Online Auctions in IaaS Clouds: Welfare and Profit Maximization With Server Costs

Xiaoxi Zhang, *Student Member, IEEE*, Zhiyi Huang, *Member, IEEE*, Chuan Wu, *Senior Member, IEEE*, Zongpeng Li, *Senior Member, IEEE*, and Francis C. M. Lau, *Senior Member, IEEE*

Abstract—Auction design has recently been studied for dynamic resource bundling and virtual machine (VM) provisioning in IaaS clouds, but is mostly restricted to one-shot or offline setting. This paper targets a more realistic case of online VM auction design, where: 1) cloud users bid for resources into the future to assemble customized VMs with desired occupation durations, possibly located in different data centers; 2) the cloud provider dynamically packs multiple types of resources on heterogeneous physical machines (servers) into the requested VMs; 3) the operational costs of servers are considered in resource allocation; and 4) both social welfare and the cloud provider's net profit are to be maximized over the system running span. We design truthful, polynomial time auctions to achieve social welfare maximization and/or the provider's profit maximization with good competitive ratios. Our mechanisms consist of two main modules: 1) an online primal-dual optimization framework for VM allocation to maximize the social welfare with server costs, and for revealing the payments through the dual variables to guarantee truthfulness and 2) a randomized reduction algorithm to convert the social welfare maximizing auctions to ones that provide a maximal expected profit for the provider, with competitive ratios comparable to those for social welfare. We adopt a new application of Fenchel duality in our primal-dual framework, which provides richer structures for convex programs than the commonly used Lagrangian duality, and our optimization framework is general and expressive enough to handle various convex server cost functions. The efficacy of the online auctions is validated through careful theoretical analysis and trace-driven simulation studies.

Index Terms—Cloud computing, auction, resource allocation, pricing, online algorithms, truthful mechanisms.

I. INTRODUCTION

AS A major model in cloud computing services, Infrastructure-as-a-Service (IaaS) clouds are proliferating in today's Internet. An IaaS cloud meets users' realtime resource demands through virtualization technologies, which pack resources (*e.g.*, CPU, RAM, disk) into virtual machines (VMs). Major IaaS providers today typically offer pre-configured VM instances of fixed types, with the number of types increasing over the years. For example, Amazon EC2 currently provides 9 categories and 39 types of VMs [1],

a substantial growth from a few years ago. Recently emerged cloud platforms start to allow customized VMs that bundle various resources at user-specified amounts [2], [3]. Although the granularity of resource provisioning keeps improving, fixed pricing is still dominant in practice, charging customers a fixed amount for each pre-configured VM [1] or each unit of resources [2], [3]. Despite the apparent simplicity, fixed-price policies inherently lack market agility and efficiency, jeopardizing the provider's profit and customers' utility. Amazon EC2's Spot Instance is a first-step attempt to apply a market-based auction mechanism on VM provisioning, but its prices have been discovered to be often not market-driven [4], which can lead to untruthful bidding [5].

Towards better market-based pricing, auctions have recently been designed for cloud resource allocation, for pre-configured VMs of limited types [6], or for customized VMs with user-specified resource bundles [7]–[9]. Most of the mechanisms have focused on the *one-shot* or *offline* setting, assuming that the bids are given all at once. Online VM auctions, where bidders come and go at wish and allocations and charging decisions are to be made on the spot, have only been investigated in very limited setups. Wang *et al.* [6] and Zhang *et al.* [10] consider only one type of VMs in their online auctions. Shi *et al.* [7] study auctions over multiple rounds, which are coupled together by an overall budget at each user, while each acquired VM is still used for a single round. The social welfare and revenue optimization in [9] is achieved based on an assumption of allowing preemption of resources reserved by a user, which is arguable at best in practice.

This work targets a more realistic and general setup in online VM auction design with the following features. (I) Each cloud user submits a bid containing multiple options, each demanding a VM assembled with a customized bundle of resources, which can start execution at any future time for any specified duration. The cloud provider accepts at most one option in each bid. Besides, the resource composition of each VM can vary over its duration, according to the projected need of user workload. (II) The cloud provider dynamically packs multiple types of resources on heterogeneous physical machines (servers) into the requested VMs. Various server cost functions under different server operational models are considered in resource allocation, which have *not* been modeled in previous cloud auctions. (III) We design auctions that maximize the social welfare on aggregate gain of the cloud provider and the users (system efficiency), and auctions that maximize the cloud provider's net profit in expectation (another realistic objective), while guaranteeing other properties including truthful bidding, individual rationality and computational efficiency over the entire system running span.

The design of online allocation algorithms in our setup is indeed challenging, when one aims to pursue social

X. Zhang, Z. Huang, C. Wu, and F. C. M. Lau are with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: xxzhang2@cs.hku.hk; zhiyi@cs.hku.hk; cwu@cs.hku.hk; fcmlau@cs.hku.hk).

Z. Li is with the Department of Computer Science, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: zongpeng@ucalgary.ca).

welfare or profit that closely approaches that in the optimal offline solution, computed using complete information in the system span. Even in the offline setting, packing multiple types of resources on heterogeneous servers into customized VMs of time-varying resource composition and different durations involves NP-hard combinatorial optimization problems. In the online algorithm, the decision on packing or reserving resources for the requested VM should be made upon receipt of each bid, without the assistance of any future information. What's more, even when an online approximate allocation algorithm is in place, it can be difficult to design a payment rule that works with the allocation algorithm to guarantee desired properties such as truthfulness. The classic VCG mechanism, essentially the only type of auction that guarantees both truthfulness and economic efficiency in the offline setting [11], does not directly work in the online case, since it requires the computation of exact optimal allocation to guarantee truthfulness, which cannot be calculated for the future requests.

The challenge further escalates when our auction model involves operational costs of servers in the computation of social welfare and profit (even in expectation). Most existing auction designs ignore such (production) costs of resources, but consider social welfare as only the overall value of accepted bids and profit as the overall payment. Significant difficulties are involved, preventing good results, when the costs of resources are deducted in calculating social welfare and profit, especially in the online setting: The allocation problem with server costs contains a mixture of packing and covering constraints—such problems are known to be more challenging than problems with only packing constraints such as the previous models without server costs [12]. Further, we seek to consider more generic server cost functions that are convex instead of linear, and there were no appropriate techniques for handling such non-linear costs until very recently (see Sec. II for details).

Our Contributions: This paper leverages a recent development in primal-dual online algorithm design and randomized reduction techniques, to design a set of truthful, polynomial-time online auctions for social welfare maximization or profit maximization in expectation with good competitive ratios. Our mechanisms consist of two main modules: (1) an online primal-dual optimization framework for VM allocation to maximize the social welfare with server costs, and for revealing the payments through the dual variables to guarantee truthfulness; and (2) a randomized reduction algorithm to convert the social welfare maximizing auctions to ones that glean a maximal expected profit for the provider, with competitive ratios comparable to those for social welfare.

First, we model the social welfare (profit) maximization problem using a primal-dual optimization framework, and adopt a new application of Fenchel duality [13] for the dual, which provides richer structures for convex programs that guide the design and analysis of online auctions, than the commonly used Lagrangian duality [14]. Our optimization framework is general and expressive enough to handle various convex server cost functions, *e.g.*, cubic, linear, or zero infinity, representing different server operation models in IaaS clouds.

Second, we design efficient primal-dual online auctions for social welfare maximization, which extend the existing online primal-dual resource allocation framework to handle departures of resource requests, such that resources released by completed VM requests can be reused by later bids.

Existing online primal-dual resource allocation algorithms (e.g., [15], [16]) do not handle resource re-use and time-varying resource demands in each request. To the best of our knowledge, the only online primal-dual algorithms that address departure of resource requests are those for online scheduling (e.g., [11]), which is structurally different from our problem and the techniques cannot be easily translated to our setting. Other highlights of the design include meticulously designed pricing functions for updating the marginal prices per unit resource, which play a key role in achieving truthfulness and good competitive ratio.

Third, we extend our social welfare maximizing auctions to profit maximizing in expectation ones using randomized reduction, with minor losses in competitive ratios. To obtain good competitive ratios in terms of the expected profit with super-linear server costs (Sec. IV), we introduce a new online primal-dual analysis for the expected profit. Previous techniques usually compare the profit of online auctions with the optimal social welfare, and do not easily generalize to our model with server costs. In contrast, we compare the the expected profit of our auctions with the dual objective of the social welfare maximization convex program. To our knowledge, using the online primal dual analysis for profit is novel in the literature and may be of further interest for other profit maximization problems.

Organization: We discuss related work in Sec. II, and define the model in Sec. III. Sec. IV presents our online auction under super-linear server cost functions. Sec. V discusses the online auctions under linear server cost models. Simulations are presented in Sec. VI. Sec. VII concludes the paper.

II. RELATED WORK

Allocating pre-configured types of VMs in an IaaS cloud has been extensively studied with different focuses. Beloglazov *et al.* [17] study energy-efficient allocation algorithms for scheduling VMs to serve computing tasks.

Auction mechanisms have been at the focal point of recent literature on VM pricing. On-demand VM allocation through a truthful auction is studied, and an achievement of a higher revenue for the cloud provider is shown. Their models do not include tailor-made VM assembling. Zhang *et al.* [8] study customized VM provisioning within one data center, and design a truthful auction by applying an LP decomposition technique, achieving a 2.72-approximation in social welfare in typical scenarios. All these studies consider only one-round auctions, neglecting the dynamical nature of users' demands.

Wang *et al.* [6] investigates an online truthful auction where bidders may occupy a VM for multiple interval. Wu *et. al.* [19] design greedy algorithms for VM allocation and pricing for online bids with departures. However, their models are over-simplified and consider one type of VM only. In the cloud market investigated by Mashayekhy *et al.* [20], heterogeneous VMs are investigated but serving delay is allowed for each bid. Shi *et al.* [7] propose auction mechanisms over multiple rounds which are coupled together by the overall budget of each user, while each acquired VM is used for only one round. We investigate a more practical setup, where each VM can be running for various durations into the future, where it is significantly more challenging to guarantee truthfulness and efficiency. Online VM allocation is also studied in [9], but a strong assumption is made on allowing preemption of resources already occupied. In addition, none of existing

cloud auctions consider server costs in the social welfare or provider's profit, which is included in our model.

The online primal dual framework (see [15] for a survey) has been used to design online algorithms and auctions for various problems, such as the ski rental problem, metrical task system problem and ad auctions. The original online primal dual framework focuses on linear programs, which does not naturally model the convex cost functions considered in this work. Recently, there have been studies on extending the online primal dual framework to problems modeled by convex programs, such as online scheduling on speed-scalable machines [11], and online combinatorial auctions with production costs [16], [21]. The former is structurally different from our problem. The latter does not handle departures of resource requests, while in practical scenarios of VM allocations, each VM only occupies the requested resources for a limited period of time, and the resources can be released and reused afterwards. This work extends the primal-dual framework to handle VM departures and resource recycling.

III. PROBLEM MODEL

A. Cloud System and the Auction

We consider an IaaS cloud system consisting of K data centers (DCs) offering users R types of resources, as exemplified by CPU, RAM, disk storage and bandwidth. We use $\mathcal{X}$ to denote the set $\{1, 2, \ldots, X\}$. Let S_k denote the set of servers in data center $k \in \mathcal{K}$. Each server $s \in \mathcal{S}_k$ can provide a maximum amount C_{rs} of resource $r \in \mathcal{R}$. Cloud users arrive over time, and each requests one tailor-made VM for workload execution, with the amount of resources needed for each VM specified. The IaaS cloud provider acts as the auctioneer and sells the VMs through an online auction.

There are N cloud users participating in the auction, arriving one by one within the whole time span $1, 2, \ldots, T$. Suppose user $n \in \mathcal{N}$ arrives at time t_n and submits a bid B_n consisting of I_n options of VMs with corresponding data center locations. In each option $i \in \mathcal{I}_n$ of B_n, user n requests a VM which comprises a bundle of resources in a future usage duration. Suppose t_{ni}^- is the start time to run the VM in option i of B_n and t_{ni}^+ is the end time, where $t_n \leq t_{ni}^- < t_{ni}^+$. Let $d_{nir}(t)$ denote the amount of type-r resource to be occupied at time t specified in option i of user n. We allow $d_{nir}(t)$ to resume different values over the customized usage duration $t \in [t_{ni}^-, t_{ni}^+]$, which enables each VM to consume a different amount of each type of resource over time. $d_{nir}(t)$'s are specified in the bid based on the projected resource need of the bidder's workload at different times. Dynamic scaling of resources occupied by a running VM is practically feasible through "hotplug" technologies that adjust CPU cycles, memory and disks allocated to a running VM, as supported in various virtualization environments including Xen, VMWare and VirtualBox [22]. Without loss of generality, we assume in the following discussions that the demands $d_{nir}(t)$ are much smaller compared to the server capacity C_{rsk}. Moreover, we allow each user n to specify a data center k_{ni} for in each option $i \in \mathcal{I}_n$, as well as a willingness-to-pay b_{ni} for option i for the respective tailor-made VM to run between t_{ni}^- and t_{ni}^+ in the specified data centre (v_{ni} be the respective true valuation), which indicates his preference of obtaining the corresponding VM in the data center. For example, a user may specify a larger bidding price for a VM allocated in a data centre that is close to him, than a VM in a remote data center.

