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

Xiaoxi Zhang
Dept. of Computer Science
The University of Hong Kong
xxzhang2@cs.hku.hk

Zhiyi Huang
Dept. of Computer Science
The University of Hong Kong
zhiyi@cs.hku.hk

Chuan Wu
Dept. of Computer Science
The University of Hong Kong
cwu@cs.hku.hk

Zongpeng Li
Dept. of Computer Science
University of Calgary
zongpeng@ucalgary.ca

Francis C.M. Lau
Dept. of Computer Science
The Univeristy of Hong Kong
fcmlau@cs.hku.hk

ABSTRACT

Auction design has recently been studied for dynamic resource bundling and VM provisioning in IaaS clouds, but is mostly restricted to the one-shot or offline setting. This work targets a more realistic case of online VM auction design, where: (i) cloud users bid for resources into the future to assemble customized VMs with desired occupation durations; (ii) the cloud provider dynamically packs multiple types of resources on heterogeneous physical machines (servers) into the requested VMs; (iii) the operational costs of servers are considered in resource allocation; (iv) 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.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design studies; Modeling techniques; I.1.2 [**Algorithms**]: Analysis of algorithms

General Terms

Algorithms; Design; Economics

SIGMETRICS'15, June 15–19, 2015, Portland, OR, USA.
Copyright © 2015 ACM 978-1-4503-3486-0/15/06 ...$15.00.
DOI: http://dx.doi.org/10.1145/2745844.2745855.

Keywords

Cloud Computing; Auction; Resource Allocation; Pricing; Online Algorithms; Truthful Mechanisms

1. 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 7 categories and 23 types of VMs [1], a substantial growth from a few years ago. A few 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 policies are still dominant in practice, charging customers a fixed amount for each pre-configured VM [1] or each unit of resources [2][3]. Despite their apparent simplicity, fixed-price policies inherently lack market agility and efficiency, jeopardizing both 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.

Towards better market-based pricing, auctions have recently been designed for cloud resource allocation, for pre-configured VMs of limited types [5][6], or for customized VMs with user-specified resource bundles [7][8][9]. Most of the mechanisms have focused on the *one-shot* or *offline* setting, solving for resource allocation and payments 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.* [10] and Zhang *et al.* [11] consider only one type of VMs available to all users 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 a strong assumption of allowing preemption of resources already occupied 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) User-specified VM (future) start/end time and time-varying resource bundle: Each cloud user bids for a customized bundle of resources to constitute her VM, which can start execution at any future time for any specified duration. Besides, the resource composition of the VM can vary over its duration, according to the projected need of user workload. (II) Heterogeneous servers with various resource capacities and operational costs: 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) Social welfare maximization as well as profit maximization: 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 (another realistic objective), while guaranteeing other desirable properties including truthful bidding, individual rationality and computationally 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 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 (if the VM is to start at a future time) 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 [12]. The classic VCG mechanism, essentially the only type of auction that guarantees both truthfulness and economic efficiency in the offline setting [13], 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. 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 (packing VM requests within resource capacities, and covering accepted requests by producing enough resources and paying server costs) —such problems are known to be more challenging than problems with only packing constraints such as the previous models without server costs [14]. 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. 2 for details).

Our Contributions. This paper leverages a set of latest, novel primal-dual online optimization techniques and randomized reduction techniques, to design a set of truthful, polynomial-time online auctions for social welfare maximization or 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 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 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. 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 real-world 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., [17]), 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 according to the current resource usage levels, which play a key role in achieving truthfulness and good competitive ratio.

Third, we extend our social welfare maximizing auctions to profit maximizing ones using randomized reduction, with minor losses in competitive ratios. To obtain good competitive ratios in terms of profit with super-linear server costs (Sec. 4), we introduce a new online primal-dual analysis for 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 profit of our auctions with the dual objective of the social welfare maximization convex program. To our knowledge, our 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. 2, and define the problem model in Sec. 3. Sec. 4 presents our online auction design under super-linear server cost functions. Sec. 5 discusses the online auctions under linear server cost models. Simulations are presented in Sec. 6. Sec. 7 concludes the paper.

2. RELATED WORK

Allocating pre-configured types of VMs in an IaaS cloud to serve user jobs has been extensively studied with different focuses. Beloglazov *et al.* [18] study energy-efficient allocation algorithms for scheduling VMs to serve computing tasks. Joe-Wong *et al.* [19] seek to balance efficiency and fairness when allocating VMs to users. Maguluri *et al.* [20] investigate stochastic models and algorithms for load balancing and VM allocation to handle randomly arriving workloads. None of them investigates online optimized packing of customized VMs at user-specified resource amounts, which is the focus of this work.

Auction mechanisms have been at the focal point of recent literature on VM pricing. Lin *et al.* [6] propose a second price auction for computing capacity allocation and pricing. Zaman and Grosu [5] study on-demand VM allocation through a truthful auction, and show through experiments that a higher revenue can be achieved for the cloud provider. Their models do not include tailor-made VM assembling. Wang *et al.* [21] model VM allocation and pricing as a multi-unit combinatorial auction, apply the critical value method, and derive a mechanism that is truthful and collusion-resistant. 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 with all bids given, neglecting the dynamical nature of users' demands.

Wang *et al.* [10] model a dynamic auction where bidders may request to occupy a VM for more than one decision interval, and propose an online auction mechanism. With simulations, they show their mechanism to be truthful and asymptotically optimal in provider revenue when demand is sufficiently high. However, the auction model is over-simplified and considers one type of VM only. In the VM auction of Zhang *et al.* [11], the bidding language and the user valuation models capture the heterogeneous demands in a cloud market. However, only a single type of VM is considered, significantly simplifying the underlying social welfare maximization. Shi *et al.* [7] investigate auctions over multiple rounds which are coupled together by the overall budget of each user, while each acquired VM is still used for only one round. We investigate a more practical online 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 by a user. In addition, none of existing cloud auctions consider server costs in the social welfare or provider's profit, which will be 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 [17][22], and online combinatorial auctions with production costs [16][23]. 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.

3. PROBLEM MODEL

3.1 Cloud System and the Auction

We consider an IaaS cloud system consisting of S servers, offering users R types of resources, as exemplified by CPU, RAM, disk storage and bandwidth. We use $[X]$ to denote the set $\{1, 2, ..., X\}$. Each server $s \in [S]$ can provide a maximum amount C_{rs} of resource $r \in [R]$. Cloud users arrive over time, and each requests one or multiple tailor-made VMs 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 auctions.

A user submits a bid for each VM she requests upon her arrival. A total number of I bids are submitted in a large time span $1, 2, \ldots, T$. Let B_i denote the ith bid submitted at t_i (we allow multiple bids to be submitted at the same time and order simultaneous bids randomly). t_i^- is the start time to run the VM if B_i is accepted and t_i^+ is the end time, where $t_i \leq t_i^- < t_i^+$.

Let $d_{ir}(t)$ denote the amount of type-r resource required by bid i at time t. By allowing $d_{ir}(t)$ to resume different values over $t \in [t_i^-, t_i^+]$, we enable each VM to consume a different amount of each type of resource over time. For example, at different stages of a MapReduce job, different CPU and bandwidth capacities are needed [24]. $d_{ir}(t)$'s are specified in the bid based on the projected resource need of the bidder's workload at different times, *e.g.*, according to previous execution of similar workloads. 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 [25][26]. Let b_i be the overall willingness-to-pay submitted in bid i for the tailor-made VM to run between t_i^- and t_i^+, and v_i be the true valuation of the respective bidder. A bid can be expressed as (*bidding language*):

$$B_i = \{t_i^-, t_i^+, \{d_{ir}(t)\}_{r\in[R], t\in[t_i^-, t_i^+]}, b_i\}. \tag{1}$$

Upon receiving each bid, the cloud provider decides whether to accept it, and on which server to provision the requested VM if accepted. A binary variable x_{is} is set to 1 if bid i is accepted with resources allocated on server s, and 0 otherwise. The provider also computes a payment $\hat{p}_i$ for each winning bid i.

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 [27]. The decisions of server provisioning happen at a much larger time scale than those for VM allocation, *e.g.*, Amazon EC2 adjusts its server provisioning roughly once per month, according to discussions with Amazon employees. Therefore, we realistically assume that all S servers are turned on in the span 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 $f_{rs}(\cdot)$ to denote the cost function of server s on the amount of type-r resource used on the server, as indicated by $y_{rs}(t)$. The cost function is defined as follows:

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

Parameter $\beta_{rs} \geq 0$ decides the shape of the cost function, according to different operational models of the server. 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 [28]. When the CPU voltage is elevated with the utilization, the CPU power usage renders a cubic increase with the CPU voltage [29], and hence we can approximately use $\beta_{rs} = 2$ 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 [30], and hence we set $\beta_{rs} = 0$ in this case. h_{rs} indicates the relative weight of the cost due to each type of resource in the overall server cost. 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 [31], and the power usage due to different resources is additive [30], confirming our additive model of the costs due to different resources.

3.2 Mechanism Design Goals

We target the following properties in our auction design.

(i) *Truthfulness*: For any bidder, declaring her true valuation of the VM and true information (*e.g.*, bid arrival time) in her bid always maximizes her 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.*,

$$\sum_{i\in[I]} \hat{p}_i \sum_{s\in[S]} x_{is} - \sum_{t\in[T]} \sum_{s\in[S]} \sum_{r\in[R]} f_{rs}(y_{rs}(t)). \quad (3)$$

Bid i's utility is $v_i - \hat{p}_i$. The social welfare over system span T is the sum of the provider's profit and the bidders' aggregate utility, $\sum_{i\in[I]}(v_i - \hat{p}_i)\sum_{s\in[S]} x_{is}$ (valuation minus payment of all winning bids), which equals the aggregate valuation of the winning bids minus the server costs,

$$\sum_{i\in[I]} v_i \sum_{s\in[S]} x_{is} - \sum_{t\in[T]} \sum_{s\in[S]} \sum_{r\in[R]} f_{rs}(y_{rs}(t)),$$

and

$$\sum_{i\in[I]} b_i \sum_{s\in[S]} x_{is} - \sum_{t\in[T]} \sum_{s\in[S]} \sum_{r\in[R]} f_{rs}(y_{rs}(t))$$

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 I bids within system span 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).

$$\text{maximize} \quad \sum_{i\in[I]} \sum_{s\in[S]} b_i x_{is} - \sum_{t\in[T]} \sum_{s\in[S]} \sum_{r\in[R]} f_{rs}(y_{rs}(t)) \quad (4)$$

subject to:

$$\sum_{s\in[S]} x_{is} \leq 1, \quad \forall i \in [I] \; (4a)$$

$$\sum_{\substack{i\in[I]:\\ t_i^- \leq t \leq t_i^+}} d_{ir}(t) x_{is} \leq y_{rs}(t), \quad \forall r \in [R], s \in [S], t \in [T] \; (4b)$$

$$x_{is} \in \{0,1\}, y_{rs}(t) \geq 0, \quad \forall r \in [R], s \in [S], i \in [I], t \in [T] \; (4c)$$

Here, constraint (4a) indicates that each requested VM is provisioned on at most one server. (4b) sums up the amount of type-r resource needed by all accepted bids on server s at t (counting only bids whose VMs are running at t) into $y_{rs}(t)$. Recall the definition of the cost function in (2): by setting $f_{rs}(y_{rs}(t)) = +\infty$ when $y_{rs}(t)$ exceeds C_{rs} (the overall capacity of type-r resource on server s), 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 $x_{is} \in \{0,1\}$ to $x_{is} \geq 0$ (constraint (4a) guarantees $x_{is} \leq 1$), and apply Fenchel duality [32] 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.

