RSMOA: A Revenue and Social Welfare Maximizing Online Auction for Dynamic Cloud Resource Provisioning

Weijie Shi
Dept. of Computer Science
The University of Hong Kong
wjshi@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

***Abstract*—We study online cloud resource auctions where users can arrive anytime and bid for heterogeneous types of virtual machines (VMs) assembled and provisioned on the fly. The proposed auction mechanism RSMOA, to the authors' knowledge, represents the first truthful online mechanism that timely responds to incoming users' demands and makes dynamic resource provisioning and allocation decisions, while guaranteeing efficiency in both the provider's revenue and system social welfare. RSMOA consists of two components: (1) an online mechanism that computes resource allocation and users' payments based on a global, non-decreasing pricing curve, and guarantees truthfulness; (2) a judiciously designed pricing curve, which is derived from a threat-based strategy and guarantees a competitive ratio $O(\ln(p))$ in both system social welfare and the provider's revenue, as compared to the celebrated offline Vickrey-Clarke-Groves (VCG) auction. Here p is the ratio between the upper and lower bounds of users' marginal valuation of a type of resource. The efficacy of RSMOA is validated through extensive theoretical analysis and trace-driven simulation studies.**

I. Introduction

Cloud computing, a recently emerged computing paradigm, enables convenient and on-demand access to a virtually unlimited pool of computing resources, such as CPU, RAM and disk storage. Cloud providers exemplified by Amazon EC2 [1] and Microsoft Azure [2] manage the resources by assembling them into virtual machines (VMs), and pursue maximized revenue through properly pricing and allocating these VMs to cloud users. Cloud users enjoy the convenience, scalability and flexibility of the cloud service, and pay the provider a monetary remuneration that is usually a fraction of their utility gained from the cloud services.

The *de facto* standard in selling cloud computing resources used to be charging fixed prices for VM access [3]. The provider sets an hourly price for each type of VMs provisioned, and charges users by usage time. While relatively simple to implement in practice, fixed-price policies suffer from a clear drawback: they cannot effectively reflect the supply-demand relationship that is varying across the temporal domain. Consequently, both (a) the revenue that the provider can glean and (b) system-side social welfare achieved by the cloud ecosystem as a whole are suboptimal.

Towards more efficient pricing and allocation of cloud resources, auction-style mechanisms have been proposed and implemented in real-world cloud systems, as exemplified by the Amazon Spot Instance market [4], with a series of subsequent work on its enhancement [5]–[7]. Unfortunately, the following flexibilities in user demand and resource allocation are still insufficient in existing cloud auction designs. (1) *On-demand VM assembly*: pre-assembled VMs with fixed configurations are the existing norm in VM provisioning, which does not address users' heterogenous VM demand well; dynamic assembly of cloud resources into desired combination of VM instances is often preferred in practice. (2) *Elastic resource demands*: Cloud user's demand is typically elastic, allowing acquisition of different amounts of resources for corresponding levels of utility gains. Existing auctions mostly allow only a fixed static resource demand from each user, and an auction mechanism that efficiently supports elastic user demands is missing. (3) *VM termination at user's will*: A cloud user should have the right to keep an acquired VM as long as she likes and to terminate it at any time she wishes, without unexpected preemption or the need to give *a priori* notification.

978-1-4799-0913-1/14/$31.00 © 2014 IEEE

Aiming at an online auction mechanism that are general and expressive enough to provide these flexibilities, we propose RSMOA, the first online combinatorial auction that timely responds to the incoming users' demands and makes dynamic resource provisioning decisions. RSMOA has the following salient features. (1) An online VM auction, in which cloud users come and go on the fly, without *a priori* notification; demands from newly arrived users are addressed instantly. (2) Expressive bidding language and dynamic VM assembly. Users' elastic demands are expressed and heterogeneous types of VMs are assembled and allocated on the fly, tailored to the demands and resource availability, guaranteeing that each user's utility never decreases over time. (3) Truthfulness, the quintessential property in auction mechanism design, which elicits voluntary truthful VM valuation reports from selfish cloud users. As a result, the cloud provider is guaranteed to receive correct information based on which optimization decisions are made. Besides truthfulness in VM valuations, RSMOA further guarantees truthfulness in demand arrival times. (4) Simultaneous optimization of the cloud provider's revenue and system-wide social welfare. Our online auction achieves a competitive ratio $O(\ln p)$ in both social welfare and provider revenue, when benchmarked against the well-known offline Vickrey-Clarke-Groves (VCG) auction. Here p is the ratio between the upper and lower bounds of users' marginal valuation of any type of resource.

More specifically, our contributions along the design of RSMOA are three-fold:

First, we identify a set of necessary conditions (namely bid independence, bundle monotonicity, and user-utility-maximizing allocation) for an auction to achieve individual rationality and truthfulness in both VM valuation and demand arrival times. Bid independence isolates the price that a user pays from her own bid, such that the user cannot manipulate her bid to achieve a higher utility. With bundle monotonicity, larger demand bundles containing larger amounts of resources are priced no lower than smaller ones. According to user-utility-maximizing allocation, the cloud provider allocates resources to maximize each user's utility, instead of her own revenue. While this appears to contradict provider revenue maximization, it is proven necessary for truthfulness, which eventually leads to maximal provider revenue together with other guarantees of the auction design.

Second, we design an online auction framework that satisfies the above three conditions. Upon arrival of a user's bid expressed using a number of XORed bundles, the provider identifies the bundle that maximizes the user's utility, and charges the user according to a pricing curve. The pricing curve maintains the supply-and-demand relationship across the board, providing a higher price per unit resource when more resources are consumed. Upon termination of a user's VM usage, the resources are returned to the cloud pool and re-allocated to other existing users. The adjustment of resources to the existing users guarantees a non-decreasing utility at each user, and priorities for assigning more resources are given to users who arrive early.

Third, we carefully design the explicit form of the pricing curve following a threat-based strategy, which targets a competitive ratio c of the provider's revenue by setting prices that allow resource transactions only when necessary, *i.e.*, when not selling the resource leads to violation of the targeted competitiveness of the auction mechanism. We formulate a set of differential equations to describe this threat-based strategy, and solve them to derive the closed form of the pricing curve. The best competitive ratio is found to be $c = O(\ln(p))$, according to a set of necessary, boundary conditions that guarantees correct strategy execution. Though this competitive ratio is computed for the provider's revenue, we prove that, interestingly, the same competitiveness is achieved in terms of social welfare.

In the rest of the paper, we discuss related work in Sec. II, and define the system model in Sec. III. Sec. IV and Sec. V present the framework of RSMOA and the design of the pricing curve, respectively. Simulation results are presented in Sec. VI. Sec. VII discusses possible extensions and future work, as well as concludes the paper.

II. Related Work

Auction as an efficient resource pricing and allocation method has been extensively studied in a number of fields, including in particular cloud computing. The celebrated VCG auction mechanism [8] constitutes the only type of auction that concurrently achieves truthfulness and maximum social welfare, by directly solving the social welfare maximizing allocation and charging each user the opportunity cost she brings. The VCG mechanism can be efficiently applied only if two conditions are met: all required information is available, and the optimal solution can be calculated efficiently (in polynomial time). When the social welfare maximization problem at the auctioneer involves online decision making or is NP-hard in nature, the VCG mechanism becomes impractical [9].

One solution to overcome the computational difficulty is to design an approximation algorithm for solving the underling social welfare maximization problem, and a customized payment rule that works in concert with the approximation algorithm. Note that the payment rule in VCG auctions, charging an opportunity cost, works with only some approximation algorithms, and loses truthfulness in general [10]. Although there is no universal truthful payment rule, some instructive ideas have been investigated, for example by exploiting the concept of critical bids [11], or using the LP decomposition technique [12] if the underlying social welfare maximization problem exhibits a packing or covering structure [7].