A bid can be expressed as (*bidding language*):

$$B_n = \{t_{ni}^-, t_{ni}^+, \{d_{nir}(t)\}_{r \in \mathcal{R}, t \in [t_{ni}^-, t_{ni}^+]},$$
$$\{k_{ni}\}_{i \in \mathcal{I}_n}, \{b_{ni}\}_{i \in \mathcal{I}_n}\}. \quad (1)$$

Upon receiving each bid, the cloud provider decides whether to accept it and for which option as well as on which server in desired DC to provision the requested VM if accepted. A binary variable x_{nis} is set to 1 if option i of user n is accepted with resources allocated on server s in DC k_{ni}, and 0 otherwise. The provider also computes a payment $\hat{p}_{ni\star}$ for each winning bid n.

In practice, most cloud data centers keep their servers on, which remain in the low-power idle mode if no jobs are running, to avoid time-consuming booting up if switched completely off [23]. Therefore, we realistically assume that all $\sum_{k \in \mathcal{K}} S_k$ servers are turned on in the span $\mathcal{T}$ under our investigation. Each server consumes a basic amount of power with no VM running, and the power usage increases with the increase of resource occupied on the server. The operational cost of a server is mainly due to the power cost, following a similar increasing trend with power consumption. We use $f_{rsk}(\cdot)$ to denote the cost function of server s in DC k on the amount of type-r resource used on the server, as indicated by $y_{rsk}(t)$. The cost function is defined as follows:

$$f_{rsk}(y_{rsk}(t)) = \begin{cases} h_{rsk} y_{rsk}(t)^{1+\beta_{rsk}}, & y_{rsk}(t) \in [0, C_{rsk}] \\ +\infty, & y_{rsk}(t) > C_{rsk} \end{cases}$$
$$(2)$$

Parameter $\beta_{rsk} \geq 0$ decides the shape of the cost function, according to different operational models of the server. h_{rsk} indicates the relative weight of the cost due to each type of resource in the overall server cost. For example, Dynamic Voltage Frequency Scaling (DVFS) has been widely supported in virtualization platforms, which adjusts the frequency or voltage of a CPU on the fly (often in response to the workload) to conserve its power consumption [25]. When the CPU voltage is elevated with the utilization, the CPU power usage renders a cubic increase with the CPU voltage [24], and hence we can approximately use $\beta_{rsk} = 2$ in (2) where r denotes the CPU. If DVFS is not enabled, measurements have shown that the server power consumption increases roughly linearly with the utilization of CPU, memory, disk I/O and network I/O [25], and hence we set $\beta_{rsk} = 0$ in this case. It has also been shown that power consumption of memory, disk I/O and network I/O are significantly lower than that of the CPU, further ranked in a decreasing order among themselves [26], and the power usage due to different resources is additive [25], confirming our additive model of the costs due to different resources.

B. Mechanism Design Goals

We target the following properties in our auction design. (i) *Truthfulness*: For any bidder, declaring his true valuation of the VM and true information (*e.g.*, bid arrival time) in his bid always maximizes his utility, regardless of other users' bids. (ii) *Computational efficiency*: Polynomial-time algorithms for resource allocation and payment calculation are needed for the auction to run efficiently in an online fashion. (iii) *Individual rationality*: Each bidder obtains a non-negative utility by participating in the auction, and the cloud provider receives a non-negative net profit. (iv) *Social welfare maximization*

or provider's profit maximization: The cloud provider's profit equals the aggregate user payment minus server costs, *i.e.*,

$$\sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}_n} \hat{p}_{ni} \sum_{s \in \mathcal{S}_{k_{ni}}} x_{nis} - \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{r \in \mathcal{R}} f_{rsk}(y_{rsk}(t)). \tag{3}$$

Bid n's utility is $v_{ni} - \hat{p}_{ni}$ if accepted in option $i \in \mathcal{I}_n$, and 0 otherwise. The social welfare over system span $\mathcal{T}$ is the sum of the provider's profit and the bidders' aggregate utility, $\sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}_n} (v_{ni} - \hat{p}_{ni}) \sum_{s \in \mathcal{S}_{k_{ni}}} x_{nis}$ (valuation minus payment of all winning bids), which equals the aggregate valuation of the winning bids minus the server costs,

$$\sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}_n} v_{ni} \sum_{s \in \mathcal{S}_{k_{ni}}} x_{nis} - \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{r \in \mathcal{R}} f_{rsk}(y_{rsk}(t)),$$

and $\displaystyle \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}_n} b_{ni} \sum_{s \in \mathcal{S}_{k_{ni}}} x_{nis} - \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{r \in \mathcal{R}} f_{rsk}(y_{rsk}(t))$

under truthful bidding. A cloud system operates at the maximal efficiency if social welfare is maximized over the running span, benefiting both the cloud provider and users. An equally natural goal is to maximize the provider's profit, which we will also pursue with efficient online auction design.

The offline VM allocation and winner determination problem can be formulated as follows, supposing all the information of the N bids in total within system span $\mathcal{T}$ are known and truthful bidding is guaranteed. The objective in (4) indicates social welfare maximization, and can be easily changed to profit maximization by replacing the social welfare with the provider's profit in (3).

$$\text{maximize} \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}_n} \sum_{s \in \mathcal{S}_{k_{ni}}} b_{ni} x_{nis}$$
$$- \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{r \in \mathcal{R}} f_{rsk}(y_{rsk}(t)) \tag{4}$$

subject to:

$$\sum_{i \in \mathcal{I}_n} \sum_{s \in \mathcal{S}_{k_{ni}}} x_{nis} \leq 1, \quad \forall n \in \mathcal{N} \tag{4a}$$

$$\sum_{\substack{n \in \mathcal{N}, i \in \mathcal{I}_n: \\ s \in \mathcal{S}_{k_{ni}} t_{ni}^- \leq t \leq t_{ni}^+}} d_{nir}(t) x_{nis}$$
$$\leq y_{rsk}(t), \quad \forall r \in \mathcal{R}, \ s \in \mathcal{S}_k, \ k \in \mathcal{K}, \ t \in \mathcal{T} \tag{4b}$$

$$x_{nis} \in \{0, 1\}, \quad \forall s \in \mathcal{S}_{k_{ni}}, \ i \in \mathcal{I}_n, \ n \in \mathcal{N} \tag{4c}$$

$$y_{rsk}(t) \geq 0, \ \forall r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}, t \in \mathcal{T} \tag{4d}$$

Here, constraint (4a) indicates that at most one option is accepted for a user and the requested VM is provisioned on at most one server of the data center. (4b) sums up the amount of type-r resource needed by all accepted bids on server s of DC k at t (counting only bids whose VMs are running at t) into $y_{rsk}(t)$. Recall the definition of the cost function in (2): by setting $f_{rsk}(y_{rsk}(t)) = +\infty$ when $y_{rsk}(t)$ exceeds C_{rsk} (the overall capacity of type-r resource on server s in DC k), it is (implicitly) guaranteed that the allocation of each type of resource on any server will not go beyond the capacity limit.

The offline problem in (4) is a convex mixed integer program with a concave objective function and linear constraints. We relax the integrality constraints $x_{nis} \in \{0, 1\}$

to $x_{nis} \geq 0$ (constraint (4a) guarantees $x_{nis} \leq 1$), and apply Fenchel duality [13] to the relaxed convex program. As compared to the well-known Lagrangian duality defined generically for convex and non-convex programs, Fenchel duality is defined only for convex programs, and the derived Fenchel dual problems typically present richer structures that guide the design and analysis of primal-dual online algorithms.

Let u_n and $p_{rsk}(t)$ be the dual variables associated with (4a) and (4b), respectively. The Fenchel dual [13] of the relaxed convex program is as follows:

$$\text{minimize} \sum_{n \in \mathcal{N}} u_n + \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{r \in \mathcal{R}} f_{rsk}^\star(p_{rsk}(t)) \tag{5}$$

subject to:

$$u_n \geq b_{ni} - \sum_{t=t_{ni}^-}^{t_{ni}^+} \sum_{r \in \mathcal{R}} d_{nir}(t) p_{rsk_{ni}}(t),$$
$$\forall s \in \mathcal{S}_{k_{ni}}, \ i \in I_n, \ n \in \mathcal{N} \tag{5a}$$

$$p_{rsk}(t) \geq 0, \quad \forall r \in \mathcal{R}, \ s \in [S], \ t \in \mathcal{T} \tag{5b}$$

$$u_n \geq 0, \ \forall n \in \mathcal{N} \tag{5c}$$

where $f_{rsk}^\star(p_{rsk}(t))$ is the conjugate of the cost function $f_{rsk}(\cdot)$, defined as

$$f_{rsk}^\star(p_{rsk}(t)) = \sup_{y_{rsk}(t) \geq 0} \{ p_{rsk}(t) y_{rsk}(t) - f_{rsk}(y_{rsk}(t)) \} \tag{6}$$

Proposition 1: The conjugate of $f_{rsk}(y_{rsk}(t))$ in (2) is

$$f_{rsk}^\star(p_{rsk}(t))$$
$$= \begin{cases} (\frac{p_{rsk}(t)}{1 + \beta_{rsk}})^{\frac{1 + \beta_{rsk}}{\beta_{rs}}} \frac{\beta_{rsk}}{(h_{rsk})^{\frac{1}{\beta_{rsk}}}}, & y_{rsk}^0(t) \leq C_{rsk} \\ C_{rsk} p_{rsk}(t) - h_{rsk}(C_{rs})^{1 + \beta_{rsk}}, & y_{rsk}^0(t) > C_{rsk} \end{cases} \tag{7}$$

where $y_{rsk}^0(t) = (\frac{p_{rsk}(t)}{h_{rsk}(1 + \beta_{rsk})})^{\frac{1}{\beta_{rsk}}}$.

The offline allocation problem and its Fenchel dual are established assuming complete knowledge about the system over its entire lifespan. In practice, with the arrival of bids, the variables and constraints emerge gradually. For example, on the arrival of user n, there is a set of new primal variables x_{nis} for all $i \in \mathcal{I}_n$ and $s \in \mathcal{S}_{k_{ni}}$ subject to $\sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{k_{ni}}} x_{nis} \leq 1$. The cloud provider must decide immediately whether to reject $\mathcal{B}_n$ or accept one of the options of the bid. For the chosen option $i^\star$ of a winning bid n, the provider will select a server in the customized DC $k_{ni^\star}$ and calculate the user's payment. In the following, we design online primal-dual allocation algorithms and payment schemes based on (4) and (5).

IV. ONLINE AUCTIONS FOR SOCIAL WELFARE AND PROFIT MAXIMIZATION

In this section, we focus on the case of superlinear server cost functions, *i.e.*, $h_{rsk} > 0$ and $\beta_{rsk} > 0$ in (2), and design online auctions for social welfare maximization (Sec. IV-A) and profit maximization in expectation (Sec. IV-B). We will discuss the case of linear server cost functions (with zero server cost as a special case) in Sec. V.

A. Online Auction for Social Welfare Maximization

1) Auction Design: Deciding whether to serve a new bid set $\mathcal{B}_n$ and how to choose an option $i \in \mathcal{I}_n$ as well as on which server with the user's location requirement fulfilled is equivalent to choosing a feasible assignment for the new primal variables x_{nis} ($i \in \mathcal{I}_n, s \in \mathcal{S}_{k_{ni}}$). If the cloud provider decides to serve bid i on some server s_i in data center k_{ni}, then let $x_{nis} = 1$, and increase the amount of allocated resources $y_{rs^\star k_{ni^\star}}(t)$ ($i^\star$ is the chosen option and $s^\star$ is the assigned server) by $d_{ni^\star r}(t)$ on server $s^\star$ in data center k_{ni} for all resources $r \in \mathcal{R}$ and for all time slots $t \in [t_{ni}^-, t_{ni}^+]$. As a result, the total server cost of $s^\star$ in k_{ni}, $\sum_{t \in \mathcal{T}} \sum_{r \in \mathcal{R}} f_{rs^\star k_{ni}}(y_{rs^\star k_{ni}}(t))$, increases accordingly. Otherwise, x_{nis} will be zero for all servers $s \in \mathcal{S}_{k_{ni}}$.

