When Smart Grid Meets Geo-distributed Cloud: An Auction Approach to Datacenter Demand Response

Zhi Zhou\(^1\) Fangming Liu\(^*1\) Zongpeng Li\(^2\) Hai Jin\(^1\)
\(^1\)Key Laboratory of Services Computing Technology and System, Ministry of Education, School of Computer Science and Technology, Huazhong University of Science and Technology, China.
\(^2\)Department of Computer Science, University of Calgary, Canada.

Abstract—Datacenter demand response is envisioned as a promising tool for mitigating operational stability issues faced by smart grids. It enables significant potentials in peak load reduction and facilitates the incorporation of distributed generation. Monetary refund from the smart grid can also alleviate the cloud’s burden in escalating electricity cost. However, the current demand response paradigm is inefficient towards incentivizing a cloud that runs over geo-distributed datacenters. Leveraging auction theory, this work presents an efficient incentive mechanism to elicit demand response from geo-distributed clouds. To determine the winning bids and their corresponding payments, the cloud that acts as the auctioneer needs to solve a set of winner determination problems that are highly challenging. By integrating techniques from the Gibbs sampling method and the alternating direction method of multipliers, we propose a decentralized algorithm for each datacenter to make autonomous decisions on winning bid selection and workload management, striking a balance among the economic efficiency, truthfulness and the computational efficiency. Through extensive trace-driven evaluations, we demonstrate that our incentive mechanism constitutes a win-win mechanism for both the geo-distributed cloud and the smart grid.

I. INTRODUCTION

The recent years have witnessed new emerging technology advances in the ICT sector, among which two are widely recognized as having great significance. The first is the internet-scale cloud services that are deployed over geographically distributed datacenters, indispensable for a wide variety of applications and serving both enterprises and end users. The second is the evolution from the traditional power grid to the smart grid, enabling sustainable, cost-effective, and environmental-friendly electric power generation and consumption.

It is readily acknowledged, however, that the further developments of both cloud computing and smart grid are facing their respective challenges. Specifically, for large-scale cloud service providers, the annual electricity bill can be as high as \(67\text{M}\) [1], a number continuing to rise with the flourishing of cloud computing services as well as the rise of electricity price. Meanwhile, the smart grid that integrates a large number of distributed generations such as solar arrays and wind turbines also faces severe operation stability and hence economic issues, due to the intermittent nature of distributed generation. For example, the enormous wind generation in May 2014 has incurred continuous negative electricity price in Germany [2].

The aforementioned concerns of both the cloud and the smart grid can be alleviated through appropriate cooperation between the two sides. It has been widely recognized that, datacenters can provide a great potential for demand response, since power consumption at a datacenter is often of very large volumes yet exhibiting an elastic nature. Specifically, datacenters are estimated to consume about 8\% of the worldwide electricity by 2020, while an individual datacenter can make up 50\% of the power load of a distribution grid nowadays [3] (e.g., Facebook’s datacenter in Crook County, Oregon). Despite its sheer volume, datacenter power consumption is a natural target in demand response as it comes from not only interactive workloads driven by user requests that can be split to geo-distributed datacenters, but also back-end batch workloads (e.g., indexing and web crawling) that are elastic to resource allocation and thus power consumption. This feasibility can be exploited to adjust the power consumption of the geo-distributed datacenters when demand response is required. While for the cloud, participation in demand response program can help to ease the burden from its extraordinarily high electricity cost.

Unfortunately, despite the fact that calculated demand response can lead to a win-win solution for both the cloud and the smart grid, in reality the cloud contributes little to demand response, due to challenges and hurdles from both technical and economic aspects. On the technical side, ensuring the availability and desired performance for the risk-sensitive datacenters is being addressed [4]. While on the economic side, the most commonly adopted demand response program, Coincident Peak Pricing (CPP), is known to be inefficient and poorly designed for datacenter participation [4], for cost savings may be insufficient to incentivize datacenters to respond. Furthermore, existing demand response schemes are designed for the stability of each regional smart grid; when a cloud runs on top of geo-distributed datacenters and thus couples multiple smart grids, the competition in the price of demand response would no doubt incur a loss in social efficiency. Consequently, an economically efficient demand response program tailored for geo-distributed datacenters to realize the true potential is needed.

To the author’s knowledge, this work is among the first that aims to design trading mechanisms for the demand response from a geo-distributed cloud. Towards the goal of effectively stimulating the geo-distributed cloud to respond to the smart grids, we take an auction approach to discover the true value of datacenter demand response, while maximizing the aggregate satisfaction from both the geo-distributed cloud and regional smart grids. The proposed combinational auction framework

\(^*\)The Corresponding Author is Fangming Liu (fmliu@hust.edu.cn). The research was support in part by a grant from National Basic Research Program (973 program) under grant No.2014CB347800.
is illustrated in Fig. 1: each regional smart grid first submits a number of sealed power demand bids to the corresponding datacenter to express its willingness to pay for different levels of power demand. After receiving the bids from all the smart grids, the cloud jointly optimizes the winning bid of each smart grid as well as workload management, i.e., how much interactive workload and batch workload to be allocated to each datacenter. This is achieved through maximizing the social welfare, which is defined as the aggregate satisfaction from both the cloud and smart grids. Finally, the cloud computes a truthful payment for each smart grid, in order to incentivize them to reveal their real utility.

The mechanism design in our demand response setting is rather challenging, with two salient differences from the design of conventional auction mechanisms: (i) each smart grid wins one and only one bid, and (ii) the utility of the cloud includes not only the payments from the smart grids, but also the electricity charges, the dis-utility from interactive workloads and the revenue loss from the batch workloads. Thus, instead of relying on existing centralized approaches for mechanism design, we propose to effectively settle the winner determination problem by exploiting the distributed computing capacity of the cloud. Combining the advantages of the Gibbs sampling method and the alternating direction method of multipliers (ADMM), we propose DORA, a Decentralized sQcial welfaRe mAximization scheme for each datacenter to first make autonomous decisions on winning bid selection, and then conduct the workload management in a fully distributed manner. We show that DORA converges to the optimal winning bids and workload management that yield the maximum social welfare with arbitrarily high probability. By adapting the classic VCG mechanism to determine the payment of each smart grid, DORA further strikes a balance among economic efficiency, truthfulness and computational efficiency.

