equilibrium is the optimal strategy profile in the Bayesian game in the sense that no player can find better strategy if he deviates from the current strategy unilaterally [34]. In other words, no player (mobile device) will have incentive to leave this strategy. Therefore, the Bayesian-Nash equilibrium is of particular interest to a non-cooperative Bayesian game. In this section, we first prove the existence and uniqueness of the Bayesian-Nash equilibrium in the Bayesian game derived in Section III for the MCC system. Next, we provide the optimization method for each mobile device in the MCC system in order to achieve such Bayesian-Nash equilibrium.

In the Bayesian game, each mobile device *i* determines its portion $p_i^{\mathbf{M}}$ $(0 \le p_i^{\mathbf{M}} \le 1)$ of service requests for remote processing in the cloud, in order to minimize the objective function $COST_i\left(\theta_i, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\cdot)\right)$ as defined in (11). The constraints on $p_i^{\mathbf{M}}$ are given as follows:

$$0 \le p_i^{\mathbf{M}} \le 1, \qquad \text{for } \forall i \tag{12}$$

$$\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i \le \mu_i^{\mathbf{M}} - \varepsilon, \qquad \text{for } \forall i \tag{13}$$

$$p_i^{\mathbf{M}} \cdot \lambda_i \le \mu_i^{\mathbf{S}} - \varepsilon, \qquad \text{for } \forall i \tag{14}$$

$$p_j^{\mathbf{C}} \cdot \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i \le \phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}} - \varepsilon, \qquad \text{for } \forall j \tag{15}$$

$$\sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i \le \mu^{\mathbf{R}} - \varepsilon \tag{16}$$

where $\varepsilon \ll 1$ is a small positive number, which is incorporated to make the domain of $\boldsymbol{p}^{\mathbf{M}}$ a closed set (important for the proof of the existence of the Bayesian-Nash equilibrium.) Constraints (13), (14), (15), and (16) are derived from equations (1), (2), (6), and (7), respectively. We name the Bayesian game the *Offloading Probability Decision* (OPD) game for each mobile device.

In the following, we prove the existence and uniqueness of the pure strategy Bayesian-Nash equilibrium in the OPD game.

Theorem I (*Pure Strategy Bayesian-Nash Equilibrium in the OPD Game*): The pure strategy Bayesian-Nash equilibrium in the OPD game exists and is unique.

Proof: The original OPD game satisfies: (i) The strategy spaces and type spaces are continuous; (ii) The strategy sets (constrained by (13) – (16)) and the type sets are compact (since both sets are in the Euclid space and are closed) and convex; (iii) The cost function $Cost_i\left(\theta_i, \theta_{-i}, p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}(\theta_{-i})\right)$ is continuous and strictly convex in each player's own strategy $p_i^{\mathbf{M}}$. Then the agent-normal form of the OPD game is a strictly concave game with (i) a strictly concave utility (payoff) function for each player to maximize, and (ii) a

[2] It may be not obvious to see how $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda})$ depends on w_{-i}. In fact, $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda})$ depends on $p_{-i}^{\mathbf{M}}(\theta_{-i})$, which further depends on w_{-i}.

closed convex domain for the strategy profile. In this case, the existence and uniqueness of the pure strategy Nash equilibrium are directly resulted from the first and third theorem in [24]. We know from [36] that this conclusion leads to the existence and uniqueness of the pure strategy Bayesian-Nash equilibrium in the original OPD game. ■

Each mobile device finds its optimal strategy achieving the Bayesian-Nash equilibrium of the corresponding OPD game using standard convex optimization technique [32][33], with detailed procedure shown in Algorithm 1.