Table 1: Notation

I	# of bids	T	# of time slots
R	# of resource types	S	# of servers
t_i	arrival time of bid i	t_i^- (t_i^+)	start (end) time of bid i

$d_{ir}(t)$	demand of resource r at t of bid i
C_{rs}	capacity of resource r of server s
x_{is}	serve bid i on server s (1) or not (0)
α_1, $2\alpha_2$	competitive ratios of algorithms PD$_1$ and PD$_2$
U_r	$\max_{i\in[I]} b_i / \sum_{t\in[t_i^-, t_i^+]} d_{ir}(t)$
L_r	$\min_{i\in[I]} b_i / \sum_{t\in[t_i^-, t_i^+]} d_{ir}(t)$
$y_{rs}(t)$	amount of allocated resource r on s at t
u_i	utility of bid i
$p_{rs}(t)$	marginal payment of type-r resource on s at t
$f_{rs}(y_{rs}(t))$	cost function of type-r resource on s
$f^*_{rs}(p_{rs}(t))$	conjugate of $f_{rs}(\cdot)$
$\hat{p}_i$	payment of bid i in PD$_1$ and PD$_2$

Let u_i and $p_{rs}(t)$ be the dual variables associated with (4a) and (4b), respectively. The Fenchel dual [32] of the relaxed convex program is as follows:

$$\text{minimize} \quad \sum_{i\in[I]} u_i + \sum_{t\in[T]} \sum_{r\in[R]} \sum_{s\in[S]} f^*_{rs}(p_{rs}(t)) \quad (5)$$

subject to:

$$u_i \geq b_i - \sum_{t\in[t_i^-, t_i^+]} \sum_{r\in[R]} d_{ir}(t) p_{rs}(t), \quad \forall s \in [S], i \in [I] \; (5a)$$

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

$$u_i \geq 0, \quad \forall i \in [I] \; (5c)$$

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

$$f^\star_{rs}(p_{rs}(t)) = \sup_{y_{rs}(t)\geq 0} \{p_{rs}(t) y_{rs}(t) - f_{rs}(y_{rs}(t))\} \quad (6)$$

PROPOSITION 1. *The conjugate of $f_{rs}(y_{rs}(t))$ defined in (2) is*

$$f^*_{rs}(p_{rs}(t)) = \begin{cases} (\frac{p_{rs}(t)}{1+\beta_{rs}})^{\frac{1+\beta_{rs}}{\beta_{rs}}} \frac{\beta_{rs}}{(h_{rs})^{\frac{1}{\beta_{rs}}}}, & y^0_{rs}(t) \leq C_{rs} \\ C_{rs} p_{rs}(t) - h_{rs}(C_{rs})^{1+\beta_{rs}}, & y^0_{rs}(t) > C_{rs} \end{cases} \quad (7)$$

where $y^0_{rs}(t) = (\frac{p_{rs}(t)}{h_{rs}(1+\beta_{rs})})^{\frac{1}{\beta_{rs}}}$.

See Appendix A for the proof.

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 bid i, there is a set of new primal variables x_{is} for $s \in [S]$ subject to $\sum_{s\in[S]} x_{is} \leq 1$. The cloud provider must decide immediately whether to serve bid i and on which server, as well as the bidder's payment if served. In the following, we design online primal-dual allocation algorithms and payment schemes based on (4) and (5).

4. ONLINE AUCTIONS FOR SOCIAL WELFARE AND PROFIT MAXIMIZATION

In this section, we focus on the case of superlinear server cost functions, *i.e.*, $h_{rs} > 0$ and $\beta_{rs} > 0$ in (2), and design online auctions for social welfare maximization (Sec. 4.1) and profit maximization (Sec. 4.2). We will discuss the case of linear server cost functions (with zero server cost as a special case) in Sec. 5.

4.1 Online Auction for Social Welfare Maximization

4.1.1 Auction Design

Deciding whether to serve a new bid i and on which server is equivalent to choosing a feasible assignment for the new primal variables x_{is}. If the cloud provider decides to serve bid i on some server s_i, then let $x_{is_i} = 1$, and increase the amount of allocated resources $y_{rs_i}(t)$ by $d_{ir}(t)$ on server s_i for all resources $r \in [R]$ and for all time slots $t \in [t_i^-, t_i^+]$. As a result, the total server cost of s_i, $\sum_{t\in[T]}\sum_{r\in[R]} f_{rs_i}(y_{rs_i}(t))$, increases accordingly. Otherwise, x_{is} will be zero for all servers $s \in [S]$.

VM Allocation. The question is how to decide whether to serve bid i and on which server. For this we will look into the dual program and resort to the KKT conditions [33]. With bid i, there is a new dual variable $u_i \geq 0$ subject to constraints (5a), that is, $u_i \geq b_i - \sum_{t\in[t_i^-,t_i^+]}\sum_{r\in[R]} d_{ir}(t)p_{rs}(t)$ for all $s \in [S]$. The KKT conditions indicate that in the offline primal and dual solutions to (4) and (5), x_{is} must be zero unless constraint (5a) is tight for server s. Thus, we let u_i be the maximal of 0 and the right hand side (RHS) of constraints (5a), that is,

$$u_i = \max\left\{0, \max_{s\in[S]}\left\{b_i - \sum_{t\in[t_i^-,t_i^+]}\sum_{r\in[R]} d_{ir}(t)p_{rs}(t)\right\}\right\}. \tag{8}$$

Accordingly, we adopt the following method to decide whether to accept a bid and on which server: the cloud provider does not serve bid i, *i.e.*, $x_{is} = 0$ for all $s \in [S]$, if no server s achieves a nonnegative value on the RHS of (5a) (*i.e.*, $u_i = 0$), and otherwise serves bid i on the server s_i that maximizes the RHS (*i.e.*, $x_{is_i} = 1$, and $x_{is} = 0, \forall s \neq s_i$).

The rationale is as follows: If we interpret $p_{rs}(t)$ as the marginal price (a.k.a. payment) per unit of type-r resource on server s at time t, then the second term on the RHS of (5a) becomes the total payment that bid i should pay for the requested resources, *i.e.*,

$$\hat{p}_i = \sum_{t\in[t_i^-,t_i^+]}\sum_{r\in[R]} d_{ir}(t)p_{rs}(t). \tag{9}$$

So the RHS of (5a) for a given s is the utility of bid i if it would be served on server s (valuation minus payment, assuming truthful bidding). Therefore, the above method effectively assigns bid i to the server that maximizes its utility, and u_i is bid i's utility. In this way, we target utility maximization for each bid, which leads to truthfulness and social welfare maximization.

Payment Design. Furthermore, the cloud provider updates the marginal prices $p_{rs}(t)$ as the amounts of allocated resources $y_{rs}(t)$ change, after calculating payment $\hat{p}_i$ of accepted bid i using (9), to ensure that: (i) the gain in social welfare when a bid is served outweighs the 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. Indeed, designing the online pricing rules is the key to obtain a good competitive ratio in social welfare, as compared to the offline optimum.

Again, our method of pricing is driven by the structure of the offline primal and dual solutions. Let $\ddot{y}_{rs}(t)$ be total demand of resource r on server s at time t, and $\ddot{p}_{rs}(t)$ be the respective marginal prices in the offline primal and dual solutions. If $\ddot{y}_{rs}(t)$ is smaller than the capacity C_{rs}, the marginal price $\ddot{p}_{rs}(t)$ shall be equal to the marginal cost per unit of the resource, $f'_{rs}(\ddot{y}_{rs}(t))$; 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.

Algorithm 1: Primal-Dual Online Auction PD$_1$

```
Input: $S$, $R$, $\mathbf{C}$, $\mathbf{h}$, $\beta$
Output: $\mathbf{x}, \mathbf{p}$
DEFINE $f_{rs}(y_{rs}(t))$ according to (2), $\forall s \in [S], r \in [R]$;
DEFINE $p_{rs}(y_{rs}(t))$ according to (10), $\forall s \in [S], r \in [R]$;
INITIALIZE $x_{is} = 0, y_{rs}(t) = 0, u_i = 0, p_{rs}(t) = 0, \forall i \in [I], s \in [S], r \in [R], t \in [T]$;
Upon the arrival of the $i$th bid
Get bidding language $B_i$ according to (1);
$(\mathbf{x}_i, \hat{p}_i, \mathbf{p}, \mathbf{y})$ = CORE$(S, R, B_i, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
if $\exists s \in [S], x_{is} = 1$ then
    Accept the $i$th bid and allocate resources on server $s$ to provision the VM requested in bid $i$;
    Charge the $i$th bid at $\hat{p}_i$;
else
    Reject the $i$th bid;
end
```