VM Allocation: The question is whether and how to select the option $i^\star \in \mathcal{I}_n$ for user n, as well as how to choose on which server in the user specified data center $k_{ni^\star}$ for the chosen option $i^\star$ if accepting $\mathcal{B}_n$. For this we will look into the dual program and resort to the KKT conditions [14]. With bid set $\mathcal{B}_n$, there is a new dual variable $u_n \geq 0$ subject to constraints (5a), *i.e.*, $u_n \geq b_{ni} - \sum_{t \in [t_{ni}^-, t_{ni}^+]} \sum_{r \in \mathcal{R}} d_{nir}(t) p_{rsk}(t)$ for all $s \in \mathcal{S}_{k_{ni}}$ and all $i \in \mathcal{I}_n$. The KKT conditions indicate that in the offline primal and dual solutions to (4) and (5), x_{nis} must be zero unless constraint (5a) is tight for server $s \in \mathcal{S}_{k_{ni}}$ and $i \in \mathcal{I}_n$. Thus, we let u_n be the maximal of 0 and the right hand side (RHS) of constraints (5a), *i.e.*, $u_n =$

$$\max \left\{ 0, \max_{\substack{s \in \mathcal{S}_{k_{ni}}, \\ i \in \mathcal{I}_n}} \left\{ b_{ni} - \sum_{t \in [t_{ni}^-, t_{ni}^+]} \sum_{r \in \mathcal{R}} d_{nir}(t) p_{rsk}(t) \right\} \right\}. \quad (8)$$

Accordingly, we adopt the following method to decide whether to choose an option of a bid and on which server: the cloud provider does not accept any bid $i \in \mathcal{I}_n$, *i.e.*, $x_{nis} = 0$ for all $s \in \mathcal{S}_{k_{ni}}$ and $i \in \mathcal{I}_n$, if no server $s \in \mathcal{S}_{k_{ni}}$ for any bid $i \in \mathcal{I}_n$ achieves a non-negative value on the RHS of (5a) (*i.e.*, $u_n = 0$). Otherwise, we select a bid $i^\star \in \mathcal{I}_n$ and a server $s^\star \in \mathcal{S}_{k_{i^\star n}}$ that maximizes the RHS.

Rationale: If we interpret $p_{rsk}(t)$ as the marginal price (a.k.a. payment) per unit of type-r resource on server s in DC k at time t, then the second term on the RHS of (5a) becomes the tentative payment that bid n should pay for the requested resources in option $i \in \mathcal{I}_n$ on server $s \in \mathcal{S}_{k_{ni}}$, *i.e.*,

$$\hat{p}_{ni} = \sum_{t \in [t_{ni}^-, t_{ni}^+]} \sum_{r \in \mathcal{R}} d_{nir}(t) p_{rsk}(t). \quad (9)$$

So the RHS of (5a) for a given s and i is the utility of $\mathcal{B}_n$ if option i is accepted and would be served on server s in k_{ni} (valuation minus payment, assuming truthful bidding). Since each bid have at most one accepted option, $\hat{p}_{ni^\star}$ is the final payment of user n, denoted by $\hat{p}_{ni^\star}$. Therefore, the above method effectively selects an option $i \in \mathcal{I}_n$ and assigns it to the server s that maximizes the utility of $\mathcal{B}_n$, and u_n is user n's utility. In this way, we target utility maximization for each user, leading to truthfulness and social welfare maximization.

Payment Design: Furthermore, the cloud provider updates the marginal prices $p_{rsk}(t)$ as the amounts of allocated resources $y_{rsk}(t)$ change, before calculating tentative payment $\hat{p}_{ni}$ of each $i \in \mathcal{I}_n$ in B_n using (9), to ensure that: (i) the gain in social welfare when a bid is served outweighs the

---

Algorithm 1 Primal-Dual Online Auction PD$_1$

Input: K, S, R, $\mathbf{C}$, $\mathbf{h}$, β
Output: $\mathbf{x}$, $\mathbf{p}$
1 Define $f_{rsk}(y_{rsk}(t))$ and $p_{rsk}(y_{rsk}(t))$ according to (2) and (10);
2 Initialize $x_{nis} = 0, y_{rsk}(t) = 0, u_n = 0, p_{rsk}(t) = 0$;
3 **Upon the arrival of the nth bid**
4 Get *bidding language* B_n according to (1);
5 $(\mathbf{x}_{ni}, \hat{p}_{ni^\star}, \mathbf{p}, \mathbf{y}) = \text{CORE}(K, S, R, B_n, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
6 **if** $\exists s^\star \in \mathcal{S}_{k_{ni^\star}}, x_{ni^\star s^\star} = 1$ **then**
7   Accept bid n and allocate resources on server $s^\star$ in $k_{ni^\star}$;
8   Charge bid n at $\hat{p}_{ni^\star}$;
9 **else**
10   Reject bid n;
11 **end**

---

loss in total server cost, and (ii) the cloud does not allocate all resources to low value bids that come early at the risk of having no capacity left for high value bids in the future. Let $\tilde{y}_{rsk}(t)$ be total demand of resource r on server s in DC k at time t, and $\tilde{p}_{rs}(t)$ be the respective marginal prices in the offline primal and dual solutions. If $\tilde{y}_{rsk}(t)$ is smaller than the capacity C_{rsk}, the marginal price $\tilde{p}_{rsk}(t)$ shall be equal to the marginal cost per unit of the resource, $f'_{rsk}(\tilde{y}_{rsk}(t))$; if the demand meets its capacity, the marginal price shall serve as a threshold to filter out the low value bids, such that the demand subject to the prices is equal to the capacity.

In this online setting, however, the cloud provider can see only the current demands of resources, not the final demands. Therefore, it predicts the final demands based on the current ones and set prices accordingly. Our approach is predicting the final demand of a resource r on server s in DC k to be δ_{rsk} times its current demand if the predicted final demand does not exceed the capacity, for a fixed parameter $\delta_{rsk} > 1$ to be determined later, and set the marginal price to be $p_{rsk}(t) = f'_{rsk}(\delta_{rsk} y_{rsk}(t))$. If the above predicted final demand exceeds the capacity, the cloud provider needs to predict the final threshold price (subject to which the demand equals the capacity) and use it as the marginal price. For this we use the state-of-the-art technique in online resource allocation (e.g., [27]), and let the marginal price increase exponentially in the current demand, *i.e.*, $p_{rsk}(t) = f'_{rsk}(C_{rsk}) \exp(\theta_{rsk}(y_{rsk}(t) - C_{rsk}/\delta_{rsk}))$, where θ_{rsk} is a parameter to be determined later, if the predicted final demand exceeds the capacity. In summary, the marginal price $p_{rsk}(t)$, is defined to be a function on $y_{rsk}(t)$ as follows (for all $r \in \mathcal{R}$, $s \in \mathcal{S}_k$, $k \in \mathcal{K}$ and $t \in \mathcal{T}$):

$$p_{rsk}(y_{rsk}(t)) = \begin{cases} f'_{rsk}(\delta_{rsk} y_{rsk}(t)), & y_{rsk}(t) \leq \frac{C_{rsk}}{\delta_{rsk}} \\ f'_{rsk}(C_{rsk}) e^{\theta_{rsk}(y_{rsk}(t) - \frac{C_{rsk}}{\delta_{rsk}})}, & \text{otherwise} \end{cases} \quad (10)$$

We will show in Sec. IV-A2 that such a price function has nice properties, guaranteeing competitiveness of the online auction.

Auction Mechanism: Directed by the discussions above, we design the online auction algorithm PD$_1$, as given in Alg. 1. Upon arrival of each user n, we calculate the payment for a winning bid with the chosen option $i^\star \in \mathcal{I}_n$ by summing up the product of resource demand and current marginal

---

Algorithm 2 One-Round Auction Algorithm CORE

Input: K, S, R, B_i, $\mathbf{p}$, $\mathbf{y}$, $\mathbf{p}(\mathbf{y})$

Output: $\mathbf{x}_{ni}$, $\hat{p}_{ni}$, $\mathbf{p}$, $\mathbf{y}$, $(n, i^\star, s^\star)_{n \in \mathcal{N}}$

1 Update the dual variable u_n (utility) according to (8);

2 **if** $u_n > 0$ **then**

3    $(i^\star, s^\star) = \operatorname{argmax}_{\substack{s \in \mathcal{S}_{k_{ni}} \\ i \in \mathcal{I}_n}} \big\{ b_{ni} -$

     $\sum_{t \in [t_{ni}^-, t_{ni}^+]} \sum_{r \in \mathcal{R}} d_{nir}(t) p_{rsk}(t) \big\};$

4   **Calculate the payment of B_n if accepted in option** $i^\star$:

5    $\hat{p}_{ni^\star} = \sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} \sum_{r \in \mathcal{R}} d_{ni^\star r}(t) p_{rsk_{ni^\star}}(t).;$

6   Update the primal variables:

7    $x_{nis^\star} = 1$, and $x_{nis} = 0$ for all $(i, s) \neq (i^\star, s^\star);$

8    $y_{rs^\star k_{ni^\star}}(t) = y_{rs^\star k_{ni^\star}}(t) + d_{nir}(t)$

     $\forall r \in \mathcal{R}, t \in [t_{ni}^-, t_{ni}^+];$

9   Update the dual marginal price variables:

10    $p_{rs^\star k_{ni^\star}}(t) = p_{rs^\star k_{ni^\star}}(y_{rs^\star k_{ni^\star}}(t))$

     $\forall r \in \mathcal{R}, t \in [t_{ni}^-, t_{ni}^+];$

11 **else**

12    $x_{nis} = 0$ for all $s \in \mathcal{S}_{ni}$ and $i \in \mathcal{I}_n;$

13 **end**

---

price on the selected server $s^\star$ for all resources over the occupation period (line 5). We update primal variables by setting $x_{ni^\star s^\star}$ of the selected bid and server to 1 (line 7), and increase the utilization $y_{rs^\star k_{ni^\star}}(t)$ for different resources for future time slots $t \in [t_{ni^\star}^-, t_{ni^\star}^+]$. Here, $i^\star$ labels the accepted option for each winning bid and $s^\star$ denotes the chosen server to provision resources for the corresponding bid. They are both updated each time the algorithm accepts a bid. We update the dual marginal price variables $p_{rs^\star k_{ni^\star}}(t)$'s according to (10) (line 10). We note that recycling of resources is implicitly handled by increasing $y_{rs^\star k_{ni^\star}}(t)$'s with bid n's resource demand, only for time slots within specified duration $[t_{ni^\star}^-, t_{ni^\star}^+]$.

2) Analysis: **(i) Truthfulness, Individual Rationality, and Polynomial Time**

Theorem 1: The online auction PD_1 *in Alg. 1 is truthful and individually rational, and processes each bid in* $O(RSIT)$ *time, where* $S = \max_{k \in \mathcal{K}} S_k$ *and* $I = \max_{n \in \mathcal{N}} I_n$.

The detailed proof can be found in Appendix A.

(ii) Competitiveness in Social Welfare

We next analyze the competitive ratio of our online auction, *i.e.*, the worst-case upper-bound ratio of the social welfare achieved by the offline solution of (4) to the overall social welfare achieved by our online auction at the end of time T.

Let $P^0 = 0$ and D^0 be the initial values of the primal and dual objectives. Let P^n and D^n denote the primal and dual objective values after handling user n. $y_{rsk}^n(t)$ denotes the amount of allocated type-r resource on server s in DC k after handling user n, and $p_{rsk}^n(t)$ denotes the corresponding marginal price. Note that P^N and D^N are the primal and dual objective values at the end of time T. Let OPT be the optimal primal objective value of (4), *i.e.*, the offline optimal social welfare.

Lemma 1: If (i) there is a constant $\alpha \geq 1$ *such that the incremental increase of the primal and dual objective values differ by at most an* α *factor, i.e.,* $P^n - P^{n-1} \geq \frac{1}{\alpha}(D^n - D^{n-1})$, *for every step* n, *and (ii) the initial dual*

objective value D^0 is at most $\frac{1}{2}OPT$, then the algorithm is 2α competitive.

Proof: Summing up the inequality of each step n, we have

$$P^N = \sum_n (P^n - P^{n-1}) \geq \frac{1}{\alpha} \sum_n (D^n - D^{n-1})$$
$$= \frac{1}{\alpha}(D^N - D^0).$$

Here, we use the fact that $P^0 = 0$. By weak duality [14], $D^N \geq OPT$. Further by our assumption that $D^0 \leq \frac{1}{2}OPT$, we have $P^N \geq \frac{1}{2\alpha}OPT$. So the algorithm is 2α competitive. □

In fact, the initial dual value of our algorithm PD_1 is $D^0 = 0$. Thus our algorithm is α (instead of 2α) competitive, if we can find the smallest $\alpha \geq 1$ such that $P^n - P^{n-1} \geq \frac{1}{\alpha}(D^n - D^{n-1})$ for all steps n, since it achieves $P^N = \frac{1}{\alpha}(D^N - D^0) = \frac{1}{\alpha}D^N \geq \frac{1}{\alpha}OPT$. The key to identify this α is to show that our constructed marginal pricing function $p_{rsk}(t)$ in (10) satisfies an *Allocation-Payment Relationship* contingent on this α, which is sufficient to guarantee the inequality in Lemma 1, based on Theorem 2.

Definition 1: The Allocation-Payment Relationship *for a given parameter* $\alpha \geq 1$ *is*

$$p_{rsk}^{n-1}(t)\big(y_{rsk}^n(t) - y_{rsk}^{n-1}(t)\big)$$
$$- \big(f_{rsk}(y_{rsk}^n(t)) - f_{rsk}(y_{rsk}^{n-1}(t))\big)$$
$$\geq \frac{1}{\alpha}\big(f_{rsk}^\star(p_{rsk}^n(t)) - f_{rsk}^\star(p_{rsk}^{n-1}(t))\big), \qquad (11)$$

for all $r \in \mathcal{R}$, $s \in \mathcal{S}_k$, $t \in [t_{n_i}^-, t_{n_i}^+]$, $i \in \mathcal{I}_n$ *and* $n \in \mathcal{N}$.

