Coordination of the Smart Grid and Distributed Data Centers: A Nested Game-Based Optimization Framework

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

***Abstract*—The emergence of cloud computing has established a trend towards building energy-hungry and geographically distributed data centers. Due to their enormous energy consumption, data centers are expected to have major impact on the electric power grid by significantly increasing the load at locations where they are built. Dynamic energy pricing policies in the recently proposed smart grid technology can incentivize the cloud computing controller to shift their computation load towards data centers in regions with cheaper electricity. On the other hand, distributed data centers also provide opportunities to help the smart grid to improve load balancing and robustness. To shed some light into these opportunities, this paper considers an interaction system of the smart grid, the cloud computing system, and other load devices. A nested two stage game based formulation is proposed based on the location-dependent real-time pricing policy of the smart grid. The leading player in this game is the smart grid controller that announces the relationship between the electricity price at each power bus and the total load demand at that bus. In the second stage, the cloud computing controller performs resource allocation as response to the pricing functions, whereas the other load devices perform demand side management. The objective of the smart grid controller is to maximize its own profit and perform load balancing among power buses, whereas the objective of the cloud computing controller is to maximize its own profit with respect to the location-dependent pricing functions. The optimal strategies are derived based on the backward induction principle for the smart grid controller, the cloud computing controller, and the other load devices, using convex optimization and heuristic search.**

I. INTRODUCTION

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

The underlying infrastructure of cloud computing consists of data centers and clusters of servers that are monitored and maintained by the cloud service providers [5]. Service providers often end up over-provisioning their resources in these servers in order to meet the clients' service level agreements (SLAs) [4]. Such over-provisioning may increase both the electrical energy cost and the carbon footprint incurred on the servers. Hence, optimal provisioning or allocation of the resources in the cloud is imperative in order to reduce the energy cost incurred on the servers as well as the environmental impact while satisfying the clients' SLAs.

The major cloud service providers such as Google, Microsoft, and Amazon have built and are working on building the world's largest data centers with enormous energy consumption. Each data center includes hundreds of thousands of computer servers, cooling equipments, and substation power transformers. For example, Microsoft's data center in Quincy, Washington consumes 48 megawatts that is enough to power 40,000 homes [5]. Data centers are expected to have a major impact on the electric grid by significantly increasing the load consumption at locations where they are built. Therefore, integration of large-scale data centers may degrade the reliability and robustness of traditional power grid with respect to load demand variations and link breakage.

The recently proposed smart grid technology takes advantage of the modern communication system to gather information from consumers and suppliers in order to improve efficiency, reliability, and sustainability of the power grid, thereby minimizing the overall cost of electrical power delivered to the end users [6]. Utitlity companies can employ *time-dependent* or *location-dependent* dynamic pricing policies, incentivizing consumers to perform *demand side management* (DSM) [7] e.g., shifting their loads from the peak time periods to off-peak periods or from one physical location to another location. *Real-time pricing* is an important dynamic pricing policy and fits very well for applications such as vehicle-to-grid (V2G) systems [8]. When real-time pricing is applied, the utility company announces the relationship (usually a superlinear function) between the electricity price and the total load demand over the next time slot (usually a few minutes to one hour.) The pricing function may be different at different locations or power buses (also known as locational marginal real-time pricing [9]).

Because the electricity cost dominates all the other cost aspects in the cloud, the central controller of the cloud should develop resource management algorithms among data centers that account for the variations of electricity price at different regions by dynamically shifting the computation load towards data centers that are located in regions with cheaper electricity [10]. Developing such resource management algorithms and location-dependent dynamic electricity pricing strategies is important in order for mitigating the negative impacts on the smart grid from integrating large-scale data centers. With appropriately designed dynamic pricing policies, it is even possible that data centers could actually help the smart grid design in terms of *load balancing* and robustness thanks to the flexibility in *service request dispatching* to various data centers [11][12].

In this paper, we consider a smart grid system comprised of multiple *power buses*. We consider a set of distributed data centers in this system. Each data center is comprised of potentially heterogeneous servers in terms of request processing ability, and is connected to a power bus in the smart grid to obtain the electricity for operation. Service requests from a common *request pool* are free to be dispatched to any server in the cloud computing system. The total profit in the cloud computing system is the total price obtained from serving the service requests, which depends on the average request response time defined in the *utility function* of the SLA, subtracted by the energy cost of the active datacenters.