Algorithm 2: One-Round Auction Algorithm CORE

```
Input: $S, R, B_i, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y})$
Output: $\mathbf{x}_i, \hat{p}_i, \mathbf{p}, \mathbf{y}$
Update the dual variable $u_i$ (utility) according to (8);
if $u_i > 0$ then
    $s_i = \mathrm{argmax}_{s\in[S]}\{b_i - \sum_{t\in[t_i^-,t_i^+]}\sum_{r\in[R]} d_{ir}(t)p_{rs}(t)\}$;
    Calculate the payment of the $i$th bid:
    $\hat{p}_i = \sum_{t\in[t_i^-,t_i^+]}\sum_{r\in[R]} d_{ir}(t)p_{rs_i}(t)$.;
    Update the primal variables:
    $x_{is_i} = 1$, and $x_{is} = 0$ for all $s \neq s_i$;
    $y_{rs_i}(t) = y_{rs_i}(t) + d_{ir}(t) \quad \forall r \in [R], t \in [t_i^-, t_i^+]$;
    Update the dual marginal price variables:
    $p_{rs_i}(t) = p_{rs_i}(y_{rs_i}(t)) \quad \forall r \in [R], t \in [t_i^-, t_i^+]$;
else
    $x_{is} = 0$ for all $s \in [S]$;
end
```

In this online setting, however, the cloud provider can see only the current demands of resources, not the final demands. Therefore, it will predict 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 to be δ_{rs} times its current demand if the predicted final demand does not exceed the capacity, for a fixed parameter $\delta_{rs} > 1$ to be determined later, and set the marginal price to be $p_{rs}(t) = f'_{rs}(\delta_{rs}y_{rs}(t))$. 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., [34]), and let the marginal price increase exponentially in the current demand, *i.e.*, $p_{rs}(t) = f'_{rs}(C_{rs})\exp(\theta_{rs}(y_{rs}(t) - C_{rs}/\delta_{rs}))$, where θ_{rs} is a parameter to be determined later, if the predicted final demand exceeds the capacity. In summary, the marginal price for each resource on each server at each time, $p_{rs}(t)$, is defined to be a function on $y_{rs}(t)$ (for all $r \in [R]$, $s \in [S]$, and $t \in [T]$):

$$p_{rs}(y_{rs}(t)) = \begin{cases} f'_{rs}(\delta_{rs}y_{rs}(t)), & y_{rs}(t) \leq \frac{C_{rs}}{\delta_{rs}} \\ f'_{rs}(C_{rs})e^{\theta_{rs}(y_{rs}(t) - \frac{C_{rs}}{\delta_{rs}})}, & y_{rs}(t) > \frac{C_{rs}}{\delta_{rs}} \end{cases} \tag{10}$$

We will show in Sec. 4.1.2 that such a price function leads to 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, with the one-round algorithm CORE given in Alg. 2. Upon arrival of each bid, we calculate the payment for an accepted bid by summing up the product of resource demand and current marginal price on the selected server s_i to serve bid i for all resources over the occupation period (line 5, Alg. 2). We update primal variables by setting x_{is_i} to 1 (line 7), and increase the utilization $y_{rs_i}(t)$ for different resources for future time slots $t \in [t_i^-, t_i^+]$, according to the demand of bid i (line 8). We update the dual marginal price variables $p_{rs_i}(t)$'s according to (10) (line 10). Recycling of resources is implicitly handled by increasing $y_{rs}(t)$'s with bid i's resource demand, only for time slots within specified duration $[t_i^-, t_i^+]$.

4.1.2 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 $O(RST)$ time.*

The detailed proof can be found in Appendix B.

(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 T. We start by introducing an online primal-dual analysis framework, which bounds the ratio according to a bound between the increase of primal objective value and the increase of dual objective value at each step of the online algorithm.

Let $P^0 = 0$ and D^0 be the initial values of the primal and dual objectives. Let P^i and D^i denote the primal and dual objective values after handling bid i. $y_{rs}^i(t)$ denotes the amount of allocated type-r resource on server s after handling bid i, and $p_{rs}^i(t)$ denotes the corresponding marginal price. Note that P^I and D^I are the primal and dual objective values at the end of 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., *$P^i - P^{i-1} \geq \frac{1}{\alpha}(D^i - D^{i-1})$, for every step i, 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 i, we have

$$P^I = \sum_i (P^i - P^{i-1}) \geq \frac{1}{\alpha}\sum_i (D^i - D^{i-1}) = \frac{1}{\alpha}(D^I - D^0).$$

Here, we use the fact that $P^0 = 0$. By weak duality [33], $D^I \geq OPT$. Further by our assumption that $D^0 \leq \frac{1}{2}OPT$, we have $P^I \geq \frac{1}{2\alpha}OPT$. So the algorithm is 2α competitive. □

In fact, the initial dual value of our algorithm PD$_1$ is $D^0 = 0$. According to the above proof, we can show that our algorithm is α (instead of 2α) competitive, if we can find the smallest $\alpha \geq 1$ such that $P^i - P^{i-1} \geq \frac{1}{\alpha}(D^i - D^{i-1})$ for all steps i, since it achieves $P^I = \frac{1}{\alpha}(D^I - D^0) = \frac{1}{\alpha}D^I \geq \frac{1}{\alpha}OPT$.

The key to identify this α is to show that our constructed marginal pricing function $p_{rs}(t)$ 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*

$$p_{rs}^{i-1}(t)\big(y_{rs}^i(t) - y_{rs}^{i-1}(t)\big) - \big(f_{rs}(y_{rs}^i(t)) - f_{rs}(y_{rs}^{i-1}(t))\big) \\ \geq \frac{1}{\alpha}\big(f_{rs}^\star(p_{rs}^i(t)) - f_{rs}^\star(p_{rs}^{i-1}(t))\big), \quad (11)$$

for all $i \in [I]$, $r \in [R]$, $s \in [S]$, and $t \in [t_i^-, t_i^+]$.

The *Allocation-Payment Relationship* shows that the difference between payment for type-r resource on server s (according to the old price before handling i) and the incremental cost of server s due to bid i's additional use of resource r, is no smaller than $\frac{1}{\alpha}$ of the value increase of the conjugate $f_{rs}^\star$ due to the adjustment of the marginal price. Since the payment of bid i according to the adjusted price is no smaller than $p_{rs}^{i-1}(t)d_{ir}(t)$, 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 net profit is lower bounded by a constant fraction of the increase in the dual objective due to the increase of dual prices $p_{rs}(t)$.

LEMMA 2. *If the Allocation-Payment Relationship holds for $\alpha \geq 1$, then for every accepted bid i and the corresponding server s_i*

$$\hat{p}_i - \sum_{t\in[t_i^-, t_i^+]}\sum_{r\in[R]}\big(f_{rs_i}(y_{rs_i}^i(t)) - f_{rs_i}(y_{rs_i}^{i-1}(t))\big) \\ \geq \frac{1}{\alpha}(D^i - D^{i-1} - u_i)$$

PROOF. Recall that $\hat{p}_i = \sum_{t\in[t_i^-, t_i^+]}\sum_{r\in[R]} d_{ir}(t)p_{rs_i}^{i-1}(t) = \sum_{t\in[t_i^-, t_i^+]}\sum_{r\in[R]}\big(y_{rs}^i(t) - y_{rs}^{i-1}(t)\big)\,p_{rs_i}^{i-1}(t)$ and that

$$D^i - D^{i-1} = u_i + \sum_{t\in[t_i^-, t_i^+]}\sum_{r\in[R]}\big(f_{rs_i}^\star(p_{rs_i}^i(t)) - f_{rs_i}^\star(p_{rs_i}^{i-1}(t))\big)$$

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

THEOREM 2. *If the Allocation-Payment Relationship holds for $\alpha \geq 1$, then $P^i - P^{i-1} \geq \frac{1}{\alpha}(D^i - D^{i-1})$ for all $i \in [I]$.*

PROOF. If bid i is rejected, then $P^i - P^{i-1} = D^i - D^{i-1} = 0$. Next, assume bid i is accepted and let s_i be the server to which bid i is allocated ($x_{is_i} = 1$). The change of primal objective is

$$P^i - P^{i-1} = b_i - \sum_{t\in[t_i^-, t_i^+]}\sum_{r\in[R]}\big(f_{rs_i}(y_{rs_i}^i(t)) - f_{rs_i}(y_{rs_i}^{i-1}(t))\big)$$

Note that $b_i = u_i + \hat{p}_i$. By Lemma 2, we get that

$$P^i - P^{i-1} \geq u_i + \frac{1}{\alpha}(D^i - D^{i-1} - u_i)$$

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

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 $\alpha_{rs} \geq 1$ for each pair of r and s, such that the marginal pricing functions on r and s satisfy the *Differential Allocation-Payment Relationship* decided by this α_{rs} (Definition 2). We can then take α as the maximum of all α_{rs}'s, *i.e.*, $\alpha = \max_{r\in[R],s\in[S]}\alpha_{rs}$, to satisfy (11) for all r and s.

Without loss of generality, we assume in the following discussions that the demands $d_{ir}(t)$ are much smaller compared to the server capacity C_{rs}. Then we can derive a differential version of the Allocation-Payment Relation (11) based on $d_{ir}(t) = \mathrm{d}y_{rs}(t)$ for all $s \in [S], t \in [t_i^-, t_i^+]$, as follows:

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

$$p_{rs}(t)\mathrm{d}y_{rs}(t) - f_{rs}'(y_{rs}(t))\mathrm{d}y_{rs}(t) \geq \frac{1}{\alpha_{rs}}{f_{rs}^\star}'(p_{rs}(t))\mathrm{d}p_{rs}(t), \\ \forall i \in [I], r \in [R], s \in [S], t \in [t_i^-, t_i^+]. \quad (12)$$

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

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

with parameters

$$\begin{aligned}\delta_{rs} &= \max\left\{2, (1+\beta_{rs})^{\frac{1}{\beta_{rs}}}\right\},\\ \theta_{rs} &= \max\left\{\tfrac{\delta_{rs}}{C_{rs}}\beta_{rs}, \tfrac{\delta_{rs}}{C_{rs}(\delta_{rs}-1)}\ln\left(\tfrac{U_r}{h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}}}\right)\right\},\end{aligned}$$

where
$$U_r = \max_{i\in[I]} \frac{b_i}{\sum_{t\in[t_i^-,t_i^+]} d_{ir}(t)}$$

is the maximum value per unit of resource r per unit of time.

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

LEMMA 4. *For $y_{rs}(t) \in [0, C_{rs}]$ and the corresponding $p_{rs}(t)$,*

$$p_{rs}(t) \geq (1+\beta_{rs})f'_{rs}\big(y_{rs}(t)\big) = h_{rs}(1+\beta_{rs})^2 y_{rs}(t)^{\beta_{rs}} \quad ,$$

for the parameters in Lemma 3.