Another solution approach attempts to first decide the payment rule instead, and then seek a good approximation ratio with the allocation algorithm design by fine-tuning the payment rule. Along this direction, Goldberg *et al.* [13] propose an auction that sells single items according to a threshold-based pricing rule. Ravi *et al.* [14] extend such threshold-based payment rule to a *pricing curve*-based solution, and their auction is applicable to more general types of goods.

Online auctions bring a new dimension of challenges into auction mechanism design, due to the lack of future information to solve the underlying resource allocation problem. It is in general difficult to design a payment rule with an online approximation algorithm that achieves nice properties [15]. For example, truthfulness is usually compromised when an auction is extended from a single round to multiple rounds in a straightforward fashion [16]. A pricing-curve based method is advantageous in the online scenario, since the pricing curve maintains global information over time, and payments can be naturally computed round by round.

Recently, a number of auctions have been designed for VM provisioning in cloud computing, using techniques mentioned above. Zhang *et al.* [7] design a truthful single-round auction using the LP decomposition technique. Wang *et al.* [5] apply the critical bid approach, and add a collusion-resistant property to their single-round auction. Zhang *et al.* [6] utilize the pricing-curve method, and design a truthful online auction for single-type VMs and different types of users. Our mechanism, RSMOA, distinguishes itself in four aspects: (1) Our VM allocation is efficient for arbitrary patterns of user valuation, instead of specified types of users. (2) Our auction is combinatorial, and applies to multiple types of VMs, which are assembled on demand. (3) The users in [6] must reveal their departure times to the provider, while no such information is needed in RSMOA, allowing users to leave at any time without *a priori* notification. (4) Not only social welfare, but also the cloud provider's revenue are approximately maximized in RSMOA. Most existing work target at maximizing only social welfare or provider's revenue [17], but not both.

III. System Model and Definitions

A. Auction Model

We consider a cloud provider who owns a pool of R types of resources (*e.g.*, CPU, RAM, disk) that can be dynamically assembled into M different types of virtual machines (VMs),

$VM_1, \cdots, VM_M$. Let $[X]$ be the set of integers $\{1, 2, \dots, X\}$. One instance of VM_m is constituted by $\alpha_{m,r}$ units of type-r resource, for all $r \in [R]$. There are A_r units of resource r in total, and $A = \sum_{r \in [R]} A_r$ is the overall number of units of all resources. The cloud operates over a continuous, potentially large time interval $[0, T]$.

There are N users in the cloud, where user n ($n \in [N]$) learns her VM demand at time t_n. Without loss of generality, assume $t_1 \leq t_2 \leq \dots \leq t_N$. The valuation of user n for a possible VM bundle $\boldsymbol{d}_n = (d_{n,1}, \cdots, d_{n,M})$ is $b_n(\boldsymbol{d}_n)$ per time unit. Here $d_{n,m}$ represents the number of VM_m instances in the bundle. $b_n(\boldsymbol{d}_n)$ is user n's valuation function, mapping a possible VM bundle to a real value. A user can express her valuation function by enumerating the values corresponding to all the possible VM bundles, or more succinctly using a formula that reflects the need of her job. Consider an example system with two types of VMs. User n running a MapReduce job requires several instances of VM_1 for the mapping tasks and VM_2 for the reducing tasks. Assume that from past experience, the number of instances of VM_1 should be 3 times that of VM_2. Her valuation for different bundles of the two types of VMs is: $b_n(3, 1) = 6, b_n(6, 2) = 9, b_n(9, 3) = 10$. The valuation function can hence be expressed as:

$$b_n(3x, x) = 10 - (3 - x)^2, \text{for } 1 \leq x \leq 3$$

In the online auction, user n who knows her VM demand at time t_n sends her bid $b_n(\cdot)$ to the provider. Here we assume she bids her real valuation function $b_n(\cdot)$, and later we will show that this assumption is reasonable for any rational users, by proving truthfulness in VM valuation. Upon receiving user n's bid, and prior to opening the next user's bid, the provider decides the bundle $\boldsymbol{d}_n(t)$ for allocation to user n at time t, along with a per-time-unit price $\pi_n(t)$. User n uses the allocated resources until she decides to leave and terminate the VMs at time $\bar{t}_n$.

The provider does not know the arrival and departure times t_n and $\bar{t}_n$ of each user in advance. A user's true valuation (the valuation function) is private information, and can be different from her bid. The user's utility per time unit is the difference between her valuation and the payment on the allocated bundle $\boldsymbol{d}_n(t)$:

$$u_n(t) = b_n(\boldsymbol{d}_n(t)) - \pi_n(t) \tag{1}$$

During user n's residence time $[t_n, \bar{t}_n]$, the provider can adjust the bundle $\boldsymbol{d}_n(t)$ of VMs allocated to the user, as well as the price $\pi_n(t)$, under the guarantee that the user's utility never decreases due to such adjustment. The adjustment is typically done when more sources become available, and is reasonable due to the elastic demand of each user, as expressed in her valuation function.

At the provider side, the total amount of resources from the VMs provided to all the users at any time cannot exceed the resource capacity:

$$\sum_{n \in [N]} \sum_{m \in [M]} d_{n,m}(t) \alpha_{m,r} \leq A_r, \forall r \in [R], t \in [1, T] \tag{2}$$

We assume that the provider can gain a residual value $\underline{p}$ per time unit if a unit of resource r is not used, which can be considered as the operational cost that is saved. Note here we adjust the units of different types of resources so that all types of resources have a uniform residual value $\underline{p}$.[1] The revenue of the provider at time t is the sum of users' payments and the overall residual value for all the unallocated resources:[2]

$$u_P(t) = \sum_{n \in [N]} \pi_n(t) + \sum_{r \in [R]} \underline{p} \widetilde{A}_r(t) \tag{3}$$

Here $\widetilde{A}_r(t)$ is the remaining units of resource r at time t:

$$\widetilde{A}_r(t) = A_r - \sum_{n \in [N]} \sum_{m \in [m]} \alpha_{m,r} d_{n,m}(t) \tag{4}$$

The achieved social welfare at time t is the sum of the provider's revenue and the users' utilities:

$$S(t) = \sum_{n \in [N]} b_n(\boldsymbol{d}_n(t)) + \sum_{r \in [R]} \underline{p} \widetilde{A}_r(t) \tag{5}$$

We assume the marginal valuation of any type of resource r of any user is lower bounded by $\overline{p}$ (intuitively, the valuation of the resource should be no lower than the residual value when the resource is not sold), and has an upper bound $\overline{p}$ as well, *i.e.*,

$$\sum_{r \in [R]} \alpha_{m,r} \underline{p} \leq \frac{\partial b_n(\boldsymbol{d}_n)}{\partial d_{n,m}} \leq \sum_{r \in [R]} \alpha_{m,r} \overline{p}.$$

We denote the ratio between the upper bound and the lower bound by $p = \overline{p} / \underline{p}$. We also assume $b_n(\cdot)$ is concave and non-decreasing on the number of any type of VMs, *i.e.*, $\frac{\partial^2 b_n(\boldsymbol{d}_n)}{\partial d_{n,m}^2} \leq 0$, $\frac{\partial b_n(\boldsymbol{d}_n)}{\partial d_{n,m}} \geq 0$.

For ease of reference, we summarize important notation in Table I.

B. Economic Properties

The following properties are pursued in our auction design.