The *Allocation-Payment Relationship* shows that the difference between payment for type-r resource on server s in DC k (according to the old price before handling n) and the incremental cost of server s in DC k due to bid n's additional use of resource r, is no smaller than $\frac{1}{\alpha}$ of the value increase of the conjugate $f_{rsk}^\star$ due to the adjustment of the marginal price. Since the payment of user n according to the adjusted price is no smaller than $p_{rsk}^{n-1}(t)d_{nir}(t)$, it guarantees that the payment from a bid if served can cover the loss in the server cost to a guaranteed extent. Concretely, the following lemma shows that the expected net profit is lower bounded by a constant fraction of the increase in the dual objective due to the increase of dual prices $p_{rsk}(t)$.

Lemma 2: If the Allocation-Payment Relationship holds for $\alpha \geq 1$, *then for each user* n, *his accepted bid* $i_n \in \mathcal{B}_n$ *and the corresponding server* $s^\star \in \mathcal{S}_{k_{ni^\star}}$ *selected for serving option* $i^\star$ *of* B_n *satisfy*

$$\hat{p}_{ni^\star} - \sum_{t \in [t_{ni}^-, t_{ni}^+]} \sum_{r \in \mathcal{R}}$$
$$\times \big(f_{rs^\star k}(y_{rs^\star k}^n(t)) - f_{rs^\star k}(y_{rs^\star k}^{n-1}(t))\big)$$
$$\geq \frac{1}{\alpha}(D^n - D^{n-1} - u_n)$$

Proof: Recall that

$$\hat{p}_{ni^\star} = \sum_{t \in [t_{i_n}^-, t_{i_n}^+]} \sum_{r \in \mathcal{R}} \big(y_{rs^\star}^n(t) - y_{rs^\star}^{n-1}(t)\big) p_{rs^\star k}^{n-1}(t)$$

and that $D^n - D^{n-1} =$

$$u_n + \sum_{t \in [t_{i_n}^-, t_{i_n}^+]} \sum_{r \in \mathcal{R}} \big(f_{rs^\star k}^\star(p_{rs^\star k}^n(t)) - f_{rs^\star k}^\star(p_{rs^\star k}^{n-1}(t))\big).$$

The lemma follows by summing the Allocation-Payment Relationship (11) over all $r \in \mathcal{R}$ and $t \in [t_{i_n}^-, t_{i_n}^+]$. □

Theorem 2: If the Allocation-Payment Relationship holds for $\alpha \geq 1$, then $P^n - P^{n-1} \geq \frac{1}{\alpha}(D^n - D^{n-1})$ for all $n \in \mathcal{N}$.

Proof: If user n is rejected, then $P^n - P^{n-1} = D^n - D^{n-1} = 0$. Otherwise, The change of primal objective is $P^n - P^{n-1} =$

$$b_{ni} - \sum_{t \in [t_{i_n}^-, t_{i_n}^+]} \sum_{r \in \mathcal{R}} \left(f_{rs^\star k}(y_{rs^\star k}^n(t)) - f_{rs^\star k}(y_{rs^\star k}^{n-1}(t)) \right)$$

Note that $b_{ni} = u_n + \hat{p}_{ni^\star}$. By Lemma 2, we get that

$$P^n - P^{n-1} \geq u_n + \frac{1}{\alpha}(D^n - D^{n-1} - u_n)$$

By $u_n \geq 0, \alpha \geq 1$, we have $P^n - P^{n-1} \geq \frac{1}{\alpha}(D^n - D^{n-1})$.   □

We next find the smallest $\alpha \geq 1$ for which our marginal price functions satisfy the respective *Allocation-Payment Relationship*. Note that each inequality (11) in Definition 1 involves only quantities about the same resource r and the same server s. Therefore, we will identify a separate ratio $\alpha_{rsk} \geq 1$ for each combination of r, s, and k such that the marginal pricing functions on r, s and k satisfy the *Differential Allocation-Payment Relationship* decided by this α_{rsk} (Definition 2). We can then take α as the maximum of all α_{rsk}'s, *i.e.*, $\alpha = \max_{r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}]} \alpha_{rsk}$, to satisfy (11) for all r, s and k. Then we can derive a differential version of the Allocation-Payment Relation (11) based on $d_{nir}(t) = \mathrm{d}y_{rsk}(t)$ for all $k \in \mathcal{K}, s \in \mathcal{S}_k, t \in [t_{ni}^-, t_{ni}^+]$, as follows:

Definition 2: The Differential Allocation-Payment Relationship *for a given parameter $\alpha_{rs} \geq 1$ is*

$$p_{rsk}(t)\mathrm{d}y_{rsk}(t) - f'_{rsk}(y_{rsk}(t))\mathrm{d}y_{rsk}(t)$$
$$\geq \frac{1}{\alpha_{rsk}} f_{rsk}^{\star}{}'(p_{rsk}(t))\mathrm{d}p_{rsk}(t), \quad \forall r \in \mathcal{R}, \ s \in \mathcal{S}_k,$$
$$k \in \mathcal{K}, \quad t \in [t_{ni}^-, t_{ni}^+], \ i \in \mathcal{I}_n, \ and \ n \in \mathcal{N}. \quad (12)$$

Lemma 3: The marginal payment function defined in (10) *satisfies the Differential Allocation-Payment Relationship for*

$$\alpha_{rsk} = \max \Big\{ 4(1 + \beta_{rsk}), \frac{2(1 + \beta_{rsk})}{\beta_{rsk}} \\ \ln\Big(\frac{U_r}{h_{rsk}(1 + \beta_{rsk})C_{rsk}^{\beta_{rsk}}}\Big) \Big\},$$

with parameters

$$\delta_{rsk} = \max \Big\{ 2, (1 + \beta_{rsk})^{\frac{1}{\beta_{rsk}}} \Big\},$$
$$\theta_{rsk} = \max \Big\{ \frac{\delta_{rsk}}{C_{rsk}} \beta_{rsk}, \frac{\delta_{rsk}}{C_{rsk}(\delta_{rsk} - 1)} \\ \times \ln\Big(\frac{U_r}{h_{rsk}(1 + \beta_{rsk})C_{rsk}^{\beta_{rsk}}}\Big) \Big\},$$

where

$$U_r = \max_{i \in \mathcal{I}_n, n \in \mathcal{N}, t \in \mathcal{T}} \frac{b_{ni}}{d_{nir}(t)}$$

is the upper bound of the value per unit of resource r per unit of time. It serves as the upper bound of the unit price of resource r such that no bid could be accepted at price U_r.

We will need the following lemma, which states that the marginal payment is greater than the marginal cost by at least a $1 + \beta_{rsk}$ factor.

Lemma 4: For $y_{rsk}(t) \in [0, C_{rsk}]$ and the corresponding $p_{rsk}(t)$, we have that $p_{rsk}(t) \geq (1 + \beta_{rsk})f'_{rsk}(y_{rsk}(t)) = h_{rsk}(1 + \beta_{rsk})^2 y_{rsk}(t)^{\beta_{rsk}}$, for the parameters in Lemma 3.

The proofs of Lemma 4 and Lemma 3 are given in Appendices B and C. We next obtain the competitive ratio of online auction PD_1.

Theorem 3: The online auction PD_1 in Alg. 1 is α_1-competitive in social welfare, for $\alpha_1 = \max_{r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}} \alpha_{rsk}$ with the parameters given in Lemma 3.

Proof: By Lemma 3, the marginal prices used by online auction PD_1 satisfies the Differential Allocation-Payment Relationship defined in (12) for α_1. By the assumption that each $\mathrm{d}y_{rs^\star k} = y_{rs^\star k}^i(t) - y_{rs^\star k}^{n-1}(t) = d_{nir}(t)$ is very small compared to the capacity (and that $\mathrm{d}y_{rsk}(t) = y_{rsk}^n(t) - y_{rsk}^{n-1}(t) = 0$ for any $s \neq s^\star$ or $k \neq k_{ni}$), we get that

$$f_{rsk}\big(y_{rsk}^n(t)\big) - f_{rsk}\big(y_{rsk}^n(t)\big)$$
$$= f'_{rsk}\big(y_{rsk}^{n-1}(t)\big)\big(y_{rsk}^n(t) - y_{rsk}^{n-1}(t)\big)$$
$$f_{rsk}^\star\big(p_{rsk}^n(t)\big) - f_{rsk}^\star\big(p_{rsk}^{n-1}(t)\big)$$
$$= f_{rsk}^{\star}{}'\big(p_{rsk}^{n-1}(t)\big)\big(p_{rsk}^n(t) - p_{rsk}^{n-1}(t)\big).$$

So (12) holds implies that the Allocation-Payment Relationship (11) also holds for α_1. Then, by Lemma 1, Theorem 2, the theorem follows.   □

B. Online Auction for Profit Maximization in Expectation

Next, we present an online auction for profit maximization in expectation based on the social welfare maximizing online auction PD_1, inspired by a randomized reduction technique [28]. While the idea of increasing the payment by a randomly chosen power-of-2 factor is similar to [28], the analysis of [28] does not extend to our setting with server costs and our analysis is fundamentally different. In particular, we will use an online primal-dual analysis, comparing the expected profit of our online auction to the dual objective value of the social welfare optimization problem. To the best of our knowledge, using the online primal dual framework to analyze profit is new in the literature, and the only known technique for analyzing profit with resource costs is the work by Blum *et al.* [21]. However, the competitive ratio achieved by their technique grows logarithmically in the number of bids, which is undesirable as we are interested in systems involving a large number of bids. In contrast, our competitive ratio is independent on the number of bids. Further, the technique of [21] incurs an additive loss in expected profit.

1) Auction Design: We first introduce a few parameters: $\beta_{\min}$ is the minimum of β_{rsk}; L_r and U_r are the lower and upper bounds of a user's value per unit of resource per unit of time, respectively.

$$\beta_{\min} = \min_{r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}} \beta_{rsk}$$
$$L_r = \min_{i \in \mathcal{I}_n, n \in \mathcal{N}} \frac{b_{in}}{R \sum_{t \in [t_i^-, t_i^+]} d_{nir}(t)}$$
$$U_r = \max_{i \in \mathcal{I}_n, n \in \mathcal{N}, t \in \mathcal{T}} \frac{b_{in}}{d_{nir}(t)} \quad (13)$$

The online auction RPD_1 is presented in Alg. 3. The idea is to use PD_1 to first obtain a tentative allocation and payment for each bid, and then re-examine each tentatively accepted bid with a boosted payment to improve profit in expectation. If the bidding price of the bid is larger than the respective boosted payment, it will be accepted; otherwise it will be rejected. Here, PD_1 serves as a pre-screening step that filters out low-value bids, giving us a set of bids whose total value is comparable to the offline optimal social welfare (according

to the competitive analysis of PD_1). The problem is that the payment chosen by PD_1 for, say, the bid B_{i_n}, could be much smaller than its true value b_{i_n}, leaving us with little profit. To resolve this problem, RPD_1 increases the tentative payment in two steps to guarantee that it is close to the true value v_i with non-trivial probability. First note that for any resource r, by the definition of L_r, the true value (bid price) of bid B_{i_n} is at least

$$b_{ni} \geq \sum_{t \in [t_{i_n}^-, t_{i_n}^+]} d_{nir}(t) L_r.$$

Since the above holds for any resource, we further get that

$$b_{nin} \geq \sum_{r \in \mathcal{R}} \sum_{t \in [t_{i_n}^-, t_{i_n}^+]} d_{nir}(t) \frac{L_r}{R}$$

By this observation, RPD_1 raises the tentative payment to

$$\tilde{p}_{ni^\star} = \max \Big\{ \sum_{r \in \mathcal{R}} \sum_{t \in [t_{i_n}^-, t_{i_n}^+]} d_{nir}(t) \frac{L_r}{R}, \hat{p}_{ni^\star} \Big\}$$

The above payment can still be very far from the true value $v_{ni^\star}$, where $i^\star$ is the chosen option of the tentatively accepted bid n. Hence, RPD_1 further multiply $\tilde{p}_{ni^\star}$ by a randomly chosen power of 2, denoted as η, which is drawn from a carefully chosen distribution (*i.e.*, (14)). By doing so, the final payment is within a factor of 2 (and lower than) the true value $v_{ni^\star}$ with non-trivial probability. We note that Alg. 3 requires U_r and L_r as input, whose exact values are not known before all bids have arrived. Instead, we adopt estimated values of the upper and lower bounds as input to our online algorithm, *e.g.*, based on past experience. We will show in the simulations good performance of the auction even if the estimation is quite different from the actual value.

2) Analysis: **(i) Truthfulness, Individual Rationality, and Polynomial Time**

Theorem 4: The randomized online auction RPD_1 in Alg. 3 is truthful and individually rational, and processes each bid in $O(RSIT)$ time, where $S = \max_{k \in \mathcal{K}} S_k$ and $I = \max_{n \in \mathcal{N}} I_n$.

The proof of truthfulness and time complexity of bid processing follows similarly to the proof of Theorem 1, and is hence omitted.

(ii) Competitiveness in Expected Profit

Theorem 5: The randomized online auction RPD_1 in Alg. 3 is $O(\alpha_1 + \log_2 \chi)$-competitive in expected profit, where α_1 is given in Theorem 3, $\chi = \max_{r \in \mathcal{R}} \frac{R U_r}{L_r}$ with U_r and L_r defined in (13).