In Algorithm 1, deriving the $COST_{i'}\left(\theta_{i'}, p_{i'}^{\mathbf{M}}, p_{-i'}^{\mathbf{M}}(\cdot)\right)$ function will have NP complexity when we integrate $Cost_{i'}\left(\theta_{i'}, \theta_{-i'}, p_{i'}^{\mathbf{M}}, p_{-i'}^{\mathbf{M}}(\theta_{-i'})\right)$ over $\theta_{-i'}$ using equation (11). We make this problem polynomial-time solvable as follows. Remember that only the term $R_{i'}^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{\lambda})$ in $Cost_{i'}\left(\theta_{i'}, \theta_{-i'}, p_{i'}^{\mathbf{M}}, p_{-i'}^{\mathbf{M}}(\theta_{-i'})\right)$ depends on $\theta_{-i'}$ and $p_{-i'}^{\mathbf{M}}(\theta_{-i'})$ since it depends on $\sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i = p_{i'}^{\mathbf{M}} \cdot \lambda_{i'} + \sum_{i \neq i'} p_i^{\mathbf{M}} \cdot \lambda_i$. Therefore, we first calculate the probability distribution of $\sum_{i \neq i'} p_i^{\mathbf{M}} \cdot \lambda_i$ from $\theta_{-i'}$ and $p_{-i'}^{\mathbf{M}}(\theta_{-i'})$, and then integrate $Cost_{i'}\left(\theta_{i'}, \theta_{-i'}, p_{i'}^{\mathbf{M}}, p_{-i'}^{\mathbf{M}}(\theta_{-i'})\right)$ over $\sum_{i \neq i'} p_i^{\mathbf{M}} \cdot \lambda_i$ to derive the $COST_{i'}\left(\theta_{i'}, p_{i'}^{\mathbf{M}}, p_{-i'}^{\mathbf{M}}(\cdot)\right)$ function with given $p_{-i'}^{\mathbf{M}}(\cdot)$. This overall procedure has polynomial time complexity.

Algorithm 1: Finding the Bayesian-Nash Equilibrium in the OPD Game for Each Mobile Device *i*.

```
Initialize $p_i^{\mathbf{M}}(\cdot)$ (the offloading probability of the $i$-th mobile device itself for different $\theta_i$) as well as $p_{-i}^{\mathbf{M}}(\cdot)$ (the anticipation of the offloading probabilities of other mobile devices for different $\theta_{-i}$.)
Do the following procedure iteratively:
    For each $1 \leq i' \leq N$:
        For each type $\theta_{i'}$ (discretization is needed here):
            Derive the $COST_{i'}\left(\theta_{i'}, p_{i'}^{\mathbf{M}}, p_{-i'}^{\mathbf{M}}(\cdot)\right)$ function using equation (11).
            Find the optimal $p_{i'}^{\mathbf{M}}(\theta_{i'})$ (i.e., the best response of the $i'$-th mobile device) when the type is $\theta_{i'}$, by solving the convex optimization problem for the $i'$-th mobile device with objective function $COST_{i'}\left(\theta_{i'}, p_{i'}^{\mathbf{M}}, p_{-i'}^{\mathbf{M}}(\cdot)\right)$ and constraints (12) – (16).
            Update $p_{i'}^{\mathbf{M}}(\theta_{i'})$ to be the new value.
        End
    End
Until the solution converges.
```

V. Experimental Results

In this section, we implement the interaction system of multiple mobile devices and the cloud computing system and demonstrate the effectiveness of the proposed game theory-based optimization framework.

We consider an MCC interaction system comprised of $N = 20$ (we will change this parameter later) mobile devices, as well as a cloud computing infrastructure. The data center in the cloud computing system consists of 10 servers. We use normalized amounts of most of the parameters in the MCC interaction system instead of their actual values. To make the solution practical, we assume that the type $\theta_i = \{\lambda_i, w_i\}$ of each mobile device takes discrete levels. The average service request generating rate λ_i of each mobile device is a uniformly distributed random variable over values 0.5, 0.75, 1, 1.25, 1.5. The relative weight w_i is uniformly distributed over values 0, 0.1, 0.2, 0.4, 1. Each mobile device is only aware of its own type θ_i, while it knows the probability distribution of the types of other mobile devices. The average service request processing rate $\mu_i^{\mathbf{M}}$ in the mobile CPU is 1.6. The average service request sending rate $\mu_i^{\mathbf{S}}$ in every mobile device is 2. The maximum dynamic power consumption values in each mobile CPU and RF components, $P_{CPU,i}^{dyn,max}$ and $P_{RF,i}^{dyn,max}$, are uniformly distributed between 4 and 6, and between 1 and 1.5, respectively. The static power consumption values in each mobile CPU and RF components, $P_{CPU,i}^{sta}$ and $P_{RF,i}^{sta}$, are uniformly distributed between 2 and 3, and between 1 and 1.5, respectively. In the cloud computing system, the maximum average service request processing rate $\mu_j^{\mathbf{C}}$ in each server (i.e., when all its resources are allocated for request processing) is 3. The average response sending rate $\mu^{\mathbf{R}}$ in the cloud computing system is 50. We assume that all the resources in the 10 servers are allocated for mobile request processing. The resource allocation results in the cloud computing system are announced to the mobile devices.