We consider the location-dependent real-time pricing scenario. In this pricing scenario, the smart grid controller announces the relationship between the electricity price at each power bus and total load demand at that power bus. The cloud computing controller as well as the other load devices will perform demand side management as response to these pricing functions. Please note that the reactive power is neglected in this modeling. We consider the interaction system of the smart grid, the cloud, and other load devices. The objective of the smart grid controller is to maximize its own profit and perform load balancing among power buses. The objective of the cloud computing controller is to maximize its own profit with respect to the location-dependent pricing functions. We provide a nested two stage (Stackelberg) game-based formulation for the interaction system. The leading player in this game is the smart grid controller. In the second stage of the game, the cloud computing controller determines the resource allocation results in the cloud, whereas the other load devices determine their corresponding power consumption values. The second stage forms a non-cooperative *subgame* of the nested two stage game, because the electricity price at each power bus depends on the load power consumption levels.

Based on the *backward induction* principle in Stackelberg games [13], we start with the optimization in the second stage and find the *subgame perfect equilibrium* (SPE) of the cloud computing controller and the other devices using convex optimization [14]. Based on the SPE, we derive the optimal strategy for the smart grid controller using the simulated annealing approach [15]. Experimental results demonstrate the effectiveness of the proposed nested game-based optimization framework on profit maximization and load balancing. Since this current approach is only valid for systems where network constraints can be neglected, in future works, network constraints will be incorporated in the formulation.

II. System Model and Problem Formulation

A. The Smart Grid Model

Consider a smart power grid with N *power buses*, indexed by the integer i. The power buses are interconnected through *branches* forming the grid topology. Each i^{th} ($1 \le i \le N$) power bus is connected to various load devices. In our system model, some load devices of the smart grid may include large data centers that support cloud computing facilities. There are M distributed data centers in this infrastructure, indexed by the integer j. Each data center is connected to one power bus in the smart grid to obtain the electricity required for its operation. Let $bus(j)$ denote the index of the power bus that the j^{th} data center is connected to.

The total load power consumption at the i^{th} power bus, denoted by $P_{bus,i}^{Load}$, is calculated via:

$$P_{bus,i}^{Load} = P_{bus,i}^{DC} + P_{bus,i}^{Dev} \tag{1}$$

where $P_{bus,i}^{DC}$ denotes the total power consumption of the data centers (if any) connected to bus i; the term $P_{bus,i}^{Dev}$ denotes the total power consumption of any load devices other than the data centers at the i^{th} power bus. Let $P_{DC,j}$ denote the power consumption of the j^{th} data center. Then $P_{bus,i}^{DC}$ is calculated by:

$$P_{bus,i}^{DC} = \sum_{bus(j)=i} P_{DC,j} \tag{2}$$

If there is no data center connected to the i^{th} power bus, $P_{bus,i}^{DC} = 0$. Please note that the reactive power is neglected in this modeling.

We consider the location-dependent real-time pricing scenario in this paper. In this pricing scenario, the smart grid controller announces the relationship between the unit electricity price at each i^{th} power bus, denoted by $price_i$, and the total load demand $P_{bus,i}^{Load}$ at that power bus. The relationship is denoted by the function $price_i\left(P_{bus,i}^{Load}\right)$. The smart grid controller performs effective load balancing by (i) incentivizing the cloud computing controller to shift the loads among data centers, and (ii) incentivizing the other load devices to perform demand side management. Hence, the unit energy price $price_i$ at the i^{th} ($1 \le i \le N$) power bus is set in the following way:

$$price_i\left(P_{bus,i}^{Load}\right) = c_i \cdot \left(P_{bus,i}^{Load} - P_{bus,i}^{Gen}\right) + price_i^{\mathbf{B}} \tag{3}$$

where $P_{bus,i}^{Gen}$ is the amount of power generation at the i^{th} power bus; $price_i^{\mathbf{B}}$ represents the base electricity price at the i^{th} power bus; c_i is the coefficient determined by the smart grid controller. As we can see from (3), if the load demand $P_{bus,i}^{Load}$ is higher than $P_{bus,i}^{Gen}$, $price_i$ will be increased to discourage the users from consuming more energy from the i^{th} power bus, and vice versa. This pricing policy will reduce the amounts of electric power flowing from one power bus to another (through the branches) by trying to match the power consumption $P_{bus,i}^{Load}$ with power generation $P_{bus,i}^{Gen}$ at each power bus.