In the rest of the paper, we review related work in Sec. II and introduce our demand response model in Sec. III. Sec. IV is the distributed social welfare maximization algorithm. Sec. V describes a payment mechanism that works in concert with the winner selection algorithm. Simulation studies are presented in Sec. VI, and Sec. VII concludes the paper.

II. RELATED WORK

Datacenter demand response lies in the intersection of two emerging areas: energy efficiency for the datacenter and demand response in the smart grid. A large body of recent research were devoted to exploiting the opportunities or addressing the challenges in datacenter demand response.

A majority of the existing literature focused on the cost minimization, when datacenter participates in demand response. For example, Liu et al. [5] explored the dynamic scheduling of interactive/batch workload and local power generation, in order to avoid coincident peaks and thus to reduce the energy expenditure. Mohsenian-Rad et al. [6] investigated the benefit of offering voluntary load reduction to a smart grid by load shedding, resulting in a tradeoff between service quality and energy consumption. Recent works have also studied the potential of flexible datacenter power management through geographic load balancing [7], [8], energy storage coordination [9], local fuel cell generation [10], and virtual machine (VM) scheduling [11], [12]. Given progresses made in the above literature, our study is new and complementary, in that we focus on the design of an efficient demand response auction mechanism that stimulates the truthful participation of a geo-distributed cloud.

While significant progress has been made on developing strategies for managing datacenters’ participation in demand response, the market design for datacenters is only beginning to receive attention. Liu et al. [3] designed an efficient prediction-based pricing rule for the demand response of multiple independent datacenters in one smart grid. Our study differs substantially in that we consider a geo-distributed cloud that couples multiple regional smart grids. Though the interaction between geo-distributed cloud and smart grids has been studied by Wang et al. [13], through a Stackelberg game model based on the assumption that the smart grids know the exact payoff function of the cloud. This assumption is practically limiting. In our auction, the private valuation of each smart grid is extracted by a truthful payment mechanism. Demand response auction with the smart grid as the auctioneer has appeared in the literature [14]. However, these auctions may be inefficient since the distributed nature of cloud services was not considered, and more importantly, they were not proven to be truthful.

III. DEMAND RESPONSE AUCTION MODEL

A. Overview of the Geo-distributed Cloud Platform

Consider a cloud provider running cloud services on a set of \( N \) geographically dispersed datacenters, \( \mathcal{D} = \{1, 2, \ldots, N\} \). Each datacenter \( j \in \mathcal{D} \) consists of \( S_j \) processing servers, which are assumed to be homogeneous in this work. We also assume that the cloud deploys a set of \( M \) front-end servers, \( \mathcal{S} = \{1, 2, \ldots, M\} \), in various geographical regions, to direct the interactive workloads driven by user requests to appropriate datacenters. In a specific scheduling period, the total amount of incoming interactive workload (in number of processing servers required) at the front-end server \( i \) is \( D_i \), which can be predicted rather accurately in practice, by employing techniques such as statistical machine learning and time series analysis [15]. The amount of interactive workload distributed from front-end server \( i \) to datacenter \( j \) is \( d_{ij} \), satisfying the following load balance constraint:

\[
\sum_{j \in \mathcal{D}} d_{ij} = D_i, \forall i \in \mathcal{S}. \tag{1}
\]
Besides interactive workload driven by front-end user requests, datacenters nowadays commonly process delay tolerant and resource elastic batch workload, as exemplified by indexing and data mining jobs, running at the back-end [16]. Let $\lambda_j$ represents the amount of batch workload (also in terms of the number of processing servers required) to be executed in datacenter $j$ during the scheduling period. The aggregated workload at datacenter $j$ is then $\sum_{i \in S} d_{ij} + \lambda_j$, and we have:

$$\sum_{i \in S} d_{ij} + \lambda_j \leq S_j. \quad (2)$$

which ensures that aggregated workload processed at each datacenter does not exceed the latter’s capacity.

### B. Datacenter Power Consumption

In our energy cost model, we first focus on power consumption of the processing servers; energy consumptions of other equipment such as cooling devices and power distribution units are roughly proportional [17]. A series of recent empirical studies [17] show that, the aggregated power consumption of homogeneous servers can be modelled as a linear function of the total workload, $s P_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}})\mu$. Here $s$ and $\mu$ denote the number of running servers and the amount of workload, respectively. $P_{\text{idle}}$ is the server power when idle, while $P_{\text{peak}}$ is the server power when fully utilized.

For a datacenter $j \in D$ that hosts $S_j$ homogeneous active servers, manages a total workload of $\sum_{i \in S} d_{ij} + \lambda_j$ and possesses a power usage efficiency of $\text{PUE}_j$, its server power consumption is: $S_j P_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}})(\sum_{i \in S} d_{ij} + \lambda_j)$. Furthermore, the total power demand at datacenter $j$ is:

$$e_j = (S_j P_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}})(\sum_{i \in S} d_{ij} + \lambda_j)) \cdot \text{PUE}_j. \quad (3)$$

The power usage efficiency metric $\text{PUE}$ represents the ratio between (i) the total amount of power used by the entire datacenter facility and (ii) the power delivered to the computing equipment. Inefficient datacenters have a $\text{PUE} \in [2.0, 3.0]$, while industry-leading datacenters are known to approach a $\text{PUE}$ of 1.1 [17]. We next reformulate the above equation of $e_j$ for concise presentation:

$$e_j = \alpha_j \sum_{i \in S} d_{ij} + \lambda_j + \beta_j, \quad (3)$$

where $\alpha_j = (P_{\text{peak}} - P_{\text{idle}}) \cdot \text{PUE}_j$ and $\beta_j = S_j P_{\text{idle}} \cdot \text{PUE}_j$.

