A Truthful and Efficient Incentive Mechanism for Demand Response in Green Datacenters

Zhi Zhou, *Member, IEEE,* Fangming Liu, *Senior Member, IEEE,* Shutong Chen,
Zongpeng Li, *Senior Member, IEEE*

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 service provider (CSP) that operates geo-distributed datacenters. To incentivize CSP participation, this work presents an auction mechanism that enables smart grids to voluntarily submit bids to the CSP to procure diverse amounts of demand response with different payments. To maximize the social welfare of the auction, the CSP that acts as the auctioneer needs to solve the winner determination problem at large-scale. By applying the proximal Jacobian alternating direction method of multipliers, we propose a distributed algorithm for each datacenter to solve a small-scale problem in a parallel fashion. Desirable properties of the proposed auction, such as social welfare maximization and truthfulness are achieved through Vickrey-Clarke-Groves (VCG) payment. Through extensive evaluations based on real datacenter workload traces and IEEE 14-bus test systems, we demonstrate that our incentive mechanism constitutes a win-win mechanism for both the geo-distributed cloud and the smart grid.

Index Terms—Geo-distributed datacenters, smart grid, demand response, incentive mechanism, distributed algorithm.

1 Introduction

The recent years have witnessed new emerging technology advances in the ICT sector, among which two are widely recognized as having profound impacts. 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, distribution 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 \$67M [1], a number continuing to rise with the flourishing of cloud services and the rise of electricity prices. Meanwhile, the smart grid that integrates a large number of distributed generations such as solar arrays and wind turbines also faces severe operation instability and hence economic issues, due to the intermittent nature of distributed generation. For example, the enormous wind generation in May 2014 in Germany incurred 5 hours of continuous negative electricity prices [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 world wide 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). Besides 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 the demand response program can help to ease the burden from its growing 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, as pointed out by the Green Grid Association, *"utilities and datacenters do not mix yet" [4]*. This is 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 data-

- *Z. Zhou is with Guangdong Key Laboratory of Big Data Analysis and Processing, School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China. E-mail:* zhouzhi9@mail.sysu.edu.cn.
- *F. Liu and S. Chen are with the Services Computing Technology and System Lab, Cluster and Grid Computing Lab in the School of Computer Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074, China. E-mail:* {fmliu, shutongchen}@hust.edu.cn.
- *Z. Li is with School of Computer Science, Wuhan University, Wuhan 430072, China. E-mail:*zongpeng@whu.edu.cn.

This work was supported in part by the National Key Research & Development (R&D) Plan under grant 2017YFB1001703, in part by NSFC under Grant 61722206, 61802449, 61761136014 (NSFC-DFG) and 61520106005, and in part by the Fundamental Research Funds for the Central Universities under Grant 2017KFKJXX009 and 2017LGJC40. The corresponding author is F. Liu.

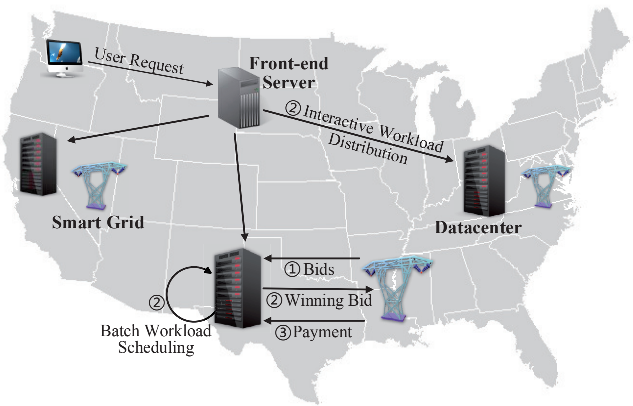


Fig. 1: The smart grid and datacenter system overview, as well as the process of our proposed demand response auction.

centers is being addressed [5]. 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 [5], for cost savings may be insufficient to incentivize datacenters to automatically respond. Furthermore, most existing demand response schemes (see [5] and the references therein) 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, in line with the suggestion of the Green Grid Association — "*energy efficiency programs need to place greater focus on marketing and outreach*" [4], an economically efficient demand response program tailored for geo-distributed datacenters to realize their potential is needed.

This work aims to design an efficient mechanism for trading demand response between the CSP and smart grids. The proposed solution for incentivizing CSP participation and social cost minimization for demand response is an auction with multiple buyers (smart grids) and a single seller (CSP). Unlike the current demand response programs (e.g., [5]–[8]) in which the CSP passively accepts the prices given by the smart grids, auction enables each smart grid to voluntarily submit multiple bids to express its willingness to pay for different levels of power demand, and then the CSP autonomously determines the winning bids and real payment of each smart to maximize the social welfare and attain its own benefit. The proposed divisible auction framework is illustrated in Fig. 1: each regional smart grid first submits the bidding function, *i.e.,* the collection of all its feasible bids to the corresponding datacenter, to express its willingness to pay for different levels of power demand. After receiving the bidding functions from all the smart grids, the CSP 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, defined as the aggregate satisfaction from both the CSP and smart grids. Finally, the CSP computes a truthful payment for each smart grid, which incentivizes them to reveal their true 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 has a non-monotonic bidding function, and (ii) the utility of the CSP includes not only the payments from the smart grids, but also the electricity charges, the revenue from both interactive workloads and batch workloads. Thus, instead of relying on existing centralized approaches for mechanism design, we propose to effectively settle the large-scale winner determination problem by exploiting the distributed computing capacity of the cloud. Leveraging the recently developed multi-block alternating direction method of multipliers (ADMM), we efficiently maximize the social welfare, *i.e.*, solve the winner determine problem, in a parallel and fully distributed manner. Though different from conventional auctions, by adapting the classic VCG mechanism to determine the payment of each smart grid, the proposed mechanism is proven to achieve desirable economic properties that include truthfulness, economic efficiency and individual rationality.

As an extension of the preliminary conference version [9], this work extends the combinatorial auction in [9] to a divisible auction which better captures the divisible nature of power demand. Specifically, for general bidding setup, the combinatorial auction approximately maximizes the social welfare in a decentralized and near-optimal manner using Gibbs sampling and 2-block ADMM. While for convex bidding function, the divisible auction optimally maximizes the social welfare in a fully distributed manner by leveraging the recently developed multi-block ADMM approach. Extensive trace-driven simulations in Sec. 6 verify the effectiveness of both the combinatorial auction and the divisible auction.

2 Background and Related Work

2.1 Background and motivation

Datacenter demand response. Demand response is identified as a priority for the future smart grid, by both the National Institute of Standards and Technology (NIST) and the Department of Energy (DoE) [5]. Among all the end users, such as factories, buildings, residential houses and electrical vehicles, datacenters constitute a particularly promising segment, as verified by the Lawrence Berkeley National Laboratories (LBNL): "Datacenters are excellent candidates with great potential for smart grid demand response" [5]. This is because (1) datacenters are extremely large loads (typically tens of MW) in the smart grid, as exemplified by Facebook's datacenter in Crook County, Oregon, which makes up 50% of the load in the power grid [3]. (2) Datacenter power drawn from the grid is highly elastic, since it can be flexibly rectified by a wide variety of approaches including hardware-based, software-based and power source-based, as summarized in Table 1.

Demand response from datacenter and other loads: similarities and differences. As discussed, datacenters can take various approaches to do demand response; in reality, similar approaches are also applicable to other power consumers. For example, a factory or a residential home can shutdown devices (*e.g., lamps*) when not needed. For the air conditioners widely deployed in buildings, the temperature set points may be flexibly adjusted within a tolerable

TABLE 1: Demand response strategies and their advantages/disadvantages.

Category	Strategy	Advantages	Disadvantages
Hardware-based	Idle Server shutdown/sleep [10]	Idle power reduction	Vulnerability to traffic bursty
	Temperature adjustment [11]	Cooling power reduction	Hardware reliability degradation
Software-based	Workload management [11]	Quick response time	Service quality degradation
Power source-based	Local Generation [1]	Large controlling range	High cost and emission
	Energy Storage [1]	Quick response time	Vulnerability to power emergency

range. Similarly, local generation is becoming standard for power consumers sensitive to power emergencies, and can be started when demand response is required. Finally, due to the promotion of commercial companies such as Tesla, energy storage is gaining popularity in electrical vehicles, buildings and homes, which can jointly make a great difference in demand response.

Perhaps the most significant difference between datacenter demand response and demand response from other loads is the freedom in workload (or power load) management. For other loads, power load management is typically performed through *temporal load shifting*, *i.e.*, a power user may advance or postpone its load to avoid the peak-hours. For geo-distributed datacenters, workload management can be realized in the form of *spatial load shifting* (*i.e.*, geographical load balancing) and batch workload scheduling. In terms of geographical load balancing, the interactive workload driven by front-end user requests can be dispatched to datacenters with sufficient power supply. Except for the interactive workload, datacenters nowadays commonly process batch workload such as data mining and web crawling frameworks that reside constantly at the back-end. Different from interactive workload that requests fixed amount of resources, batch workload is elastic to resource allocation and power consumption. Unfortunately, both geographical load balancing and batch workload scheduling would incur service quality degradation, since the former may increase the RTTs and the latter may compromise output accuracy.

Why a new market for datacenter demand response? Power load shaping is traditionally achieved via real-time electricity pricing. Unfortunately, for geo-distributed datacenters, the above traditional scheme may not work well, due to the following distinct features of datacenters: (1) datacenters are extremely large energy consumers, they typically sign long-term energy purchase contracts with generators at locked-in electricity price, in order to avoid the risk of electricity price volatility. For example, in 2010 and 2011, Google secured two 20-years contracts with wind farms to power its datacenters in Iowa and Oklahoma, respectively [12]. (2) In traditional dynamic electricity pricing program, the smart grid dynamically adapts the power price to shape the power consumption of price-sensitive users. As the power demand of traditional users exhibits a strong periodic pattern, the smart grid can readily predict the behaviour of the users, by using some machine learning based methods that have been extensively studied and verified in literature [13]. However, for a geo-distributed cloud, the workload is allocated to multiple geo-distributed datacenters based on multi-criteria as exemplified by energy cost, performance, carbon emission, etc [10], [11], [14]. Nevertheless, since the smart grids are unaware of the private parameters such as power usage efficiency, server capacity and performance coefficient of those geo-distributed datacenters [14], the decision-making process of geographical load balancing is a black-box for those smart grids. As a result, the smart grids are unable to confidently evaluate the datacenters' response to their electricity prices.