The proofs of Lemma 4 and Lemma 3 are given in Appendices C and D. 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*

$$\alpha_1 = \max\nolimits_{r\in[R],s\in[S]} \alpha_{rs}$$

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 $\mathrm{d}y_{rs_i} = y^i_{rs_i}(t) - y^{i-1}_{rs_i}(t) = d_{ir}(t)$ is very small compared to the capacity (and that $\mathrm{d}y_{rs}(t) = y^i_{rs}(t) - y^{i-1}_{rs}(t) = 0$ for $s \neq s_i$), we get that

$$f_{rs}\big(y^i_{rs}(t)\big) - f_{rs}\big(y^{i-1}_{rs}(t)\big) = f'_{rs}\big(y^{i-1}_{rs}(t)\big)\big(y^i_{rs}(t) - y^{i-1}_{rs}(t)\big)$$

$$f^\star_{rs}\big(p^i_{rs}(t)\big) - f^\star_{rs}\big(p^{i-1}_{rs}(t)\big) = f^{\star\prime}_{rs}\big(p^{i-1}_{rs}(t)\big)\big(p^i_{rs}(t) - p^{i-1}_{rs}(t)\big) \quad .$$

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

4.2 Online Auction for Profit Maximization

Next, we present an online auction for profit maximization based on the social welfare maximizing online auction PD_1, inspired by a randomized reduction technique [35]. While the idea of increasing the payment by a randomly chosen power-of-2 factor is similar to [35], the analysis of [35] 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.* [23]. 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 [23] incurs an additive loss in profit (other than the multiplicative competitive ratio) while our technique does not.

4.2.1 Auction Design

We first introduce a few parameters: $\beta_{\min}$ is the minimum of β_{rs}; 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.

Algorithm 3: Randomized Online Auction RPD_1

```
Input: $S$, $R$, $\mathbf{C}$, $\mathbf{h}$, $\beta$, $L_r$, $U_r$
Output: $\mathbf{x}$, $\mathbf{p}$
DEFINE $\chi = \max_{r\in[R]} \frac{RU_r}{L_r}$;
INITIALIZE $x_{is} = 0, y_{rs}(t) = 0, u_i = 0, p_{rs}(t) = 0, \forall i \in [I], s \in [S], r \in [R], t \in [T]$;

Upon the arrival of the $i$th bid
Get bidding language $B_i$ according to (1);
$(\mathbf{x}_i, \hat{p}_i, \mathbf{p}, \mathbf{y}) = \mathrm{CORE}(S, R, B_i, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
if $\exists s \in [S], x_{is} = 1$ then
    Update
        $\tilde{p}_i = \max\left\{\sum_{r\in[R]}\sum_{t\in[t_i^-,t_i^+]} d_{ir}(t)\frac{L_r}{R}, \hat{p}_i\right\}$
    Generate a random number $\eta$ such that:
        $\eta = \begin{cases}1 & \text{with prob. } \frac{1}{2}\\ 2^j & \text{with prob. } \frac{1}{2\log_2\chi} \text{ for } j = 1,\ldots,\log_2\chi\end{cases}$    (13)
    if $b_i \geq \eta\tilde{p}_i$ then
        Accept the $i$th bid, allocate the VM on server $s$, and charge $\eta\tilde{p}_i$;
    else
        Reject the $i$th bid;
    end
else
    Reject the $i$th bid;
end
```

$$\beta_{\min} = \min_{s\in[S],r\in[R]} \beta_{rs}$$

$$L_r = \min_{i\in[I]} \frac{b_i}{\sum_{t\in[t_i^-,t_i^+]} d_{ir}(t)} \qquad U_r = \max_{i\in[I]} \frac{b_i}{\sum_{t\in[t_i^-,t_i^+]} d_{ir}(t)} \quad (14)$$

(assuming truthful bidding such that $v_i = b_i$).

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. 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 ith bid, could be much smaller than its true value b_i, 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 the ith bid is at least

$$b_i \geq \sum\nolimits_{t\in[t_i^-,t_i^+]} d_{ir}(t)L_r \quad .$$

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

$$b_i \geq \sum\nolimits_{r\in[R]}\sum\nolimits_{t\in[t_i^-,t_i^+]} d_{ir}(t)\tfrac{L_r}{R}$$

In light of this observation, RPD_1 raises the tentative payment to

$$\tilde{p}_i = \max\left\{\sum\nolimits_{r\in[R]}\sum\nolimits_{t\in[t_i^-,t_i^+]} d_{ir}(t)\tfrac{L_r}{R}, \hat{p}_i\right\}$$

The above payment can still be very far from the true value v_i. Hence, RPD$_1$ further multiply $\tilde{p}_i$ by a randomly chosen power of 2, denoted as η, which is drawn from a carefully chosen distribution (*i.e.*, (13)). By doing so, the final payment is within a factor of 2 (and lower than) the true value v_i 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.

4.2.2 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* $O(RST)$ *time.*

The proof is given in Appendix E.

(ii) Competitiveness in 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,* $\chi = \max_{r\in[R]} \frac{RU_r}{L_r}$ *with* U_r *and* L_r *defined in* (14).

PROOF. Let R^i denote the expected profit that RPD$_1$ generates from the first i bids ($R^0 = 0$). We will use an analysis similar to the online primal dual approach, and show that for any $i \in [I]$,

$$O\big(\alpha_1 + \log_2 \chi\big) \cdot (R^i - R^{i-1}) \geq D^i - D^{i-1} \qquad (15)$$

where D_i is dual objective of primal dual auction PD$_1$ after handling the first i bids. The theorem follows because the expected profit of RPD$_1$ at the end of the instance satisfies $O\big(\alpha_1 + \log_2 \chi\big) R^I \geq D^I - D^0 \geq OPT$, where the last inequality holds because $D^I \geq OPT$ (weak duality) and $D^0 = 0$ (definition of PD$_1$).

It remains to prove (15). If bid i is not accepted by PD$_1$, it is not accepted by RPD$_1$ either. Therefore, both sides of (15) are zero.

Next, suppose bid i is accepted by PD$_1$. Let

$$c_i = \textstyle\sum_{t\in[t_i^-, t_i^+]} \sum_{r\in[R]} f'_{rs_i}\big(y_{rs_i}(t)\big) \mathrm{d}y_{rs_i}(t) \qquad (16)$$

denote the increase of server costs if bid i is accepted. Then,

$$\begin{aligned} & R^i - R^{i-1} \\ &= \Pr[\eta = 1]\big(\tilde{p}_i - c_i\big) + \sum_{j=1}^{\log_2 \chi} \Pr[\eta = 2^j] \mathbf{1}_{b_i \geq 2^j \tilde{p}_i} \cdot 2^j \big(\tilde{p}_i - c_i\big) \\ &= \tfrac{1}{2}\big(\tilde{p}_i - c_i\big) + \tfrac{1}{2\log_2 \chi} \textstyle\sum_{j=1}^{\log_2 \chi} \mathbf{1}_{b_i \geq 2^j \tilde{p}_i} \cdot \big(2^j \tilde{p}_i - c_i\big) \qquad (17) \end{aligned}$$

where $\mathbf{1}_{b_i \geq 2^j \tilde{p}_i}$ equals 1 if $b_i \geq 2^j \tilde{p}_i$, and equals 0 otherwise.

By Lemma 2, 3 (recall the definition of c_i in (16)), we have

$$\tilde{p}_i - c_i \geq \hat{p}_i - c_i \geq \tfrac{1}{\alpha_1}(D^i - D^{i-1} - u_i) \qquad (18)$$

So the first term of (17) alone is almost sufficient for showing (15) modulo the u_i term.

The rest of the proof is divided into two cases depending on whether $b_i < 2\tilde{p}_i$ (only the first term in (17) is non-zero, but b_i and, thus, u_i are small), or not (b_i and, thus, u_i are large, but we get positive contribution from the second term of (17)).

Case 1: $b_i < 2\tilde{p}_i$. Note that

$$\tilde{p}_i \geq \hat{p}_i = \textstyle\sum_{r\in[R]} \sum_{t\in[t_i^-, t_i^+]} p_{rs_i}(t) \mathrm{d}y_{rs_i}(t) \ .$$

By Lemma 4, $\tilde{p}_i$ is at least

$$\textstyle\sum_{r\in[R]} \sum_{t\in[t_i^-, t_i^+]} (1+\beta_{rs}) f'_{rs_i}\big(y_{rs}(t)\big) \mathrm{d}y_{rs_i}(t) \geq (1+\beta_{\min}) c_i$$

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

$$\begin{aligned} & R^i - R^{i-1} \\ &= \tfrac{1}{2}\big(\tilde{p}_i - c_i\big) \\ &\geq \tfrac{1}{6}\big(\hat{p}_i - c_i\big) + \tfrac{1}{3}\big(\tilde{p}_i - c_i\big) && (\tilde{p}_i \geq \hat{p}_i) \\ &\geq \tfrac{1}{6}\big(\hat{p}_i - c_i\big) + \tfrac{\beta_{\min}}{3(1+\beta_{\min})} \tilde{p}_i && (\tilde{p}_i \geq (1+\beta_{\min}) c_i) \\ &\geq \tfrac{1}{6}\big(\hat{p}_i - c_i\big) + \tfrac{\beta_{\min}}{6(1+\beta_{\min})} u_i && (u_i \leq b_i < 2\tilde{p}_i) \\ &\geq \tfrac{1}{6\alpha_1}(D^i - D^{i-1} - u_i) + \tfrac{\beta_{\min}}{6(1+\beta_{\min})} u_i && \text{(by (18))} \end{aligned}$$

By definition, $\alpha_{rs} \geq 1 + \frac{1}{\beta_{rs}}$. Recall $\alpha_1 = \max_{r\in[R], s\in[S]} \alpha_{rs}$. So $\alpha_1 \geq 1 + \frac{1}{\beta_{\min}}$. Putting together, (15) follows because

$$R^i - R^{i-1} \geq \tfrac{1}{6\alpha_1}(D^i - D^{i-1} - u_i) + \tfrac{\beta_{\min}}{6} u_i \geq \tfrac{1}{6\alpha_1}(D^i - D^{i-1})$$

Case 2: $b_i \geq 2\tilde{p}_i$. On one hand, by our choice of $\tilde{p}_i$, it is at least

$$\tilde{p}_i \geq \textstyle\sum_{r\in[R]} \sum_{t\in[t_i^-, t_i^+]} d_{ir}(t) \frac{L_r}{R} \ .$$