### C. Demand Response Bidding

Before presenting our datacenter demand response auction, we first state two assumptions on the smart grid. First, a single datacenter locates within the geographical span of a regional smart grid, in line with the fact that regional smart grids usually cover a moderate-size district. Second, non-datacenter power demands in a smart grid, such as residential power demands, are not elastic and can be accurately predicted for the near future [18].

At the beginning of each scheduling period, each regional smart grid first predicts both the level of renewable generation and inelastic power demand during this period. Based on the predictions, the smart grid corresponding to datacenter $j$ can compute: (i) desired power consumption by datacenter $j$, $\hat{E}_j$, which minimizes the voltage violation possibility of the smart grid corresponding to datacenter $j$, and (ii) the lower-bound $E_j^-$ and upper-bound $E_j^+$ of the actual power consumption $e_j$, which bound the voltage violation possibility within an acceptable range, e.g., below 3% [3]. In practice, when given the structure of the power distribution network (e.g., the SCE 47 bus network in [4]), the above parameters can be computed by taking the “branch flow” model [19].

Based on the calculated $\hat{E}_j$, $E_j^-$ and $E_j^+$, each smart grid corresponding to datacenter $j$ — as the bidder in the datacenter demand response auction — computes its valuation (i.e., maintenance cost savings from potential voltage violation) on the actual power consumption $e_j$ of datacenter $j$. As illustrated in Fig. 2, when $e_j \in [E_j^-, \hat{E}_j)$, the valuation is increasing and strictly concave; when $e_j \in [\hat{E}_j, E_j^+]$, the valuation is decreasing and strictly concave. The strictly concavity assumption is widely adopted in literatures [3], as it captures increased marginal maintenance cost in practice. Furthermore, when $e_j$ is out of the acceptable interval $[E_j^-, E_j^+]$, we assume the valuation is 0. Note that this assumption reflects the reality that the smart grid will not pay for the datacenter demand response, if the resulted power consumption is out of the acceptable interval.

![Smart grid's valuation curve on the actual power consumption of the datacenter.](image)

The regional smart grid corresponding to datacenter $j$ submits a number of sealed valid power demand bids $\{(e_j^1, b_j^1), (e_j^2, b_j^2), \ldots, (e_j^K, b_j^K)\}$ to the datacenter $j$, sorted in ascending order according to $e_j^k$. Here valid means that each $e_j^k$ is among $[\text{PUE}_j S_j P_{\text{idle}}, \text{PUE}_j S_j P_{\text{peak}}]$, the interval of feasible power consumption of datacenter $j$. Each bid $(e_j^k, b_j^k)$ specifies the willingness of the smart grid corresponding to datacenter $j$ to pay $b_j^k$ for the datacenter’s power consumption $e_j^k$. In practice, $b_j^k$ can be sampled from a discretization of the valuation curve (e.g., solid dots in Fig. 2).

Among bids submitted by each local smart grid, one and only one of them wins. This represents a departure from the conventional wireless spectrum auction and cloud virtual machine (VM) auction literature [20], [21] where a bidder may not win any item at all. The rationale behind such a new assumption is the following. In conventional auction, the auctioneer may not allocate any item to a bidder, i.e., this bidder loses his bid. While in our demand response auction, to power the hosted servers, each datacenter would always cause a none-zero energy consumption that corresponds to a certain winning bid of the regional smart grid.

Let $(e_j, b_j)$ denote the winning bid of the smart grid corresponding to datacenter $j$, we have

$$(e_j, b_j) \in \{(e_j^1, b_j^1), (e_j^2, b_j^2), \ldots, (e_j^K, b_j^K)\}. \quad (4)$$

Note that since each bid $(e_j^k, b_j^k)$ is assumed to be valid, then the datacenter capacity constraint (2) in Sec. III-A is naturally embodied into the above constraint (4), and can be omitted
in the later formulations. The operator of each smart grid corresponding to datacenter \( j \) has a quasi-linear utility:

\[
    u_j = b_j - r_j,
\]

where \( r_j \) is the payment of smart grid corresponding to datacenter \( j \) for winning a bid.

D. Utility of Geo-distributed Cloud to Participate in Demand Response Auction

While the cloud can benefit from demand response to the smart grids (i.e., the payment \( r_j \) from each smart grid), such benefit does not come without compromising the cloud operator’s utility, including the dis-utility or revenue loss due to the latency of its interactive workload, reduced revenue from its batch workload, and higher electricity cost of each datacenter. We next elaborate on these three terms.

Dis-utility of interactive workload. For interactive applications, latency is arguably the most important performance metric of a cloud service. A moderate increase in user-perceived latency translates into substantial revenue loss for the provider [15]. The wide-area network propagation latency from the front-end server to the datacenter largely accounts for user-perceived latency in cloud services, shadowing other factors including queuing and processing delays in datacenters. The propagation latency \( L_{ij} \) between the front-end server \( i \) and datacenter \( j \) can be obtained through empirical approaches such as active measurements. The dis-utility of the interactive workload aggregated at front-end server \( i \) depends on experienced mean propagation latency \( \sum_{j \in D} d_{ij}L_{ij}/D_i \) through a generic utility function \( U_i \) that is increasing and convex. A commonly adopted dis-utility function is a quadratic function that reflects a user’s increased tendency to leave the service with an increased latency [15]:

\[
U_i(d_i) = qD_i \left( \sum_{j \in D} d_{ij}L_{ij}/D_i \right)^2, \tag{5}
\]

where \( d_i = (d_{i1}, d_{i2}, \ldots, d_{iN})^T \), and \( q \) is the price that converts dis-utility to a monetary term.