B. The Cloud Computing System Model

Figure 1 shows the structure of the target resource allocation system in the cloud with a service request pool, M distributed and heterogeneous data centers as well as a central resource manager. Each j^{th} data center consists of K_j potentially heterogeneous servers. We use k as the index of servers in a data center.

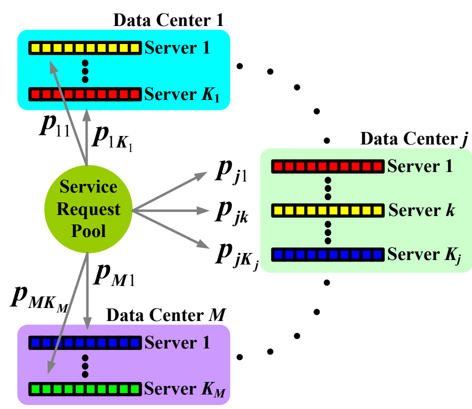

Fig. 1. Architecture of the resource allocation system in the cloud.

The service request pool contains service requests generated from all the clients. A service request is free to be dispatched to any server in the cloud. The request dispatcher dispatches a request to the k^{th} server in the j^{th} data center with probability $p_{j,k}$. These probability values are the optimization variables in the resource allocation framework.

In order to find the analytical form of the average response time, service requests (in the request pool) are assumed to follow a Poisson process with an average generating rate of λ, which is predicted based on the behavior of the clients. According to the properties of the Poisson distribution, service requests that are dispatched to the k^{th} server in the j^{th} data center follow a Poisson process with an average rate of $p_{j,k} \cdot \lambda$, which is the average request arrival rate of that server. By using the well-known formula in M/M/1 queues [17], the average response time of service requests dispatched to each k^{th} server in the j^{th} data center is calculated as

$$R_{j,k}(p_{j,k}) = \begin{cases} \dfrac{1}{\mu_{j,k} - p_{j,k} \cdot \lambda} & \text{if } p_{j,k} > 0 \\ 0 & \text{if } p_{j,k} = 0 \end{cases} \tag{4}$$

where $\mu_{j,k}$ denotes the average request processing speed of the k^{th} server in the j^{th} data center.

The average power consumption in each k^{th} server in the j^{th} data center is proportional to the portion of time that the server is active, which is given by $\left(p_{j,k} \cdot \lambda\right)/\mu_{j,k}$:

$$P_{j,k}^{Serv}(p_{j,k}) = \frac{p_{j,k} \cdot \lambda}{\mu_{j,k}} \cdot P_{j,k}^{Serv,max} \quad (5)$$

where $P_{j,k}^{Serv,max}$ is the server's power consumption when it is active.

The power consumption of the j^{th} data center is the sum of the total power consumption of all its servers, i.e.,

$$P_j^{DC} = \sum_{1 \le k \le K_j} P_{j,k}^{Serv}(p_{j,k}) \quad (6)$$

We use $U^{DC}(R) = \beta - \gamma \cdot R$ to represent the utility function of the cloud computing system with the average request response time equal to R. Then the total profit of the cloud, which accounts for both the price obtained from servicing requests and the energy cost, is calculated by:

$$\lambda \cdot \left(\beta - \gamma \cdot \sum_{j=1}^{M} \sum_{k=1}^{K_j} \frac{p_{j,k}}{\mu_{j,k} - p_{j,k} \cdot \lambda} \right) - \sum_{i=1}^{N} price_i \cdot \sum_{bus(j)=i} \sum_{k=1}^{K_j} P_{j,k}^{Serv}(p_{j,k}) \quad (7)$$

C. Other Load Devices

Let L_i denote the number of other load devices than data centers that are connected to the i^{th} power bus in the smart grid. Let $P_{i,l}^{Dev}$ denote the power consumption of the l^{th} $(1 \le l \le L_i)$ load device connected to the i^{th} power bus, and we have:

$$P_{bus,i}^{Dev} = \sum_{l=1}^{L_i} P_{i,l}^{Dev} \quad (8)$$

We use $U_{i,l}^{Dev}(P_{i,l}^{Dev})$ to denote the utility function (i.e., the satisfaction level) of the l^{th} load device connected to the i^{th} power bus, as a function of the power consumption level $P_{i,l}^{Dev}$. The utility function should satisfy the following three requirements:

- $U_{i,l}^{Dev}(0) = 0$.
- The first derivative of the utility function is positive, i.e., $U_{i,l}^{Dev}(P_{i,l}^{Dev})$ is a increasing function, when $P_{i,l}^{Dev} = 0$.
- The utility function is a concave function.