The above analysis shows how long-term energy contracts and geo-distributed nature of datacenters inhibit the effectiveness of current demand response programs, this is inline with recent literature on datacenter demand response. For example, both the utility program influence survey conducted by the Green Grid Association [4] and a recent report [15] show that datacenters contribute little to demand response. To address this problem, the Green Grid Association [4] and Lawrence Berkeley National Laboratory [16] have suggested that market design for datacenter demand response deserves more attention. Following this thread, we proposed an auction-based incentive mechanism tailored for geo-distributed datacenters. In the proposed mechanism, each smart grid first voluntarily submits a bidding function to the CSP to express its willingness to pay for different levels of power demand, then the CSP determines the power load and the real payment of each datacenter by maximizing the social welfare.

Since the payment for demand response is determined by the CSP, some readers may wonder whether the proposed program subverts traditional incentive programs and it would be manipulated by the CSP. Actually, our proposed market only introduces moderate changes to the traditional incentive program and would not harm the position of the smart grids. First, our proposed program can co-exist with and supplement to the existing demand response programs such as dynamic electricity pricing. Specifically, by enhancing dynamic electricity pricing with our proposed program, the inefficiency of the former on demand response can be alleviated. Besides, in our proposed program, each smart grid submits a bidding function to the CSP, though the final payment is determined by the CSP, it could not exceed the bidding price which is controlled by the smart grid. For the above reasons, we believe that our proposed market is friendly to the stringent regulation of electricity market.

2.2 Related work

To curb the escalating energy cost, a large body of recent research has been devoted to improve energy efficiency of datacenters. To this end, power proportional techniques such as server right-sizing (*i.e.*, shutting down idle servers) and dynamic voltage and frequency scaling (DVFS) can be applied to reduce the server power consumption [5], [10], [17], geographical load balancing for workload can be adopted to exploit the spatial diversities of electricity, carbon emission and cooling efficiency at different datacenter locations [10], [11], [18]. While for delay tolerant workload, joint optimization on geographical load balancing and temporal workload scheduling can be applied to reduce the energy cost of geo-distributed datacenters [19], [20]. Our work is supplementary to the above literature, since an efficient market for datacenter demand response further reduces the energy-related cost achieved by the above strategies.

The interaction between a geo-distributed cloud and a smart grid has been studied by Wang *et al.* [6], through a single-leader single-follower Stackelberg game model, assuming that the geo-distributed datacenters are covered by a single smart grid. More recently, Zhou *et al.* [7] and Tran *et al.* [8] studied the pricing for the geo-distributed cloud which trades to completing smart grids, both through a multi-leader single-follower pricing game approach. For demand response of geo-distributed colocation datacenters that serves delay-tolerant workload, an online auction mechanism has been proposed by Sun *et al.* [21]. Our work is supplementary to the above literature in the following aspects: (1) the above studies assume that the smart grids know the exact utility function of the CSP, which is not always practical. In contrast, in our auction, the private valuation of each smart grid is extracted through a truthful payment mechanism. (2) Rather than only consider interactive or delay-tolerant workload, we consider the more practical and challenging scenario of a mix of both interactive and batch workload.

As an effective pricing mechanism, auction theory has witnessed a wave of successful applications in a broad range of IT problems, as exemplified by cloud resource provisioning [22], spectrum management [23] and network bandwidth allocation [24], etc. Unfortunately, our problem is shown to departure from standard auction in two crucial aspects (elaborated in Sec. 3.3 and Sec. 3.4). In response to our non-standard setup, we verify that some well-known mechanisms such as dimensional Kelly mechanism [24] and monotonic allocation [23] are not applicable, and further theoretically prove the availability of VCG mechanism in such a non-standard setup. To address the challenge of prohibitively high computation complexity incurred by VCG mechanism, we take advantage of distributed computing power of the CSP to accelerate the implementation the VCG auction mechanism in a fully parallel and distributed manner, based on the recently proposed multi-block alternating direction method of multipliers (ADMM).

3 Demand Response Auction Model

3.1 Overview of the Geo-distributed Cloud Platform

Consider a CSP running cloud services on a set of N geographically dispersed datacenters, $\mathcal{D} = \{1, 2, ..., N\}$. 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, $\mathcal{S} = \{1, 2, ..., M\}$, here front-end servers represent mapping nodes deployed at distinct geographical regions with moderate size (e.g., a city), such as authoritative DNS servers as used by Akamai and most CDNs, or HTTP ingress proxies used by Google and Yahoo [25]. The function of a front-end server is to aggregate the user requests of the region and then route them to appropriate datacenters based on certain criteria.

To capture and follow the system dynamics such as time-varing electricity price and workload, we adopt a discrete time-slotted model where the length of a time slot matches the time scale at which demand response pricing and workload management decisions are periodically updated, e.g., hourly. At each time slot $t = (0, 1, 2, \cdots)$, the total amount of incoming interactive workload (in number of processing servers required) at the front-end server i is $D_i(t)$, and the amount of interactive workload distributed from front-end server i to datacenter j is $d_{ij}(t)$, which is to be determined. Considering the enormous amount of user requests of typical applications such as web-service, we allow a front-end server to arbitrarily split the user requests among the set of datacenters. In practice, such a flexibility can be achieved by commercial DNS servers and HTTP proxies that use the hash-based scheme to split and route the user requests to different datacenters [25].

In this paper, we assume that the interactive requests are computation-intensive rather than data-intensive. Therefore, the back-end input data for the cloud services can be fully replicated across the geo-distributed datacenters with very slight overhead, and an interactive request can be distributed to any datacenter for execution. While for data-intensive workload such as analytic jobs, running them across geo-distributed datacenters is far from trivial. Since it requires a joint optimization on the task distribution and input data movement to reduce the additional overheads such as WAN-bandwidth cost and data transmission delay incurred by the cross-datacenter bulk data transfer [26]. Dealing with data-intensive workload is out of the scope of this paper, and we leave it as our future work.

Besides interactive workload, datacenters nowadays commonly process back-end batch workload that has elastic resource demand, as exemplified by maching learning model training, web crawling and data mining jobs, running at the back-end [11]. Let $\lambda_j(t)$ represents the amount of batch workload (also in the number of processing servers allocated) to be executed in datacenter j at time t. The aggregated workload at datacenter j and time slot t is then $\sum_{i\in\mathcal{S}} d_{ij}(t) + \lambda_j(t)$. Note that for interactive requests, since they typically require to be finished immediately, i.e., within a tolerable delay up to several seconds which is far smaller than the time length of a slot, therefore all the requests arrived at each time slot can be finished within the time slot. While for batch workload, it is resource flexible as it does not predefine a resource quota that temporally couples resource allocation. Thus, the decision making of the CSP is uncoupled from slot to slot, and we can focus on a single time slot and drop index t henceforth.

3.2 Datacenter Power Consumption

A series of recent empirical studies [14] show that, the aggregated power consumption of homogeneous servers can be modelled as a linear function of the total workload, $sP_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}})\mu$. Here s and μ denote the number of running servers and the amount of workload, respectively. P_{idle} is the server power when idle, while P_{peak} is the server power when fully utilized. For a datacenter $j \in \mathcal{D}$ that hosts S_j homogeneous active servers, manages a total workload of $\sum_{i\in\mathcal{S}} d_{ij} + \lambda_j$ and possesses a power usage efficiency of PUE_j, its server power consumption is: $S_j P_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}})(\sum_{i\in\mathcal{S}} d_{ij} + \lambda_j)$. Furthermore, the total power demand at datacenter j is:

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

The *power usage efficiency metric* 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. We next reformulate the above equation of e_j for concise presentation:

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

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

In the above formulation, we assume that all the servers at each datacenter are powered on and thus the parameter S_j is a constant. The rationale is that for risk-sensitive commercial cloud services such as Amazon, reliability and wear-and-tear are more of the concern than shutting down the idle servers to reduce energy [27], [28]. Nevertheless, our model is quite general and can be easily extended to incorporate the choice of shutting down the idle servers for further energy savings [29]. Specifically, if we use $S_j^{\max}$ to denote the number of total available servers in datacenter j, then, we further need to determine the number of active server S_j which satisfies the additional constraint $S_j \leq S_j^{\max}$ for each datacenter j.

3.3 Demand Response Bidding

Before presenting our datacenter demand response auction, we assume that a single datacenter locates within the geographical span of a smart grid, in line with the fact that regional smart grids usually cover a moderate-size district. Table 2 below shows that Google's 6 datacenters in USA purchase electricity from different power utilities.

TABLE 2: Power utilities for Google datacenters in America [18].

Location	**Utility**
Council Bluffs, IA	MidAmerican Energy
Berkeley County, SC	South Carolina Electric
The Dalles, OR	Northern Wasco County PUD
Lenoir, NC	Duke Energy
Mayes County, OK	The Grand River Dam Authority
Douglas County, GA	Georgia Power

At the beginning of each time slot, each smart grid corresponding to datacenter j first computes the demand response goal, *i.e.*, the desired power consumption by datacenter j, $\hat{e}_j$, which minimizes the voltage violation frequency of that smart grid. In practice, the power usage behavior of traditional users (non-datacenter users) in the smart grid exhibits strong periodical pattern and price-sensitivity, thereby can be readily predicted by taking some machine learning methods that have been extensively researched in literature [13]. Then, based on the predicted power demand of non-datacenter users and the real-time supply of renewable generation, the smart grid can compute the demand response target $\hat{e}_j$ for the datacenter. Specifically, given the profile of the power distribution network (*e.g.*, the SCE 47-bus network in [3], and the IEEE 14-bus network used in performance evaluation in Sec. 6), by applying the "branch flow" model [3] to explore the feasible region of datacenter power consumption, $\hat{e}_j$ can be found.