In the first experiment, we test on the MCC interaction system and compare the expected average power consumption and the expected average request response time of all the mobile devices using the Bayesian game-based optimization framework and using the two baseline methods. These expectation values are averaged over all the mobile devices, where the type θ_i of each mobile device is a random variable as discussed before. In the first baseline system, service requests generated from all the mobile devices are processed locally. In the second baseline system, the mobile devices send all the service requests to the cloud computing system for remote processing.

Table I illustrates comparison results on the expected average power consumption, the expected average request response time, as well as the expected value of the objective function (9) in the mobile devices. This objective function shows a desirable tradeoff of power consumption and service request response time in the mobile devices. We have the following observations: (i) When comparing with Baseline 1, the mobile devices can achieve simultaneous reduction in expected average power consumption and expected average service request response time, by 27.3% and 63.7%, respectively, when using the proposed Bayesian game-based optimization framework. This is because the mobile devices have high power consumption in the mobile CPU and large service request response time due to congestion in request processing, if all the service requests are processed locally. (ii) The mobile device can achieve significant expected average service request response time reduction, by 50.7%, compared with Baseline 2. However, the mobile devices cannot achieve reduction in expected average power consumption compared with Baseline 2 because offloading all the service requests for remote processing turns out to be the most energy-efficient policy for the mobile devices, although it may incur significant delay. (iii) With respect to the expected value of the objective function (9), the proposed Bayesian game-based optimization framework achieves 52.4% and 29.0% reduction compared with the two baseline systems, respectively, demonstrating the effectiveness of the proposed Bayesian game-based optimization framework.

Table I: Comparison on the expected average power consumption, expected average response time, and expected value of objective function (9) among the proposed system and baselines.

	Proposed System	Baseline 1	Baseline 2
Power	5.3544	7.3580	4.4120
Response time	1.4819	4.0833	3.0083
Objective function	2.4741	5.1967	3.4855

In the second experiment, we change the number N of mobile devices in the MCC interaction system. We test on the MCC interaction system under the Bayesian game-based optimization framework, and compare the expected average power consumption and the expected average service request response time of each mobile device with respect to the number N. In this experiment, the average service requests generating rate λ_i of each mobile device is uniformly distributed over 0.8, 1, 1.2, 1.4, and 1.5. Figure 3 illustrates the results of this experiment. We can observe from Figure 3 that the expected average power consumption only slightly increases with the increase of N (also subject to the random fluctuation of parameters such as $P_{CPU,i}^{dyn,max}$ and $P_{RF,i}^{dyn,max}$ of each mobile device.) On the other hand, the expected average service request response time increases with the increase of N. This is because of the increasing in the congestion level and therefore the increasing in the average service request response time in the data center when N is increased. The mobile devices are aware of the congestion and assign more service requests for local processing, which in turn increases their power consumption levels.

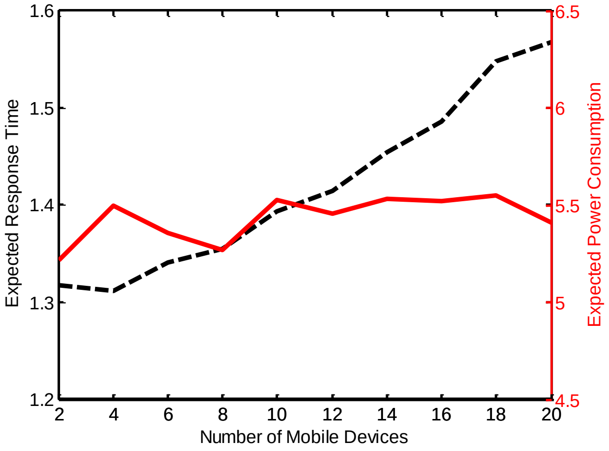

Figure 3. The expected average power consumption and the expected average response time versus the number of mobile devices using the proposed Bayesian game-based optimization framework.

VI. Conclusion

Cloud computing and virtualization techniques provide mobile devices with battery energy saving opportunities by allowing them to offload computation and execute applications remotely. In this paper, we consider an MCC interaction system consisting of multiple mobile devices and the cloud computing system. We provide a Bayesian game formulation for the MCC interaction system. In this game, each mobile device determines the portion of its service requests for remote processing in the cloud computing system. All the mobile devices compete for the allocated resources in the data center. Each mobile device is aware of its own service request generating rate through effective prediction methods. It has only partial information about the other mobile devices. The objective of each mobile device is to minimize its power consumption as well as the service request response time. We prove that the pure strategy Bayesian-Nash equilibrium in this game always exists and is unique. We derive the optimal strategies for all the mobile devices in the Bayesian game achieving such equilibrium using convex optimization approach. Experimental results demonstrate the effectiveness of the proposed Bayesian game-based optimization framework of the MCC system.