Revenue loss of allocating resources for batch workload. Batch workload is elastic to both resource allocation and revenue for the datacenters. Intuitively, the more computing capacity is fully allocated to batch workload: a differentiable, decreasing convex function \( V_j(\lambda_j) \) (as exemplified by the following affine function) can be adopted, as the loss is zero when the computing capacity is fully allocated to batch workload:

\[
V_j(\lambda_j) = \theta(S_j - \lambda_j), \tag{6}
\]

where \( \theta \) is the price that translates a resource amount to monetary terms.

Electricity cost. Energy cost constitutes a substantial portion (e.g., > 40%) of the operational cost for cloud providers. Given the power consumption \( e_j \) of datacenter \( j \), and power price \( p_j \) at the smart grid corresponding to datacenter \( j \), the total electricity cost of the cloud is:

\[
\sum_{j \in D} e_j p_j.
\]

In the power demand response auction, the cloud operator receives a total payment from the smart grids of \( \sum_{j \in D} r_j \).

We are now ready to formulate the utility of the cloud participating in the demand response auction:

\[
\sum_{j \in D} \left\{ r_j - V_j(\lambda_j) - e_j p_j \right\} - \sum_{i \in S} U_i(d_i).
\]

A second difference (besides one winning bid per bidder) between our power demand response auction and a conventional auction is that the auctioneer’s utility comprises of not only payments from the bidders, but also the varying dis-utility, revenue loss and cost that depend on the allocation of the datacenter power consumption.

E. The Winner Determination Problem (WDP)

Given the utility of each smart grid and the geo-distributed cloud, we formulate the winner determination problem (WDP) that maximizes the social welfare (the aggregated utility of the cloud and all the smart grids), \( W(D) \):

\[
\max \sum_{j \in D} u_j + \sum_{j \in D} \left\{ r_j - V_j(\lambda_j) - e_j p_j \right\} - \sum_{i \in S} U_i(d_i), \tag{7}
\]

s.t. constraints (1)(3)(4).

The above WDP is a mixed integer nonlinear programming (MINLP) problem, which is NP-hard in general. While there exist literature on designing an auction mechanism that is both truthful and computationally efficient, the two differences between our problem and conventional auction design preclude direct application of such approaches. Specifically, a randomized auction framework from theoretical computer science has recently been successfully applied in spectrum and VM auctions [20], [21], whose WDPs satisfy a parking property. Unfortunately, the first difference between our auction and the conventional auction, that each bidder wins one and only one bid, invalidates the parking property. Furthermore, the classic method that exploits monotonic allocation and critical value based charging was widely adopted in truthful auction design [20]. However, the nonlinear terms in the utility of the cloud operator and thus the social welfare make it highly challenging to construct a power distribution scheme that satisfies monotonicity.

While the distinct features of our demand response auction impose great challenges on the mechanism design, it is critical to note that, the cloud who acts as the auctioneer provides enormous computing capacities distributed across its datacenters. A natural question is then, can we leverage this distributed computing power to facilitate the auction design? The answer is ‘yes’, and in the next section, we describe how to maximize the social welfare in a decentralized manner.

IV. DECENTRALIZED SOCIAL WELFARE MAXIMIZATION

A. DORA: A Decentralized Algorithm for WDP

The WDP in Sec. III-E requires a global search over all feasible combinations of the submitted power demand bids, which is a prohibitive computation task. To tackle such complexity challenges, as well as to facilitate practical distributed implementation, we develop DORA, a decentralized social welfare maximization scheme where each datacenter can autonomously decide the winning bid of its corresponding regional smart grid, utilizing the Gibbs sampling technique [22], [23]. We present DORA in Algorithm 1.
Gibbs sampling is a stochastic optimization method, achieving iterative convergence to the global optimal solution of a non-convex problem by probabilistically transiting among possible states in the solution space. In our social welfare maximization problem, during each iteration a datacenter \( j \) is randomly selected to autonomously update the chosen bid of its corresponding smart grid, by randomly selecting a bid \((e^*_j, b_j^*)\) from the bidding set, with smart grids’ bids remaining intact. If the new bid combination generates a larger social welfare, \((e^*_j, b_j^*)\) is kept with a carefully-selected high probability. The transition from one bid combination to another depends only on the current state and is irrelevant to previous states, and can hence be modelled with a Markov chain that converges to a steady state (i.e., achieving the maximal social welfare in WDP) after a finite number of transitions.

Algorithm 1 Decentralized sOcial welfaRe maximizAtion (DORA)

1: Initialization: each front-end server \( i \in S \) chooses a feasible workload distribution \( d^*_i \), and sends it to all the datacenters. Then each datacenter \( j \in D \) sets the bid \((e^*_j, b_j^*) = (e^{K_j}_j, b^{K_j}_j)\) that requires the most power consumption, to ensure that all the interactive workload can be served, and then chooses a feasible amount of batch workload \( \lambda^*_j \). Set \((\bar{\tau}_j, \bar{b}_j) = (e^*_j, b^*_j)\) and compute the corresponding social welfare \( W^* \).

2: Obtain \( \bar{d} \) and \( \bar{\lambda} \) by solving the following workload management problem:

\[
\begin{align*}
\max \quad & \sum_{j \in D} (b^*_j - e^*_j p_j) - \sum_{j \in D} V_j(\lambda^*_j) - \sum_{i \in S} U_i(d_i), \\
\text{s.t.} \quad & \text{constraint (1) and (3)}.
\end{align*}
\]

Set \( \bar{W} \) to be the maximal value of socal welfare maximization problem (8).