Based on the calculated $\hat{e}_j$, 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. Here we use a *continuous* function $V_j(\cdot)$ to capture the maintenance cost savings brought by demand response from datacenter j. In particular, the valuation function is $V_j(e_j - \hat{e}_j)$. We assume this function is concave, non-negative, incresing when $e_j - \hat{e}_j \leq 0$ and decreasing when $e_j - \hat{e}_j \geq 0$. The continuity captures the *divisible nature of power consumption*. The concavity assumption is widely adopted in literature [3], for capturing the fact that the marginal maintenance cost increases as e_j deviates the target $\hat{e}_j$ more. For illustration, based on the quadratic *Taguchi loss function* which has been used to evaluate the cost of a demand response mechanism in [30], we take the following valuation function as an example:

$$V_j(e_j - \hat{e}_j) = b_j - \frac{c_j}{2}(e_j - \hat{e}_j)^2, \tag{1}$$

where c_j is the price that converts the power deviation $e_j - \hat{e}_j$ to a monetary term, and b_j denotes the maximal cost saving that is achieved when the deviation $e_j - \hat{e}_j$ is equal to 0.

A primary difference between the proposed demand response auction and conventional auctions such as combinatorial auctions for cloud VMs [22], divisible auction for network bandwidth [24], is the *non-monotonicity* of the valuation function $V_j(e_j - \hat{e}_j)$. For each smart grid, its valuation on the power consumption of the corresponding datacenter first increases and then decreases as the power consumption grows. Such non-monotonicity captures the fact that neither too high or too low datacenter power consumption is beneficial to demand response.

The operator of each smart grid corresponding to datacenter j has a *quasi-linear utility*:

$$u_j(e_j) = V_j(e_j - \hat{e}_j) - r_j, \tag{2}$$

where r_j is the payment of the smart grid corresponding to datacenter j for winning a bid of power consumption e_j.

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

While the CSP 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 CSP's utility, including the 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 cloud applications such as web search and social networking, latency is a critical performance metric. Even a small increase in latency can significantly impact the revenue of service providers, as demonstrated by measurements conducted by internet giants [31]. For Google, an additional 400 ms latency in search responses reduces search volume by 0.74%. For Amazon, a 100 ms latency increase implies a 1% sales loss. For Bing, a 500 ms latency increase leads to 1.2% revenue loss, while a one second latency increase can lead to 2.8% revenue loss.

The round-trip times (RTT) L_{ij} between the front-end server i and datacenter j can be obtained through active measurements in practice [14]. Empirical studies have also demonstrated that, in backbone networks, L_{ij} can be approximated by geographical distance l_{ij} between the front-end server i and datacenter j as: $L_{ij} = l_{ij} \times 0.02\text{ms/km}$ [14]. The revenue from serving the interactive workload aggregated at front-end server i depends on the experienced mean propagation latency $\sum_{j \in \mathcal{D}} d_{ij} L_{ij} / D_i$, through a generic utility function U_i that is decreasing and concave. A commonly adopted revenue function is quadratic on the mean latency, reflecting a user's increased tendency to leave the service with an increased latency [18]:

$$U_i(d_i) = qD_i\Big(w_i - \Big(\frac{\sum_{j\in\mathcal{D}} d_{ij}L_{ij}}{D_i}\Big)^2\Big), \tag{3}$$

where $d_i = (d_{i1}, d_{i2}, \cdots, d_{iN})^{\mathsf{T}}$, q is the price that converts latency to a monetary term, and qD_iw_i can be treated as the revenue when the latency approaches zero.

Revenue from resource-flexible batch workload. System-level batch workload, such as indexing, data mining and machine learning jobs that are originated from the back-end datacenters rather than the front-end clients, has witnessed great proliferation in datacenters with the booming of big data and artificial intelligence (AI). One salient and appealing feature of batch workload is that it is resource-flexible and thus can be *partially served* with an arbitrary amount of resource allocation. For instance, for some approximated big data analytics [32] applications, an analytic job can be executed with only a subset of its tasks to be completed, leading to various levels of resource allocation. While for system level AI applications (e.g., recommendation system of an e-commerce website, and the click rates prediction for internet ads), inference models with different sizes can be employed [33], also resulting in various amounts of resource consumption. Intuitively, the more computing resources allocated to batch workload, the better performance and thus higher revenue the datacenters can glean. The marginal improvement of performance however is diminishing as the allocated resources increase. A typical application that exhibits this property is web-page indexing: as demonstrated by 200K queries in a production trace of Microsoft Bing, the indexing quality profile is concave on the amount of allocated computing resources [34]. To model the revenue of allocating processing servers in datacenter j for batch workload, a differentiable, increasing and concave function $H_j(\lambda_j)$, as exemplified by the logarithmic function, can be adopted, as the revenue is zero when no computing capacity is allocated to batch workload:

$$H_j(\lambda_j) = \theta\log(1+\lambda_j), \tag{4}$$

where θ 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: $\sum_{j\in\mathcal{D}} e_jp_j$. Here p_j can be the real-time electricity price dynamically adjusted by the smart grid, or a wholesale price signed with the smart grid in advance.

In the demand response auction, the CSP receives a total payment of $\sum_{j\in\mathcal{D}} r_j$ from the smart grids. We are now ready to formulate **the revenue of the CSP in the demand response auction**: $\sum_{j\in\mathcal{D}}\Big\{r_j + H_j(\lambda_j) - e_jp_j\Big\} + \sum_{i\in\mathcal{S}} U_i(d_i)$.

A second difference (besides *the non-monotonic valuation function*) 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 revenue from serving both interactive and batch workloads, and energy cost that depends on the allocation of the datacenter power consumption.

3.5 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), $\mathrm{W}(\mathcal{D}) =$

$$\begin{aligned}
\max \quad & \sum_{j\in\mathcal{D}}\Big\{V_j(e_j-\hat{e}_j) + H_j(\lambda_j) - e_jp_j\Big\} + \sum_{i\in\mathcal{S}} U_i(d_i), && (5)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, \forall i\in\mathcal{S}, && (6)\\
& \sum_{i\in\mathcal{S}} d_{ij} + \lambda_j \le S_j, \forall j\in\mathcal{D}, && (7)\\
& e_j = \alpha_j(\sum_{i\in\mathcal{S}} d_{ij} + \lambda_j) + \beta_j, \forall j\in\mathcal{D}, && (8)\\
& d_{ij}\ge 0, \lambda_i \ge 0, e_j \ge 0, \forall i\in\mathcal{S}, \forall j\in\mathcal{D}. && (9)
\end{aligned}$$

The load balance constraint (6) ensures the interactive workload from each front-end server is satisfied. The capacity constraint (7) ensures the aggregated workload processed at each datacenter does not exceed the latter's capacity. (8) is the relation between power consumption and aggregated workload at each datacenter, as derived in Sec. 3.2. Finally, (9) is the nonnegativity constraint. Note that the functions $V_j(\cdot)$, $H_j(\cdot)$ and $U_i(\cdot)$ are not restricted to the specific forms presented in the equations (1), (3) and (4). Instead, our model and algorithm to be presented are generally applicable to those functions with mild assumptions such as concavity.

In our model, the auctioneer maximizes the social efficiency (i.e., social welfare) rather than its own utility. Note that this is a de facto setup in many auctions. The rationale is that, when maximizing the social efficiency, the happiness of both the auctioneer and the bidders can be improved. By doing so, the sustainability and attractiveness of the auction to the bidders can be firmly maintained, which in return improve the long-term market share and thus the revenue of the auctioneer.

The above model makes simplifying assumptions to focus on the most fundamental features of a datacenter demand response auction. We now briefly discuss how this model can be further extended to accommodate a variety of practical operational conditions of a geo-distributed cloud.

Delay-tolerant Workload. Delay-tolerant workload, as exemplified by scientific computing and video transcoding, represents another important segment of datacenter workload. Since this type of workload is tolerant to a soft deadline typically ranging from minutes to days, it greatly complicates our mechanism design by temporally coupling the optimization problem across consecutive slots. Fortunately, efficient management of delay-tolerant workload has been well-studied in literature [21]. By taking such schemes, we can decouple the long-term optimization problem into a series of one-shot optimization problems. Then, the proposed mechanism in this paper can be readily incorporated into each one-shot optimization problem.

Extending the 1-to-1 trading. The basic auction model assumes that each datacenter trades with only one smart grid. In practice, multiple datacenters may locate within one smart grid, and one datacenter may trade with multiple smart grids. The underlying problem does not change fundamentally. For the case that multiple datacenters locate within one smart grid, the smart grid needs only to submit one bid to each of the datacenters. For the case that one datacenter trades with multiple smart grids, one may introduce an additional constraint to ensure that the sum of the winning bids of these smart grids equals the power consumption of the datacenter.

Multiple CSPs. The basic auction model assumes a single CSP that operates multiple geo-distributed datacenters. A practical direction of extending this model is to consider multiple CSPs (e.g., Google, Microsoft and Facebook) co-existing and trading with multiple smart grids. To accommodate such a complicated multi-CSP multi-smart-grid scenario, the well-studied double auction model [35] can be applied, in which a third-party auctioneer first collects the sell bids from CSPs and buy bids from smart grids. Then, the auctioneer determines the received/payed payment of each CSP/smart grid by maximizing the social welfare.