$a_{i,l} P_{i,l}^{Dev} - b_{i,l} (P_{i,l}^{Dev})^2$ is one type of utility function that satisfies the above three requirements, where $a_{i,l}$ and $b_{i,l}$ are positive coefficients and are device specific. This type of utility function is suitable for air conditioning or water heating applications, where neither of too low or too high power consumption is desirable.

The overall objective function of the l^{th} load device connected to the i^{th} power bus is given as follows, accounting for both the utility function of the device and the electricity cost:

$$U_{i,l}^{Dev}(P_{i,l}^{Dev}) - price_i \cdot P_{i,l}^{Dev} \quad (9)$$

III. FORMULATION AND OPTIMIZATION

In this section, we consider the interaction system of the smart grid and cloud computing systems under the location-dependent real-time pricing scenario, and provide a nested game-based formulation. The leading player is the smart grid controller, which announces the electricity price function $price_i(P_{bus,i}^{Load})$ at each power bus, instead of the actual price values. In the second stage, the cloud computing controller determines the resource allocation results, i.e., the $p_{j,k}$ values, whereas other load devices determine the power consumption levels $P_{i,l}^{Dev}$'s, as response to the price functions. The second stage forms a non-cooperative subgame of the nested two stage game, because the electricity price $price_i$ at each power bus depends on the locational load power consumption levels.

Based on the backward induction principle in Stackelberg games [13], we start with the optimization procedure in the second stage. We find the *subgame perfect equilibrium* (SPE) of the cloud computing controller and the other devices in the second stage, as stated in Section III.A. Based on the SPE, we derive the optimal strategy of the smart grid controller in the first stage as described in Section III.B.

A. The Subgame Perfect Equilibrium in the Second Stage

Suppose that the price function at each power bus is provided by the smart grid controller. Then the second stage of the nested two stage game forms a non-cooperative *normal-form game*, where all the players take action simultaneously. We name the normal-form game the *Resource Allocation and Demand Side Management* (RA-DSM) game. The players include the cloud computing controller and the other load devices. The optimization variables (i.e., the strategy) of the cloud computing controller are the $p_{j,k}$ values, whereas the strategy chosen by the l^{th} other load device connected to the i^{th} power bus is the power consumption level $P_{i,l}^{Dev}$. The payoff functions (i.e., the objective functions) of the cloud computing controller and the other load devices are given by Eqns. (7) and (9), respectively. Please note that the payoff of each player in the RA-DSM game also depends on the strategies of the other players since the electricity price $price_i(P_{bus,i}^{Load})$ at each power bus depends on the total load demand of data centers and the other load devices at that power bus.

The constraints in the RA-DSM game include:

$$0 \le p_{j,k} \le \mu_{j,k}/\lambda, \quad \text{for } \forall j,k \quad (10)$$

$$\sum_{j=1}^{M} \sum_{k=1}^{K_j} p_{j,k} = 1 \quad (11)$$

$$P_{i,l}^{Dev} \ge 0, \quad \text{for } \forall i,l \quad (12)$$

As the cloud computing controller and the other load devices are considered to be non-cooperative among each other, we are interested in the existence and uniqueness of the Nash equilibrium [13]. The Nash equilibrium of the RA-DSM game is the SPE of the overall nested two stage game for the interaction system. The Nash equilibrium is the optimal strategy profile for all the players in the sense that no player can find a better strategy if he deviates from the current strategy unilaterally. In other words, no player (the cloud computing controller or the other load devices) will have incentive to leave this strategy in the Nash equilibrium. Therefore, the Nash equilibrium is of particular interest to a non-cooperative normal-form game. We prove the existence and uniqueness of the Nash equilibrium in the RA-DSM game.

Theorem I (*Nash Equilibrium in the RA-DSM Game*): The Nash equilibrium of the RA-DSM game exists and is unique.

Proof: We are going to prove that the RA-DSM game is a strictly concave *n*-player game. We need to prove (i) the domain of the strategy profile for all the players, which is constrained by (10) - (12), is a closed convex set, and (ii) the objective (payoff) function for each player to maximize is a concave function with respect to the optimization variable of that player, assuming that the optimization variable values of the other players are given. One can easily observe that (i) is true. In the following, we prove that statement (ii) is also true.