On the other hand, by the definition of U_r, the true value is at most

$$b_i \leq \textstyle\sum_{r\in[R]} \sum_{t\in[t_i^-, t_i^+]} d_{ir}(t) U_r$$

So $\frac{b_i}{\tilde{p}_i} \leq \frac{RU_r}{L_r} \leq \chi$. Since η is a randomly chosen power of 2 between 1 and χ, with probability $\frac{1}{2\log_2 \chi}$, η satisfies $\frac{1}{2} b_i \leq \eta \tilde{p}_i \leq b_i$ and, thus, RPD$_1$ accepts bid i and obtains profit at least $\frac{1}{2} b_i$. Therefore, $R^i - R^{i-1} \geq \frac{1}{2}(\tilde{p}_i - c_i) + \frac{1}{2\log_2 \chi} \frac{1}{2} b_i$ (recall (17)). By (18), the first term is at least $\frac{1}{2\alpha_1}(D^i - D^{i-1} - u_i)$. 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

$$\begin{aligned} R^i - R^{i-1} &\geq \tfrac{1}{2\alpha_1}(D^i - D^{i-1} - u_i) + \tfrac{1}{4\log_2 \chi} u_i \\ &\geq \tfrac{1}{2\alpha_1 + 4\log_2 \chi}(D^i - D^{i-1} - u_i) + \tfrac{1}{2\alpha_1 + 4\log_2 \chi} u_i \\ &= \tfrac{1}{2\alpha_1 + 4\log_2 \chi}(D^i - D^{i-1}) \end{aligned}$$

So the theorem holds in case 2 as well. □

(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$_1$. By the definition if RPD$_1$, it will tentatively accept the bid and then reexamine it with randomly chosen boosted payment. With probability $\frac{1}{2}$, $\eta = 1$. Further note that when $\eta = 1$, $b_i \geq \eta \tilde{p}_i$ and, thus, RPD$_1$ will accept bid i. 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$_1$. □

5. EXTENSION TO LINEAR SERVER COSTS

We extend the auction design in Sec. 4 to the setting of linear server cost functions, *i.e.*, $\beta_{rs} = 0$ in (2) such that

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

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

Algorithm 4: Primal-Dual Online Auction PD$_2$

```
**Input**: $S$, $R$, $\mathbf{C}$, $\mathbf{h}$, $\sigma$
**Output**: $\mathbf{x}, \mathbf{p}$
DEFINE $f_{rs}(y_{rs}(t))$ according to (19);
DEFINE $p_{rs}(y_{rs}(t))$ according to (21);
INITIALIZE $x_{is} = 0, y_{rs}(t) = 0, u_i = 0, p_{rs}(t) = \frac{L_r - h_{rs}}{2\sigma RS} + h_{rs}, \forall i \in [I], s \in [S], r \in [R], t \in [T]$;
**Upon the arrival of the $i$th bid**
Get *bidding lauguage* $B_i$ according to (1);
$(\mathbf{x}_i, \hat{p}_i, \mathbf{p}, \mathbf{y})$ = CORE$(S, R, B_i, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
**if** $\exists s \in [S], x_{is} = 1$ **then**
    Accept the $i$th bid on server $s$, and charge it at $\hat{p}_i$;
**else**
    Reject the $i$th bid;
**end**
```

PROPOSITION 2. *The conjugate of $f_{rs}(y_{rs}(t))$ defined in* (19) *is*

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

The proof can be found in Appendix F.

5.1 Social Welfare Maximization

We adapt online auction PD$_1$ to online auction PD$_2$ for linear cost functions, as given in Alg. 4. Among the input, $\sigma > 0$ is a parameter such that $\frac{T}{\sigma}$ is the minimum resource occupation time span among all bids (which can be an estimated lower bound). We define a new marginal payment function:

$$p_{rs}(y_{rs}(t)) = \tfrac{L_r - h_{rs}}{2\sigma RS}\big(\tfrac{2\sigma RS(U_r - h_{rs})}{L_r - h_{rs}}\big)^{\frac{y_{rs}(t)}{C_{rs}}} + h_{rs} \ , \qquad (21)$$

where L_r and U_r are defined in (14). For any bid i such that $b_i / \sum_{t \in [t_i^-, t_i^+]} d_{ir}(t) \leq h_{rs}$, 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 $L_r > h_{rs}$.

The marginal price $p_{rs}(t)$ is a function of the amount of currently allocated resource $y_{rs}(t)$. $p_{rs}(t)$ is higher than marginal operational cost h_{rs} 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 $p_{rs}(t)$ increases as $y_{rs}(t)$ 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 $y_{rs}(t) > C_{rs}$ to make sure a server will not allocate more resources than its capacity.

PD$_2$ also uses CORE 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 the proof 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*

$$\alpha_2 = \max_{r \in [R], s \in [S]} \ln\big(\tfrac{2\sigma RS(U_r - h_{rs})}{L_r - h_{rs}}\big) \ , \qquad (22)$$

assuming that the offline optimal social welfare is at least

$$OPT \geq \tfrac{1}{RS} \textstyle\sum_{r \in [R]} \sum_{s \in [S]} \tfrac{T}{\sigma}(L_r - h_{rs})C_{rs} \ .$$

Before we get to the proof of the theorem, let us first explain how to interpret the assumed lower bound on the offline optimal social welfare. Recall that $\frac{T}{\sigma}$ is the minimum time span that a bid occupies the requested resources for. $L_r - h_{rs}$ is the minimum social welfare generated by a bid that demands resource r and is allocated to server s, per unit of resource r and per unit of time. Thus, $\frac{T}{\sigma}\big(L_r - h_{rs}\big)C_{rs}$ is the minimal social welfare generated by all bids that demand resource r and are allocated to server s if the entire capacity of resource r on server s is occupied for each time slot. So the assumption in the above theorem is essentially saying that in the offline solution, there are enough workloads to exhaust at least one resource on one server at a time slot, which is easily satisfied in real-world cloud systems.

PROOF. Let $\alpha_{rs} = \ln \frac{2\sigma RS(U_r - h_{rs})}{L_r - h_{rs}}$. 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_{rs}, i.e.,

$$p_{rs}(t)\mathrm{d}y_{rs}(t) - f'_{rs}(y_{rs}(t))\mathrm{d}y_{rs}(t) \geq \tfrac{1}{\alpha_{rs}} f^{\star\prime}_{rs}(p_{rs}(t))\mathrm{d}p_{rs}(t)$$

and that the initial value of the dual objective is at most $\frac{1}{2}OPT$. Then, given $\alpha_2 = \max_{r \in [R], s \in [R]} \alpha_{rs}$, the theorem will follow from Lemma 1 and Theorem 2. According to (19) and (20),

$$f'_{rs}(y_{rs}(t)) = \begin{cases} h_{rs}, & 0 \leq y_{rs}(t) \leq C_{rs} \\ +\infty, & y_{rs}(t) > C_{rs} \end{cases}$$

$$f^{\star\prime}_{rs}(p_{rs}(t)) = \begin{cases} 0, & 0 \leq p_{rs}(t) < h_{rs} \\ C_{rs} & p_{rs}(t) \geq h_{rs} \end{cases}$$

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

Finally, according to (5) and (20),

$$\begin{aligned} D^0 &= \textstyle\sum_{t \in [T]} \sum_{s \in [S]} \sum_{r \in [R]} C_{rs}\big(p_{rs}(0) - h_{rs}\big) \\ &= \textstyle\sum_{t \in [T]} \sum_{s \in [S]} \sum_{r \in [R]} C_{rs}\big(\frac{L_r - h_{rs}}{2\sigma RS}\big) \\ &= \tfrac{1}{2}\tfrac{1}{RS} \textstyle\sum_{r \in [R]} \sum_{s \in [S]} \frac{T}{\sigma}(L_r - h_{rs})C_{rs} \\ &\leq \tfrac{1}{2}OPT. \end{aligned}$$

The last inequality is based on the assumption in the theorem. □

5.2 Profit Maximization

Next, we present an extension of PD$_2$ *i.e.*, Alg. 5, that achieves a competitive ratio comparable to $2\alpha_2$ in provider's profit. Similar to how RPD$_1$ extends PD$_1$, RPD$_2$ first use PD$_2$ as a black box to obtain a tentative allocation x_i and payment $\hat{p}_i$ for each bid i. 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 the proof of Theorem 4 and thus omitted.

Algorithm 5: Randomized Online Auction RPD$_2$

```
Input: $S$, $R$, $\mathbf{C}$, $\mathbf{h}$, $\sigma$, $L_r$, $U_r$
Output: $\mathbf{x}, \mathbf{p}$
DEFINE $f_{rs}(y_{rs}(t))$ according to (19);
DEFINE $p_{rs}(y_{rs}(t))$ according to (21);
DEFINE $\chi = \max_{r\in[R],s\in[S]} \frac{2\sigma RS(U_r-h_{rs})}{L_r-h_{rs}}$;
INITIALIZE $x_{is} = 0, y_{rs}(t) = 0, u_i = 0, p_{rs}(t) = \frac{L_r-h_{rs}}{2\sigma RS} + h_{rs}, \forall i \in [I], s \in [S], r \in [R], t \in [T]$;
Upon the arrival of the $i$th bid
Get bidding language $B_i$ according to (1);
$(\mathbf{x}_i, \hat{p}_i, \mathbf{p}, \mathbf{y}) = \text{CORE}(S, R, B_i, \mathbf{p}, \mathbf{y}, \mathbf{p}(\mathbf{y}))$;
if $\exists s \in [S], x_{is} = 1$ then
    Generate a random number $\eta$ such that:
        $\eta = \begin{cases} 1 & \text{with prob. } \frac{1}{2} \\ 2^j & \text{with prob. } \frac{1}{2\log_2 \chi} \text{ for } j = 1, \ldots, \log_2 \chi \end{cases}$    (23)
    if $b_i \geq \eta\hat{p}_i$ then
        Accept the $i$th bid on server $s$, and charge $\eta\hat{p}_i$;
    else
        Reject the $i$th bid;
    end
else
    Reject the $i$th bid;
end
```

The profit guarantee will be presented in Theorem 10. We first show a few technical lemmas that are needed for the proof of Theorem 10 (proofs of lemmas given in appendices G—I). Let $A^\star$ be the set of bids accepted by social welfare maximizing online auction PD$_2$, and $O^\star$ be the set of bids accepted by the offline omniscient optimal algorithm for social welfare maximization. c_i is the cost occurs by an accepted bid i in (16).