(i) Non-decreasing user utility. The solution space in our auction design includes re-adjusting the bundle allocated to a user n at a time $t \in (t_n, \bar{t}_n)$. A corresponding desirable property that such re-adjustment should satisfy, for encouraging user participation, is that a user always gains a higher utility from the new bundle allocated and is hence happy with the adjustment. Formally, we require $\forall t_1 > t_2 \in [t_n, \bar{t}_n]$, $u_n(t_1) \geq u_n(t_2), \forall n \in [N]$.

We continue with our MapReduce example to illustrate this property. Suppose user n is allocated 6 instances of VM_1 and 2 instances of VM_2 upon bidding, at a per-unit-time price 6. Her utility is therefore $9 - 6 = 3$. At a later time point, the provider decides to give her 3 more instances of VM_1 and 1

[1] For instance, suppose 1 CPU unit has a residual value \$1, and 1 GB memory has a residual value of \$4. We then use 0.25GB as the unit of memory, and the residual values of CPU and memory are now both \$1.

[2] Another way to formulate the provider's revenue: Let $\underline{p}$ be the unit operational cost of the resources. The provider's revenue is more naturally expressed as $\sum_{n \in [N]} \pi_n(t) - \sum_{r \in [R]} \underline{p}(A_r - \widetilde{A}_r(t)) = \sum_{n \in [N]} \pi_n(t) - \sum_{r \in [R]} \underline{p} A_r + \sum_{r \in [R]} \underline{p} \widetilde{A}_r(t)$. Here $\sum_{r \in [R]} \underline{p} A_r$ is a constant, removing which we obtain (3). Using the revenue formulation in (3) and interpreting $\underline{p}$ as the residual value instead, provide us convenience in proving properties of our auction, as well as connect better to the pricing curve design in Sec. V.

TABLE I. NOTATION

N	# of users	$[X]$	integer set $\{1, 2, \ldots, X\}$
M	# of VM types	t_n	user n's bidding time
R	# of resource types	$\bar{t}_n$	user n's leaving time

T	total time length
VM_m	VM type m
$\alpha_{m,r}$	# of resource r required by VM_m
A_r	available units of resource r
A	total available units of all resources
$\widetilde{A}_r(t)$	total units of remaining resource r at time t
$\pi_n(t)$	user n's payment per time unit at time t
$u_n(t)$	user n's utility per time unit at time t
$\underline{p}$	lower bound of marginal valuation of any resource, as well as the provider's residual value
$\overline{p}$	upper bound of marginal valuation
p	ratio between the upper and lower bounds of the marginal valuation
D	set of all the possible bundles
$\boldsymbol{d}_n(t)$	user n's allocated bundle at time t
$d_{n,m}(t)$	# of VM_m in user n's bundle at time t
$b_n(\boldsymbol{d}_n)$	user n's valuation function
τ	queue containing all the current users
$u_P(t)$	the provider's revenue at time t
$S(t)$	social welfare at time t
c_S	social welfare competitive ratio
c_R	provider revenue competitive ratio

more instance of VM_2, at a price 6.5 per unit time for all the $9+3$ instances. The user's new utility becomes $10-6.5 = 3.5$.

The valuation function allows a user to express her elastic demand on the resources. For example, a user can configure the upper bound of resources desired, by having the valuation function reach a plateau at the upper bound, such that further excess supply results in zero valuation increment. In this way, a user can impose a cap on the amount of received resources by specifying her valuation function accordingly.

We would like to emphasize that the non-decreasing utility is a property achieved by our algorithm, rather than a constraint or restriction imposed on the model. The design of online algorithms typically enables dynamic adjustment over time in order to improve the algorithm efficiency. Existing online mechanisms, such as the Amazon Spot Instance market [4], may terminate the leases at any time and bring significant uncertainty to the users. We seek to improve the existing mechanisms by providing guarantee of basic resources to the users, while retaining flexibility in adjusting resource allocation, which is essential to the efficiency of the online auction. By guaranteeing a non-decreasing utility (calculated based on the valuation function), the dynamic adjustment of allocated resources is welcomed by a user who targets utility maximization.

(ii) Individual Rationality. An online auction is individually rational if no user has a negative utility over her staying time: $\forall n \in [N], t \in [t_n, \bar{t}_n]$, $u_n(t) \geq 0$. This property provides a basic incentive for users to participate in the auction.

(iii) Truthfulness in VM Valuation. Each user in the auction achieves her maximum utility if she bids her valuation $b_n(\cdot)$ truthfully regardless of other users' bids. The user's utility gain by reporting $b_n(\cdot)$ should be no smaller than the utility gain with a false bid $b'_n(\boldsymbol{d}_n) \neq b_n(\boldsymbol{d}_n)$.

(iv) Truthfulness in Demand Arrival Time. A user may try to delay her bid upon learning her resource demand, in order to maximize her total utility across her staying in the cloud. Truthfulness in demand arrival time guarantees that bidding at the time user n learns her demand, *i.e.*, t_n, maximizes her total utility.

(v) Competitiveness in Social Welfare. We compare the overall social welfare of the online auction with the social welfare under the offline VCG auction, which is the optimal offline social welfare [8]. The offline VCG auction calculates the maximum offline social welfare by collecting all the bids over $[0, T]$ and find the optimal allocation satisfying all the constraints introduced in Sec. III-A and guaranteeing a non-decreasing utility at each user. Suppose the social welfare at t under the offline VCG is $S_{vcg}(t)$. Denote the total social welfare under our auction mechanism and the offline VCG auction by S and S_{vcg} respectively. The supremum of

$$c_S = \frac{S_{vcg}}{S} = \frac{\int_0^T S_{vcg}(t)dt}{\int_0^T S(t)dt}$$

is referred to as the social welfare competitive ratio of the online auction.

(vi) Competitiveness in Provider's Revenue. We compare the provider's revenue from the online auction with the revenue under the offline VCG auction as well. Suppose the revenue at t under the offline VCG is $u_{Pvcg}(t)$. The supremum of $c_R = \frac{u_{Pvcg}}{u_P} = \frac{\int_0^T u_{Pvcg}(t)dt}{\int_0^T u_P(t)dt}$ is the revenue competitive ratio of our online auction.

IV. ONLINE AUCTION DESIGN

A. Conditions on Truthfulness and Individual Rationality

We start our auction mechanism design by investigating conditions that must be satisfied to guarantee truthfulness of the online auction in both the VM valuation and demand arrival time, as well as individual rationality.

Assume the provider calculates user n's payment $\pi_n(t)$ by a function $\pi_n(b_n(\cdot), \boldsymbol{d}_n, t)$, which is dependent on user n's valuation function, her received bundle, and the current time, (and implicitly the bids of previously arriving users). Here we slightly abuse the notation π_n to denote both the user's actual payment, and the payment calculation function used by the provider. We will show that, for truthfulness in VM valuation, the payment $\pi_n(b_n(\cdot), \boldsymbol{d}_n, t)$ should be independent from a user's bid $b_n(\cdot)$, to prevent the user from influencing the payment by modifying her bid. Therefore, the payment calculation function can be simplified to $\pi_n(\boldsymbol{d}_n, t)$. Another intuitive requirement on the payment is that it should be bundle-monotonic, *i.e.*, larger bundles with more resources are priced higher. The idea is simple: if smaller bundles are more expensive, the mispricing will be exploited by dishonest users. After a pricing function $\pi_n(\boldsymbol{d}_n, t)$ is established, the exact bundle $\boldsymbol{d}_n(t)$ to be allocated to user n can be decided based on the pricing function and the user's valuation. A general rule in truthful auction design is to allocate the amount of resource which maximizes the user's utility. Since maximizing her utility is a user's objective, such a mechanism encourages users to bid truthfully by giving them the maximum utility. We give the definitions of the above three requirements and prove them as necessary conditions for truthfulness in VM valuation in Theorem 1. The detailed proof can be found in Appendix A.