In this paper, our focus lies mainly on the basic auction problem which is inherently difficult, as we show as follows. The WDP is a large-scale convex optimization problem, as the number of front-end servers is typically $O(10^5) - O(10^6)$, and number of datacenters is typically $O(10)$ for business production systems such as Google [11]. Thus, our problem can have tens of millions of variables and million of constraints, solving such large-scale problem with standard convex solvers is unpractical since it may take hours. 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, to efficiently share the divisible network resources, proportional allocation mechanisms were introduced in [24], and VCG mechanisms with dimensional bids were introduced in [36]. Unfortunately, the first difference between our auction and the conventional auction, that the valuation function of each bid is non-monotonic, invalidates these approaches. Furthermore, the classic method that exploits monotonic allocation and critical value based charging was widely adopted in truthful auction design [23]. 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 as well as the large-scale 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 abundant server resource distributed across its datacenters. A natural question is then, can we leverage this enormous computing power to facilitate the auction design? The answer is 'yes', and in the next section, we describe how to parallelize the winner determination problem to significantly reduce the computation time.

4 Distributed Social Welfare Maximization

Since conventional auction design approaches are not directly applicable in our problem, in this section, we propose a distributed algorithm to efficiently address the challenge of solving the WDP at large-scale. Our algorithm is based on the *alternating direction method of multipliers* (ADMM), a simple yet powerful algorithm for convex optimization that witnessed successful recent applications in a broad spectrum of problems from image processing to machine learning and applied statistics. The performance of ADMM in solving large-scale convex problems has been extensively studied.

4.1 The ADMM Method

ADMM works for linearly constrained convex problems whose objective function is separable into multiple convex functions with non-overlapping variables:

$$\min \sum_{i=1}^{m} f_i(x_i), \quad \text{s.t.} \sum_{i=1}^{m} A_i x_i = z, \tag{10}$$

with variables $x_i \in \mathbb{R}^{n_i}$ $(i = 1, ..., m)$, where $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ $(i = 1, ..., m)$ are closed proper convex functions, $A_i \in \mathbb{R}^{l \times n_i}$ are relation matrices, and $z \in \mathbb{R}^l$ is a relation vector. Note that the model (10) can easily accommodate general linear inequality constraints $\sum_{i=1}^{m} A_i x_i \leq z$ by adding one extra block. In particular, we can introduce a slack variable $x_{m+1} \geq 0$ and rewrite the inequality constraint as $\sum_{i=1}^{m} A_i x_i + x_{m+1} = z$.

A commonly adopted approach to the separable convex optimization (10) is to form the augmented Lagrangian by introducing an extra $L2$ norm term $\| \sum_{i=1}^{m} A_i x_i - z\|^2$ to the objective:

$$\begin{aligned}\mathcal{L}_\rho(x_1, ..., x_m; y) \quad = \quad & \sum_{i=1}^{m} f_i(x_i) + y^{\mathsf{T}}(\sum_{i=1}^{m} A_i x_i - z) \\ & + \frac{\rho}{2} \| \sum_{i=1}^{m} A_i x_i - z\|_2^2, \end{aligned} \tag{11}$$

where $\rho \geq 0$ is the penalty parameter. Clearly the minimization of $\mathcal{L}_\rho(x_1, ..., x_m; y)$ is equivalent to the original problem (10).

When there are two blocks of variables, *i.e.*, $m = 2$, the iterative scheme of ADMM method decomposes the two blocks of variables in the following Gauss–Seidel *sequential* manner

$$\begin{aligned} x_1^{k+1} &= \arg\min_{x_1} \mathcal{L}_\rho(x_1, x_2^k; y^k), \\ x_2^{k+1} &= \arg\min_{x_2} \mathcal{L}_\rho(x_1^{k+1}, x_2; y^k), \\ y^{k+1} &= y^k + \rho(\sum_{i=1}^{m} A_i x_i^{k+1} - z). \end{aligned} \tag{12}$$

Note that in each iteration, the augmented Lagrangian is minimized over x_1 and x_2 separately. Consequently, the generated subproblems are of smaller-scale and are easier to solve. The convergence of ADMM with two blocks of variables has been well understood.

However, when there are more than two blocks of variables, the Gauss–Seidel type direct extension of the 2–block ADMM does not necessarily converge [37], unless the functions f_i $(i = 1, \cdots, m)$ are strongly convex [11]. Some recent progresses have been made to establish the global convergence of ADMM with $m \geq 3$. Among which, the *proximal Jacobian ADMM* [38] is an iterative scheme for the m–block ADMM with Jacobian update and , it updates each block of variables in the following *parallel* coordinate fashion:

$$x_i^{k+1} = \arg\min_{x_i} \mathcal{L}_\rho(x_i, \{x_j^k\}_{j \neq i}; y^k) + \frac{1}{2}\|x_i - x_i^k\|_{Q_i}^2. \tag{13}$$

Here $\frac{1}{2}\|x_i - x_i^k\|_{Q_i}^2$ represents a proximal term for each x_i-update to improve the convergence, $Q_i \succeq 0$ is a symmetric and positive semi-definite matrix, and we let $\|x_i\|_{Q_i}^2 = x_i^\intercal Q_i x_i$. A commonly adopted setting is $Q_i = \tau_i \mathbf{I}$ ($\tau_i > 0$). The involvement of the proximal term can make the subproblem of x_i strictly or strongly convex and thus make the problem more stable. Besides, a damping parameter $\gamma > 0$ is introduced for the update of y: $y^{k+1} = y^k + \gamma\rho(\sum_{i=1}^m A_i x_i^{k+1} - z)$. Iterations in the proximal Jacobian ADMM can execute in a parallel fashion, making the proximal Jacobian ADMM amenable for parallel implementation. Finally, the global convergence of the proximal Jacobian ADMM in *general cases* has been proven [38], with a convergence rate of $o(1/k)$.

4.2 Transforming the WDP to ADMM form

In the original WDP (5), two different blocks of variables, d_{ij} and λ_j are constrained by an inequality, rather than an equality constraint required by the ADMM method. To address this challenge, we might introduce an additional block of slack variables to transform inequality into equality. However, the introduction of a new block of variables would no doubt increase the complexity of the algorithm. For our problem, a more appropriate approach is to replace the inequality constraint on capacity allocation with the bounds on datacenter power consumption, given the relationship between the power consumption and the computing capacity. Specifically, by jointly considering the inequality $\sum_{i\in\mathcal{S}} d_{ij} + \lambda_j \le S_j$ and the equality $e_j = \alpha_j(\sum_{i\in\mathcal{S}} d_{ij} + \lambda_j) + \beta_j$, we have $\beta_j \le e_j \le \alpha_j S_j + \beta_j$. Thus, We re-write the WDP (5) as $\mathrm{W}(\mathcal{D}) =$

$$\begin{aligned}
\max \quad & \sum_{j\in\mathcal{D}} \Big\{ V_j(e_j - \hat{e}_j) + H_j(\lambda_j) - e_j p_j \Big\} + \sum_{i\in\mathcal{S}} U_i(d_i), \quad (14)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, \forall i \in \mathcal{S},\\
& e_j = \alpha_j(\sum_{i\in\mathcal{S}} d_{ij} + \lambda_j) + \beta_j, \forall j \in \mathcal{D},\\
& d_{ij} \ge 0, \lambda_i \ge 0, \beta_j \le e_j \le \alpha_j S_j + \beta_j, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}.
\end{aligned}$$

Is ADMM directly applicable? However, directly applying ADMM to problem (14) will lead to a centralized algorithm with high complexity, since the workload utility $\sum_{i\in\mathcal{S}} U(d_i)$ couples d_{ij}'s across j, while the penalty term $\sum_{j\in\mathcal{D}} \big(\alpha_j(\sum_{i\in\mathcal{S}} d_{ij} + \lambda_j) + \beta_j - e_j\big)^2$ couples d_{ij}'s across i. Thus, the workload utility ought to be separated from the penalty term if we pursue a distributed algorithm.

To this end, we continue to introduce a set of auxiliary variables $a_{ij} = d_{ij}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}$, and reformulate problem (14) as $\mathrm{W}(\mathcal{D}) =$

$$\begin{aligned}
\max \quad & \sum_{j\in\mathcal{D}} \Big\{ V_j(e_j - \hat{e}_j) + H_j(\lambda_j) - e_j p_j \Big\} + \sum_{i\in\mathcal{S}} U_i(d_i), \quad (15)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, \forall i \in \mathcal{S},\\
& a_{ij} = d_{ij}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}, \quad (16)\\
& e_j = \alpha_j(\sum_{i\in\mathcal{S}} a_{ij} + \lambda_j) + \beta_j, \forall j \in \mathcal{D}, \quad (17)\\
& d_{ij}, a_{ij}, \lambda_i \ge 0, \beta_j \le e_j \le \alpha_j S_j + \beta_j, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}.
\end{aligned}$$

Insight: Problem (15) is equivalent to problem (14), where d_{ij} controls the workload utility only with the load balance constraint (6), while a_{ij} ensures that constraint (17) is enforced when the coupling happens across the front-end server i. This is the key idea that enables both the d and a minimization to be decomposable, as we will demonstrate in the next section.

The augmented Lagrangian $\mathcal{L}_\rho$ of problem (15) can be readily obtained from (11) as follows:

$$\begin{aligned}
\mathcal{L}_\rho \;=\; & \sum_{j\in\mathcal{D}} \Big\{ e_j p_j - V_j(e_j - \hat{e}_j) - H_j(\lambda_j) \Big\} - \sum_{i\in\mathcal{S}} U_i(d_i)\\
& + \sum_{j\in\mathcal{D}} \phi_j \Big(\alpha_j\big(\sum_{i\in\mathcal{S}} a_{ij} + \lambda_j\big) + \beta_j - e_j \Big)\\
& + \frac{\rho}{2} \sum_{j\in\mathcal{D}} \Big(\alpha_j\big(\sum_{i\in\mathcal{S}} a_{ij} + \lambda_j\big) + \beta_j - e_j \Big)^2\\
& + \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{D}} \varphi_{ij}(a_{ij} - d_{ij}) + \frac{\rho}{2} \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{D}} (a_{ij} - d_{ij})^2,
\end{aligned}$$

where ϕ_j is the dual variable for constraint (17), φ_{ij} is the dual variable for the new constraint (16).