Proof: Let R^n denote the expected profit that RPD_1 generates from the first n bids ($R^0 = 0$). We will use an analysis similar to the online primal dual approach, and show that for any $n \in \mathcal{N}$, we have

$$O(\alpha_1 + \log_2 \chi) \cdot (R^n - R^{n-1}) \geq D^n - D^{n-1} \quad (15)$$

where D_n is dual objective of primal dual auction PD_1 after handling user n. The theorem follows because the expected profit of RPD_1 at the end of the instance satisfies $O(\alpha_1 + \log_2 \chi) R^N \geq D^N - D^0 \geq OPT$, where the last inequality holds because $D^N \geq OPT$ (weak duality) and $D^0 = 0$ (definition of PD_1). It remains to prove (15). If bid n is not accepted by PD_1, it is not accepted by RPD_1 either. Therefore, both sides of (15) are zero. Next, suppose option $i^\star$ of bid n is accepted by PD_1. Let

$$c_{ni^\star} = \sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} \sum_{r \in \mathcal{R}} f_{rs^\star k}'(y_{rs^\star k}(t)) \mathrm{d}y_{rs^\star k}(t) \quad (16)$$

Algorithm 3 Randomized Online Auction RPD_1

Input: $\mathbf{S}, K, R, \mathbf{C}, \mathbf{h}, \beta, L_r, U_r$
Output: $\mathbf{x}, \mathbf{p}$

1 DEFINE $\chi = \max_{r \in [R]} \frac{R U_r}{L_r}$, $\tilde{\beta} = \min_{s \in \mathcal{S}_k, r \in \mathcal{R}} \beta_{rsk}$;
2 INITIALIZE $x_{is} = 0, y_{rs}(t) = 0, u_i = 0, p_{rs}(t) = 0$;

3 **Upon the arrival of the ith bid**
4 Get *bidding language* B_i according to (1);
5 $(\mathbf{x}_{ni}, \hat{p}_{ni}, \mathbf{p}, \mathbf{y}) = \mathrm{CORE}(K, S, R, B_n, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
6 **if** $\exists s^\star, i^\star, x_{ni^\star s^\star} = 1$ **then**
7   Update

$$\tilde{p}_{ni^\star} = \max \Big\{ \sum_{r \in \mathcal{R}} \sum_{t \in [t_i^-, t_i^+]} d_{nir}(t) \frac{L_r}{R}, \hat{p}_{ni^\star} \Big\}$$

8   Generate a random number η such that:

$$\eta = \begin{cases} 1 & \text{with prob. } \frac{1}{2} \\ 2^j & \text{with prob. } \frac{1}{2 \log_2 \chi} \text{ for } j = 1, \ldots, \log_2 \chi \end{cases} \quad (14)$$

9   **if** $b_i \geq \eta \tilde{p}_i$ **then**
10   Accept option $i^\star$ of bid n, serve it on $s^\star$ of $k_{ni^\star}$, and charge $\eta \tilde{p}_{ni^\star}$;
11   **else**
12   Reject bid n;
13   **end**
14 **else**
15   Reject bid n;
16 **end**

---

denote the increase of server costs if option $i^\star$ of bid n is accepted. Then,

$$
\begin{aligned}
&R^n - R^{n-1} \\
&= \Pr[\eta = 1] \cdot (\tilde{p}_{ni^\star} - c_{ni^\star}) \\
&\quad + \sum_{j=1}^{\log_2 \chi} \Pr[\eta = 2^j] \cdot \mathbf{1}_{b_{ni^\star} \geq 2^j \tilde{p}_{ni^\star}} \cdot 2^j (\tilde{p}_{ni^\star} - c_{ni^\star}) \\
&= \frac{1}{2} (\tilde{p}_{ni^\star} - c_{ni^\star}) + \frac{1}{2 \log_2 \chi} \sum_{j=1}^{\log_2 \chi} \mathbf{1}_{b_{ni^\star} \geq 2^j \tilde{p}_{ni^\star}} \\
&\quad \cdot (2^j \tilde{p}_{ni^\star} - c_{ni^\star})
\end{aligned} \quad (17)
$$

where $\mathbf{1}_{b_{ni^\star} \geq 2^j \tilde{p}_{ni^\star}}$ equals 1 if $b_{ni^\star} \geq 2^j \tilde{p}_n$, and equals 0 otherwise. By Lemma 2 and Lemma 3 (also recall the definition of c_{ni} in (16)), we have

$$\tilde{p}_{ni^\star} - c_{ni^\star} \geq \hat{p}_{ni^\star} - c_{ni^\star} \geq \frac{1}{\alpha_1}(D^n - D^{n-1} - u_n) \quad (18)$$

So the first term of (17) alone is almost sufficient for showing (15) modulo the u_n term. The rest of the proof is divided into two cases depending on whether $b_{ni^\star} < 2 \tilde{p}_{ni^\star}$ (only the first term in (17) is non-zero, but $b_{ni^\star}$ and, thus, u_n are small), or not ($b_{ni^\star}$ and, thus, u_n are large, but we get positive contribution from the second term of (17)).

Case 1: $b_{ni^\star} < 2 \tilde{p}_{ni^\star}$. Note that

$$\tilde{p}_{ni^\star} \geq \hat{p}_{ni^\star} = \sum_{r \in \mathcal{R}} \sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} p_{rs^\star k}(t) \mathrm{d}y_{rs^\star k}(t).$$

By Lemma 4, $\tilde{p}_{ni^\star}$ is at least

$$\sum_{r \in \mathcal{R}} \sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} (1 + \beta_{rsk}) f'_{rs^\star k}(y_{rsk}(t)) \mathrm{d}y_{rs^\star k}(t)$$
$$\geq (1 + \beta_{\min}) c_{ni^\star}$$

Note that $\mathbf{1}_{b_{ni^\star} \geq 2^j \tilde{p}_{ni^\star}} = 0$ for all $j \geq 1$. We get that

$$
\begin{aligned}
R^n &- R^{n-1} \\
&= \frac{1}{2}\big(\tilde{p}_{ni^\star} - c_{ni^\star}\big) \\
&\geq \frac{1}{6}(\hat{p}_{ni^\star} - c_{ni^\star}) + \frac{1}{3}\big(\tilde{p}_{ni^\star} - c_{ni^\star}\big) \quad (\tilde{p}_{ni^\star} \geq \hat{p}_{ni^\star}) \\
&\geq \frac{1}{6}(\hat{p}_{ni^\star} - c_{ni^\star}) + \frac{\beta_{\min}}{3(1 + \beta_{\min})}\tilde{p}_{ni^\star} \quad (\tilde{p}_{ni^\star} \geq (1 + \beta_{\min})c_{ni^\star}) \\
&\geq \frac{1}{6}(\hat{p}_{ni^\star} - c_{ni^\star}) + \frac{\beta_{\min}}{6(1 + \beta_{\min})}u_n \quad (u_n \leq b_{ni^\star} < 2\tilde{p}_{ni^\star}) \\
&\geq \frac{1}{6\alpha_1}(D^n - D^{n-1} - u_n) + \frac{\beta_{\min}}{6(1 + \beta_{\min})}u_n \quad \text{(by (18))}
\end{aligned}
$$

By definition of α_{rsk}, $\alpha_{rsk} \geq \frac{1}{\beta_{rsk}}$. Recall $\alpha_1 = \max_{r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}} \alpha_{rsk}$. So $\alpha_1 \geq 1 + \frac{1}{\beta_{\min}}$. Putting together, (15) follows because

$$R^n - R^{n-1} \geq \frac{1}{6\alpha_1}(D^n - D^{n-1})$$

Case 2: $b_{ni^\star} \geq 2\tilde{p}_{ni^\star}$. On one hand, by our choice of $\tilde{p}_{ni^\star}$, it is at least $\tilde{p}_{ni^\star} \geq \sum_{r \in \mathcal{R}} \sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} d_{nir}(t)\frac{L_r}{R}$.

On the other hand, by the definition of U_r, the true value is at most $b_{ni^\star} \leq \sum_{r \in \mathcal{R}} \sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} d_{ni^\star r}(t)U_r$

So $\frac{b_{ni^\star}}{\tilde{p}_{ni^\star}} \leq \frac{RU_r}{L_r} \leq \chi$. Since η is a randomly chosen power of 2 between 1 and χ, with probability $\frac{1}{2\log_2 \chi}$, η satisfies $\frac{1}{2}b_{ni^\star} \leq \eta\tilde{p}_{ni^\star} \leq b_{ni^\star}$ and, thus, RPD$_1$ accepts option $i^\star$ of bid n and obtains profit at least $\frac{1}{2}b_{ni^\star}$ in expectation. Therefore, $R^n - R^{n-1} \geq \frac{1}{2}\big(\tilde{p}_{ni^\star} - c_{ni^\star}\big) + \frac{1}{2\log_2 \chi}\frac{1}{2}b_{ni^\star}$ (recall (17)). By (18), the first term is at least $\frac{1}{2\alpha_1}(D^n - D^{n-1} - u_n)$. The second term is at least $\frac{1}{4\log_2 \chi}u_i$ because bid i's utility is at most its value. Putting together, the expected profit from bid i is at least

$$
\begin{aligned}
R^n - R^{n-1} &\geq \frac{1}{2\alpha_1 + 4\log_2 \chi}(D^n - D^{n-1} - u_n) \\
&\quad + \frac{1}{2\alpha_1 + 4\log_2 \chi}u_n \\
&= \frac{1}{2\alpha_1 + 4\log_2 \chi}(D^n - D^{n-1})
\end{aligned}
$$

So the theorem holds in case 2 as well. $\qquad\square$

(iii) Competitiveness in Social Welfare

Theorem 6: Randomized online auction RPD$_1$ *in Alg. 3 is* $2\alpha_1$-*competitive in expected social welfare, where* α_1 *is given in Theorem 3.*

Proof: Consider any bid i that is accepted by PD$_1$, RPD$_1$ will tentatively accept the bid and then re-examine it with randomly chosen boosted payment. With probability $\frac{1}{2}$, $\eta = 1$. Further note that when $\eta = 1$, $b_{ni^\star} \geq \eta\tilde{p}_{ni^\star}$ and, thus, RPD$_1$ will accept option $i^\star$ of bid n. In sum, for every bid i that is accepted by PD$_1$, RPD$_1$ would accepted it with probability at least $\frac{1}{2}$. Therefore, the expected social welfare of RPD$_1$ is at least a half of the social welfare of PD$_1$. $\qquad\square$

Algorithm 4 Primal-Dual Online Auction PD$_2$

   Input: S, R, $\mathbf{C}$, $\mathbf{h}$, σ
   Output: $\mathbf{x}$, $\mathbf{p}$
1 DEFINE $f_{rsk}(y_{rsk}(t))$ and $p_{rsk}(y_{rsk}(t))$ according to (19) and (21);
2 INITIALIZE $x_{nis} = 0, y_{rsk}(t) = 0, u_n = 0, p_{rsk}(t) = \frac{L_r - h_{rsk}}{2R\sum_{k \in \mathcal{K}} S_k} + h_{rsk}$;
3 **Upon the arrival of the ith bid**
4 Get *bidding lauguage* B_i according to (1);
5 $(\mathbf{x}_{ni}, \hat{p}_{ni}, \mathbf{p}, \mathbf{y}) = $ CORE$(K, S, R, B_n, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
6 **if** $\exists i^\star, s^\star, x_{ni^\star s^\star} = 1$ **then**
7   Accept bid n on server $s^\star$ of $k_{ni^\star}$, and charge it at $\hat{p}_{ni^\star}$;
8 **else**
9   Reject bid n;
10 **end**

V. EXTENSION TO LINEAR SERVER COSTS

In this section, we extend the auction design in Sec. IV to the setting of linear server cost functions, *i.e.*, $\beta_{rs} = 0$ in (2) such that

$$f_{rsk}(y_{rsk}(t)) = \begin{cases} h_{rsk}y_{rsk}(t), & y_{rsk}(t) \in [0, C_{rsk}] \\ +\infty, & y_{rsk}(t) > C_{rsk} \end{cases} \quad (19)$$

which says that the marginal cost per unit of resource usage is a constant (h_{rsk}) within the capacity constraint.