Definition 1: (Bid-independent Payments) Let $\pi_n(b_n(\cdot), \boldsymbol{d}_n, t)$ denote the payment of user n at time t if receives bundle $\boldsymbol{d}_n$ by bidding $b_n(\cdot)$. The payment is bid-independent if for any two different bids $b_n(\cdot) \neq b'_n(\cdot)$ which lead to the same bundle allocation result, the payments are always the same, *i.e.*, $\pi_n(b_n(\cdot), \boldsymbol{d}_n, t) = \pi_n(b'_n(\cdot), \boldsymbol{d}_n, t)$.

When bid-independent, the payment only depends on a user's allocated bundle and previous user bids.

Definition 2: (Bundle-monotonic Payments) Define $\boldsymbol{d}_n \geq \boldsymbol{d'_n}$ if $d_{n,m} \geq d'_{n,m}, \forall m \in [M]$. A bid-independent payment is further called bundle-monotonic if $\pi_n(\boldsymbol{d}_n, t) \geq \pi_n(\boldsymbol{d'_n}, t), \forall \boldsymbol{d}_n \geq \boldsymbol{d'_n}$.

Definition 3: (Utility-maximizing Allocation Rule) An allocation rule is utility-maximizing if the provider always calculates a payment function $\pi_n(\boldsymbol{d}_n, t)$ independent of user n's bid, and decides the VM allocation by finding the bundle $\boldsymbol{d}^*_n(t)$ which maximizes this user's utility $u_n(t)$, *i.e.*,

$$\boldsymbol{d}^*_n(t) = \arg\max_{\boldsymbol{d} \in D}(b_n(\boldsymbol{d}_n) - \pi_n(\boldsymbol{d}_n, t)) \quad (6)$$

where D is the set of all possible VM bundles.

In the special cases where the maximum utility is negative, the provider allocates an empty bundle to the user (resulting in zero utility).

Theorem 1: (Necessary Conditions on Truthfulness in VM valuation) For any VM valuation truthful deterministic online auction, its payment must be bid-independent, bundle-monotonic, and the allocation should be utility-maximizing.

Next we study truthfulness in the demand arrival time. To incentive users to bid as soon as their demands arrive (t_n), the auction mechanism should "punish" users who postpone their bidding, by charging higher prices for the same bundles. We define a time-monotonic payment as follows, and prove that by adding this requirement, we have identified a set of sufficient conditions for an online auction to be truthful in VM valuation and demand arrival time, as well as individually rationality, in Theorem 2, with a detailed proof in Appendix B.

Definition 4: (Time-monotonic Payment) The payment is time-monotonic if user n never gets a lower price for the same bundle $\boldsymbol{d}_n$ by delaying her bidding time: if user n bids at time t_1, her payment is $\pi_n(\boldsymbol{d}_n, t)$ at time $t \geq t_1$; if she bids at time $t_2 > t_1$, her payment at time $t \geq t_2$ is $\pi'_n(\boldsymbol{d}_n, t) \geq \pi_n(\boldsymbol{d}_n, t)$. Given that the users are differentiated by their arrival times, the property is equivalent to the following: for any users $n_1 < n_2, n_1 \in [N], n_2 \in [N]$, $\pi_{n_1}(\boldsymbol{d}, t) \leq \pi_{n_2}(\boldsymbol{d}, t)$, $\forall t \in [0, T], \forall \boldsymbol{d} \in D$.

Theorem 2: (Sufficient Conditions on Truthfulness and Individual Rationality) An online auction with a bid-independent, bundle-monotonic, time-monotonic payment and a utility-maximizing allocation rule is truthful in VM valuation, truthful in demand arrival time, and individual rational.

B. Online Auction Mechanism

Based on Theorem 2, we are able to design an online auction framework that achieves the three important properties: At time t upon receiving a user n's bid, the provider prepares a payment function $\pi_n(\boldsymbol{d}_n, t)$ that depends on the bundle to be allocated to user n ($\boldsymbol{d}_n$) and all previous user bids before t (but independent of user n's bid), and is bundle-monotonic and time-monotonic. Then the provider selects the bundle $\boldsymbol{d}^*_n$ that maximizes user n's utility, and determines her payment by $\pi_n(\boldsymbol{d}^*_n, t)$.

The key lies in designing the payment function. To obtain a payment function that is bundle-monotonic, we first evaluate the total number of units of resources that VMs in the bundle consume, as $x(\boldsymbol{d}_n) = \sum_{m \in [M]} \sum_{r \in [R]} d_{n,m}\alpha_{m,r}$. We then set up a marginal price function $P(x) : \mathbb{R}^+ \to \mathbb{R}^+$, which gives the marginal price of one more unit of resource, when x units of resource have been allocated. The payment function is defined as: $\pi_n(\boldsymbol{d}_n, t) = \int_{x_0}^{x_0 + x(\boldsymbol{d}_n)} P(y)dy$, where x_0 is the total number of units of allocated resources before allocating bundle $\boldsymbol{d}_n$ to user n. The payment function is apparently bid-independent and bundle-monotonic.

The detailed design of the pricing curve, $P(x)$, which is a core component of RSMOA, is presented in the next section. For now, we know that $P(x)$ is non-decreasing, such that: (1) a lower marginal price $P(x)$ for smaller x leads to a lower payment for a user who arrives earlier, which implies time monotonicity; (2) when more units of resources have been allocated (*i.e.*, x is larger), a higher marginal price is desirable, such that VM bundles are allocated to users who value them more – a common technique in resource allocation to pursue competitiveness in provider revenue and social welfare. Thus the payment mechanism helps achieve a higher overall valuation (and also revenue) when the resources are constrained.

To pursue competitiveness, our online auction also enables dynamical adjustment of bundles (prices) allocated (charged) to the existing users who have arrived earlier, upon arrival of a new user or departure of a user at time t. Specifically, users are maintained in a queue τ: a user is added to the tail of τ upon arrival, and removed from the queue when she departs. The provider always adjusts bundle allocation to existing users according to their order in queue τ (the oldest user is handled first), in order to guarantee time monotonicity in Definition 4 (*i.e.*, at any time t, the price for the same bundle to a user who arrived earlier is lower). For each user, the best bundle which maximizes her utility is selected, and the payment is computed using the payment function, according to how many units of resources have been allocated so far.

The non-decreasing pricing curve $P(x)$, combined with the handling order τ, also guarantees a non-decreasing utility at each user. Consider when a user n' is removed from τ, then for any user n who is after n' in τ, her request is handled earlier than before. Consequently her payment is the integral of $P(x)$ on smaller value of x, and with smaller value of $P(x)$. So assuming her receiving bundle is unchanged, her payment is non-increasing and her utility is non-decreasing. Further note that the provider always finds the user's utility-maximized bundle among all the possible bundles, which include her original receiving bundle. As a result, the utility after adjustment is non-decreasing.

Our online auction mechanism, RSMOA, is summarized in Alg. 1. Theorem 3 proves that RSMOA achieves all the desired properties except competitiveness, which we will show in the next section. The detailed proof is given in Appendix C.