For the j^{th} load device connected to the i^{th} power bus, Eqn. (9) is a concave objective function of the optimization variable $P_{i,l}^{Dev}$ because:

- The first term $U_{i,l}^{Dev}(P_{i,l}^{Dev})$ of Eqn. (9) is a concave function of $P_{i,l}^{Dev}$ according to the definition.
- The second term $price_i \cdot P_{i,l}^{Dev}$ is a convex function of $P_{i,l}^{Dev}$ because $price_i$ is a linearly increasing function of $P_{i,l}^{Dev}$.

For the cloud computing controller, Eqn. (7) is a concave objective function of the optimization variables $p_{j,k}$'s because:

- The first term in Eqn. (7) is a concave function of $p_{j,k}$ values because $\frac{p_{j,k}}{\mu_{j,k}-p_{j,k}\cdot\lambda}$ is a convex function of $p_{j,k}$.
- The second term in Eqn. (7) is a convex function of the $p_{j,k}$ values because $price_i$ is a linear function of $\sum_{bus(j)=i}\sum_{k=1}^{K_j} P_{j,k}^{Serv}(p_{j,k})$, which is furthermore a linear function of the $p_{j,k}$ values.

After we have proved that the RA-DSM game is a strictly concave *n*-player game, the existence and uniqueness of Nash equilibrium are directly resulted from the first and third theorem in [18]. ■

Each player of the RA-DSM game finds the Nash equilibrium, which is the SPE of the overall nested two stage game, using standard convex optimization technique [14]. The detailed procedure is illustrated in Algorithm 1.

Algorithm 1: Finding the Nash Equilibrium in the RA-DSM Game for the Cloud Computing Controller and Other Devices.

Initialize the $p_{j,k}$ values for the cloud computing system as well as the $P_{i,l}^{Dev}$ values for the other load devices.

Do the following procedure iteratively:

> Find the optimal $p_{j,k}$ values (i.e., the *best response* of the cloud computing system), by solving the convex optimization problem with the objective function (7) and constraints (10), (11).
>
> Update the $p_{j,k}$ values.
>
> **For each** $1 \le i \le N$, $1 \le l \le L_i$:
>
> > Find the optimal $P_{i,l}^{Dev}$ value (i.e., the *best response* of the l^{th} mobile device connected to the i^{th} power bus), by solving the convex optimization problem with objective function (9) and constraint (12).
> >
> > Update $P_{i,l}^{Dev}$ to be the new value.
>
> **End**

Until the solution converges.

B. Optimization of the Pricing Functions

Based on the backward induction principle in Stackelberg games [13], we find the optimal strategy of the smart grid controller in this section after deriving the SPE in the second stage of the nested game. In the first stage, the smart grid controller (the leading player) determines the optimal c_i values (and therefore, the optimal real-time pricing functions) for $1 \le i \le N$, which are the optimization variables. The objective of the smart grid controller is to achieve an optimal balance between maximizing its own profit and performing load balancing among power buses, with an anticipation of the demand side managements performed by various load devices including data centers. More specifically, the smart grid controller finds the optimal c_i values in order to maximize the following objective function:

$$w_1 \cdot \sum_{i=1}^{N} price_i \cdot P_{bus,i}^{Load} - w_2 \cdot \max_{1\le i\le N}\left|P_{bus,i}^{Load} - P_{bus,i}^{Gen}\right|^2 \qquad (13)$$

where w_1 and w_2 are the relative weights greater than or equal to zero. In Eqn. (13), $\sum_{i=1}^{N} price_i \cdot P_{bus,i}^{Load}$ is the total revenue of the smart grid controller from selling electricity, whereas $\max_{1\le i\le N}\left|P_{bus,i}^{Load} - P_{bus,i}^{Gen}\right|^2$ is the metrics of load imbalance (because the power flowing through the branches will be minimized when $P_{bus,i}^{Load}$ matches with $P_{bus,i}^{Gen}$.) Less load imbalance among all the power buses is preferred for the robustness concerns. Therefore, we maximize the former term and minimize the latter term in (13). We name this optimization problem performed in the smart grid controller the *Optimal Pricing with Anticipation of Demand side managements* (OPAD) problem.