References

[1] B. Hayes, “Cloud Computing,” *Communications of the ACM*, 2008.

[2] R. Buyya, “Market-oriented cloud computing: vision, hype, and reality of delivering computing as the 5th utility,” in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid* (CCGrid), 2009.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Pabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *Communications of the ACM*, 2010.

[4] M. Pedram, “Energy-efficient datacenters”, *IEEE Trans. on CAD*, 2012.

[5] L. A. Barroso and U. Holzle, “The case for energy-proportional computing,” *IEEE Computer*, 2007.

[6] R. H. Katz, “Tech Titans Building Boon,” *IEEE Spectrum*, 2009.

[7] Y. Wang, S. Chen, H. Goudarzi, and M. Pedram, "Resource allocation and consolidation in a multi-core server cluster using a Markov decision process model," *ISQED*, 2013.

[8] Y. Wang, S. Chen, and M. Pedram, “Service level agreement-based joint application environment assignment and resource allocation in cloud computing systems,” *GreenTech*, 2013.

[9] K. Krauter, R. Buyya, and M. Maheswaran, “A taxonomy and survey of grid resource management systems for distributed computing,” *Software Practice and Experience*, 2002.

[10] L. Zhang and D. Ardagna, “SLA based profit optimization in autonomic computing systems,” in *2nd Int. Conf. on Service Oriented Computing*, 2004.

[11] D. Ardagna, M. Trubian, and L. Zhang, “SLA based resource allocation policies in autonomic environments,” *Journal of Parallel and Distributed Computing*, 2007.

[12] A. Chandra, W. Gongt, and P. Shenoy, “Dynamic resource allocation for shared clusters using online measurements,” *International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS), 2003.

[13] M. N. Bennani and D. A. Menasce, “Resource allocation for autonomic clusters using analytic performance models,” in *Proc. of the 2nd Int. Conf. on Autonomic Computing*, 2005.

[14] H. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," in *Wireless Commun. and Mobile Comput.*, 2011.

[15] A. Khan and K. Ahirwar, “Mobile cloud computing as a future of mobile multimedia database,” in *International Journal of Computer Science and Communication*, 2011.

[16] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Proc. of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, 2010.

[17] http://aws.amazon.com/s3/.

[18] K. Kumar and Y. Lu, "Cloud computing for mobile users: can offloading computation save energy?" in *IEEE Computer*, 2010.

[19] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” in *Proceedings of the 19th ACM Symposium on Operating System Principles* (SOSP), 2003.

[20] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. of International Conference on Mobile Systems, Applications, and Services*, 2010.

[21] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: elastic execution between mobile device and cloud," in *Proc. of EuroSys*, 2011.

[22] Y. Ge, Y. Zhang, Q. Qiu, and Y. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. of ISLPED*, 2012.

[23] H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid", in *Proc. of IEEE PES ISGT*, 2010.

[24] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave *n*-person games," *Econometrica*, vol. 33, pp. 347-351, 1965.

[25] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[26] S. Srikantaiah, A. Kansal, and F. Zhao, “Energy aware consolidation for cloud computing,” in *Workshop on Power Aware Computing and Systems* (HotPower’08), 2008.

[27] I. Hwang, T. Kam, and M. Pedram, “A study of the effectiveness of CPU consolidation in a virtualized multi-core server system,” in *Proc. of International Symposium on Low Power Electronics and Design* (ISLPED), 2012.

[28] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. Dick, Z. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proc. of Int. Conf. on HW/SW Codesign and System Synthesis* (CODES+ISSS), 2010.

[29] D. Shin, W. Lee, K. Kim, Y. Wang, Q. Xie, M. Pedram, and N. Chang, “Online estimation of the remaining energy capacity in mobile systems considering system-wide power consumption and battery characteristics,” to appear in *Proc. of Asia South Pacific Design Automation Conference* (ASP-DAC), 2013.

[30] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 3rd edition, 1991.

[31] L. Kleinrock, *Queueing Systems, Volume I: Theory*, New York: Wiley, 1975.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[33] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 1.21.” http://cvxr.com/cvx, Feb, 2011.

[34] K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*, Morgan & Claypool Publishers, 2008.

[35] J. C. Harsanyi, "Games with Incomplete Information Played by "Bayesian" Players, I-III: Part I. The Basic Model", *Management Science*, Nov. 1967, Vol. 14, Num. 3, pp. 159-182.

[36] http://ocw.mit.edu/courses/economics/14-126-game-theory-spring-2010/lecture-notes/MIT14_126S10_lec01.pdf