Algorithm 1 RSMOA: The Online Auction Mechanism

```
Do the following at $t$ if there is user arrival or departure:
Update queue $\tau$
Initialize the amount of allocated resource $x_0 \leftarrow 0$
**for all** user $n$ in queue $\tau$ **do**
    Prepare the payment function $\pi_n(\boldsymbol{d_n}) = \int_{x_0}^{x_0+x(\boldsymbol{d_n})} P(y)dy$
    Compute the best bundle to be allocated to user $n$ $\boldsymbol{d}_n^* \leftarrow \arg\max_{\boldsymbol{d}_n \in D}\{b_n(\boldsymbol{d}_n) - \pi_n(\boldsymbol{d}_n)\}$
    Compute user $n$'s payment $\pi_n(\boldsymbol{d}_n^*, t)$
    Update the amount of allocated resource $x_0 \leftarrow x_0 + x(\boldsymbol{d}_n^*)$
**end for**
```

Theorem 3: RSMOA, as described in Alg. 1, is truthful in VM valuation, truthful in demand arrival time, individually rational, and guarantees a non-decreasing utility at any user over time.

V. Pricing Curve and Competitive Ratio Analysis

In this section, we show that the efficiency of Alg. 1 can be achieved by carefully designing the global, non-decreasing pricing function $P(x)$. We design $P(x)$ that guarantees a competitive ratio $O(\ln p)$ in the provider's revenue as well as in social welfare, based on a *threat-based* approach proposed in recent literature [14].

Assume we target a competitive ratio c between the provider's revenue obtained with Alg. 1 and that with the offline VCG mechanism. Recall in Alg. 1, the provider sells resource units (in the form of VM bundles) at the non-decreasing marginal price $P(x)$ per extra unit. With the threat-based approach, $P(x)$ is set in a way that the next units of resources are sold to a user (in the form of a VM bundle, when the user's valuation of the bundle is no lower than the payment the provider is asking for), only if not selling these units leads to an immediate violation of the competitive ratio c in the provider's revenue, in the case that the auction immediately terminates. In other words, $P(x)$ should lead to rejection of a user unless her bid is so high that not accepting this bid results in a threat to the revenue competitiveness of the algorithm. We only consider such immediate threats (*i.e.*, the target competitive ratio will be violated if the auction immediately terminates), because only in this scenario can the competitive ratio be calculated accurately with hitherto available information. Otherwise, if we wish to consider the competitive ratio at a future time, we need assumptions about the future events, which complicates the strategy and is less practically feasible.

Such a threat-based approach is a conservative strategy in maintaining a target competitive ratio c. Intuitively, a more aggressive strategy is to set the prices so that more resources are sold when a user's valuation is higher. There is a catch though: it may mistakenly sell too many resources to a user and miss a better opportunity to sell the same amount of resources with a higher price in the future. How to ensure a good competitive ratio with this strategy requires future studies. In comparison, the threat-based strategy is guaranteed to achieve the target ratio c [14].

In order to derive the mathematical expression of $P(x)$ according to the threat-based strategy, we investigate its inverse function $Q(z) = P^{-1}(z)$.[3] Since $P(x)$ is the marginal price per extra unit of resource when x units have been allocated, $Q(z)$ represents the total number of units of resources allocated when the marginal price reaches z. From the perspective of the cloud provider, designing function $P(x)$ and designing function $Q(z)$ are two equivalent problems: if the provider focuses on $P(x)$, she tries to quote a reasonable price for the next unit of resource; if she focuses on $Q(z)$, she tries to decide the additional number of units sold to users, when the marginal price increases. We next focus on deriving $Q(z)$ by setting up a number of equations on $Q(z)$, according to the threat-based strategy. We use $V(z)$ to denote the total payment collected by the provider for $Q(z)$ units of resources sold, when the marginal price goes up to z. We derive $Q(x)$ based on the following conditions:

$$Q(z) = 0 \text{ and } V(z) = 0, \forall z \le c\underline{p} \tag{7}$$

Recall that the provider gains a residual value $\underline{p}$ for each unit of unallocated resource. If the marginal price z is not higher than $c\underline{p}$, the provider does not need to allocate any unit of resource, since the competitive ratio c will not be violated anyway: the revenue of selling all the resources is no higher than $Ac\underline{p}$, but not selling anything can achieve a total residual value $A\underline{p}$.

$$V'(z) = zQ'(z), \forall z \in [c\underline{p}, \overline{p}] \tag{8}$$

The number of resource units sold at marginal price z is $Q'(z)$, and the product of z and $Q'(z)$ is the additional payment $V'(z)$.

$$zA/c = (A - Q(z))\underline{p} + V(z), \forall z \in [c\underline{p}, \overline{p}] \tag{9}$$

When the marginal price is z, the worst case is that all the resources, at the total number of units A, are sold at price z. To maintain a competitive ratio c, the overall revenue gleaned by the provider should be at least $1/c$ fraction of the revenue collected under VCG. The marginal price reaching z shows that there exists a user, who is the last user under our immediate termination assumption, valuing one more unit of resources by no more than z (recall the assumption on non-increasing marginal valuation). None of the previous users has a higher marginal valuation, since $P(x)$ is non-decreasing. So the total revenue collected under VCG is no more than Az. Therefore by setting the revenue target Az/c equal to the sum of total payment $V(z)$ and residual value of the remaining resources $(A - Q(z))\underline{p}$, we can guarantee a competitive ratio c when comparing our mechanism to VCG.

$$Q(\overline{p}) = A \tag{10}$$

All the resources should have been sold at the upper bound of the marginal price.

Solving these four groups of equations (7)-(10), we can derive the solution of $Q(z)$ as follows (detailed steps are given

[3] Since $P(x)$ may not be strictly increasing, we define $P^{-1}(z)$ as the maximum value y satisfying $z = P(y)$

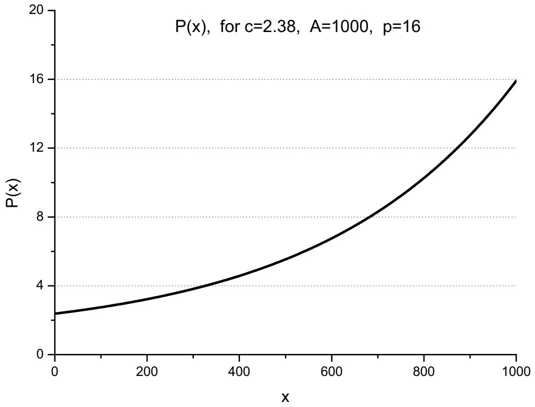

Fig. 1. An example of $P(x)$, where $\underline{p} = 1, p = 16$, $c = 2.38$, $A = 1000$.

in Appendix D):

$$Q(z) = A \int_{c\underline{p}}^{z} \frac{1}{c(y - \underline{p})} dy, z \in [c\underline{p}, \overline{p}] \tag{11}$$

Then we obtain the marginal pricing function $P(x)$ by inverting $Q(z)$:

$$P(x) = \underline{p}(1 + (c-1)e^{cx/A}) \tag{12}$$

The pricing function is non-decreasing and concave, *i.e.*, the marginal price increases more significantly when less available resource remains. Fig. 1 shows an example of the pricing curve $P(x)$, where $\underline{p} = 1, p = 16, A = 1000$, and $c = 2.38$. The marginal price starts from $c\underline{p} = 2.38$, and increases exponentially to the upper bound $\overline{p} = 16$, with the increase of the number of allocated resource units.