3: Compute the transition probability

\[
Pr = \frac{\exp(T \times \bar{W})}{\exp(T \times \bar{W}) + \exp(T \times W^*)}.
\]

4: With a probability of \( Pr \), each front-end server \( i \) sets \( d^*_i \leftarrow \bar{d}_i \), each datacenter \( j \) sets:

\[
\lambda^*_j \leftarrow \bar{\lambda}_j, (\bar{\tau}_j, \bar{b}_j) \leftarrow (e^*_j, b^*_j), \quad \text{and} \quad W^* \leftarrow \bar{W}.
\]

With a probability of \( 1 - Pr \), each datacenter sets \((e^*_j, b^*_j) \leftarrow (\bar{\tau}_j, \bar{b}_j)\).

5: Randomly select a datacenter \( j \in D \) to randomly select a bid \((\bar{e}_j, \bar{b}_j^*) \in \{(e^*_j, b^*_j), (e^*_j, b^*_j), \cdots, (e^{K_j}_j, b^{K_j}_j)\}\) that satisfies:

\[
\sum_{i \in D, i \neq j} \frac{e^*_j - \beta_i}{\alpha_i} + \bar{e}_j \geq \sum_{i \in S} D_i,
\]

to ensure that all the interactive workload can be served, and sets \((e^*_j, b^*_j) \leftarrow (\bar{e}_j, \bar{b}_j^*)\).

6: Return to step 2 until the stopping criteria is met.

DORA does not always choose the current best decision. The rationale is that for combinatorial optimization problems, the greedy approach that chooses the current best solution may lead to an arbitrarily bad outcome in the long run, such as the case of the classic knapsack problem [24]. To avoid such inefficiency, our proposed scheme carefully explores new solutions by introducing randomness to decision making. Specifically, when the new solution is better than the current solution (i.e., \( \bar{W} > W^* \), and thus \( Pr > 1/2 \)), the new solution is adopted with a higher probability \( Pr \), while the current solution will be kept with a lower probability \( 1 - Pr \).

The degree of randomness is controlled by a tunable smoothing parameter \( T > 0 \) in (9) that arbitrates the tradeoff between exploring better solutions and exploiting the current solution. As \( T \) increases, a new solution is kept with a greater probability if it is better than the current solution, leading to a more aggressive scheme. However, this greater probability is achieved at the cost of taking more iterations to explore the solution space, since our scheme emphasizes more on exploitation and may be stuck in a local optimum for a long time before successfully exploring other more efficient solution. Furthermore, since exploring better solution improves the optimality but also takes a longer time, the parameter controls the tradeoff between welfare maximization and computational complexity in our WDP solution.

B. Optimality Analysis

We now proceed to analyze and prove the optimality of our proposed algorithm DORA. Theorem 1 shows that it can solve WDP with an arbitrarily high probability.

Theorem 1: As \( T \) increases, the proposed algorithm DORA converges to the global optimal social welfare with a higher probability. When \( T \to \infty \), the algorithm converges to the global optimal social welfare with probability 1.

Due to space limit, interested readers can refer to our technical report [25] for the detailed proof.

C. Distributed Workload Management

The workload management problem. When determining the winning bids of each smart grid by randomly updating a bid, the optimization problem (8), i.e., workload management decision, needs to be solved multiple times, since there is a large number of combinations of feasible power demand bids. Consequently, it is desirable to parallelize the computation of workload management for efficient winner determination.

Once the power demand bids \((e^*_j, b^*_j)\) was selected, the remaining workload management problem can be reformulated into the following convex optimization problem:

\[
\begin{align*}
\min \quad & \sum_{i \in S} U_i(d_i) + \sum_{j \in D} V_j(\lambda_j), \\
\text{s.t.} \quad & \text{constraint (1)}, \\
& \sum_{i \in S} d_i + \lambda_j = w_j, \forall j \in D, \\
& \lambda_j = (e^*_j - \beta_j)/\alpha_j.
\end{align*}
\]

The ADMM Method. In this subsection, we propose a novel distributed algorithm to solve the workload management problem efficiently. Our algorithm is based on alternating direction method of multipliers (ADMM), a simple yet powerful algorithm that witnessed successful applications in a broad spectrum of problems from image processing to machine learning and applied statistics [15]. ADMM works well for linearly constrained convex problems whose objective function
is separable into two individual convex functions with non-overlapping variables. It alternatively optimizes part of the objective with one block of variables to reach the optimum with fast convergence.

Given a convex optimization problem in the form:

$$\min \ f(x) + g(z)$$

s.t. \quad \begin{align*}
Ax + Bz &= c, \\
x &\in C_1, \ z \in C_2,
\end{align*}$$

with variables \(x \in \mathbb{R}^m\) and \(z \in \mathbb{R}^n\), where \(A \in \mathbb{R}^{l \times m}\) and \(B \in \mathbb{R}^{l \times n}\) are relation matrices, \(A \in \mathbb{R}^l\) is a relation vector. \(f : \mathbb{R}^m \rightarrow \mathbb{R}\) and \(g : \mathbb{R}^n \rightarrow \mathbb{R}\) are convex functions, and \(C_1, C_2\) are non-empty polyhedral sets. Note that in ADMM, functions \(f\) and \(g\) are not required to be strictly convex.

The augmented Lagrangian of the above problem (11) can be written as:

$$L_{\rho}(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|^2,$$ \hspace{1cm} (14)

where \(\rho > 0\) is the penalty parameter. The augmented Lagrangian can be viewed as the unaugmented Lagrangian with a penalty term, thus the minimization of \(L_{\rho}(x, z, y)\) is equivalent to the original problem (11).

A generalized ADMM algorithm updates the original variables \(x, z\) and the dual variable \(y\) in an alternating fashion:

$$x^{t+1} = \text{argmin}_{x \in C_1} L_{\rho}(x, z^t, y^t),$$

$$z^{t+1} = \text{argmin}_{z \in C_2} L_{\rho}(x^{t+1}, z, y^t),$$

$$y^{t+1} = y^t + \rho(Ax^{t+1} + Bz^{t+1} - c).$$ \hspace{1cm} (15)

The optimality and convergence of ADMM is guaranteed under mild assumptions, as shown in the following lemma.