4.3 Distributed Winner Determination

For our transformed winner determination problem (15), the commonly adopted revenue functions $U_i(d_i)$ and $H_j(\lambda_j)$ (in the form of (3) and (4), respectively) are not strongly concave. Thus, for computation efficiency and provable convergence, we adopt the proximal Jacobian ADMM to facilitate the distributed winner determination.

Now we show how the update of each block of variables can be performed in a distributed fashion. Here we take the d-update as an example: by omitting the irrelevant terms in $\mathcal{L}_\rho$ and setting the matrices $Q = \tau\mathbf{I}$ ($\tau > 0$) for each block, at each iteration, the d-update step involves solving the following problem according to the iteration process (13) of the proximal Jacobian ADMM method:

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{S}} \bigg(\sum_{j\in\mathcal{D}} \Big(\frac{\rho}{2}(a_{ij}^k - d_{ij})^2 + \frac{\tau}{2}(d_{ij} - d_{ij}^k)^2 - \varphi_{ij}^k d_{ij} \Big)\\
& - U(d_i) \bigg)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, d_{ij} \ge 0, \forall i \in \mathcal{S}.
\end{aligned}$$

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 update steps for e, λ and a are decomposable over datacenters. Therefore, the winner determination problem can be efficiently computed in a *parallel* and *fully distributed* fashion.

The detailed iterative scheme for distributed winner determination is shown as follows:

Distributed Winner Determination. Initialize the variables λ, e, a, d and multipliers ϕ, φ to 0. For each iteration $k = 0, 1, \cdots$, perform the following five updates in a parallel fashion.

1. $\underline{d}$-update: Each front-end server i solves the following sub-problem for d_i^{k+1}:

$$\begin{aligned}
\min \quad & \sum_{j\in\mathcal{D}} \Big(\frac{\rho}{2}(a_{ij}^k - d_{ij})^2 + \frac{\tau}{2}(d_{ij} - d_{ij}^k)^2 - \varphi_{ij}^k d_{ij} \Big) - U(d_i),\\
\text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, d_{ij} \ge 0, \forall i \in \mathcal{S}. \quad (18)
\end{aligned}$$

2. e-update: Each datacenter j solves the following sub-problem for e_j^{k+1}:

$$\begin{aligned} \min \quad & (p_j - \phi_j^k)e_j + \frac{\rho}{2}\Big(\alpha_j\big(\sum_{i\in\mathcal{S}} a_{ij}^k + \lambda_j^k\big) + \beta_j - e_j\Big)^2 \\ & + \frac{\tau}{2}(e_j - e_j^k)^2 - V_j(e_j - \hat{e}_j), \\ \text{s.t.} \quad & \beta_j \le e_j \le \alpha_j S_j + \beta_j. \end{aligned} \tag{19}$$

This per-datacenter sub-problem (19) is over a single variable, thus, given the form of V_j, it can be readily solved. For instance, if V_j is in form of equation (1), then the above problem is a quadratic programming, and it leads to the closed-form solution: $e_j^{k+1} = \max\Big\{\min\Big\{\Psi_i, \alpha_j S_j + \beta_j\Big\}, \beta_j\Big\}$, where the constant Ψ_i is defined as $\Psi_i = \frac{\phi_j^k - p_j + \tau e_j^k + c_j \hat{e}_j + \rho\Big(\alpha_j\big(\sum_{i\in\mathcal{S}} a_{ij}^k + \lambda_j^k\big) + \beta_j\Big)}{\rho + \tau + c_j}$.

3. λ-update: Each datacenter j solves the following sub-problem for λ_j^{k+1}:

$$\begin{aligned} \min \quad & -\phi_j^k \lambda_j + \frac{\rho}{2}\Big(\alpha_j\big(\sum_{i\in\mathcal{S}} a_{ij}^k + \lambda_j\big) + \beta_j - e_j^k\Big)^2 \\ & + \frac{\tau}{2}(\lambda_j - \lambda_j^k)^2 - H_j(\lambda_j), \\ \text{s.t.} \quad & \lambda_j \ge 0. \end{aligned} \tag{20}$$

This per-datacenter sub-problem (20) is over a single variable too, and can be readily solved by following the technique used in the e-update step.

4. a-update: Each datacenter j solves the following sub-problem for $a_j^{k+1} = (a_{1j}^{k+1}, \cdots, a_{Mj}^{k+1})^\mathsf{T}$:

$$\begin{aligned} \min \quad & \frac{\rho}{2}\Big(\alpha_j\big(\sum_{i\in\mathcal{S}} a_{ij} + \lambda_j^k\big) + \beta_j - e_j^k\Big)^2 + \sum_{i\in\mathcal{S}} a_{ij}(\alpha_j\phi_j^k + \varphi_{ij}^k) \\ & + \sum_{i\in\mathcal{S}} \Big(\frac{\rho}{2}(a_{ij} - d_{ij}^k)^2 + \frac{\tau}{2}(a_{ij} - a_{ij}^k)^2\Big), \\ \text{s.t.} \quad & a_{ij} \ge 0. \end{aligned} \tag{21}$$

5. Dual update: Each datacenter j updates ϕ_j for the constraint $e_j = \alpha_j(\sum_{i\in\mathcal{S}} a_{ij} + \lambda_j) + \beta_j$ with $\phi_j^{k+1} = \phi_j^k + \gamma\rho\Big(\alpha_j\big(\sum_{i\in\mathcal{S}} a_{ij}^k + \lambda_j^k\big) - e_j^k\Big)$. Each front-end server i updates φ_{ij} for the equality constraint $d_{ij} = a_{ij}$ with $\varphi_{ij}^{k+1} = \varphi_{ij}^k + \gamma\rho(a_{ij}^k - d_{ij}^k)$.

Implementation issues: The computation of the distributed ADMM algorithm can be undertaken by each local facility. However, such a fully localized implementation would make the broadcast operations at each iteration travel across the WAN that interconnects the front-end servers and datacenters, incurring heavy usage of expensive WAN bandwidth and prolonged communication time. To address this issue, we can push computation into a designated datacenter that has abundant server resources, and split those subproblems to the numerous servers, on which the subproblems can be solved in a parallel manner.

5 Payment Design

In the previous section, we have computed the winning bid of each smart grid to maximize the social welfare in a fully distributed manner. When implementing an auction, the goal is naturally two-fold: besides pursuing economic efficiency by maximizing the social welfare, truthfulness in terms of determining the payment of each winning bid to elicit truthful bids from each smart grid is equally important. Since only when the truthfulness is guaranteed, the dominant strategy of each economically-motivated selfish bidder is to report her truthful valuation to the auctioneer, and thus the true rather than a fake social welfare is maximized. We design the payment scheme for each smart grid by leveraging the celebrated VCG mechanism. Though our setting differs from conventional auctions, we rigorously prove that the VCG mechanism still preserves nice properties including truthfulness, economic efficiency and individual rationality.

5.1 The VCG Payment Mechanism

The VCG mechanism is a well known type of auction 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. 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.

An alternative to VCG auctions, core-selecting auctions [39], have gained increasing attention in the literature recently. However, it is based on the hypothesis that seller utility is only composed of all the bidders' payments, which is just not our case as highlighted in Sec. 3.4. 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 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 the auction. Specifically, the VCG payment of each smart grid corresponding to datacenter $m \in \mathcal{D}$ for its winning bid is:

$$r_m = \mathsf{W}(\mathcal{D}\backslash\{m\}) - (\mathsf{W}(\mathcal{D}) - V_m(e_m^* - \hat{e}_m)), \tag{22}$$

here $\mathsf{W}(\mathcal{D}) - V_m(e_m^* - \hat{e}_m))$ computes the social welfare of the geo-distributed cloud and all the other smart grids when the smart grid corresponding to datacenter $m \in \mathcal{D}$ joins the auction, while $\mathsf{W}(\mathcal{D}\backslash\{m\})$ is the same aggregated social welfare when the smart grid corresponding to datacenter $m \in \mathcal{D}$ is absent from the auction, which can be formulated as $\mathsf{W}(\mathcal{D}\backslash\{m\}) =$

$$\begin{aligned} \max \quad & \sum_{j\in\mathcal{D}, j\ne m} V_j(e_j - \hat{e}_j) + \sum_{j\in\mathcal{D}} \Big\{H_j(\lambda_j) - e_j p_j\Big\} + \sum_{i\in\mathcal{S}} U_i(d_i), \\ \text{s.t.} \quad & \text{constraints } (6)(7)(8)(9). \end{aligned} \tag{23}$$

Problem (23) can be solved by applying the distributed algorithm proposed in Sec. 4, by setting $V_m(e_m - \hat{e}_m) = 0$.

Furthermore, there are in total $N+1$ WDPs to be solved in the 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.

5.2 Economical Properties of the Payment Mechanism

When applying the VCG mechanism to classical combinatorial auctions such as spectrum auction and cloud auction, the aforementioned economic properties can be guaranteed. However, the demand response auction in our paper has two distinguish features with both conventional combinatorial auction and divisible auction in (1) the bidding function submitted by each smart grid maybe non-monotone, and (2) the auctioneer's utility comprises of not only payments from the bidders, but also the varying revenue from serving both interactive and batch workloads, and energy cost that depends on the allocation of the datacenter power consumption. Thus, it is unclear whether those economic properties still hold in the presence of these two differences. These are examined in the following theorems.

Theorem 1: *The proposed auction mechanism is truthful, i.e., a smart grid can not increase its utility by misreporting its private valuation function, whatever other smart grids report.*

Proof: Suppose that the submitted bidding functions of the smart grids are $(B_1,\cdot\cdot\cdot,B_m,\cdot\cdot\cdot,B_N)$. To prove that the VCG payment is truthful for our auction, we need to prove that for any smart grid correspond to datacenter $m\in\mathcal{D}$, it utility $u_m(e_m)$ is maximized when $B_m=V_m$. When applying the VCG mechanism to compute the payment of the smart grid corresponding to datacenter m, its bidding function is not included in problem (23). Thus, the value of the term $\mathrm{W}'(\mathcal{D}\backslash\{m\}) = \max\Big\{\sum_{i\in\mathcal{S}}U_i(d_i) + \sum_{j\in\mathcal{D},j\neq m}B_j(e_j-\hat{e}_j)+\sum_{j\in\mathcal{D}}\{H_j(\lambda_j)-e_jp_j\}\Big\}$ is independent of the submitted valuation function of the smart grid corresponding to datacenter m.