Applying such a pricing function in Alg. 1 guarantees a competitive ratio c in the provider's revenue, achieved by our online auction. We seek to derive the best ratio c under which the threat-based strategy remains feasible, which is given as the solution to the following equation (detailed steps are given in Appendix D). We can prove the best ratio is $c = O(\ln p)$ (recall $p = \overline{p}/\underline{p}$), since $c = W(\frac{p-1}{e}) + 1$, where $W(\cdot)$ is the product log function, *a.k.a.* the Lambert W function, and $W(n) = O(\ln n)$.

$$c = \ln \frac{p-1}{c-1} \tag{13}$$

Theorem 4 proves that applying the pricing function $P(x)$ in (12), a competitive ratio $O(\ln p)$ is achieved in terms of the provider's revenue. The detailed proof can be found in Appendix E. Surprisingly, though the threat-based strategy focuses on the provider's revenue, the same competitive ratio is achieved in terms of social welfare as well, as stated in Theorem 5, and proved in Appendix F.

Theorem 4: Applying the pricing function $P(x)$ in (12) in Alg. 1, the competitive ratio c_R of the algorithm in the provider's revenue is $O(\ln(p))$, if the resources are not exhausted, *i.e.*, $\widetilde{A}_r(t) > 0, \forall t \in [0, T]$.

Theorem 5: Applying the pricing function $P(x)$ in (12) in Alg. 1, the competitive ratio c_S of the algorithm in social welfare is also $O(\ln(p))$, if the resources are not exhausted, *i.e.*, $\widetilde{A}_r(t) > 0, \forall t \in [0, T]$.

VI. Performance Evaluation

A. Simulation setup

We evaluate our online auction design using trace-driven simulations. We consider 6 types of VMs and 3 types of resources. The configurations of VMs ($\alpha_{m,r}$) are based on the instances of Amazon EC2, as shown in Table II. Users' resource demands are extracted from Google cluster data [18], which is a record of computational tasks submitted to the Google cluster, with information on their resources demands (CPU, RAM, disk). We convert the resource demands into VM demands by calculating the number of instances that make up the same amount of resources. The VM demand constructed in this way corresponds to one VM bundle of a user in our model. Since a user's demand is elastic in our model, we create elastic demand of a user in the following manner: Suppose the basic VM bundle needed by user n from the Google data is $\boldsymbol{d}_n$, the user can receive bundles $2\boldsymbol{d}_n$, $3\boldsymbol{d}_n$, $\ldots$, $\lambda\boldsymbol{d}_n$ at different valuations as well, where λ is uniformly distributed in $[1, 5]$. A user by default applies a linear, increasing valuation function $b_n(\cdot)$, with a marginal valuation (*i.e.*, slope of the linear valuation function) uniformly distributed within $[1, p]$. The default number of users is $N = 10000$. The system runs for $T = 10000$. The arrival time t_n of a user is uniformly distributed within $[0, T]$, and the departure time of the user is uniformly distributed within $(t_n, T]$. The total number of units of resources of each type is by default 60% of the overall maximum amount demanded by all users. We run each experiment for 10 times, and present the average result.

TABLE II. VM Instances of Amazon EC2

VM Type	CPU	RAM	Disk
m1.medium	2	3.75GB	410GB
c1.medium	5	1.7GB	350GB
m2.2xlarge	13	34.2GB	850GB
m1.large	4	7.5GB	840GB
m1.xlarge	8	15GB	1.68TB
c1.xlarge	20	7GB	1.68TB

B. Simulation results

We first compare the ratio of the social welfare achieved by the offline VCG auction and that achieved by our online auction in the experiments, with the theoretical worst-case competitive ratio $O(\ln p)$, at different values of p. Fig. 2 shows that the ratio in practice is much better than the worst-case bound, and is close to the optimum 1. We also observe that our online auction performs close to the offline VCG auction regardless of the value of p.

Next we suppose a concave valuation function for each user (with linearly decreasing marginal valuation), instead of a linear valuation function. Fig. 3 compares the ratio between the social welfare achieved by the offline VCG auction and that achieved by our online auction under the two types of valuation functions. The ratio is slightly larger when the concave valuation function is used. With the concave valuation function, the marginal valuation of each additional VM is variable, which makes it more difficult for the algorithm to find an optimal allocation.

We further evaluate the impact of different number of users on the performance of our online auction in Fig. 4, where p is fixed at 8. It can be concluded that the performance of RSMOA

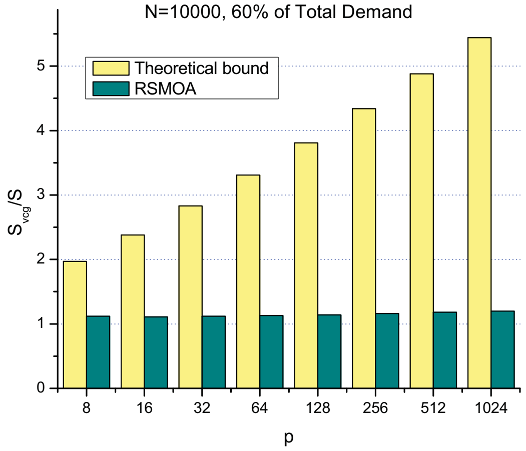

Fig. 2. Social welfare of RSMOA compared with offline VCG auction under different values of p

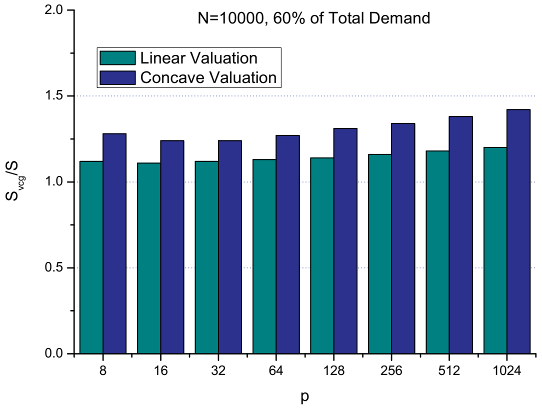

Fig. 3. Social welfare of RSMOA compared with offline VCG auction under concave valuation and linear valuation

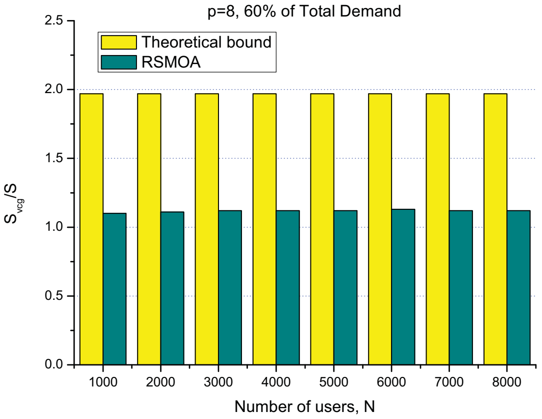

Fig. 4. Social welfare of RSMOA compared with offline VCG auction under different numbers of users

is not affected by the number of the users, which is consistent with our analysis of the worst-case ratio, which is only related to p.