**Lemma 1:** Assume that the set \(C_i\) is bounded or else the matrix \(A^TA\) is nonsingular, and the solution set of optimization (11) is nonempty, then the sequence generated by ADMM converges to the optimal solution of the problem (11).

**Is ADMM directly applicable to problem (10)?** It is clear that \(d_{ij}, \forall i, j\) and \(\lambda_j, \forall j\) can be treated as two independent blocks of variables, while the objective function is separable for each block of variables. It seems that ADMM can be directly applied to problem (10). However, such a direct application results in a centralized algorithm with high complexity, since the coupling of \(d_{ij}\) happens on two orthogonal dimensions simultaneously: the dis-utility term \(\sum_{i \in S} U_i(d_i)\) couples \(d_{ij}\) across \(j\), while the penalty term \(\sum_{j \in D} \left(\sum_{i \in S} d_{ij} + \lambda_j - w_j\right)^2\) couples \(d_{ij}\) across \(i\).

Thus, the dis-utility function should be separated from the penalty term, if a distributed algorithm is pursued. To this end, we introduce a set of auxiliary variables \(d_{ij} = d_{ij}, \forall i \in S, \forall j \in D\), replace \(\lambda_j\) with \(w_j - \sum_{i \in S} a_{ij}\) in the utility term \(V_j(\lambda_j)\), and reformulate problem (10) as follows:

$$\min \ \sum_{i \in S} U_i(d_i) + \sum_{j \in D} V_j \left( w_j - \sum_{i \in S} a_{ij} \right),$$ \hspace{1cm} (16)

s.t. \quad \begin{align*}
\sum_{j \in D} d_{ij} &= D_i, \forall i \in S, \\
d_{ij} &= a_{ij}, \forall i \in S, \forall j \in D, \\
d_{ij} &= a_{ij} \geq 0, \forall i \in S, \forall j \in D.
\end{align*}

**Insight:** Problem (16) is equivalent to problem (10), where \(d_{ij}\) controls the dis-utility of processing interactive workload with only the load balance constraint, while \(a_{ij}\) determines the utility of running batch workload. \(d_{ij}\) and \(a_{ij}\) are connected through an equality constraint. That is the key idea that enables the minimization of both \(\lambda\) and \(a\) to be decomposable, as we will demonstrate in the following.

The augmented Lagrangian \(L_{\rho}\) of problem (16) can be readily obtained from (14). By omitting the irrelevant terms, at each iteration \(k + 1\), the \(d\)-minimization step involves solving the following problem according to (15):

$$\min \ \sum_{i \in S} \left( U(d_i) + \sum_{j \in D} \left( y_{ij}^t d_{ij} + \frac{\rho}{2} \left(d_{ij}^2 - 2a_{ij}^t d_{ij}\right)\right)\right)$$

s.t. \quad \begin{align*}
\sum_{j \in D} d_{ij} &= D_i, d_{ij} \geq 0, \forall i \in S,
\end{align*}$$

**Insight:** The problem described above is clearly decomposable over \(i\) into \(M\) per-front-end server sub-problems, since the objective function and constraint are separable over \(i\). Similarly, we also find that the minimization step for \(a\) is decomposable over datacenters. Therefore, the reformulated workload management problem (16) can be efficiently computed in a fully distributed manner.

The detailed iterative scheme of our proposed distributed ADMM algorithm is as follows:

**Algorithm 2 Distributed Workload Management**

1. Each datacenter \(j \in D\) initializes \(d_{ij}\) and \(y_{ij}\) to 0, \(\forall i \in S\). Each front-end server \(i \in S\) initializes \(a_{ij}\) to 0, \(\forall j \in D\).

2. **\(d\)-minimization:** Each front-end server \(i\) solves the following sub-problem for \(d_{ij}^{t+1}\), and broadcasts it to all the datacenters:

$$\min \ U(d_i) + \sum_{j \in D} \left( y_{ij}^t d_{ij} + \frac{\rho}{2} \left(d_{ij}^2 - 2a_{ij}^t d_{ij}\right)\right)$$

s.t. \quad \begin{align*}
\sum_{j \in D} d_{ij} &= D_i, d_{ij} \geq 0, \forall j \in S.
\end{align*}$$

3. **\(a\)-minimization:** Each datacenter \(j\) solves the following sub-problem for \(a_{ij}^{t+1}\), and broadcasts it to all the front-end servers:

$$\min \ \left( w_j - \sum_{i \in S} a_{ij}\right) + \sum_{i \in S} \left( \frac{\rho}{2} \left(a_{ij}^2 - 2a_{ij} a_{ij}^{t+1}\right) - y_{ij}^t a_{ij}\right)$$

s.t. \quad \begin{align*}
a_{ij} &\geq 0, \forall i \in S,
\end{align*}$$

4. **Dual update:** Each datacenter \(j\) updates \(y_{ij}\) for the equality constraint \(d_{ij} = a_{ij}\), and broadcasts \(y_{ij}^{t+1}\) to all the front-end servers:

$$y_{ij}^{t+1} = y_{ij}^t + \rho(a_{ij}^{t+1} - a_{ij}^{t+1}).$$

5. Return to step 2 until convergence.
load balance constraint (1), and hence the algorithm converges to the optimal solution according to Lemma 1.

V. PAYMENT DESIGN

In the previous section, we have computed the winning bid of each smart grid to maximize the social welfare in a decentralized manner. When implementing an auction, the goal is naturally two-fold: besides pursuing the economic efficiency by maximizing the social welfare, determining the payment of each winning bid to elicit truthful bids from each smart grid is equally important. We design the payment scheme for each smart grid by leveraging the VCG mechanism.