Proposition 2: The conjugate of $f_{rsk}(y_{rsk}(t))$ *defined in (19) is*

$$f^\star_{rsk}(p_{rsk}(t)) = \begin{cases} 0, & p_{rsk}(t) \leq h_{rsk} \\ (p_{rsk}(t) - h_{rsk})C_{rs}, & p_{rsk}(t) > h_{rsk} \end{cases} \quad (20)$$

A. Social Welfare Maximization

We adapt online auction PD$_1$ to online auction PD$_2$ for linear cost functions, as given in Alg. 4. We define a new marginal payment function:

$$
\begin{aligned}
&p_{rsk}(y_{rsk}(t)) \\
&= \frac{L_r - h_{rsk}}{2R\sum_{k \in \mathcal{K}} S_k}\Big(\frac{2R\sum_{k \in \mathcal{K}} S_k(U_r - h_{rsk})}{L_r - h_{rsk}}\Big)^{\frac{y_{rsk}(t)}{C_{rsk}}} + h_{rsk},
\end{aligned}
\quad (21)
$$

where L_r and U_r are defined in (13). For each i of bid n such that $b_{ni}/\sum_{t \in [t_{ni}^-, t_{ni}^+]} d_{nir}(t) \leq h_{rsk}$, its value is smaller than the server cost needed to serve it and, thus, will be rejected. So we can assume without loss of generality that $L_r > h_{rsk}$. The marginal price $p_{rsk}(t)$ is a function of the amount of currently allocated resource $y_{rsk}(t)$. $p_{rsk}(t)$ is higher than marginal operational cost h_{rsk} to guarantee non-negative utility of the provider. The initial marginal price is low enough such that any bid (subject to L_r which lower bounds a bid's value per unit of resource per unit of time) will be accepted. Then, the marginal price $p_{rsk}(t)$ increases as $y_{rsk}(t)$ increases to ensure that a server will not allocate all capacity of a resource to low value bids. Finally, the marginal price is high enough (larger than U_r – upper bound of a bid's value per unit of resource per unit of time) when $y_{rsk}(t) > C_{rsk}$ to make sure a server

will not allocate more resources than its capacity. PD_2 also uses PD_1 as a sub-routine. The only difference between PD_1 and PD_2 are the marginal price functions and initial values of the marginal prices, due to different server cost functions.

Theorem 7: The online auction PD_2 in Alg. 4 is truthful and individually rational, and runs in polynomial time.

The proof is similar to that of Theorem 1 and thus omitted.

Theorem 8: The online auction PD_2 in Alg. 4 is $2\alpha_2$-competitive in social welfare with

$$\alpha_2 = \max_{r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}} \ln\Big(\frac{2R\sum_{k \in \mathcal{K}} S_k(U_r - h_{rsk})}{L_r - h_{rsk}}\Big), \quad (22)$$

assuming the offline optimal social welfare (OPT) is at least

$$\frac{1}{R\sum_{k \in \mathcal{K}} S_k} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{r \in \mathcal{R}} (L_r - h_{rsk})C_{rsk}.$$

Proof: Let $\alpha_{rsk} = \ln\frac{2R\sum_{k \in \mathcal{K}} S_k(U_r - h_{rsk})}{L_r - h_{rsk}}$. We will show that the marginal payment function defined in (21) satisfies the Differential Allocation-Payment Relationship for all resource r, server s, and time t with parameter α_{rsk}, i.e., $p_{rsk}(t)\mathrm{d}y_{rsk}(t) - f'_{rsk}(y_{rsk}(t))\mathrm{d}y_{rsk}(t) \geq \frac{1}{\alpha_{rsk}}f^{\star\prime}_{rsk}(p_{rsk}(t))\mathrm{d}p_{rsk}(t)$, and that the initial value of the dual objective is at most $\frac{1}{2}OPT$. Then, given $\alpha_2 = \max_{r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}} \alpha_{rsk}$, the theorem will follow from Lemma 1 and Theorem 2.

By (21), we get $p_{rsk}(t) > U_r$ when the demand exceeds the capacity, i.e., $y_{rsk}(t) > C_{rsk}$. So by our choice of $p_{rsk}(t)$, the cloud provider will never allocate more resources than its capacity, *i.e.*, $y_{rsk}(t) \leq C_{rsk}$. In addition, the initial marginal price $p_{rsk}(0) = \frac{L_r - h_{rsk}}{2R\sum_{k \in \mathcal{K}} S_k} + h_{rsk} > h_{rsk}$, and the marginal prices are non-decreasing. Therefore in the rest of the discussion, we may assume that $f'_{rsk}(y_{rsk}(t)) = h_{rsk}$ and $f^{\star\prime}_{rsk}(p_{rsk}(t)) = C_{rsk}$. Putting them into the above differential inequality, it becomes $(p_{rsk}(t) - h_{rsk})\mathrm{d}y_{rsk}(t) \geq \frac{C_{rsk}}{\alpha_{rsk}}\mathrm{d}p_{rsk}(t)$, which holds with equality by our choice of $p_{rsk}(t)$ and that $\alpha_{rsk} = \ln\frac{2R\sum_{k \in \mathcal{K}} S_k(U_r - h_{rsk})}{L_r - h_{rsk}}$. Finally, according to (5) and (20),

$$D^0 = \frac{1}{2}\frac{1}{R\sum_{k \in \mathcal{K}} S_k} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{r \in \mathcal{R}} (L_r - h_{rsk})C_{rsk}$$
$$\leq \frac{1}{2}OPT.$$

The last inequality is based on the assumption in the theorem. $\square$

B. Profit Maximization in Expectation

Next, we present an extension of PD_2 that achieves a competitive ratio comparable to $2\alpha_2$ in provider's expected profit. Alg. 5 gives our randomized online auction for profit maximization in expectation. Similar to how RPD_1 extends PD_1, RPD_2 first use PD_2 as a black box to obtain a tentative allocation x_{nis} and payment $\hat{p}_{ni^\star}$ for each bid n. Then it raises the payment by a factor of $\eta \geq 1$ and accepts the bid only when its value is higher than the new payment. The difference lies in the different values of χ, which decides the distribution from which η is sampled.

Theorem 9: The online auction RPD_2 in Alg. 5 is truthful and individually rational, and runs in polynomial time.

The proof is similar to that of Theorem 4 and thus omitted.

The expected profit guarantee will be presented in Theorem 10. We first show a few technical lemmas that are

---

Algorithm 5 Randomized Online Auction RPD_2

Input: K, S, R, $\mathbf{C}$, $\mathbf{h}$, L_r, U_r
Output: $\mathbf{x}$, $\mathbf{p}$

1 DEFINE $f_{rsk}(y_{rsk}(t))$ and $p_{rsk}(y_{rsk}(t))$ according to (19) and (21);
2 DEFINE $\chi = \max_{r \in \mathcal{R}, s \in \mathcal{S}_k, k \in \mathcal{K}} \frac{2R\sum_{k \in \mathcal{K}} S_k(U_r - h_{rsk})}{L_r - h_{rsk}}$;
3 INIT. $x_{nis} = 0, y_{rsk}(t) = 0, u_n = 0, p_{rsk}(t) = \frac{L_r - h_{rsk}}{2R\sum_{k \in \mathcal{K}} S_k} + h_{rsk}$;
4 **Upon the arrival of the nth bid**
5 Get *bidding language* B_i according to (1);
6 $(\mathbf{x}_{ni}, \hat{p}_{ni}, \mathbf{p}, \mathbf{y}) = \mathrm{CORE}(S, R, B_n, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
7 **if** $\exists i^\star, s^\star, x_{ni^\star s^\star} = 1$ **then**
8   Generate a random number η such that:

$$\eta = \begin{cases} 1 & \text{with prob. } \frac{1}{2} \\ 2^j & \text{with prob. } \frac{1}{2\log_2 \chi} \text{ for } j = 1, \ldots, \log_2 \chi \end{cases} \quad (23)$$

    if $b_i \geq \eta\hat{p}_i$ **then**
9   Accept option $i^\star$ of bid n on server $s^\star$ of $k_{ni^\star}$, and charge $\eta\hat{p}_{ni^\star}$;
10   **else**
11   Reject bid n;
12   **end**
13 **else**
14   Reject bid n;
15 **end**

---

needed for the proof of Theorem 10 (proofs of lemmas given in appendices D—F). Let B_{ni} be the VM i of n. Let $A^\star$ be the set of bids with the options $i^\star$ accepted by social welfare maximizing online auction PD_2 and O be the set of bids with the options i accepted by the omniscient optimal algorithm for social welfare maximization. c_{ni} is the cost occurs by an accepted option i of bid n in (16).

Lemma 5: The optimal profit in expectation of any truthful and individually rational auction (online or offline) of our problem is upper bounded by

$$\sum_{B_{ni^\star} \in A^\star} (b_{ni^\star} - c_{ni^\star}) + \sum_{B_{ni} \in O} (\min\{p_{ni}, b_{ni}\} - c_{ni}) \quad (24)$$

Lemma 6: The first term of (24) is upper bounded by $4\log_2 \chi$ times the expected profit of online auction RPD_2.

Lemma 7: The second term of (24) can be upper bounded by α_2 times the profit of online auction PD_2, i.e.,

$$\sum_{B_{ni} \in O} (\min\{p_{ni}, b_{ni}\} - c_{ni})$$
$$\leq \alpha_2 \sum_{B_{ni^\star} \in A^\star} (\hat{p}_{ni^\star} - c_{ni^\star}) \quad (25)$$

where α_2 is defined in (22).

Lemma 8: The expected profit of RPD_2 is at least a half of the profit of PD_2.

Proof: It follows from that for every bid n, RPD_2 uses the same allocation and payment as in PD_2 with probability $\frac{1}{2}$. $\square$

Theorem 10: The randomized online auction RPD_2 in Alg. 5 is $\left(\frac{4}{\ln 2} + 2\right)\alpha_2$-competitive in terms of expected profit.

Proof: Combining Lemmas 5, 6, 7 and 8, the optimal expected profit is upper bounded by $4\log_2\chi + 2\alpha_2$ times the expected profit of RPD$_2$. By the choice of $\chi = \max_{r\in\mathcal{R},s\in\mathcal{S}_k,k\in\mathcal{K}} \frac{2R\sum_{k\in\mathcal{K}}S_k(U_r - h_{rsk})}{L_r - h_{rsk}}$ and the definition of α_2 in (22), we get $\log_2\chi = \frac{1}{\ln 2}\ln\chi = \frac{1}{\ln 2}\alpha_2$.   □

Theorem 11: The randomized online auction RPD$_2$ *in Alg. 5 is $4\alpha_2$-competitive in terms of expected social welfare.*

Finally, we note that as a special case, the online auctions in this section also handle the case with no server costs within the capacity, which is equivalent to having $h_{rsk} = 0$, and the server cost functions as the following zero infinity functions:

$$f_{rsk}(y_{rsk}(t)) = \begin{cases} 0, & y_{rsk}(t) \in [0, C_{rsk}] \\ +\infty, & y_{rsk}(t) > C_{rsk} \end{cases} \quad (26)$$

The properties shown for linear cost functions apply to zero infinity costs, with proofs omitted due to duplicity.

VI. PERFORMANCE EVALUATION

We evaluate our auctions using trace-driven simulations, exploiting Google cluster-usage data [29]. A job comprises multiple tasks, each of which is accompanied by its resource requirement (CPU, RAM and disk). Google cluster data provides the arrival time of each job, and the start time and end time of each task in a job. Thus we translate each job into a VM, requesting $R = 3$ types of resources at the demands extracted from the traces (note demand $d_{nir}(t)$ here is not much smaller than C_{rsk}, though our theoretical analysis assumed so). We randomly combine 1 to 10 VMs (as options) to compose a bid. We set $T = 5000$ and each time slot is 10 seconds long [30]. We set the arrival time according to the job arrival time in Google data, the start (end) time of VM usage in the bids, t_{ni}^- (t_{ni}^+) according to the earliest (latest) time slot in the VM durations of the corresponding tasks. We randomly choose a DC for each option. We set b_{ni} by multiplying the corresponding resource demands by unit prices randomly picked within different ranges, according to the upper and lower bounds of users' value per unit of resource per unit of time, U_r and L_r, which we will vary in different experiments. We will vary the span between bid arrival time and the VM start time. The number of data centers (K) is 13 according to the number of Google data centers [31]. We simulate servers with heterogenous resource capacities (C_{rsk}) following the distribution of server configurations summarized from the Google data as follows (CPU and Memory units are normalized so that the maximum capacity is 1):

# of machines	6732	3863	1001	795	126
(percentage)	(53%)	(30%)	(8%)	(6%)	(3%)
CPU	0.50	0.50	0.50	1.00	0.25
Memory	0.50	0.25	0.750	1.00	0.25

Since the Google data does not provide disk configurations of servers, we set the disk storage capacity of our servers randomly within $[320, 800](GB)$. The total capacity of each type of resource to provision, and hence the number of servers in each DC to simulate, is roughly according to the total amount of demand from all bids multiplying a random number in $[0.4, 0.8]$. h_{rsk} is set within $[0.4, 0.6]$ for CPU (different for different servers s), and within $[0.005, 0.02]$ for RAM and disk, roughly following the percentage measured in [26]. For superlinear cost functions, we set β_{rsk} within $[1.7, 2.2]$ for CPU and within $[0.5, 1]$ for RAM and disk [24], [26]. By default, $U_r = 50$, $L_r = 1$.
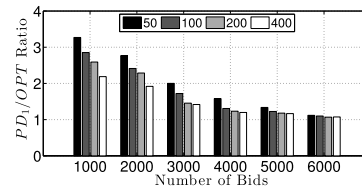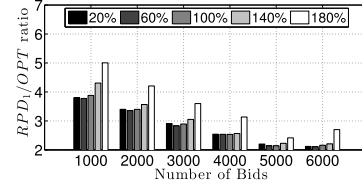

Fig. 1. PR of PD_1 in social welfare (Γ).

Fig. 2. PR of RPD_1 in profit (different values of $\tilde{U}_r/U_r$.
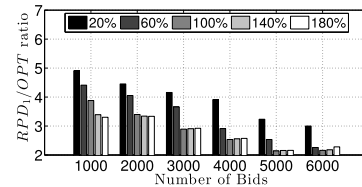

Fig. 3. PR of RPD_1 in profit (different values of $\tilde{L}_r/L_r$).