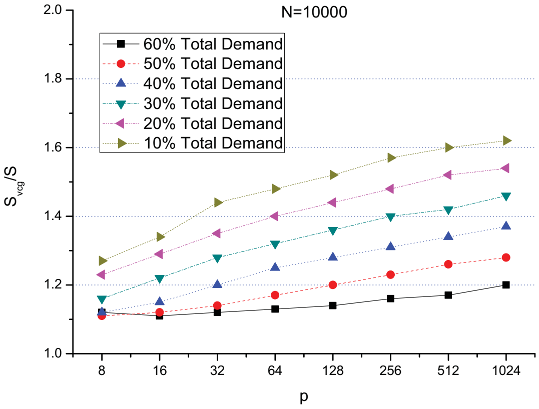

Fig. 5. Social welfare of RSMOA compared with offline VCG auction under different amounts of resources

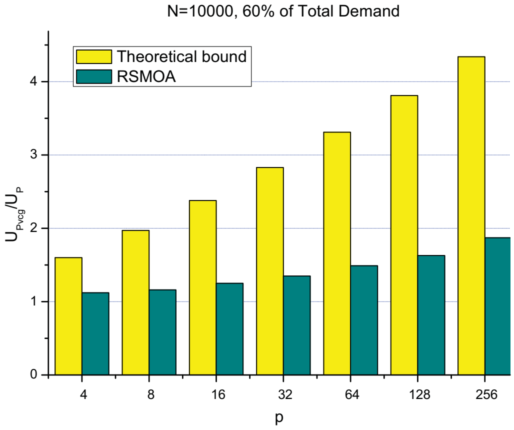

Fig. 6. Provider's revenue of RSMOA compared with offline VCG auction under different values of p

Our theoretical analysis in Theorem 4 and 5 only guarantees a competitive ratio when the total amount of resources at the provider is sufficient. As a supplement to our theoretical results, we evaluate the performance of our auction when the resources are more constrained, by setting the total amount of resources of each type to be $60\%, 50\%, \ldots, 10\%$ of the overall demand of the users. Fig. 5 shows only slight increase of the ratio when the resources are more scarce.

Finally, we compare the provider's revenue achieved with RSMOA with the revenue under the offline VCG auction. Fig. 6 shows that the revenue under our online auction is close to that of the offline VCG.

VII. Conclusion

RSMOA presented in this paper can be applied to other related models. For example, the temporal domain can be finite or infinite, continuous or slotted. Users can modify their valuation at any time during the auction. RSMOA just treats the users with modified valuation as new incoming users, whose utility may decrease. But other users are unaffected and still gain a non-decreasing utility.

RSMOA represents the first online combinatorial auction for dynamic cloud resource provisioning with guaranteed revenue and social welfare. It advances the state-of-the-art of cloud auction design in that all previous VM auction mechanisms either have fixed VM provisioning, or focus on one performance metric only. Our online auction, RSMOA, comprises of two components. First we design an online mechanism based one a set of necessary conditions of the truthful property. Second, we derive the closed form of the critical pricing curve from a threat-based strategy. RSMOA guarantees a competitive ratio of $O(\ln p)$ where p is the ratio between the upper and lower bounds of users' marginal valuation of any type of resource. Trace-driven simulation shows RSMOA achieves near-optimal performance in practical scenarios.

We plan to extend this work to more practical model settings, such as by considering the network bandwidth among VMs, or by considering the problem of packing VMs into physical machines. Furthermore, some impossibility results may complement results in this work, for instance, lower bound results on the competitive ratio in either revenue of social welfare.

Acknowledgment

The research was supported in part by a grant from Hong Kong RGC under the contract HKU 718513 and Wedge Networks Inc and National Sciences and Engineering Research Council of Canada (NSERC).

References

[1] "Amazon Elastic Compute Cloud," http://aws.amazon.com/ec2/.

[2] "Windows Azure: Microsoft's Cloud Platform," http://www.windowsazure.com/.

[3] "Amazon EC2 Pricing," http://aws.amazon.com/ec2/pricing/.

[4] "Amazon EC2 Spot Instances," http://aws.amazon.com/ec2/spot-instances/.

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

[6] 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.

[7] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *Proc. of IEEE INFOCOM*, 2014.

[8] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.

[9] M. H. Rothkopf, A. Pekeč, and R. M. Harstad, "Computationally manageable combinational auctions," *Management science*, vol. 44, no. 8, pp. 1131–1147, 1998.

[10] 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.

[11] D. Lehmann, L. I. Oćallaghan, and Y. Shoham, "Truth revelation in approximately efficient combinatorial auctions," *Journal of the ACM (JACM)*, vol. 49, 2002.

[12] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," in *Proc. of FOCS*, 2005, pp. 595–604.

[13] A. V. Goldberg, J. D. Hartline, and A. Wright, "Competitive auctions and digital goods," in *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2001, pp. 735–744.

[14] R. Lavi and N. Nisan, "Competitive analysis of incentive compatible online auctions," in *Proceedings of the 2nd ACM Conference on Electronic Commerce*. ACM, 2000, pp. 233–241.

[15] N. Buchbinder, K. Jain, and J. S. Naor, "Online primal-dual algorithms for maximizing ad-auctions revenue," in *Proceedings of the 15th Annual European Symposium on Algorithms*, 2007.

[16] N. Nisan, *Algorithmic game theory*. Cambridge University Press, 2007.

[17] W. Wang, B. Liang, and B. Li, "Revenue maximization with dynamic auctions in iaas cloud markets," in *Proc. IEEE ICDCS*, 2013.

[18] *Google Cluster Data*, https://code.google.com/p/googleclusterdata/.

Appendix A
Proof of Theorem 1

Proof: First we argue that bid-independent is necessary for truthfulness in VM valuation. We do this by induction on time t. The payment at the beginning of users' arrival t_n should be independent of users' bids $b_n()$. Otherwise, there are two different bids $b_n(), b'_n()$ such that the bundles allocated at time t_n are the same but the payments are correspondingly $\pi_n(t_n) > \pi'_n(t_n)$. Then the user with real valuation $b_n()$ can make a false bid $b'_n()$ and increase his utility at time t_n. Then he terminates the VMs in a very short time (recall that the user can arbitrarily decides the leaving time without notifying the provider in advance), and his total utility over the staying time is larger than using truthful bid, which contradicts truthfulness. Next we argue that for any $t > t_n$, the payment should still be bid-independent. Similarly there are bids $b_n() \neq b'_n()$ which leads to the same allocating bundles $\boldsymbol{d}(t)$ during time $[t_n, t]$. By induction, the payments before t are the same for both bidders. However the payment at time t is different $\pi_n(t) > \pi'_n(t)$. Again the bidder with valuation $b_n()$ can bid $b'_n()$ and leave the auction at a short time after t with higher total utility.

Second we argue that the larger bundle should never have a lower price. Otherwise, consider bids $b_n(), b'_n()$ which lead to two different bundles $\boldsymbol{d'_n} \geq \boldsymbol{d}_n$, with payments $\pi_n(\boldsymbol{d}_n, t) > \pi_n(\boldsymbol{d'_n}, t)$. But the user's valuation increases with bundle size: $b_n(\boldsymbol{d'_n}) \geq b_n(\boldsymbol{d}_n)$ since the non-negative marginal valuation. Then the user can modify his truthful valuation $b_n()$ to a false bid $b'_n()$ to increase his utility.

Finally we argue that the utility-maximized allocation is also a necessary condition. We have proved that the provider must calculate $\pi_n(\boldsymbol{d}_n)$ without the knowledge of $b_n()$. If the allocation decision $\boldsymbol{d}_n$ does not maximize user's utility, which means there is another bundle $\boldsymbol{d'_n}$ leading to higher utility but not chosen. Then the user can modify the bid to achieve the better allocation result $\boldsymbol{d'_n}$. Since the payment is independent of the user's bid, he can successfully achieve the higher utility. ■

Appendix B
Proof of Theorem 2

Proof: First, the utility-maximized allocation rule guarantees individual rationality. Next we prove the truthfulness in VM valuation: Suppose the user n's true valuation is $b_n()$, and under truthful bid, his bundle at time t is $\boldsymbol{d}^*_n(t)$, with utility $b_n(\boldsymbol{d}^*_n(t)) - \pi_n(\boldsymbol{d}^*_n(t), t)$. Assume he makes a false bid $b'_n()$, and gets bundle $\boldsymbol{d'_n}(t)$. Then the new utility is $b_n(\boldsymbol{d'_n}(t)) - \pi_n(\boldsymbol{d'_n}(t), t)$, which is not larger than the utility under truthful bid, since $\boldsymbol{d}^*_n(t)$ maximizes his utility at time t. So the total utility over time $[t_n, \bar{t}]$ is maximized under truthful valuation.