A. The VCG Payment Mechanism

The celebrated VCG mechanism is a well known type of auctions at the centre of truthful mechanism design. It is essentially the only type of auction that ensures both truthfulness and economic efficiency in terms of social welfare maximization [21]. However, the VCG mechanism suffers from vulnerability to shill bidding, precluding its direct applications in auction markets such as cloud computing platforms and secondary spectrum markets.

Recently, an alternative to VCG auctions, core-selecting auctions [26], have gained increasing attention in the literature, due to its economic efficiency, shill-proofness, and satisfactory seller revenue. To achieve these goals, core-selecting auctions make a compromise in absolute truthfulness. Furthermore, it is based on the hypothesis that seller utility is composed of all the bidders’ payments. However, as highlighted in Sec. III-D, our demand response auction is rather different from a conventional auction, in which the total utility of the auctioneer (the cloud operator) includes not only payments from the bidders, but also the varying electricity cost and dis-utilities from both interactive and batch workloads. As a result, core-selecting auctions can not be directly applied to our demand response setting.

Fortunately, when comparing our demand response auction to other auctions (e.g., virtual machine auction and wireless spectrum auction), we find that the demand response auction is naturally shielded from shill bidding. Each smart grid corresponds to a specific datacenter, and is unable to impersonate multiple bidders in the demand response auction. Therefore, applying the VCG payments to our demand response auction does not result in vulnerability to shill bidding.

The VCG payment scheme charges each winning bidder, who in our demand response auction wins exactly one bid, an amount equal to the externality or harm that it exerts on the other bidders. As a result, the utility of a winning bidder is the marginal contribution to the total values when it joins. Specifically, the VCG payment of each smart grid corresponding to datacenter \( m \in \mathcal{D} \) for its winning bid is:

\[
\begin{align*}
r_m = W(\mathcal{D}\setminus\{m\}) &- (W(\mathcal{D}) - b_m),
\end{align*}
\]

here \( W(\mathcal{D}) - b_m \) computes the social welfare of the geo-distributed cloud and all the other smart grids when smart grid corresponding to datacenter \( m \in \mathcal{D} \) joins the auction, while \( W(\mathcal{D}\setminus\{m\}) \) is the same aggregated social welfare when smart grid corresponding to datacenter \( m \in \mathcal{D} \) does not join the auction, which can be formulated as:

\[
\begin{align*}
\max & \sum_{j \in \mathcal{D}, j \neq m} b_j - \sum_{j \in \mathcal{D}} \left\{ V_j(\lambda_j) + e_j p_j \right\} - \sum_{i \in \mathcal{S}} U_i(d_i).
\end{align*}
\]

s.t. constraint (1)(3)(4).

Problem (20) can be solved by applying the decentralized method proposed in Sec. IV, a detailed solution is omitted here for sake of space. Furthermore, there are totally a number of \( N + 1 \) winner determination problems to be solved in VCG payment computation, and a multi-threaded implementation where each thread corresponds to a WDP instance can be adopted to expedite the overall payment computation process.

B. Truthfulness of the Payment Mechanism

Besides economic and computational efficiency, truthfulness represents a quite essential property commonly pursued in the design of auction mechanisms. The classic VCG mechanism ensures absolute truthfulness when the underlying WDP can be optimized precisely. In our demand response auction, as characterized in Theorem 1, optimality of the solution \( W(\mathcal{D}) \) and \( W(\mathcal{D}\setminus\{m\}), \forall m \in \mathcal{D} \) depends on the value of the tunable smoothing parameter \( T \) defined in Sec. IV-A. Based on Theorem 1, we further have the following Theorem 2, showing that our payment mechanism can achieve truthfulness with an arbitrarily high probability.

**Theorem 2:** As \( T \) increases, the proposed payment mechanism can achieve absolute truthfulness with a higher probability. When \( T \to \infty \), the VCG-based payment mechanism guarantees truthfulness with a probability of 1.

Recall that the smoothing parameter \( T \) also represents the tradeoff between social welfare maximization and computational efficiency. Together with Theorem 2, it shows that \( T \) acts as a design knob in our auction mechanism, striking a balance for the cloud provider between economic efficiency, truthfulness and computational efficiency.

VI. PERFORMANCE EVALUATION

In this section, we conduct trace-driven simulations to evaluate the practical economic benefits of the proposed incentive mechanism. The simulations are based on real-world workload traces and electricity prices.

![Fig. 3: Total request traffic of the HP workload trace.](image)

**A. Simulation Setup**

We consider Google’s six datacenters in the U.S. as a representative geo-distributed cloud. Each datacenter’s capacity is set to be \( 1.5 \times 10^9 \) processing servers. The location and electricity price of each datacenter are listed in Table I. Electricity
price $p_j$ is taken from the 2011 annual average day-ahead on peak prices at the local markets [16]. We also assume that the cloud deploys $M = 10$ front-end servers uniformly across the continental U.S. to distribute the interactive workloads.

<table>
<thead>
<tr>
<th>Datacenter Location</th>
<th>Price (USD/MWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Council Bluffs, IA</td>
<td>42.73</td>
</tr>
<tr>
<td>Berkeley County, SC</td>
<td>44.44</td>
</tr>
<tr>
<td>The Dalles, OR</td>
<td>32.57</td>
</tr>
<tr>
<td>Lenoir, NC</td>
<td>40.68</td>
</tr>
<tr>
<td>Mayes County, OK</td>
<td>36.41</td>
</tr>
<tr>
<td>Douglas County, GA</td>
<td>59.97</td>
</tr>
</tbody>
</table>

We use the one-day hourly HP request trace reported in the recent literature [8] to represent the request traffic of an interactive cloud service. The trace is scaled proportionally to the number of processing servers required, as shown in Fig. 3. We can see that the workload shows great variability and exhibits a clear diurnal pattern, typical for interactive cloud service. To imitate the geographical distribution of requests, we split this total workload among the $M = 10$ front-end servers following a normal distribution [16].