When the pricing functions (i.e., the c_i values) are given, we can derive the load power consumption levels $P_{bus,i}^{Load}$'s in (13) through finding the Nash equilibrium in the RA-DSM game (i.e., the SPE of the second stage in the nested game.) This procedure requires using convex optimization technique as stated in Section III.A. The actual electricity price $price_i$ at each i^{th} $(1 \le i \le N)$ power bus also depends on $P_{bus,i}^{Load}$ as stated in (3). Since the OPAD problem is integrated with finding the Nash equilibrium in the RA-DSM game, it is a hard problem to be solved optimally in polynomial time. Therefore, we propose to use the simulated annealing-based method [17] in order to find a near-optimal solution of the OPAD problem. The detailed procedure is provided in Algorithm 2.

Algorithm 2: Deriving a Near-Optimal Pricing Policy for the Smart Grid Controller.

Initialize the temperature T.

Initialize Obj_{max} to be a large negative number.

Do the following procedure:

> Randomly change one or multiple c_i values.
>
> Find the Nash equilibrium in the RA-DSM game using convex optimization technique, based on the updated c_i values.
>
> Calculate $P_{bus,i}^{DC}$ for $1 \le i \le N$ using (2), (5), (6), based on the derived $p_{j,k}$ values.
>
> Calculate $P_{bus,i}^{Dev}$ for $1 \le i \le N$ using (8), based on the derived $P_{i,l}^{Dev}$ values, and subsequently calculate $P_{bus,i}^{Load}$ for $1 \le i \le N$ using (1).
>
> $Obj \leftarrow$ the value of the objective function (13) based on the calculated $P_{bus,i}^{Load}$ values. Note that the actual electricity price $price_i$ for $1 \le i \le N$ also depends on $P_{bus,i}^{Load}$ as stated in (3).
>
> **If** $Obj \ge Obj_{max}$: Accept the change of the c_i values.
>
> **Else**: Accept the change with probability $e^{(Obj-Obj_{max})/T}$.
>
> $Obj_{max} \leftarrow Obj$ if the change has been accepted.
>
> Decrease the temperature T.

Until the temperature T has decreased to a certain value, i.e., the algorithm has cooled down.

IV. Experimental Results

In this section, we implement the interaction system of the smart grid, the cloud computing system, and the other load devices, and demonstrate the effectiveness of the proposed nested game-based optimization framework. We use normalized amounts of most of the parameters in the interaction system instead of their real values.

We consider a smart power grid that is comprised of 12 power buses. The amount of power generation $P_{bus,i}^{Gen}$ at each i^{th} $(1 \le i \le N)$ power bus in the smart grid system is a uniformly distributed random variable between 6 and 8. The base price over all the power buses, i.e., $price_i^{\mathbf{B}}$ for $1 \le i \le N$, are set to 1, whereas the c_i values are the optimization variables. We consider 4 distributed data centers in the interaction system. The data centers are connected to power buses 1 to 4. Each data center is connected to a power bus. The data centers are comprised of 5 servers, 6 servers, 9 servers, and 10 servers, respectively. The average service request generating rate λ is assumed to be 30. The average service request processing rate μ_{jk} in each server is a uniformly distributed random variable between 1 and 2.

The maximum power consumption of each server is uniformly distributed between 2 and 4. For the utility function in the cloud computing system, parameter β is set to 9 and γ is set to 1. For the other load devices than the data centers, we assume that each of power buses 1 to 4 is connected with 3 other load devices, whereas each of power buses 5 to 12 is connected with 5 other load devices. For each l^{th} load device connected to the i^{th} power bus, parameter $a_{i,l}$ in the utility function is set to 2 and parameter $b_{i,l}$ is a uniformly distributed random variable between 0.2 and 0.3.

In the experiment, we consider the interaction system under the location-dependent real-time pricing scenario. We compare the capabilities in profit maximization and load balancing of the smart grid system using the proposed nested game-based optimization method and the baseline method. In the baseline system, the smart grid controller simply sets the electricity price at every i^{th} power bus to be $price_i^{\mathbf{B}}$. In other words, the c_i values are all set to 0, and real-time pricing policy is not applied in the baseline system. Table I shows the normalized values (with respect to the largest value) of total profit $\sum_{i=1}^{N} price_i \cdot P_{bus,i}^{Load}$ and load imbalance $\max_{1\le i\le N}\left|P_{bus,i}^{Load} - P_{bus,i}^{Gen}\right|^2$ of the smart grid system using the proposed nested game-based optimization method and the baseline method. Different rows in Table I are obtained by adjusting the parameters w_1 and w_2.