We compare our algorithms with the offline optimum and three existing schemes *Twice-the-Cost* (TC), *Twice-the-Index* (TI) [21], and RSM [9]. TC and TI share similar basic ideas with ours, but adopt different marginal pricing functions: For TC, $p_{rsk}(y_{rsk}(t)) = 2f'_{rsk}(y_{rsk}(t))$ (the current marginal payment is twice of the current marginal cost); for TI, $p_{rsk}(y_{rsk}(t)) = f'_{rsk}(2y_{rsk}(t))$ (the current marginal payment is the marginal cost on twice of the current resource usage). RSM uses a similar price function with PD$_2$ for the problem allowing preemption without considering server cost. We disable preemption of RSM and consider corresponding server cost in the comparison with PD$_1$(RPD$_1$) and PD$_2$(RPD$_2$).

A. Comparison With Offline Optimum

We first study the performance ratios (PRs) achieved by our online auctions, computed by dividing the offline optimal social welfare obtained by solving (4) exactly using Gurobi Optimizer, by the social welfare/profit achieved by the respective online algorithm. Due to the high time complexity of solving the offline convex program with a large number of variables, we set the largest number of bids to be 6000 in this set of experiments. Fig. 1 illustrates that a smaller performance ratio comes with a larger Γ, which is the average number of time slots between the bid arrival time to its specified VM start time over all the bids. This result is interesting but quite reasonable: a group of bids apply for a future occupation duration; the earlier they arrive before their actual resource occupation starts, the more future information the system can learn, and better allocation and payment decisions can be made. In Fig. 2 and 3, we use $\tilde{U}_r$ and $\tilde{L}_r$, the estimated values of U_r or L_r as input to RPD$_1$, which are at different percentages as divided by the real U_r or L_r. We see that both overestimation and underestimation have minor influence in the performance, as compared to that achieved by the real U_r or L_r (the case of 100% in the figures), and underestimation of U_r and overestimation of L_r are more desirable.
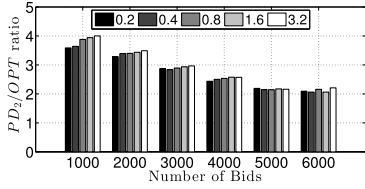
Fig. 4. Compare $PD1$, TC, TI and RSM (welfare); $RPD1$, TC, TI and RSM (profit).

B. Comparison With Existing Schemes

We next compare the social welfare and profit achieved by our auctions with TC and TI at larger scales of the system. Fig. 4 shows that (i.) the social welfare achieved by our PD_1 outperforms those by TC, TI and RSM; (ii.) the profit achieved by our RPD_1 outperforms those achieved by TC, TI, and RSM. Especially, we observe through our experiments that the marginal payment function in TC can not filter out low value bids when the used resource of a server approaches its full capacity, and with TI, maximally only half of the capacity on a server can be allocated due to the $+\infty$ part of the cost function, both leading to lower social welfare.

VII. Concluding Remarks

This work designs truthful and efficient online VM auctions where cloud users have multiple options to require future resources for her tailor-made VM with different running durations and location preferences, targeting social welfare maximization and cloud provider's profit maximization. We consider server costs in our auction model, and handle the resulting significantly more challenging mechanism design by leveraging a set of latest, novel primal-dual online optimization and randomized reduction techniques. Our primal-dual framework adopts a new application of Fenchel duality and handles various convex server cost functions. It further allows request departures and resource recycling while guaranteeing good competitive ratios. For profit maximization, we introduce a new online primal-dual analysis to obtain good competitive ratios with super-linear server costs, which is new in the literature. Trace driven simulations validate our theoretical analysis and show good performance of our mechanisms as compared to the offline solution and two existing mechanisms on similar frameworks.

Appendix A
Proof of Theorem 1

Proof: (Truthfulness in bidding price) The marginal prices are independent on the bidding price for any option in each bid. Further, we always assign bids to servers to maximize each bid's utility given the current marginal prices. Upon the arrival of each bidder, we only choose the option that yields the maximal utility for the bidder among all the options in his bid and accept it if the utility is positive, which is also what the bidder looks for. In this way, our mechanism is equivalent to one sequential posted price mechanism (e.g., [33]), where the auctioneer posts the prices to a bidder and let the bidder choose his best option, which maximizes his utility, to submit. Under the take-it-or-leave-it pricing scheme, a bidder cannot improve his utility by lying about his bidding price for his best option [33]: if the bidding price for the best option is higher than the corresponding true value, the bidder may suffer from a negative utility; if the bidding price is lower than the true

value, this option may not be selected, resulting in lower utility for the bidder. Moreover, reporting fake bidding prices for the user's other options cannot increase the utility of the user: case (i.) faking bidding price in a non-optimal option such that it becomes the best option leads to a user utility under the non-optimal option, which is not the maximum utility that the user can achieve; case (ii.) reporting a false bidding price for a non-optimal option such that it is still not the best option does no influence the user utility. Therefore, even for a multi-minded bidder, reporting truthful bidding prices for all his options is the dominant strategy.

(Truthfulness in arrival time): Since the marginal prices are non-decreasing in the amount of allocated resources, which is non-decreasing over the resource allocation time. Hence a bidder cannot decrease the total price of the resource that it requests by delaying its arrival. Note that the arrival time of a bid is the first time the bidder is aware of her demands so the arrival time can not be earlier.

(Truthfulness in resource occupation times): Dropping part of the true resource occupation duration in the request would risk failing to complete the job. On the other hand, the marginal prices are non-negative according to (10). So requesting a superset of the true resource occupation duration increases a bidder's payment and decreases her utility.

(Individually rational): By (8), the bidder's utility is non-negative. The profit under any realization of our randomized algorithms of the provider is also non-negative by (10) which implies $p_{rsk}(y_{rsk}(t)) > f'_{rsk}(y_{rsk}(t))$.

(Polynomial running time): To process a user n, we compute the payment if option i is accepted, which runs in $O(RSIT)$ time. Then, we computes u_n and decides the allocation and payment by checking the utility in $O(S_{k_{ni}}ni^\star)$ time. Finally, we update $y_{rsk}(t)$ and $p_{rsk}(t)$ in $O(RSIT)$ time. $\square$

Appendix B
Proof of Lemma 4

Proof: It is easy to verify when $y_{rsk}(t) \leq \frac{C_{rsk}}{\delta_{rsk}}$ according to (10) and the choice of δ_{rsk} in Lemma 3. Next, assume that $y_{rsk}(t) \geq \frac{C_{rsk}}{\delta_{rsk}}$. In this case, $p_{rsk}(t) = h_{rsk}(1 + \beta_{rsk})C_{rsk}^{\beta_{rsk}}e^{\theta_{rsk}(y_{rsk}(t) - \frac{C_{rsk}}{\delta_{rsk}})}$. So the inequality is equivalent to $\frac{e^{\theta_{rsk}y_{rsk}(t)}}{y_{rsk}(t)^{\beta_{rsk}}} \geq \frac{(1+\beta_{rsk})e^{\frac{\theta_{rsk}C_{rsk}}{\delta_{rsk}}}}{C_{rsk}^{\beta_{rsk}}}$. Recall that $\delta_{rsk} \geq (1 + \beta_{rsk})^{1/\beta_{rsk}}$, The above inequality holds with equality for $y_{rsk}(t) = \frac{C_{rsk}}{\delta_{rsk}}$. Next, it suffices to show that $\frac{e^{\theta_{rsk}y_{rsk}(t)}}{y_{rsk}(t)^{\beta_{rsk}}}$ is non-decreasing as $y_{rsk}(t)$ increases. Its derivative is $\frac{e^{\theta_{rsk}y_{rsk}(t)}(\theta_{rsk}y_{rsk}(t) - \beta_{rsk})}{y_{rsk}(t)^{1+\beta_{rsk}}}$. By our choice of $\theta_{rsk} \geq \frac{\delta_{rsk}\beta_{rsk}}{C_{rsk}}$ and the assumption that $y_{rsk}(t) \geq \frac{C_{rsk}}{\delta_{rsk}}$, $\theta_{rsk}y_{rsk}(t) - \beta_{rsk} \geq 0$ and the above derivative is non-negative. $\square$

Appendix C
Proof of Lemma 3

Proof: By our choice of $p_{rsk}(t)$, we have $y_{rsk}(t) \leq C_{rsk}$, due to $p_{rsk}(t) = h_{rsk}(1 + \beta_{rsk})C_{rsk}^{\beta_{rsk}}e^{\theta_{rsk}(C_{rsk} - \frac{C_{rsk}}{\delta_{rsk}})} \geq U_r$ ($y_{rsk}(t) = C_{rsk}$). Therefore, we may assume in the rest of the proof that $y_{rsk}(t) \leq C_{rsk}$ and, thus, $f'_{rsk}(y_{rsk}(t)) = h_{rsk}(1 + \beta_{rsk})y_{rsk}(t)^{\beta_{rsk}}$. Next, according to the piece-wise definition of $f^{\star\prime}_{rsk}$, the proof is divided into two cases.

Case 1: $y_{rsk}(t) \leq \frac{C_{rsk}}{\delta_{rsk}}$. By the definition of $p_{rsk}(t)$ in (10),

$$p_{rsk}(t) = f'_{rsk}(\delta_{rsk} y_{rsk}(t)) = h_{rsk}(1+\beta_{rsk})\delta_{rsk}^{\beta_{rsk}} y_{rsk}(t)^{\beta_{rsk}}. \tag{27}$$

Thus the Differential Allocation-Payment Relationship in this case is equivalent to $(\delta_{rsk}^{\beta_{rsk}} - 1) \geq \frac{1}{\alpha_{rsk}}\beta_{rsk}\delta_{rsk}^{\beta_{rsk}+1}$. If $\beta_{rsk} \geq 1$, then $\delta_{rsk} = \max\{2, (1+\beta_{rsk})^{\frac{1}{\beta_{rsk}}}\} = 2$ and $\delta_{rsk}^{\beta_{rsk}+1} = 2\delta_{rsk}^{\beta_{rsk}} = 4\delta_{rsk}^{\beta_{rsk}} - 2\delta_{rsk}^{\beta_{rsk}} \leq 4\delta_{rsk}^{\beta_{rsk}} - 4$. So we get that $\frac{\beta_{rsk}\delta_{rsk}^{\beta_{rsk}+1}}{\delta_{rsk}^{\beta_{rsk}}-1} \leq 4\beta_{rsk} < \alpha_{rsk}$. If $\beta_{rsk} < 1$, then $\delta_{rsk} = (1+\beta_{rsk})^{\frac{1}{\beta_{rsk}}}$, and $\frac{\beta_{rsk}\delta_{rsk}^{\beta_{rsk}+1}}{\delta_{rsk}^{\beta_{rsk}}-1} \leq e(1+\beta_{rsk}) < \alpha_{rsk}$.

Case 2: $\frac{C_{rsk}}{\delta_{rsk}} \leq y_{rsk}(t) \leq C_{rsk}$. Recall that when $y_{rsk}(t) \geq \frac{C_{rsk}}{\delta_{rsk}}$, the marginal payment is

$$p_{rsk}(t) = h_{rsk}(1+\beta_{rsk})C_{rsk}^{\beta_{rsk}} e^{\theta_{rsk}(y_{rsk}(t) - \frac{C_{rsk}}{\delta_{rsk}})} \tag{28}$$

By Lemma 4, to show the Differential Allocation-Payment Relationship in this case, it suffices to show $\frac{\beta_{rsk}}{1+\beta_{rsk}}p_{rsk}(t)\mathrm{d}y_{rsk}(t) \geq \frac{1}{\alpha_{rsk}}C_{rsk}\mathrm{d}p_{rsk}(t)$. On the other hand, by the definition of the marginal payment in (28), we have $\mathrm{d}p_{rsk}(t) = \theta_{rsk}p_{rsk}(t)\mathrm{d}y_{rsk}(t)$. So it remains to show that $\theta_{rsk} \leq \frac{\beta_{rsk}}{C_{rsk}(1+\beta_{rsk})}\alpha_{rsk}$. By our choice of parameters,

$$\theta_{rsk} \leq \frac{e}{C_{rsk}}\beta_{rsk} = \frac{\beta_{rsk}}{C_{rsk}(1+\beta_{rsk})}e(1+\beta_{rsk}) \; (\delta_{rsk} \leq e)$$
$$\leq \frac{\beta_{rsk}}{C_{rsk}(1+\beta_{rsk})}e(1+\beta_{rsk}) \leq \frac{\beta_{rsk}}{C_{rsk}(1+\beta_{rsk})}\alpha_{rsk}$$
$$\theta_{rsk} \leq \frac{2}{C_{rsk}}\ln\left(\frac{U_r}{h_{rsk}(1+\beta_{rsk})C_{rsk}^{\beta_{rsk}}}\right) \; (\delta_{rsk} \geq 2)$$
$$\leq \frac{\beta_{rsk}}{C_{rsk}(1+\beta_{rsk})}\frac{2(1+\beta_{rsk})}{\beta_{rsk}}\ln\left(\frac{U_r}{h_{rsk}(1+\beta_{rsk})C_{rsk}^{\beta_{rsk}}}\right)$$
$$\leq \frac{\beta_{rsk}}{C_{rsk}(1+\beta_{rsk})}\alpha_{rsk}.$$