LEMMA 5. *The optimal profit of any truthful and individually rational auction (online or offline) in our system is upper bounded by*

$$\textstyle\sum_{i\in A^\star}(b_i - c_i) + \sum_{i\in O^\star:b_i<\hat{p}_i}(b_i - c_i). \qquad (24)$$

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.,*

$$\textstyle\sum_{i\in O^\star:b_i<\hat{p}_i}(b_i - c_i) \leq \alpha_2 \sum_{i\in A^\star}(\hat{p}_i - c_i) \qquad (25)$$

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 i, RPD$_2$ uses the same allocation and payment as in PD$_2$ with probability $\frac{1}{2}$. □

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 profit is upper bounded by $4\log_2 \chi + 2\alpha_2$ times the expected profit of RPD$_2$. By the choice of $\chi = \max_{r\in[R],s\in[S]} \frac{2\sigma RS(U_r-h_{rs})}{L_r-h_{rs}}$ and the definition of α_2 in (22), we get that $\log_2 \chi = \frac{1}{\ln 2}\ln\chi = \frac{1}{\ln 2}\alpha_2$. So the theorem follows. □

We also give the competitiveness in social welfare achieved by RPD$_2$ in the following theorem, with proof given in Appendix J.

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 $h_{rs} = 0$, and the server cost functions as the following zero infinity functions:

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

All the properties that we have shown for linear cost functions apply to zero infinity costs, with proofs omitted due to duplicity.

6. PERFORMANCE EVALUATION

We evaluate our auctions using trace-driven simulations, exploiting Google cluster-usage data [36], which contains information including resource demands (CPU, RAM, Disk) for each job submitted to the Google cluster, job arrival times and durations. We translate each job into a VM bid, requesting $R = 3$ types of resources at the demands extracted from the traces (note demand $d_{ir}(t)$ here is not much smaller than C_{rs}, though our theoretical analysis assumed so). Each time slot is 10 seconds [37], and a bid arrives every $[1, 10]$ time slot(s) (we set this arrival rate relatively low to test our algorithm in extreme scenarios, since the more bids arrive concurrently, the closer our online algorithm is to the offline one). The duration of each VM is between 10 and 3600 time slots. We set the bidding price of each bid by multiplying the overall resource demands in the bid 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 will be varied in different experiments. We will vary the span between bid arrival time and the VM start time. We simulate servers with heterogenous resource capacities (C_{rs}) 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, we set the disk storage capacity of servers randomly within $[320, 800](GB)$. The total capacity of each type of resource to provision, and hence the number of servers to simulate, is roughly according to the total amount of demand from all bids multiplying a random number in $[0.4, 0.8]$. The total number of time slots T is set to $5I + 3600$, since on average a bid arrives in 5 time slots and an extra 3600 time slots are reserved to serve long-running bids that come near the end of the batch. h_{rs} 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 [31]. For superlinear cost functions, we set β_{rs} within $[1.7, 2.2]$ for CPU and within $[0.5, 1]$ for RAM and disk [29][31]. By default, $\sigma = 500$, $U_r = 50$, $L_r = 1$.

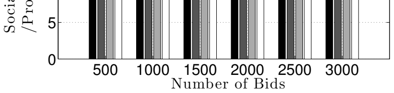

Figure 7: Comparison between $PD2$ and TC for social welfare, and between $RPD2$ and TC for profit

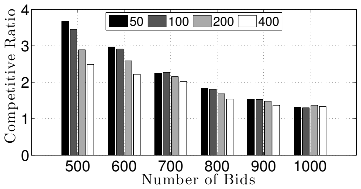

Figure 1: Competitive ratio of PD_1 in social welfare among different Γ

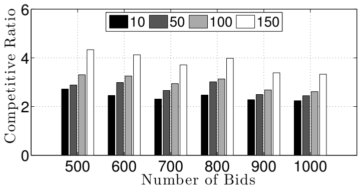

Figure 2: Competitive ratio of RPD_1 in profit among different U_r

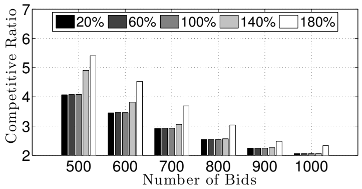

Figure 3: Competitive ratio of RPD_1 in profit over different percentages of estimation over real U_r

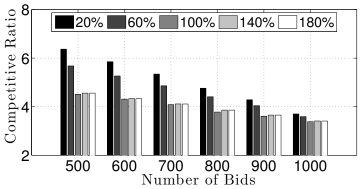

Figure 4: Competitive ratio of RPD_1 in profit over different percentages of estimation over real L_r

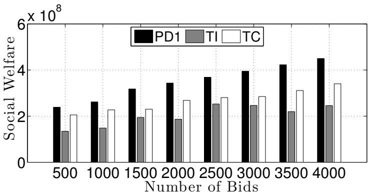

Figure 5: Comparison among $PD1$, TC and TI

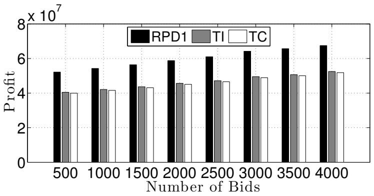

Figure 6: Comparison among $RPD1$, TC, TI

We compare our algorithms with the offline optimum, as well as two existing schemes *Twice-the-Cost* (TC) and *Twice-the-Index* (TI) [23] (we identified a lack of comparable approaches from the VM auction literature). TC and TI share similar basic ideas with our online auctions, but adopt different marginal pricing functions: For TC, $p_{rs}(y_{rs}(t)) = 2f'_{rs}(y_{rs}(t))$, *i.e.*, the current marginal payment is twice of the current marginal cost; for TI, $p_{rs}(y_{rs}(t)) = f'_{rs}(2y_{rs}(t))$, *i.e.*, the current marginal payment is the marginal cost on twice of the current resource usage.

6.1 Comparison with Offline Optimum

We first study the competitive ratios 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 or 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 1000 in this set of experiments.

Fig. 1 illustrates that a smaller competitive 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. Considering the online setting, 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. Fig. 2 shows that the competitive ratio of expected profit with RPD_1 becomes larger with the increase of U_r. According to Theorem 5, a larger U_r not only negatively influences the competitive ratio of RPD_1, but also determines the distribution of η that further increases the ratio. In Fig. 3 and 4, we use 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.

In addition, all figures show that the competitive ratio becomes smaller with the increase of the number of bids. According to our setup, when the number of bids increases, the total demands of resources increase, and more servers are provisioned. In our online auctions, we always choose a cheapest server for each coming bid in (8), and hence the solution space becomes larger when the number of servers increases, leading to better competitive ratios. The results for PD_2 and RPD_2 are similar to the cases of PD_1 and RPD_1, and we omit them here due to space constraints.

6.2 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. 5 shows that the social welfare achieved by our PD_1 outperforms those by TC and TI. 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. Fig. 6 further reveals the higher profit achieved by our RPD_1, as compared to TC and TI.

Due to space limit, we compare PD_2 and RPD_2 with TC in terms of social welfare ($W(\cdot)$) and profit ($P(\cdot)$), respectively, in the same Fig. 7 (note the values are in log scale). Our algorithms outperform TC in both cases.

7. CONCLUDING REMARKS

This work designs truthful and efficient online VM auctions where cloud users bid for resources into the future for tailor-made VMs with different running durations, 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, which existing online primal-dual resource allocation frameworks do not handle. 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 and can be of further interest for other profit maximization problems in general. Trace driven simulations validate our theoretical analysis and show good performance of our mechanisms as compared to the offline solution and existing mechanisms on similar frameworks.

8. ACKNOWLEDGMENTS

This work was supported in part by grants from Hong Kong RGC under the contracts HKU 718513 and HKU 27200214E, the Natural Sciences and Engineering Research Council of Canada (NSERC), and Wedge Networks.

9. REFERENCES

[1] "Amazon EC2 Instances," http://aws.amazon.com/ec2/instance-types/.

[2] "ProfitBricks," https://www.profitbricks.com.

[3] "CloudSigma," https://www.cloudsigma.com.

[4] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing Amazon EC2 Spot Instance Pricing," in *Proc. of IEEE CloudCom*, 2011.

[5] S. Zaman and D. Grosu, "Combinatorial Auction-based Allocation of Virtual Machine Instances in Clouds," *Journal of Parallel and Distributed Computing*, vol. 73, no. 4, pp. 495–508, 2013.

[6] W.-Y. Lin, G.-Y. Lin, and H.-Y. Wei, "Dynamic Auction Mechanism for Cloud Resource Allocation," in *IEEE/ACM CCGrid*, 2010.

[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. of ACM SIGMETRICS*, 2014.

[8] L. Zhang, Z. Li, and C. Wu, "Dynamic Resource Provisioning in Cloud Computing: A Randomized Auction Approach," in *Proc. of IEEE INFOCOM*, 2014.

[9] W. Shi, C. Wu, and Z. Li, "RSMOA: A Revenue and Social Welfare Maximizing Online Auction for Dynamic Cloud Resource Provisioning," in *Proc. of IWQoS*, 2014.

[10] W. Wang, B. Liang, and B. Li, "Revenue Maximization with Dynamic Auctions in IaaS Cloud Markets," in *Proc. of IEEE ICDCS*, 2013.

[11] H. Zhang, B. Li, H. Jiang, F. Liu, A. V. Vasilakos, and J. Liu, "A Framework for Truthful Online Auctions in Cloud Computing with Heterogeneous User Demands," in *Proc. of IEEE INFOCOM*, 2013.

[12] N. Buchbinder, K. Jain, and J. S. Naor, "Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue," in *Proc. of the 15th Annual European Symposium on Algorithms*, 2007.

[13] A. Mu'Alem and N. Nisan, "Truthful Approximation Mechanisms for Restricted Combinatorial Auctions," *Games and Economic Behavior*, vol. 64, no. 2, pp. 612–631, 2008.

[14] Y. Azar, U. Bhaskar, L. Fleischer, and D. Panigrahi, "Online Mixed Packing and Covering," in *Proc. of ACM-SIAM SODA*, 2013.

[15] N. Buchbinder and J. Naor, "The Design of Competitive Online Algorithms via a Primal-dual Approach," *Foundations and Trends in Theoretical Computer Science*, vol. 3, no. 2–3, pp. 93–263, 2009.

[16] Z. Huang and A. Kim, "Welfare Maximization with Production Costs: a Primal Dual Approach," in *Proc. of the ACM-SIAM SODA*, 2015.

[17] N. R. Devanur and Z. Huang, "Primal Dual Gives Almost Optimal Energy Efficient Online Algorithms," in *Proc. of ACM-SIAM SODA*, 2014.