For power consumption of the servers in each datacenter, we choose a state-of-the-art setting where each server has a peak power $P_{peak} = 200W$, and consumes $P_{idle} = 100W$ when idling. We set a higher energy efficiency $PUE_j = 1.1$ for the six datacenters, which is consistent with Google’s datacenter energy efficiency. The propagation latency $L_{ij}$ is calculated according to the empirical approximation $L_{ij} = 0.02ms/Km \times h_{ij}$, where the geographical distance $h_{ij}$ can be obtained via Google Maps. We take the dis-utility and revenue loss functions defined in (5) and (6), respectively, with $q = 4 \times 10^{-6}$ and $\theta = 4.4 \times 10^{-3}$ to make the value of the dis-utility and revenue loss close to that of the electricity cost, which represents an impartial consideration on the impacts among the three aspects.

We let each smart grid submit 50 bids, such that the power demands are normally distributed in the power consumption interval [16,33]W. As to the parameters $E_j$, $E_j^-$ and $E_j^+$ associated with the demand response valuation curve in Fig. 2, we set $E_j^-=E_j^-5$ and $E_j^+=E_j^+5$, and generate the power demand bids of each smart grid $j$ according to the valuation function $b_j(c_j) = \max\{3000 - 120(E_j - e_j)^2, 0\}$ that satisfies the properties described in Sec. III-C.

**B. Performance**

For comparison, we further implement and evaluate the current real world situation, in which there is no demand response auction between the geo-distributed cloud and the datacenters.

**Social welfare improvement.** Fig. 4 depicts the social welfare under different schemes over 24 hours. We have the following observations: (1) Compared to the no auction scheme, the proposed datacenter demand response auction significantly improves the social welfare, demonstrating the economic efficiency of the proposed auction mechanism. (2) The social welfare improvement promoted by the demand response auction is relatively small when the interactive workload bursts, as shown by the statistic from hour 10 to hour 15. The rationale behind this decrement of social welfare is that when the interactive workload consumes a larger amount of datacenter capacity, the flexibility on power consumption associated with the batch workload is reduced. In consequence, a loss of the demand response efficiency is incurred.

**Improvement of the cloud’s utility.** Fig. 5 compares the utility of the geo-distributed cloud under different schemes over 24 hours. Clearly, it demonstrates that the cloud can dramatically improve its utility by launching the demand response auction. We also observe that, the cloud’s utility improvement is more remarkable when the amount of interactive workload decreases. This is because the datacenter can earn more payment in the demand response auction if it provides more flexible power consumption, while this can be achieved by leaving more capacity to the elastic batch workload when the amount of interactive workload is low.

**Improvement of the smart grids’ utility.** Fig. 6 plots the total utility of the smart grids under different schemes over 24 hours. It can be seen that, by participating in the demand response auction, the total utility of smart grids can be effectively improved at most of times, except for hour 4, 7 and 17. We should note that these few exceptions do not imply that our auction mechanism is inefficient in improving the stability of the smart grid. The reasonable interpretation to the exceptions is that in the absence of a demand response auction, the smart grid may occasionally have chances to get desired power consumption from the cloud without any payment, and thus obtain a high utility.

**Improvement of the smart grids’ stability.** We further demonstrate the efficiency of the proposed auction mechanism in improving the smart grids’ utility in a more straightforward manner. Specifically, we define the aggregated demand deficit (quantified by $\sum_{j \in S} |e_j - E_j|$) as the gap between the datacenter’s actual power consumption $(e_1, e_2, \cdots, e_N)$ and the most stable power consumption profile $(E_1, E_2, \cdots, E_N)$ to capture the stability of the smart grids. Fig. 7 shows that the stability of the smart grids can be largely improved when demand response auction is introduced. In particular, the aggregated demand deficit can be quite close to the most stable value 0 when the interactive workload is off-peak, corresponding to a larger social welfare as shown in Fig. 4.

**Influence of the maximal bidding price.** We also investigate the role of the maximal bidding price, which was set to $30000$ in the valuation function. In this simulation, we let the maximal bidding price vary, and plot the mean of the aggregated demand deficit over 24 hours in Fig. 8. We observe that, as the maximal bidding price increases, the mean of the aggregated demand deficit diminishes much faster and converges to a lower level. The observation indicates that the instability of the smart grid can be eliminated by bidding via moderate higher prices.
Flexibility of batch workload on demand response. The computing-resource-elastic nature of batch workload enables substantial flexibility on its power consumption, and thus provides great potential for datacenter demand response. In this simulation, we assess the flexibility of batch workload on demand response. Specifically, by scaling the interactive workload trace while still keeping the capacity of each datacenter unchanged (corresponding to varying the available capacity for batch workload), we plot the mean of the aggregated demand deficit over the 24 hours under various interactive workload scaling ratios (i.e., how much we scale the amount of request traffic presented in Fig. 3) in Fig. 9. As expected, the aggregated demand deficit increases as the scaling ratio grows, demonstrating that a greater potential is provided if there is more capacity for batch workload.

VII. CONCLUDING REMARKS

This work presented perhaps the first study on the auction mechanism design for demand response from a geo-distributed cloud. Relying on existing approaches for the mechanism design of such market is impracticable, since the demand response auction is substantially different from conventional auctions in two significant aspects. To address this challenge, we first resort to the distributed computing capacity of the cloud and propose the decentralized social welfare maximization algorithm DORA, by incorporating techniques from the Gibbs sampling method and the alternating direction method of multipliers. The payment mechanism is then designed based on the classic VCG mechanism. Extensive trace-driven evaluations demonstrate that our incentive mechanism facilitates a win-win solution to both the geo-distributed cloud and the smart grid.

REFERENCES