Based on the utility given by (2) and the payment given by (22), for each smart grid corresponding to datacenter m, its utility can be written as

$$\begin{aligned} u_m(e) \quad &= \quad V_m(e_m-\hat{e}_m) - \Big\{\mathrm{W}'(\mathcal{D}\backslash\{m\}) - \Big\{\sum_{i\in\mathcal{S}}U_i(d_i) \\ &\quad + \sum_{j\in\mathcal{D},j\neq m}B_j(e_j-\hat{e}_j)+\sum_{j\in\mathcal{D}}\{H_j(\lambda_j)-e_jp_j\}\Big\}\Big\} \\ &= \quad V_m(e_m-\hat{e}_m) + \Big\{\sum_{i\in\mathcal{S}}U_i(d_i) + \sum_{j\in\mathcal{D},j\neq m}B_j(e_j-\hat{e}_j) \\ &\quad + \sum_{j\in\mathcal{D}}\{H_j(\lambda_j)-e_jp_j\}\Big\} - \mathrm{W}'(\mathcal{D}\backslash\{m\}). \qquad (24) \end{aligned}$$

We use $e^*=(e_1^*,\cdot\cdot\cdot,e_m^*,\cdot\cdot\cdot,e_N^*)$ to denote the outcome of the auction with the submitted bidding function $(B_1,\cdot\cdot\cdot,B_m,\cdot\cdot\cdot,B_N)$. Since the outcome out the auction always maximize the social welfare under the submitted bidding function, thus e^* maximize the term:

$$\begin{aligned} &B_m(e_m^*-\hat{e}_m)+\Big\{\sum_{i\in\mathcal{S}}U_i(d_i^*)+\sum_{j\in\mathcal{D},j\neq m}B_j(e_j^*-\hat{e}_j) \\ &+\sum_{j\in\mathcal{D}}\{H_j(\lambda_j^*)-e_jp_j\}\Big\}. \qquad (25) \end{aligned}$$

Note that when $B_m=V_m$, *i.e.*, the smart grid corresponding to datacenter m reports its true valuation function, then the right-hand-side of (24) is maximized by $e^*|_{B_m=V_m}$ since we have shown that the last term of $\mathrm{W}'(\mathcal{D}\backslash\{m\}$ is independent of B_m, and thus $e^*|_{B_m=V_m}$ exactly maximize $u_m(e)$. While if $B_m\neq V_m$, the outcome of the auction, $e^*|_{B_m\neq V_m}$ is the optimum of maximizing the term (25), but not necessarily the optimum of maximizing $u_m(e)$. To conclude, the utility of each smart grid is maximized when it reports the true valuation function to the CSP. □

Theorem 2: *The proposed auction mechanism satisfies the property of individual rationality, i.e., for each smart grid, it has a non-negative utility.*

Proof: Based on the proved truthfulness and the utility given by (2) and the payment given by (22), for each smart grid corresponding to datacenter m and wins the bid e_m^*, its utility can be written as

$$u_m(e_m^*) = \mathrm{W}(\mathcal{D}) - \mathrm{W}(\mathcal{D}\backslash\{m\}), \qquad (26)$$

Note that the WDPs (5) and (23) have the same constraints, while the objective function of problem (5) has an additional *non-negative* term $V_m(e_m-\hat{e}_m)$ not in the objective function of problem (23). Therefore, the maximum of problem (5) is no smaller than that of problem (23), *i.e.*, $\mathrm{W}(\mathcal{D}) \geq \mathrm{W}(\mathcal{D}\backslash\{m\})$, substituting this into the equation (26), we have $u_m(e_m^*)\geq 0$. This completes the proof, and it also implies that the joining of an additional smart grid would not harm the social welfare. □

Theorem 3: *The proposed auction mechanism grantees that, for the cloud, it gets a non-negative payment from each smart grid.*

Proof: Since the WDPs (5) and (23) have the same constraints, the optimal solution for problem (5), *i.e.*, $e^*=(e_1^*,e_2^*,\cdot\cdot\cdot,e_N^*)$ is also a *feasible solution* of the problem (23). Furthermore, the value of the second term in equation (22), *i.e.*, $\mathrm{W}(\mathcal{D})-V_m(e_m^*-\hat{e}_m)$ can be viewed as a feasible solution of the objective function of problem (23) under the feasible set $\mathbf{e}^*=(e_1^*,e_2^*,\cdots,e_N^*)$. While the first term in equation (22), *i.e.*, $\mathrm{W}(\mathcal{D}\backslash\{m\})$ is the optimal solution of problem (23), since the optimal solution is always no smaller than a feasible solution, we have $\mathrm{W}(\mathcal{D}\backslash\{m\}) \geq \mathrm{W}(\mathcal{D}) - V_m(e_m^*-\hat{e}_m)$, substituting this into the equation (22), we further have $r_m\geq 0$. This completes the proof. □

6 Performance Evaluation

In this section, we conduct trace-driven simulations to evaluate the practical economic benefits of the proposed incentive mechanism.

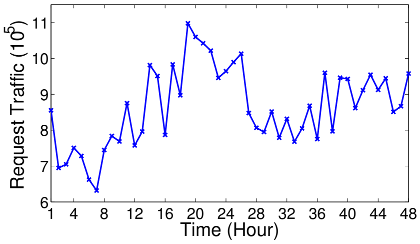


Fig. 2: Normalized CPU usage trace from a Google cluster.

6.1 Simulation Setup

Geo-distributed cloud: we consider Google's six datacenters in the USA as a representative geo-distributed cloud.

Location of each datacenter and the corresponding electricity price are listed in Table 3. Following a recent report on the number of servers owned by Google, each datacenter's capacity is set to 2×10^5 processing servers. We use the 2-day hourly CPU usage extracted from Google cluster-usage data [40] to generate the interactive workload, by proportionally scaling the CPU usage to the number of servers required, as shown in Fig. 2. To imitate the geographical distribution of requests, we split this total workload among the $M = 10$ front-end servers that are uniformly distributed across the continental U.S., following a normal distribution [11]. Each round-trip time L_{ij} is calculated according to the aforementioned empirical approximation $L_{ij} = t_{ij} \times 0.02\text{ms/Km}$, where the geographical distance t_{ij} is obtained from Google Maps. For power consumption of the servers in each datacenter, we choose a state-of-the-art setting where each server has a peak power $P_{\text{peak}} = 250W$, and consumes $P_{\text{idle}} = 125W$ when idling. We set a higher energy efficiency $\text{PUE}_j = 1.2$ for the six datacenters, which is consistent with industrial leading datacenters' energy efficiency. We use the 2011 annual average day-ahead on peak prices at the corresponding local markets as the power prices p_j for the 6 datacenter locations [11], as shown in Table 3. We take the revenue functions in the form of (3) and (4), with $q_i = 4\times10^{-6}$ and $\theta_j = 4.4\times10^{-3}$ to make the revenues from both interactive and batch workloads close to that of the electricity cost, which represents an impartial consideration on the impacts among the three aspects.

TABLE 3: The electricity price ($USD/MWh) at different datacenter locations.

Council Bluffs, IA	42.73	Berkeley County, SC	44.44
The Dalles, OR	32.57	Lenoir, NC	40.68
Mayes County, OK	36.41	Douglas County, GA	39.97

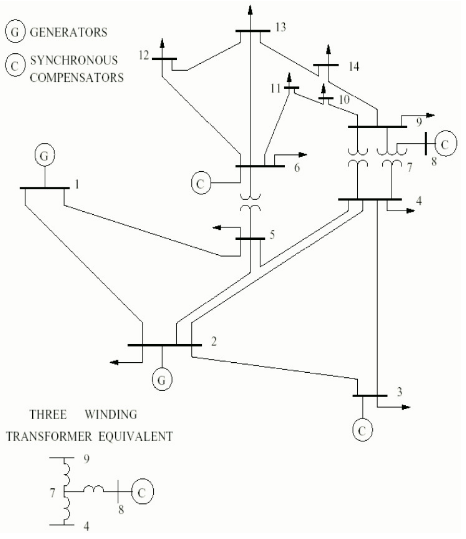

G GENERATORS
C SYNCHRONOUS COMPENSATORS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
G
G
C
C
C
THREE WINDING TRANSFORMER EQUIVALENT
9
7
8
4
C

Fig. 3: Topology of the IEEE 14-bus test system.

Smart grids: we use the IEEE 14-bus test system [41] illustrated in Fig. 3 to represent a smart grid that serves a datacenter. The arrows represent various power loads such as datacenter power demand, the synchronous condensers at buses 3, 6 and 8 can be replaced by power loads or renewable generators, and two other generators are connected to buses 1 and 2. To distinguish among the six smart grids, we place the datacenters at different buses of the test system, and use different configurations of renewable generations and power demands. The demand profiles and renewable generations are taken form the SCE load profile [3] and the NREL datasets [42], respectively. The desired levels of datacenter power consumption that minimizes the voltage violation frequency, $\hat{e}_j$, are computed by using MatPower [3]. We choose $b_j = 3000$, and set $c_j = \frac{2b_j}{(e_j-60)^2}$ for $\hat{e}_j \leq 45$, $c_j = \frac{2b_j}{(e_j-30)^2}$ for $\hat{e}_j \geq 45$ to make the valuation function V_j close the electricity bill of each datacenter.

6.2 Performance

For comparison, we further implement and evaluate other three schemes: (1) The No-In scheme in which no incentive is provided to the datacenter demand response. (2) The combinatorial auction proposed in the preliminary work [9], where each smart grid submits 10 and 20 discrete feasible bids, respectively. (3) The game-theory-based dynamic power pricing sheme (denoted as DPP) proposed in [8]. Note that DPP assumes that the utility maximization problem of the cloud admits a closed-form solution which can be derived theoretically. For comparision with DPP, here we consider a restricted case in which the cloud does not consider of utility of the interactive workload (i.e., $q_i = 0$).