[18] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[19] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multi Resource Allocation: Fairness-Efficiency Tradeoffs in a Unifying Framework," in *Proc. of IEEE INFOCOM*, 2012.

[20] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters," in *Proc. of IEEE INFOCOM*, 2012.

[21] Q. Wang, K. Ren, and X. Meng, "When Cloud meets eBay: Towards Effective Pricing for Cloud Computing," in *Proc. of IEEE INFOCOM*, 2012.

[22] S. Anand, N. Garg, and A. Kumar, "Resource Augmentation for Weighted Flow-time Explained by Dual Fitting," in *Proc. of ACM-SIAM SODA*, 2012.

[23] A. Blum, A. Gupta, Y. Mansour, and A. Sharma, "Welfare and Profit Maximization with Production Costs," in *Proc. of IEEE FOCS*, 2011.

[24] R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella, "Multi-Resource Packing for Cluster Schedulers," in *Proc. of ACM SIGCOMM*, 2014.

[25] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," in *Proc. of ACM SOSP*, 2003.

[26] "KVM CPU Hotplug," http://www.linux-kvm.org/page/CPUHotPlug.

[27] H. Chen, M. C. Caramanis, and A. K. Coskun, "Reducing the Data Center Electricity Costs Through Participation in Smart Grid Programs," in *Proc. of IGCC*, 2014.

[28] "Xen DVFS," http://lists.xen.org/archives/html/xen-devel/2009-09/msg00585.html.

[29] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware Provisioning of Virtual Machines for Real-time Cloud Services," *Concurrency and Computation: Practice and Experience archive*, vol. 23, no. 13, pp. 1491–1505, 2011.

[30] B. Krishnan, H. Amur, A. Gavrilovska, and K. Schwan, "VM Power Metering: Feasibility and Challenges," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 3, pp. 56–60, 2010.

[31] J. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual Machine Power Metering and Provisioning," in *Proc. of ACM SoCC*, 2010.

[32] N. R. Devanur and Z. Huang, "Primal dual gives optimal energy efficient online algorithms," in *Proc. of ACM-SIAM SODA*, 2014.

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

[34] N. Buchbinder and J. Naor, "Online Primal-Dual Algorithms for Covering and Packing Problems," in *Algorithms–ESA 2005*. Springer, 2005, pp. 689–701.

[35] B. Awerbuch, Y. Azar, and A. Meyerson, "Reducing Truth-telling Online Mechanisms to Online Optimization," in *Proc. of ACM STOC*, 2003.

[36] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format + schema," Google Inc., Mountain View, CA, USA, Technical Report, Nov. 2011, revised 2012.03.20. Posted at URL http://code.google.com/p/googleclusterdata/wiki/TraceVersion2.

[37] 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. of ACM SoCC*, 2012.

[38] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan, "Multi-Parameter Mechanism Design and Sequential Posted Pricing," in *Proc. of ACM STOC*, 2010.

APPENDIX

A. PROOF OF PROPOSITION 1

PROOF. $f^{\star}_{rs}(p_{rs}(t)) =$

$$\sup_{y\geq 0}\begin{cases} p_{rs}(t)y_{rs}(t) - h_{rs}y_{rs}(t)^{1+\beta_{rs}}\mathbf{1}, & y_{rs}(t)\in[0,C_{rs}] \\ p_{rs}(t)y_{rs}(t) - \infty\mathbf{2}, & y_{rs}(t) > C_{rs} \end{cases}$$

Observe that $\mathbf{2}$=$-\infty$. Then we only need to obtain the conjugate of $f_{rs}(\cdot)$ from $\mathbf{1}$.

Let $g_{rs}(y_{rs}(t)) = p_{rs}(t)y_{rs}(t) - h_{rs}y_{rs}(t)^{1+\beta_{rs}}$. Differentiate $g_{rs}(t)$ with respect to $y_{rs}(t)$ and set

$$(p_{rs}(t)\times y_{rs}(t) - h_{rs}y_{rs}(t)^{1+\beta_{rs}})' = 0. \tag{27}$$

We obtain the solution of the equation as $y^0_{rs}(t) = (\frac{p_{rs}(t)}{h_{rs}(1+\beta_{rs})})^{\frac{1}{\beta_{rs}}}$.

Note that the domain of $g_{rs}(t)$ is $y_{rs}(t)\in[0,C_{rs}]$. Hence the supremum happens at $y^0_{rs}(t)$ only if $y^0_{rs}(t)$ lies in $[0,C_{rs}]$. Otherwise when $y^0_{rs}(t) > C_{rs}$, $(p_{rs}(t)\times y_{rs}(t) - h_{rs}y_{rs}(t)^{1+\beta_{rs}})' > 0$, such that $g_{rs}(y_{rs}(t))$ monotonically increases with y_{rs} in $[0,C_{rs}]$ and the supremum happens at C_{rs}.

Merging the two cases, we derive the conjugate of the cost function:

$$f^{*}_{rs}(p_{rs}(t)) = \begin{cases} (\frac{p_{rs}(t)}{1+\beta_{rs}})^{\frac{1+\beta_{rs}}{\beta_{rs}}}\frac{\beta_{rs}}{h_{rs}^{\frac{1}{\beta_{rs}}}}, & y^0_{rs}(t)\leq C_{rs} \\ C_{rs}p_{rs}(t) - h_{rs}C_{rs}^{1+\beta_{rs}}, & y^0_{rs}(t) > C_{rs} \end{cases} \tag{28}$$

□

B. PROOF OF THEOREM 1

PROOF. *(Truthfulness in bidding price)* The marginal prices that the cloud provider presents to bid i depend only on the demands of resources before the arrival of bid i and the demand of bid i, thus, are independent on bid i's bidding price. Further, the cloud provider always assigns bids to servers to maximize each bid's utility given the current marginal prices. So it falls into the family of sequential posted price mechanisms (e.g., [38]) and, thus, a bidder cannot improve its utility by lying about its bidding price. *(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, *i.e.*, let $y_{rs}(t_i,t)$ denote the amount of resource r on s in t which has been allocated by t_i ($t_i\leq t$) and $y_{rs}(t_i,t)$ is non-decreasing over t_i. 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. So no bidder would do that. 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) According to (8), the utility of a bidder is always non-negative. The profit of the provider is also non-negative based on (10) which implies $p_{rs}(y_{rs}(t)) > f'_{rs}(y_{rs}(t))$.

(Polynomial running time) We assume that the algorithm can compute the differentials of marginal server cost functions, *i.e.*, f'_{rs} in constant time. (Otherwise, the running time will be $O(RST)$ times the time complexity of computing f'_{rs}.) To process a bid i, the algorithm first sums up the marginal prices for all requested resources over the occupation duration for each server $s\in[S]$ to compute the payment that bid i should pay if it would be served on server s. This step runs in $O(RST)$ time. Then, the algorithm computes u_i and decides the allocation and payment of bid i by checking the utility of the bid if it would be served on each server s, which can be done in $O(S)$ time. Finally, the algorithm updates the amount of allocated resources $y_{rs}(t)$ and the respective marginal prices $p_{rs}(t)$, which can be done in $O(RST)$ time. □

C. PROOF OF LEMMA 4

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

D. PROOF OF LEMMA 3

PROOF. First, we explicitly write down the differentials of the server cost functions (in (2)) and their convex conjugates (in (6)):

$$f'_{rs}(y_{rs}(t)) = \begin{cases} h_{rs}(1+\beta_{rs})y_{rs}(t)^{\beta_{rs}}, & y_{rs}(t)\in[0,C_{rs}] \\ +\infty, & y_{rs}(t) > C_{rs} \end{cases}$$

$$f^{\star\prime}_{rs}(p_{rs}(t)) = \begin{cases} (\frac{p_{rs}(t)}{(1+\beta_{rs})h_{rs}})^{\frac{1}{\beta_{rs}}}, & p_{rs}(t)\leq h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}} \\ C_{rs}, & p_{rs}(t) > h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}} \end{cases}$$

By $\theta_{rs}\geq\frac{\delta_{rs}}{C_{rs}(\delta_{rs}-1)}\ln(\frac{U_r}{h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}}})$ and the definition of marginal payment in (10), we get that when $y_{rs}(t) = C_{rs}$,

$$p_{rs}(t) = h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}}e^{\theta_{rs}(C_{rs}-\frac{C_{rs}}{\delta_{rs}})}\geq U_r\ .$$

By our choice of $p_{rs}(t)$, the cloud provider will never allocate more resources than its capacity, *i.e.*, $y_{rs}(t)\leq C_{rs}$. Therefore, we may assume in the rest of the proof that $y_{rs}(t)\leq C_{rs}$ and, thus, $f'_{rs}(y_{rs}(t)) = h_{rs}(1+\beta_{rs})y_{rs}(t)^{\beta_{rs}}$. Next, according to the piece-wise definition of $f^{\star\prime}_{rs}$, the proof is divided into two cases.

Case 1: $y_{rs}(t)\leq\frac{C_{rs}}{\delta_{rs}}$.