TABLE I. NORMALIZED VALUES OF TOTAL PROFIT AND LOAD IMBALANCE OF THE SMART GRID SYSTEM

w_2/w_1	Proposed System		Baseline System	
	Profit	Imbalance	Profit	Imbalance
0	1	0.385	0.918	1
0.5	0.938	0.319	0.890	0.900
2	0.953	0.192	0.905	0.995
10	0.908	0.180	0.866	0.763
∞	0.883	0.143	0.872	0.701

We can derive the following observations from Table I. First, simultaneously enhancement in the total profit and better load balancing can be achieved using the proposed nested game-based optimization framework. For example, when $w_2/w_1 = 2$, the proposed optimization framework outperforms the baseline method by 5.3% in total profit of the smart grid system and 79.8% reduction in load imbalance. Second, the proposed game theoretic optimization framework is extremely powerful in enhancing load balancing compared with profit maximization. As shown in Table I, the maximum reduction in the load imbalance value is 79.8% compared with the baseline method, whereas the maximum enhancement in the total profit is 8.3%. Figure 2 illustrates the tradeoff curve (obtained by adjusting parameters w_1 and w_2) between the higher profit of the smart grid and lower load imbalance at different power buses. One can observe that the load imbalance value can be significantly reduced compared with the baseline method by up to 85% when the total profit is the same, thereby demonstrating the effectiveness of the proposed game theoretic optimization framework in load balancing.

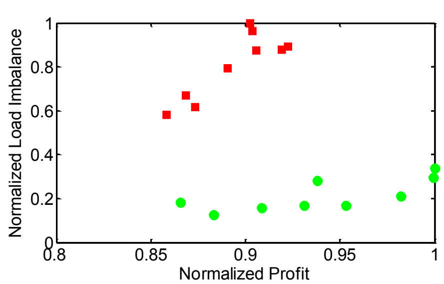

Fig. 2. The tradeoff between total profit of the smart grid system and load imbalance at different power buses.

V. CONCLUSION

In this paper, we consider an interaction system of the smart grid, the cloud computing system, and other load devices. We propose a nested two stage game-based formulation based on the location-dependent real-time pricing policy of the smart grid. The leading player is the smart grid controller that announces the relationship at each power bus between the local electricity price and the total load demand. In the second stage, the cloud computing system performs request dispatching and resource allocation as response to the pricing functions, whereas the other load devices perform demand side management. The objective of the smart grid controller is to maximize its own profit and perform load balancing among power buses, whereas the objective of the cloud computing controller is to maximize its own profit with respect to the pricing functions. We derive the optimal strategies based on backward induction for the smart grid controller, the cloud computing controller, and the other load devices, using convex optimization and simulated annealing approaches. Since this current approach is only valid for systems where network constraints can be neglected, in future works, network constraints will be incorporated in the formulation.

REFERENCES

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

[2] M. Pedram, "Energy-efficient datacenters," *IEEE Trans. on Computer-Aided Design*, 2012.

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

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

[6] S. Kishore and L. V. Snyder, “Control mechanisms for residential electricity demand in SmartGrids,” *IEEE SmartGridComm*, 2010.

[7] S. Caron and G. Kesidis, “Incentive-based energy consumption scheduling algorithms for the Smart Grid,” *Proc. of IEEE SmartGridComm*, 2010.

[8] P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings, “Agent-based micro-storage management for the smart grid,” *Proc. of AAMAS*, 2010.

[9] C. Wu, H. Mohsenian-Rad, and J. Huang, "Vehicle-to-aggregator interaction game," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, March 2012.

[10] L. Rao, X. Liu, L. Xie, and W. Liu, “Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment,” in *Proc. of IEEE Infocom*, Mar. 2010.

[11] A. H. Mohsenian-Rad and A. Leon-Garcia, “Energy-information transimission tradeoff in green cloud computing,” in *Global Communications Conference* (GLOBECOM), 2010.

[12] Y. Wang, X. Lin, and M. Pedram, “A sequential game perspective and optimization of the smart grid with distributed data centers,” *Proc. of IEEE PES Innovative Smart Grid Technologies* (ISGT), 2013.

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

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

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, Vol. 220, No. 4598, pp. 671 – 680.

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

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

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