Social welfare. Fig. 4 depicts the social welfare under different schemes over 48 hours. We have the following observations: (1) Compared to the no incentive scheme and dynamic power pricing, demand response auction significantly improves the social welfare. Specifically, the time-averaged hourly improvement over the two schemes is 8.7% and 6.3%, respectively. (2) The performance of the combinatorial auction is very close to that of the divisible auction, as the time-averaged hourly improvement of the divisible auction over the 10 bids auction and 20 bids auction is 0.8% and 1.1%, respectively. This demonstrates that the proposed combinatorial auction can achieve near-optimal economic efficiency. (3) The social welfare improvement promoted by the demand response auction is relatively small when the interactive workload bursts, as shown by statistics from hour 19 to hour 22. The reason 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. While the performance improvement from our proposed auctions is not so significant when compared to the DPP approach, we should note that our scheme is more practical than the DPP approach. Specifically, DPP assumes that the smart grids know the exact utility function of the CSP, which is not always practical. In contrast, in our auction, the private valuation of each smart grid is not given as a priori but extracted through a truthful payment mechanism, making it amenable to practical implementation.

Further discussions of the performance improvement. While our auction scheme realizes 8.7% improvement over the scenario with no incentives, we believe it still represents a fairly appealing result, due to the following reasons: (1) we optimize the social welfare which includes four terms: cost of demand deficit, dis-utility of interactive workload, revenue from batch workload and the electricity cost of datacenters, rather than the single term of cost of demand deficit. In the simulation, for fair comparison without bias,

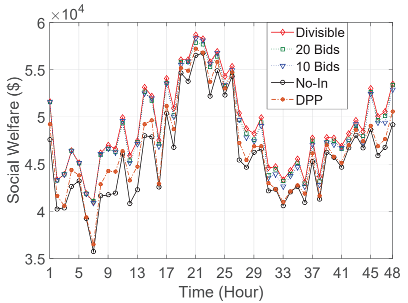


Fig. 4: Social welfare under different schemes.

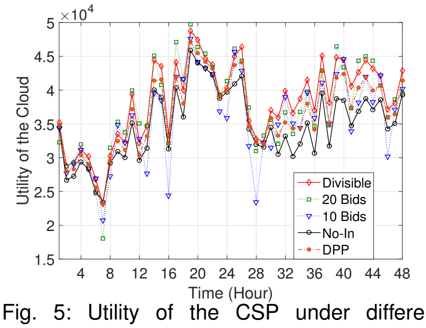


Fig. 5: Utility of the CSP under different schemes.

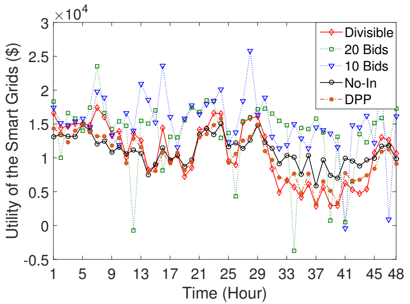


Fig. 6: Aggregated utility of the smart grids.

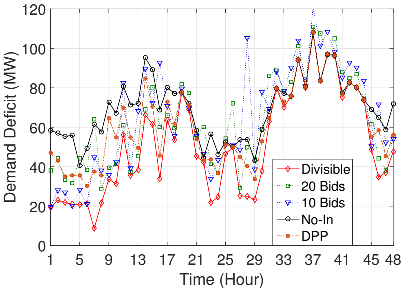


Fig. 7: Aggregated power demand deficit.

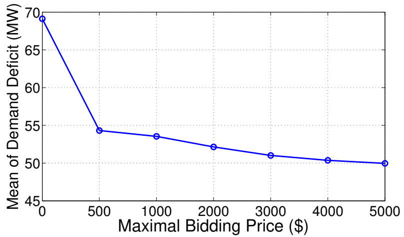


Fig. 8: Mean of demand deficit under different maximal bidding price.

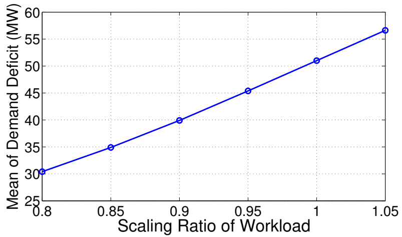


Fig. 9: Demand deficit *vs.* scaling ratio of interactive workload.

we set the cost parameters to make the above four terms are close. Therefore, even though our auction scheme significantly reduces the cost of demand deficit (as illustrated in Fig. 7), the social welfare (i.e., the aggregated cost of the four terms) may not necessarily diminish dramatically. (2) In practice, voltage violation is not the norm, instead, it happens only when the demand and supply of electricity are seriously imbalanced. Since such severe imbalance does not frequently occur, the remarkable cost reduction obtained at such imbalance time would be diluted by that at balance time, leading to a small average improvement over a long-term.

Utility of the CSP. Fig. 5 compares the utility of the geo-distributed cloud under different schemes over 48 hours. We observe that: (1) compared to the no incentive and dynamic power pricing schemes, the divisible demand response auction can always improve the utility of the CSP. (2) The performance of the combinatorial auctions is instable. This is because the optimization algorithm proposed in the preliminary work [9] is an approximated and randomized algorithm that does not search all the solution space, thus the performance of the algorithm may fluctuates in practice. (3) The utility of the CSP under combinatorial auctions may outperform that under divisible auction. This is due to the fact that our solution focus on the maximization of social welfare, thus it may sacrifice the auctioneer's utility. Hence, it is reasonable that the CSP's utility under 20 bids may larger than that under divisible auction. However, this does not mean that our proposed mechanism is not efficient, since the later evaluation will show that our solution indeed improves the stability of the smart grids.

Utility of smart grids. Fig. 6 plots the total utility of the smart grids under different schemes over 48 hours. It can be seen the total utility of smart grids could not always be improved by the divisible auction. We should note that 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. In addition, the utility of the smart grids under combinatorial auctions may outperform that under divisible auction. The explanation is that, in our previously combinatorial auction, the utility of each smart grid is denoted by a discrete function, while the utility of the CSP is denoted by a continuous function. When exploring the solution space, the adopted Gibbs sampling method focuses on tunning the discrete terms, thereby giving more emphasis on improving the utility of the smart grids. In contrast, in the divisible auction, since the utility functions of both the smart grids and the CSP are continuous, they are equally treated.

Stability of smart grids. We further demonstrate the efficiency of the proposed auction mechanism in improving the smart grids' stability in a more straightforward manner. Specifically, we define the aggregated demand deficit (quantified by $\sum_{j\in\mathcal{D}}|e_j-\hat{e}_j|$) as the gap between the datacenter's actual power consumption $(e_1, e_2, \cdots, e_N)$ and the most stable power consumption profile $(\hat{e}_1, \hat{e}_2, \cdots, \hat{e}_N)$ to capture the stability of the smart grids. Fig. 7 shows that the stability of the smart grids can be largely improved when the divisible demand response auction is introduced. In particular, the aggregated demand deficit can be very small when the interactive workload is off-peak (*e.g.*, hour 7), this is because the datacenter can provide more flexibility on power consumption, via leaving more capacity to the elastic batch workload when the amount of interactive workload is low. We also observe that from hour 30 to hour 44, the difference between the divisible auction and no auction becomes slighter. This is due to the fact that, during that interval, $\hat{e}_j$ reduces greatly for each datacenter, meaning that each datacenter is expected to use less power. However, to serve to interactive workload which has fixed amount of power demand, each datacenter becomes less responsive to the smart grid. As a result, the demand deficit keeps almost unchanged.

Influence of the maximal bidding price. We also investigate the role of the maximal bidding price, which was set to \$3000 in the valuation function. In this simulation, we let the maximal bidding price vary, and plot the mean of the aggregated demand deficit over 48 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 moderately 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 48 hours under various interactive workload scaling ratios (*i.e.,* how much we scale the amount of request traffic presented in Fig. 2) 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.

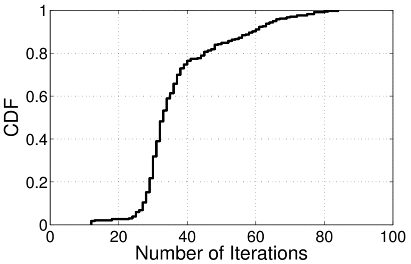


Fig. 10: CDF of the number of iterations to achieve convergence.

Convergence of the distributed algorithm. We now examine the convergence of our distributed proximal Jocabian ADMM-based algorithm. Fig. 10 plots the CDF of the number of iterations that our algorithm takes to achieve convergence for the 336 (7×48) runs. It suggests that our algorithm is able to converge within 45 iterations for 80% of the total runs. Furthermore, the fastest run uses only 12 iterations, and our algorithm takes at most 84 iterations to converge. These demonstrate the fast convergence of our distributed algorithm.

Running time. Since we do not have enough server resource to experiment with a parallel implementation, we evaluate the proposed distributed algorithm on an Intel Xeon E5-2670 server with 8-core CPU (2.6G) and 8GB DDR3 memory. By solving the per-front-end server and per-datacenter sub-problems with Matlab2014R in a sequential manner, we observe that one iteration takes 0.13 second on average. Considering that there are 22 sub-problems for one iteration (10 for d-update, 6 for λ-update and 6 for a update), one iteration takes about 0.0062 second when the algorithm is implemented in a fully parallel manner. As the scale of the geo-distributed cloud is close to the state-of-the-art level, we further scale the number of the front-end servers to examine the scalability of our solution. Specifically, when scaling the number of front-end servers from 10 to 20, 30 and 40, the running time (under fully parallel implementation) for one iteration increases from 0.0062 second to 0.0069, 0.0078 and 0.0092 second, respectively. Clearly, the running time only slightly increases as the number of front-end server scales, since the scale of each per-front-end server sub-problem remains unchanged.