$\square$

APPENDIX D
PROOF OF LEMMA 5

Proof: The expected profit of any truthful and individually rational auction is upper bounded by the optimal social welfare, i.e., $\sum_{B_{ni} \in O}(b_{ni} - c_{ni})$. For each $B_{ni} \in O$, if $b_{ni} \leq p_{ni}$, $b_{ni} = \min\{b_{ni}, p_{ni}\}$; otherwise we have $b_{ni} - c_{ni} \leq b_{ni^\star} - \hat{p}_{ni^\star} + \hat{p}_{ni} - c_{ni} = b_{ni^\star} - \hat{p}_{ni^\star} + \min\{b_{ni}, \hat{p}_{ni}\} - c_{ni}$. Plus the commonly accepted bids with options, we upper bound the optimal social welfare. $\square$

APPENDIX E
PROOF OF LEMMA 6

Proof: Due to the linear cost function in this case, the incurred cost by an accepted bid n is $c_{ni^\star} = \sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]}\sum_{r \in \mathcal{R}} d_{ni^\star r}(t)h_{rs^\star k_{ni^\star}}$, where $s^\star$ is the server that serves the winning bid n. By the definition of PD_2, for any tentatively accepted bid $B_{ni^\star} \in A^\star$, the tentative profit from bid i is at least $\hat{p}_{ni^\star} - c_n$

$$= \sum_{r \in \mathcal{R}}\sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]}$$
$$\times \left(d_{ni^\star r}(t)p_{rs^\star k_{ni}}(t) - d_{ni^\star r}(t)h_{rs^\star k_{ni}}\right)$$
$$\geq \sum_{r \in \mathcal{R}}\sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} d_{ni^\star r}(t)\left(p_{rs^\star k_{ni}}(0) - h_{rs^\star k_{ni}}(t)\right)$$

Since $p_{rsk}(0) = \frac{L_r - h_{rsk}}{2RSK} + h_{rsk}$, we further have

$$\hat{p}_{ni^\star} - c_n \geq \sum_{r \in \mathcal{R}}\sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} d_{ni^\star r}(t)\frac{L_r - h_{rs^\star k_{ni^\star}}}{2RSK}$$

On the other hand, the maximum profit under any realization of our randomized algorithm from the chosen option $i^\star \in \mathcal{I}_n$ of any bid n served on any server $s \in \mathcal{S}_{k_{ni^\star}}$ is at most $b_{ni^\star} - c_{ni^\star} \leq \sum_{r \in \mathcal{R}}\sum_{t \in [t_{ni^\star}^-, t_{ni^\star}^+]} d_{ni^\star r}(t)(U_r - h_{rs^\star k_{ni^\star}})$. We have that $\frac{b_{ni^\star} - c_{ni^\star}}{\hat{p}_{ni^\star} - c_n} \leq \frac{2RSK(U_r - h_{rsk})}{L_r - h_{rsk}} \leq \chi$. By the definition of η in (23), η is a randomly chosen power of 2 between 1 and χ and, thus, satisfies that $\frac{1}{2}(b_{ni^\star} - c_n) \leq \eta(\hat{p}_{ni^\star} - c_n) \leq (b_{ni^\star} - c_n)$ with probability at least $\frac{1}{2\log_2 \chi}$. In this case, RPD_2 will accept the bid $ni^\star$ of user n and generate profit $\eta(\hat{p}_{ni^\star} - c_n) \geq \frac{1}{2}(b_{ni^\star} - c_n)$ in expectation. Therefore, the expected profit generated by RPD_2 from every $B_{ni^\star} \in A^\star$ (that is, tentatively accepted by PD_2) is at least $\frac{1}{2\log_2 \chi} \cdot \frac{1}{2}(b_{ni^\star} - c_n)$. The expected profit of RPD_2 is at least $\frac{1}{4\log_2 \chi}\sum_{B_{ni^\star} \in A^\star}(b_{ni^\star} - c_n)$. $\square$

APPENDIX F
PROOF OF LEMMA 7

Proof: Consider a virtual instance of our online problem where the bid $ni^\star$ of user n in the virtual instance has the same demands $d_{ni^\star r}(t)$ over $[t_{ni^\star}^-, t_{ni^\star}^+]$ as in the original instance, but has bidding prices $\min\{b_{ni^\star}, \hat{p}_{ni^\star}\}$ instead of $b_{ni^\star}$. Then, the online auction PD_2 would accept the same set of bids in the virtual instance as in the original instance. The social welfare of PD_2 in the virtual instance is therefore $\sum_{ni^\star \in A^\star}(\hat{p}_{ni^\star} - c_n)$. Further note that choosing all $B_{ni} \in O$ is a feasible solution in the virtual instance and gets social welfare $\sum_{B_{ni} \in O \setminus A^\star}(\min\{b_{ni}, \hat{p}_{ni}\} - c_n)$. The lemma now follows from the competitive ratio of online algorithm PD_2. $\square$

APPENDIX G
PROOF OF THEOREM 11

Proof: The set of accepted bids of the online auction RPD_2 is a subset of that of PD_2. Let $B^\star$ denote the set of accepted bids of RPD_2. Recall $A^\star$ denotes the set of accepted bids of PD_2. Define a random variable as $b'_{ni^\star} = \{b_{ni^\star}, \text{if } b_{ni^\star} \geq \eta\hat{p}_{ni^\star}; 0, \text{otherwise }\}$. The expected social welfare of RPD_2 is: $E\left[\sum_{B_{ni^\star} \in B^\star}(b_{ni^\star} - c_n)\right] = E\left[\sum_{B_{ni^\star} \in A^\star}(b'_{ni^\star} - c_n)\right]$. Note that $A^\star = \{B_{ni^\star} : b_{ni^\star} \geq \hat{p}_{ni^\star}\}$. Thus we have $E\left[\sum_{B_{ni^\star} : b_{ni^\star} \geq \hat{p}_{ni^\star}}(b'_{ni^\star} - c_n)\right] = \sum_{B_{ni^\star} : b_{ni^\star} \geq \hat{p}_{ni^\star}} E\left[b'_{ni^\star} - c_n\right] = \sum_{B_{ni^\star} : b_{ni^\star} \geq \hat{p}_{ni^\star}}(b_{ni^\star} - c_n) \cdot \Pr[b_{ni^\star} \geq \eta\hat{p}_{ni^\star}] \geq \frac{1}{2}\sum_{B_{ni^\star} \in A^\star}(b_{ni^\star} - c_n)$. $\square$

REFERENCES

[1] *Amazon EC2 Instance Types*. Accessed on 2015. [Online]. Available: http://aws.amazon.com/ec2/instance-types/

[2] *ProfitBricks*. Accessed on 2015. [Online]. Available: https://www.profitbricks.com

[3] *CloudSigma*. Accessed on 2015. [Online]. Available: https://www.cloudsigma.com

[4] O. A. Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing amazon EC2 spot instance pricing," in *Proc. IEEE CloudCom*, Nov. 2011, pp. 304–311.

[5] Q. Wang, K. Ren, and X. Meng, "When cloud meets eBay: Towards effective pricing for cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 936–944.

[6] W. Wang, B. Liang, and B. Li, "Revenue maximization with dynamic auctions in IaaS cloud markets," in *Proc. IEEE IWQoS*, Jun. 2013, pp. 1–6.

[7] W. Shi, L. Zhang, C. Wu, Z. Li, and F. C. Lau, "An online auction framework for dynamic resource provisioning in cloud computing," in *Proc. ACM SIGMETRICS*, 2014, pp. 71–83.

[8] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 433–441.

[9] W. Shi, C. Wu, and Z. Li, "RSMOA: A revenue and social welfare maximizing online auction for dynamic cloud resource provisioning," in *Proc. IWQoS*, May 2014, pp. 41–50.

[10] H. Zhang *et al.*, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," in *Proc. IEEE INFOCOM*, Mar. 2016, pp. 805–818.

[11] A. Mu'alem and N. Nisan, "Truthful approximation mechanisms for restricted combinatorial auctions," *Games Econ. Behavior*, vol. 64, no. 2, pp. 612–631, Nov. 2008.

[12] Y. Azar, U. Bhaskar, L. Fleischer, and D. Panigrahi, "Online mixed packing and covering," in *Proc. ACM-SIAM SODA*, 2013, pp. 85–100.

[13] N. R. Devanur and Z. Huang, "Primal dual gives almost optimal energy efficient online algorithms," in *Proc. ACM-SIAM SODA*, 2014, pp. 1123–1140.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[15] N. Buchbinder and J. Naor, "The design of competitive online algorithms via a primal: Dual approach," *Found. Trends Theoretical Comput. Sci.*, vol. 3, nos. 2–3, pp. 93–263, Feb. 2009.

[16] Z. Huang and A. Kim, "Welfare maximization with production costs: A primal dual approach," in *Proc. ACM-SIAM SODA*, 2015, pp. 59–72.

[17] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[18] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 702–710.

[19] X. Wu *et al.*, "Online mechanism design for VMs allocation in private cloud," in *Network Parallel Computing*, vol. 8707. Berlin, Germany: Springer, 2014, pp. 234–246.

[20] L. Mashayekhy, M. M. Nejad, D. Grosu, and A. V. Vasilakos, "An online mechanism for resource allocation and pricing in clouds," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1172–1184, Apr. 2015.

[21] A. Blum, A. Gupta, Y. Mansour, and A. Sharma, "Welfare and profit maximization with production costs," in *Proc. IEEE FOCS*, Oct. 2011, pp. 77–86.

[22] P. Barham *et al.*, "Xen and the art of virtualization," in *Proc. ACM SOSP*, 2003, pp. 164–177.

[23] H. Chen, M. C. Caramanis, and A. K. Coskun, "Reducing the data center electricity costs through participation in smart grid programs," in *Proc. IGCC*, Nov. 2014, pp. 1–10.

[24] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of virtual machines for real-time cloud services," *Concurrency Comput., Pract. Exper.*, vol. 23, no. 13, pp. 1491–1505, 2011.

[25] B. Krishnan, H. Amur, A. Gavrilovska, and K. Schwan, "VM power metering: Feasibility and challenges," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 3, pp. 56–60, 2010.

[26] J. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proc. ACM SoCC*, 2010, pp. 39–50.

[27] N. Buchbinder and J. Naor, "Online primal-dual algorithms for covering and packing problems," in *Algorithms–ESA*. Berlin, Germany: Springer, 2005, pp. 689–701.

[28] B. Awerbuch, Y. Azar, and A. Meyerson, "Reducing truth-telling online mechanisms to online optimization," in *Proc. ACM STOC*, 2003, pp. 503–510.

[29] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: Format + schema," Google Inc., Mountain View, CA, USA, Tech. Rep. ClusterData2011_2, Nov. 2011. [Online]. Available: http://code.google.com/p/googleclusterdata/wiki/TraceVersion2

[30] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. ACM SoCC*, 2012, p. 7.

[31] *Google Data Centers*. Accessed on 2015. [Online]. Available: http://goo.gl/zJvCTH

[32] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. M. Lau, "Online auctions in IaaS clouds: Welfare and profit maximization with server costs," in *Proc. ACM SIGMETRICS*, 2015, pp. 3–15.

[33] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan, "Multi-parameter mechanism design and sequential posted pricing," in *Proc. ACM STOC*, 2010, pp. 311–320.

Xiaoxi Zhang received the B.E. degree from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, in 2013. She is currently pursuing the Ph.D. degree with the Department of Computer Science, The University of Hong Kong. Her research interests include cloud computing, mechanism design, and online algorithms.

Zhiyi Huang received the B.E. degree from the Department of Computer Science and Technology, Tsinghua University, in 2008, and the Ph.D. degree from the Department of Computer and Information Science, University of Pennsylvania, in 2013. He is currently an Assistant Professor of Computer Science with The University of Hong Kong. His research interests include theoretical computer science, algorithmic game theory, differential privacy, and online algorithms.

Chuan Wu received the B.E. and M.E. degrees from the Department of Computer Science and Technology, Tsinghua University, China, in 2000 and 2002, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Toronto, Canada, in 2008. She is currently an Associate Professor with the Department of Computer Science, The University of Hong Kong. Her research interests include cloud computing and online/mobile social network.
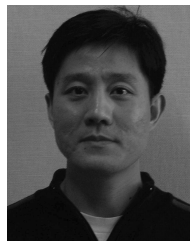
Zongpeng Li received the B.E. degree in computer science and technology from Tsinghua University in 1999, and the M.S. degree in computer science and the Ph.D. degree in electrical and computer engineering from the University of Toronto in 2001 and 2005, respectively. He is currently a Professor with the Department of Computer Science, University of Calgary. His research interests are in computer networks, particularly in network optimization, multicast algorithm design, network game theory, and network coding.

Francis C. M. Lau received the Ph.D. degree from the Department of Computer Science, University of Waterloo. He is currently a Professor in computer science with The University of Hong Kong. His research interests include computer systems, networks, programming languages, and application of computing in arts. He is the Editor-in-Chief of the *Journal of Interconnection Networks*.