Finally we prove that the user cannot gain more utility by delaying his bidding time. Suppose user n's real arriving time is t_n, with bundle $\boldsymbol{d}^*_n(t)$ and utility $b(\boldsymbol{d}^*_n(t)) - \pi(\boldsymbol{d}^*_n(t), t)$. If he delays the bid to a later time $t'_n > t_n$. According to the time-monotonic property, for any bundle $\boldsymbol{d}_n(t)$, his payment is no less than the payment under no-delaying bid: $\pi'(\boldsymbol{d}_n(t), t) \geq \pi(\boldsymbol{d}_n(t), t)$. So the maximum utility at time t with delaying bid is not larger than the utility with no-delaying bid:

$$\begin{aligned} & \max_{\boldsymbol{d}_n \in D} \{b(\boldsymbol{d}_n(t)) - \pi'(\boldsymbol{d}_n(t), t)\} \\ \leq \quad & \max_{\boldsymbol{d}_n \in D} \{b(\boldsymbol{d}_n(t)) - \pi(\boldsymbol{d}_n(t), t)\} \\ \leq \quad & b(\boldsymbol{d}^*_n(t)) - \pi(\boldsymbol{d}^*_n(t), t) \end{aligned} \tag{14}$$

So the total utility with no-delaying bid is larger than the utility with delaying bid during $[t'_n, \bar{t}_n]$. ■

Appendix C
Proof of Theorem 3

Proof: Alg. 1 has bid-independent, bundle-monotonic, time-monotonic payment and utility-maximized allocation rule. So Thm. 2 has shown that it achieves truthfulness in VM valuation, truthfulness in demand arrival time and individual rationality. We only need to prove the non-decreasing utility. Notice that a new user arrival does not affect the existing users, since the payment of old users is based on smaller value of x on $P(x)$, and their priorities are not changed. So we only consider the case when user n' leaves the auction. For the same reason, all the users $n < n'$ are not affected. For other users $\forall n > n'$, his payment function lowers on any bundle, compared with before the adjustment, because his priority is promoted. Suppose the payment before adjustment is $\pi(\boldsymbol{d}_n)$, and the payment after adjustment is $\pi'(\boldsymbol{d}_n)$. Then we have $\forall \boldsymbol{d}_n$, $\pi'(\boldsymbol{d}_n) \leq \pi(\boldsymbol{d}_n)$. Assuming his previous bundle is $\boldsymbol{d}^*_n$, and becomes $\boldsymbol{d}^{**}_n$ after adjustment. User n's utility after adjustment is:

$$\begin{aligned} & b(\boldsymbol{d}^{**}_n) - \pi'(\boldsymbol{d}^{**}_n) \\ \geq \quad & \max_{\boldsymbol{d}_n \in D} \{b(\boldsymbol{d}_n) - \pi'(\boldsymbol{d}_n)\} \\ \geq \quad & \max_{\boldsymbol{d}_n \in D} \{b(\boldsymbol{d}_n) - \pi(\boldsymbol{d}_n)\} \\ = \quad & b(\boldsymbol{d}^*_n(t)) - \pi(\boldsymbol{d}^*_n) \end{aligned} \tag{15}$$

, which implies non-decreasing utility. ■

Appendix D
Derivation of $Q(z)$, $P(x)$, and c

Proof: Taking differential on both side of (9) gives us: $A/c = -\underline{p}Q'(z) + V'(z)$. Then substitutes $V'(z)$ here by (8): $A/c = (\overline{z} - \underline{p})Q'(z)$. And we have determined $Q(c\underline{p}) = 0$ in (7). So we derive the expression of $Q(z)$ in (11). $\overline{P}(x)$ is simply derived by calculating the inverse function of $Q(z)$.

Next we use the condition in (10): $Q(\overline{p}) = A/c \cdot \ln(\frac{\overline{p}-\underline{p}}{(c-1)\underline{p}}) = A$. So $e^c = \frac{p-1}{c-1}$, which is exactly (13). ■

Appendix E
Proof of Theorem 4

Proof: We compare the revenue $u_P(t)$ at any time t. For user n let the quantity of resources allocated before him to be x_{old}, and the quantity of his bundle to be $x_n = x(\boldsymbol{d}_n(t))$. The marginal price after him is $P_n = P(x_{old} + x_n)$. Let n^* be the last user in τ that receives a non-empty bundle. Let $b'_n(x)$ denote the marginal valuation of user n's bid: $b'_n(x) = \frac{\partial b_n(\boldsymbol{d})}{\partial x}$. For all n and $x > x_n$, we have $b'_n(x) \leq P_n$. Additionally, for all $n_1 < n_2 \in \tau$, $P_{n_1} \leq P_{n_2}$. So every user in τ values additional resources Δx by no more than $P_{n^*}\Delta x$.

Since VCG auction maximizes user's valuation, there exist a user n', such that the quantity allocated under VCG $x_{n',vcg}$ is larger or equal to $x_{n'}$, which indicates $b'_{n'}(x_{n',vcg}) \leq b'_{n'}(x_{n'}) \leq P_{n^*}$. Notice VCG maximizes total valuation, so for all $n \in \tau$ and $x > x_{n,vcg}$, $b'_n(x) \leq P_{n^*}$. So we conclude that $u_{Pvcg} \leq A \cdot P_{n^*}$. According to the property of $Q(z)$, the online revenue is $A \cdot P_{n^*}/c$. Since at any time, the per time unit revenue has competitive ratio c, the total competitive ratio is also c if the offline VCG result does not satisfy the non-decreasing utility at each user. Adding the non-decreasing utility requirement decreases the actual performance of the VCG auction in our model, thus the competitive ratio is c in terms of revenue. ■

Appendix F
Proof of Theorem 5

Proof: Suppose the set of user bids is σ. Consider a new set of bids σ^* by modifying each user's marginal valuation:

$$b'^*_n(x) = \begin{cases} P_n & \text{if } x \leq x_n \\ b'_n(x) & \text{if } x > x_n \end{cases} \tag{16}$$

The allocations for σ and σ^* are the same because we only modify the bids at parts that are not allocated. Since $b'_n(x) \leq P_n \leq P_{n^*}$, we have $S_{vcg} \leq A \cdot P_{n^*}$. Note the property of threat-based rule: $S(\sigma^*) \geq u_P(\sigma^*) = P_{n^*}/c$. So $S(\sigma^*) \geq S_{vcg}(\sigma^*)/c$.

In order to utilize the above result on σ^*, we need to modify σ to σ^*, in several steps, and each step modifies one user's bid. For user n, if $b'_n(x) > b'^*_n(x)$ at x, then it is decreased to the target. Let σ^n be the bidding set after n modifications. The online auction is affected on the whole decrease part on the valuation curve, so $S(\sigma^n) - S(\sigma^{n+1}) \geq S_{vcg}(\sigma^n) - S_{vcg}(\sigma^{n+1})$. Adding these inequalities together:

$$\begin{aligned} S(\sigma) \quad & \geq \quad (S_{vcg}(\sigma) - S_{vcg}(\sigma^*))/c + S(\sigma^*) \\ & \geq \quad S_{vcg}(\sigma)/c \end{aligned} \tag{17}$$

So the total social welfare over a period of time is also lower-bounded by ratio c. ■