7 Concluding Remarks

This work studied truthful and efficient auction mechanism design for demand response from a geo-distributed cloud. Relying on existing approaches for the mechanism design in such a market is impractical, since the demand response auction is substantially different from conventional auctions in two significant aspects. To address this challenge, we first propose a distributed social welfare maximization algorithm, by incorporating techniques from the alternating direction method of multipliers. The payment mechanism is then designed based on the classic VCG mechanism. Nice properties of the proposed mechanism, such as social efficiency, individual rationality and improvement on grid stability, are verified via rigorous theoretical analysis and/or extensive evaluations based on real datacenter workload traces and IEEE 14-bus test systems. For future work, we hope to facilitate the potential extensions discussed in Sec. 3.5.

References

[1] W. Deng, F. Liu, H. Jin, B. Li, and D. Li, "Harnessing Renewable Energy in Cloud Datacenters: Opportunities and Challenges," *IEEE Network Magazine*, vol. 28, no. 1, pp. 48–55, 2014.

[2] German Power Prices Negative over Weekend. [Online]. Available: http://energytransition.de/2014/05/german-power-prices-negative-over-weekend

[3] Z. Liu, I. Liu, S. Low, and A. Wierman, "Pricing Data Center Demand Response," in *Proc. of ACM SIGMETRICS*, 2014.

[4] Green carrots: utility incentive programs and the it industry. [Online]. Available: http://www.thegreengrid.org/~/media/WhitePapers/WP57GreenCarrotspreview.pdf?lang=en

[5] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad, "Opportunities and Challenges for Data Center Demand Response," in *Proc. of IGCC*, 2014.

[6] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad, "Exploring Smart Grid and Data Center Interactions for Electric Power Load Balancings," in *Proc. of ACM GreenMetrics*, 2013.

[7] Z. Zhou, F. Liu, and Z. Li, "Bilateral electricity trade between smart grids and green datacenters: Pricing models and performance evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3993–4007, 2016.

[8] N. Tran, D. Tran, S. Ren, Z. Han, and C. Hong, "How geo-distributed data centers do demand response: A game-theoretic approach," *Smart Grid, IEEE Transactions on*, preprint.

[9] Z. Zhou, F. Liu, Z. Li, and H. Jin, "When Smart Grid Meets Geo-distributed Cloud: An Auction Approach to Datacenter Demand Response," in *Proc. of IEEE INFOCOM*, 2015.

[10] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu, and H. Jin, "Carbon-aware online control of geo-distributed cloud services," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 27, no. 9, pp. 2506–2519, 2016.

[11] H. Xu, C. Feng, and B. Li, "Temperature Aware Workload Management in Geo-distributed Datacenters," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 6, pp. 1743–1753, 2015.

[12] Google's Green PPAs: What, How, and Why. [Online]. Available: www.google.com/green/pdfs/renewable-energy.pdf

[13] A. Bracale, G. Carpinelli, P. Falco, and T. Hong, "Short-term industrial load forecasting: A case study in an Italian factory," in *Proc. of IEEE ISGT-Europe*, 2017.

[14] A. Qureshi, "Power-demand routing in massive geo-distributed systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.

[15] A data center perspective on demand response. [Online]. Available: http://www.datacenterdynamics.com/content-tracks/power-cooling/a-data-center-perspective-on-demand-response/73892.fullarticle

[16] C. Chen, B. He, Y. Ye, X. Yuan, and M. A. Piette, "Demand Response Opportunities and Enabling Technologies for Data Centers: Findings from Field Studies," 2012.

[17] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in saas clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2648–2658, 2014.

[18] H. Xu and B. Li, "Reducing Electricity Demand Charge for Data Centers with Partial Execution," in *Proc. of ACM e-Energy*, 2014.

[19] C. Joe-Wong, I. Kamitsos, and S. Ha, "Interdatacenter job routing and scheduling with variable costs and deadlines," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2669–2680, 2015.

[20] J. Li, Z. Bao, and Z. Li, "Modeling demand response capability by internet data centers processing batch computing jobs," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 737–747, 2015.

[21] Q. Sun, S. Ren, C. Wu, and Z. Li, "An Online Incentive Mechanism for Emergency Demand Response in Geo-Distributed Colocation Data Centers," in *Proc. of ACM e-Energy*, 2016.

[22] W. Shi, C. Wu, and Z. Li, "An online auction mechanism for dynamic virtual cluster provisioning in geo-distributed clouds," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 28, no. 3, pp. 677–688, 2017.

[23] Z. Li, B. Li, and Y. Zhu, "Designing truthful spectrum auctions for multi-hop secondary networks," *Mobile Computing, IEEE Transactions on*, to appear.

[24] R. Maheswaran and T. Basar, "Efficient Signal Proportional Allocation (ESPA) Mechanisms: Decentralized Social Welfare Maximization for Divisible Resources," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 5, pp. 1000–1009, 2006.

[25] H. Xu and B. Li, "Joint request mapping and response routing for geo-distributed cloud services," in *Proc. of IEEE INFOCOM*, 2013, pp. 854–862.

[26] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, J. Padhye, and G. Varghese, "Global analytics in the face of bandwidth and regulatory constraints." in *Proc. of Usenix NSDI*, 2015.

[27] Powering down servers is a calculated risk. [Online]. Available: https://www.infoworld.com/article/2640563/green-it/powering-down-servers-is-a-calculated-risk.html

[28] Power, Pollution and the Internet. [Online]. Available: https://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html

[29] I. Kamitsos, S. Ha, L. L. Andrew, J. Bawa, D. Butnariu, H. Kim, and M. Chiang, "Optimal sleeping: models and experiments for energy-delay tradeoff," *International Journal of Systems Science: Operations & Logistics*, vol. 4, no. 4, pp. 356–371, 2017.

[30] K. Ma, G. Hu, and C. J. Spanos, "A cooperative demand response scheme using punishment mechanism and application to industrial refrigerated warehouses," *Industrial Informatics, IEEE Transactions on*, vol. 11, no. 6, pp. 1520–1531, 2015.

[31] A. Singla, B. Chandrasekaran, P. Godfrey, and B. Maggs, "The Internet at the Speed of Light," in *Proc. of ACM Hotnets*, 2014.

[32] G. Ananthanarayanan, M. C.-C. Hung, X. Ren, I. Stoica, A. Wierman, and M. Yu, "Grass: Trimming stragglers in approximation analytics," 2014.

[33] S. Teerapittayanon, B. McDanel, and H. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*, 2016, pp. 2464–2469.

[34] Y. He, S. Elnikety, J. Larus, and C. Yan, "Zeta:Scheduling interactive services with partial execution," in *Proc. of ACM SoCC*, 2012.

[35] S. Parsons, M. Marcinkiewicz, J. Niu, and S. Phelps, "Everything you wanted to know about double auctions, but were afraid to (bid or) ask," *City University of New York: New York2005*, 2006.

[36] S. Yang and B. Hajek, "VCG-Kelly Mechanisms for Allocation of Divisible Goods: Adapting VCG Mechanisms to One-Dimensional Signals," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 1237–1243, 2007.

[37] G. Ghatikar, V. Ganti, and N. Matson, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, 2014.

[38] W. Deng, M. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o(1/k) convergence," *Journal of Scientific Computing*, vol. 71, pp. 712–736, 2017.

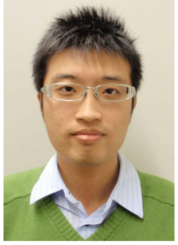
[39] Y. Zhu, B. Li, and Z. Li, "Core-selecting combinatorial auction design for secondary spectrum markets," in *Proc. of IEEE INFOCOM*, 2013.

[40] Google Cluster Data. https://code.google.com/p/googleclusterdata/.

[41] Power Systems Test Case Archive. http://www.ee.washington.edu/research/pstca/.

[42] NREL Data and Resources. http://www.nrel.gov/electricity/transmission/data resources.html.

Zhi Zhou (M'17) received the B.S., M.E. and Ph.D. degrees in 2012, 2014 and 2017, respectively, all from the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), Wuhan, China. He is currently a research associate fellow in School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. In 2016, he has been a Visiting Scholar at University of Gottingen. His research interests include cloud computing, edge computing and distributed systems.



Fangming Liu (S'08-M'11-SM'16) received the B.Eng. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2005, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2011. He was a StarTrack Visiting Faculty with Microsoft Research Asia from 2012 to 2013, and a Visiting Scholar with the University of Toronto from 2009 to 2010. He is currently a Full Professor with the Huazhong University of Science and Technology, Wuhan, China. His research interests include cloud computing and datacenter, edge computing and mobile cloud, green computing, SDN/NFV and virtualization. He received the National Natural Science Fund (NSFC) for Excellent Young Scholars, and the National Program Special Support for Top-Notch Young Professionals.

Shutong Chen received her B.Sc. degree in the College of Mathematics and Econometrics, Hunan University, China. She is currently a Ph.D. student in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. Her research interests include green cloud computing, datacenter energy management and edge computing.

Zongpeng Li received his B.E. degree in CS from Tsinghua University in 1999, his M.S. degree in CS from University of Toronto in 2001, and his Ph.D. degree in ECE from University of Toronto in 2005. Since then, Zongpeng has been a faculty member at the University of Calgary and Wuhan University. His research interests are in computer networks, network coding, cloud computing, and energy networks. Zongpeng was named an Edward S. Rogers Sr. Scholar in 2004, won the Alberta Ingenuity New Faculty Award in 2007, and was nominated for the Alfred P. Sloan Research Fellow in 2007. Zongpeng coauthored papers that received Best Paper Awards at the following conferences: PAM 2008, HotPOST 2012, and ACM e-Energy 2016. Zongpeng received the Department Excellence Award from the Department of Computer Science, University of Calgary, the "Outstanding Young Computer Science Researcher Prize from the Canadian Association of Computer Science, and the Research Excellence Award (Early Career) from the Faculty of Science, University of Calgary.