In this case, the Differential Allocation-Payment Relationship is

$$\begin{aligned}\big(p_{rs}(t) - h_{rs}(1+\beta_{rs})y_{rs}(t)^{\beta_{rs}}\big)\mathrm{d}y_{rs}(t)& \\ \geq\frac{1}{\alpha_{rs}}\big(\tfrac{1}{h_{rs}(1+\beta_{rs})}p_{rs}(r)\big)^{\frac{1}{\beta_{rs}}}\mathrm{d}p_{rs}(t)&\end{aligned} \tag{29}$$

By the marginal payment definition in (10),

$$p_{rs}(t) = f'_{rs}(\delta_{rs}y_{rs}(t)) = h_{rs}(1+\beta_{rs})\delta_{rs}^{\beta_{rs}}y_{rs}(t)^{\beta_{rs}}. \tag{30}$$

Putting them together, (29) is equivalent to (cancelling common terms of both sides) $(\delta_{rs}^{\beta_{rs}}-1)\geq\frac{1}{\alpha_{rs}}\beta_{rs}\delta_{rs}^{\beta_{rs}+1}$. If $\beta_{rs}\geq 1$, then $\delta_{rs} = \max\left\{2,(1+\beta_{rs})^{\frac{1}{\beta_{rs}}}\right\} = 2$ and $\delta_{rs}^{\beta_{rs}+1} = 2\delta_{rs}^{\beta_{rs}} = 4\delta_{rs}^{\beta_{rs}} - 2\delta_{rs}^{\beta_{rs}}\leq 4\delta_{rs}^{\beta_{rs}} - 4$. So we get that $\frac{\beta_{rs}\delta_{rs}^{\beta_{rs}+1}}{\delta_{rs}^{\beta_{rs}}-1}\leq 4\beta_{rs} <$

α_{rs}. If $\beta_{rs} < 1$, then $\delta_{rs} = (1+\beta_{rs})^{\frac{1}{\beta_{rs}}}$, and

$$\frac{\beta_{rs}\delta_{rs}^{\beta_{rs}+1}}{\delta_{rs}^{\beta_{rs}}-1} = \delta_{rs}(1+\beta_{rs}) \leq e(1+\beta_{rs}) < \alpha_{rs} \ .$$

Case 2: $\frac{C_{rs}}{\delta_{rs}} \leq y_{rs}(t) \leq C_{rs}$.
In this case, the Differential Allocation-Payment Relationship is

$$\big(p_{rs}(t) - h_{rs}(1+\beta_{rs})y_{rs}(t)^{\beta}\big)\mathrm{d}y_{rs}(t) \geq \tfrac{1}{\alpha_{rs}}C_{rs}\mathrm{d}p_{rs}(t) \quad (31)$$

Recall that when $y_{rs}(t) \geq \frac{C_{rs}}{\delta_{rs}}$, the marginal payment is

$$p_{rs}(t) = h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}}e^{\theta_{rs}(y_{rs}(t)-\frac{C_{rs}}{\delta_{rs}})} \qquad (32)$$

By Lemma 4, to show (31), it suffices to show $\frac{\beta_{rs}}{1+\beta_{rs}}p_{rs}(t)\mathrm{d}y_{rs}(t) \geq \frac{1}{\alpha_{rs}}C_{rs}\mathrm{d}p_{rs}(t)$. On the other hand, by the definition of the marginal payment in (32), we have $\mathrm{d}p_{rs}(t) = \theta_{rs}p_{rs}(t)dy_{rs}(t)$. So it remains to show that $\theta_{rs} \leq \frac{\beta_{rs}}{C_{rs}(1+\beta_{rs})}\alpha_{rs}$. By our choice of parameters, either

$$\begin{aligned}
\theta_{rs} &= \tfrac{\delta_{rs}}{C_{rs}}\beta_{rs} \leq \tfrac{e}{C_{rs}}\beta_{rs} && (\delta_{rs} \leq e)\\
&= \tfrac{\beta_{rs}}{C_{rs}(1+\beta_{rs})}e(1+\beta_{rs}) \leq \tfrac{\beta_{rs}}{C_{rs}(1+\beta_{rs})}\alpha_{rs} \ ,
\end{aligned}$$

$$\begin{aligned}
\text{or} \quad \theta_{rs} &= \tfrac{\delta_{rs}}{C_{rs}(\delta_{rs}-1)}\ln\big(\tfrac{U_r}{h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}}}\big)\\
&\leq \tfrac{2}{C_{rs}}\ln\big(\tfrac{U_r}{h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}}}\big) && (\delta_{rs} \geq 2)\\
&= \tfrac{\beta_{rs}}{C_{rs}(1+\beta_{rs})}\tfrac{2(1+\beta_{rs})}{\beta_{rs}}\ln\big(\tfrac{U_r}{h_{rs}(1+\beta_{rs})C_{rs}^{\beta_{rs}}}\big)\\
&\leq \tfrac{\beta_{rs}}{C_{rs}(1+\beta_{rs})}\alpha_{rs} \ .
\end{aligned}$$

So the lemma holds in case 2. □

E. PROOF OF THEOREM 4

PROOF. RPD$_1$ also posts a take-it-or-leave-it price for each bid i and serves the bid if and only if its bidding price is larger than the posted price, while changing the payment from $\hat{p}_i$ in PD$_1$ to $\eta_1\tilde{p}_i$, where η_1 and $\tilde{p}_i$ are independent of b_i's bidding price. So it is still a sequentially posted pricing mechanism and, thus, truthful in terms of bidding price. The proof of truthfulness in bid arrival time and duration times, individual rationality, and time complexity of bid processing follows similarly to the proof of Theorem 1, and is hence omitted. □

F. PROOF OF PROPOSITION 2

PROOF. The conjugate of $f_{rs}(y_{rs}(t))$ is defined in (6). By the definition of f_{rs}, $p_{rs}(t)y_{rs}(t) - f_{rs}(y_{rs}(t))$ is equal to $(p_{rs}(t) - h_{rs})y_{rs}(t)$ if $y_{rs}(t) \leq C_{rs}$, and is $-\infty$ if $y_{rs}(t) > C_{rs}$. If $p_{rs}(t) - h_{rs} < 0$, then the right hand side of (6) is maximized when $y_{rs}(t) = 0$ with maximum value 0; if $p_{rs}(t) - h_{rs} \geq 0$, then it is maximized when $y_{rs}(t) = C_{rs}$ with maximum value $(p_{rs}(t) - h_{rs})C_{rs}$. □

G. PROOF OF LEMMA 5

PROOF. The profit of any truthful and individually rational auction is upper bounded by its social welfare and, thus, by the optimal social welfare, *i.e.*,

$$\begin{aligned}
\textstyle\sum_{i\in O^\star}(b_i - c_i) &= \textstyle\sum_{i\in O^\star:b_i\geq\hat{p}_i}(b_i - c_i) + \sum_{i\in O^\star:b_i<\hat{p}_i}(b_i - c_i)\\
&\leq \textstyle\sum_{i:b_i\geq\hat{p}_i}(b_i - c_i) + \sum_{i\in O^\star:b_i<\hat{p}_i}(b_i - c_i)\\
&= \textstyle\sum_{i\in A^\star}(b_i - c_i) + \sum_{i\in O^\star:b_i<\hat{p}_i}(b_i - c_i)
\end{aligned}$$

□

H. PROOF OF LEMMA 6

PROOF. Due to the linear cost function in this case, the incurred cost by an accepted bid i is $c_i = \sum_{t\in[t_i^-,t_i^+]}\sum_{r\in[R]}d_{ir}(t)h_{rs_i}$, where s_i is the server that serves the ith bid. By the definition of PD$_2$, for any tentatively accepted bid $i \in A^\star$, the tentative profit from bid i is at least

$$\begin{aligned}
\hat{p}_i - c_i &= \textstyle\sum_{r\in[R]}\sum_{t\in[t_i^-,t_i^+]}\big(d_{ir}(t)p_{rs}(t) - d_{ir}(t)h_{rs}\big)\\
&\geq \textstyle\sum_{r\in[R]}\sum_{t\in[t_i^-,t_i^+]}d_{ir}(t)\big(p_{rs}(0) - h_{rs}(t)\big)
\end{aligned}$$

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

$$\hat{p}_i - c_i \geq \textstyle\sum_{r\in[R]}\sum_{t\in[t_i^-,t_i^+]}d_{ir}(t)\frac{L_r-h_{rs}}{2\sigma RS}$$

On the other hand, the maximum profit from bid i is at most $b_i - c_i \leq \sum_{r\in[R]}\sum_{t\in[t_i^-,t_i^+]}d_{ir}(t)(U_r - h_{rs})$. We have that $\frac{b_i-c_i}{\hat{p}_i-c_i} \leq \frac{2\sigma RS(U_r-h_{rs})}{L_r-h_{rs}} = \chi$. By the definition of η in (23), η is a randomly chosen power of 2 between 1 and χ and, thus, satisfies that $\frac{1}{2}(b_i - c_i) \leq \eta(\hat{p}_i - c_i) \leq (b_i - c_i)$ with probability at least $\frac{1}{2\log_2\chi}$. In this case, RPD$_2$ will accept the ith bid and generate profit $\eta(\hat{p}_i - c_i) \geq \frac{1}{2}(b_i - c_i)$ from the ith bid. Therefore, the expected profit generated by RPD$_2$ from every bid $i \in A^\star$ (that is, tentatively accepted by PD$_2$) is at least $\frac{1}{2\log_2\chi}\cdot\frac{1}{2}(b_i - c_i)$. The expected profit of RPD$_2$ is at least $\frac{1}{4\log_2\chi}\sum_{i\in A^\star}(b_i - c_i)$. □

I. PROOF OF LEMMA 7

PROOF. Consider a virtual instance of our online problem where the ith bid in the virtual instance has the same demands $d_{ir}(t)$ over $[t_i^-, t_i^+]$ as in the original instance, but has bidding prices $\min\{b_i, \hat{p}_i\}$ instead of b_i. 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 $\sum_{i\in A^\star}(\hat{p}_i - c_i)$. Further note that choosing all $i \in O^\star$ such that $b_i < \hat{p}_i$ is a feasible solution in the virtual instance and gets social welfare $\sum_{i\in O^\star:b_i<\hat{p}_i}(b_i - c_i)$. The lemma now follows from the competitive ratio of online algorithm PD$_2$. □

J. 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 b'_i as follows:

$$b'_i = \begin{cases} b_i & \text{if } b_i \geq \eta\hat{p}_i \\ 0 & \text{if } b_i < \eta\hat{p}_i \end{cases} \qquad (33)$$

The expected social welfare of RPD$_2$ is:

$$E\big[\textstyle\sum_{i\in B^\star}(b_i - c_i)\big] = E\big[\sum_{i\in A^\star}(b'_i - c_i)\big] \ . \qquad (34)$$

Note that $A^\star = \{i : b_i \geq \hat{p}_i\}$. By linearity of expectation,

$$\begin{aligned}
(34) \quad &= \quad E\big[\textstyle\sum_{i:b_i\geq\hat{p}_i}(b'_i - c_i)\big] = \sum_{i:b_i\geq\hat{p}_i}E\big[b'_i - c_i\big]\\
&= \quad \textstyle\sum_{i:b_i\geq\hat{p}_i}(b_i - c_i)\cdot\Pr[b_i \geq \eta\hat{p}_i]\\
&\geq \quad \textstyle\sum_{i:b_i\geq\hat{p}_i}(b_i - c_i)\cdot\Pr[b_i \geq \hat{p}_i]\cdot\Pr[\eta = 1]\\
&\geq \quad \tfrac{1}{2}\textstyle\sum_{i\in A^\star}(b_i - c_i)
\end{aligned}$$

Therefore, the expected social welfare of RPD$_2$ is at least half of that of PD$_2$. The theorem then follows from Theorem 8. □