Open access • Proceedings Article • DOI:10.1109/INFCOM.2013.6566946

A framework for truthful online auctions in cloud computing with heterogeneous user demands — Source link

Hong Zhang, Bo Li, Hongbo Jiang, Fangming Liu ...+2 more authors

Institutions: Huazhong University of Science and Technology, Hong Kong University of Science and Technology, Kuwait University, Simon Fraser University

Published on: 14 Apr 2013 - International Conference on Computer Communications

Topics: Cloud computing, Bidding, Common value auction and Resource allocation

A Framework for Truthful Online Auctions in Cloud Computing with Heterogeneous User Demands

Hong Zhang[1] Bo Li[2] Hongbo Jiang[1] Fangming Liu[3] Athanasios V. Vasilakos[4] Jiangchuan Liu[5]
[1]Department of Electronics and Information Engineering, Huazhong University of Science and Technology, China.
[2]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong.
[3]School of Computer Science and Technology, Huazhong University of Science and Technology, China.
[4]Department of Computer Science, Kuwait University, Kuwait [5]School of Computing Science, Simon Fraser University, Canada.
[1]{hongzhangblaze,hongbojiang2004}@gmail.com, [2]bli@cse.ust.hk, [3]fmliu@hust.edu.cn, [4]vasilako@ath.forthnet.gr, [5]jcliu@cs.sfu.ca.

***Abstract*—The paradigm of cloud computing has spontaneously prompted a wide interest in market-based resource allocation mechanisms by which a cloud provider aims at efficiently allocating cloud resources among potential users. Among these mechanisms, auction-style pricing policies, as they can effectively reflect the underlying trends in demand and supply for the computing resources, have attracted a research interest recently. This paper conducts the first work on a framework for truthful online cloud auctions where users with heterogeneous demands could come and leave on the fly. Our framework desirably supports a variety of design requirements, including (1) dynamic design for timely reflecting fluctuation of supply-demand relations, (2) joint design for supporting the heterogeneous user demands, and (3) truthful design for discouraging bidders from cheating behaviors. Concretely speaking, we first design a novel bidding language, wherein users' heterogeneous demands are generalized to regulated and consistent forms. Besides, building on top of our bidding language we propose COCA, an incentive-Compatible (truthful) Online Cloud Auction mechanism based on two proposed guidelines. Our theoretical analysis shows that the worst-case performance of COCA can be well-bounded. Further, in simulations the performance of COCA is seen to be comparable to the well-known off-line Vickrey-Clarke-Groves (VCG) mechanism [11].**

I. Introduction

Cloud computing is meant to offer on-demand network access to configurable computing resources, and promises to deliver to cloud users fast and flexible provisioning of resources with the freedom from long-term investments [10]. Such a paradigm has spontaneously prompted a wide interest in dynamic and market-based resource allocation mechanisms in order to regulate the supply and demand relationship of cloud resources at market equilibrium, and provide satisfactory resource allocation in terms of economic incentives for both cloud consumers and providers [5].

As a quick and efficient approach to selling goods at market value, auction-style pricing polices have been widely applied, reflecting the underlying trends in demand and supply for the computing resources. Indeed, an auction-style pricing policy, so called Spot Instance [1], has been adopted by Amazon to dynamically allocate cloud resources among potential users. Such a design has attracted significant attentions from the research community, and prompted a number of studies [2], [12], [13], [17] focusing on auction-style cloud pricing mechanism design.

A. Design Requirements and Related Work

Typical methods such as [1] and [17] apply a pricing policy that changes periodically, to simplify the cloud provider's operations. On the downside, cloud users often suffer from this simplicity. For example, a cloud user with an uninterruptible job which lasts for more than one period will face the threat of being outbid and losing its cloud usage in any of these periods. Plus, as the price only changes periodically (once per hour; or less frequently in many cases [1], [17]), the fluctuation of supply-demand relations, which is always drastic due to the inherent dynamics and burst nature of user demands, can not be timely and efficiently reflected.

Cloud users often have a variety of application and valuation types (i.e., with heterogeneous demands). For instance, users who have analytic or batch jobs to run are *job-oriented*: they mostly concern about whether their jobs can be finished in time [14], [16]. An SaaS provider provisioning for the peak demand is *resource-aggressive*, attaining enough cloud resources in a specific time interval (rush hours) is of his primary consideration [9], [10]. Existing studies [2], [10], [12] only consider cloud users with a single valuation type for simplicity. Moreover, users' valuations could be multi-minded: a bidder may have a valuation of $10 in total for five VMs, while having a valuation of $8 in total if he get three of them. Current designs [1], [12], [13], [17] can not reflect such complicated form of user demands.

Last but not the least, the cloud market could be vulnerable to selfish user behaviors: cloud users may manipulate auction outcomes and gain unfair advantages via untruthfully revealing their preference on cloud resources. These strategic (or so-called cheating) behaviors will hinder other qualified users, significantly degrade auction efficiency, and greatly discourage users from participation. Truthful designs [12], [13] have been proposed under one-time or periodic auction settings, which are unable to serve cloud users come on-the-fly. [2] tried to investigate truthful auction policies under Bayes-Nash Equilibrium [11] in a spot market model, which is not practical as users are assumed to have only two different valuations.

B. Overview of Our Proposed Framework

This paper conducts the first work on a framework, as shown in Fig. 1, for truthful online cloud auctions where

978-1-4673-5946-7/13/$31.00 ©2013 IEEE

users with heterogeneous demands could come and leave on the fly. First, cloud auctions are all carried out in an *online* manner, i.e., bidders can request cloud resources whenever they need, and their requests are processed by the cloud provider instantaneously. Such flexibility, in accordance with the "pay as you go" cloud paradigm, makes online auctions particularly attractive in practice [6]. Plus, a *bidding language* is implemented in the client side to translate user-specific demands into requests, by which users' heterogeneous demands can be restricted to regulated and consistent forms while the details of the requirements can still be revealed. Finally, each request is then submitted to the server side through web service interfaces, and a *truthful* (also called incentive-compatible) online auction mechanism is implemented so that cloud users can be rationally motivated to reveal their truthful valuations in their requests.

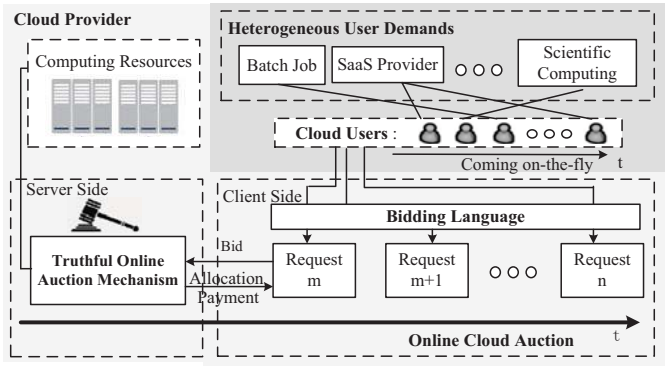


Fig. 1. Infrastructure of the framework for truthful online cloud auctions with heterogeneous user demands

To that end, our first contribution is a novel bidding language for our online cloud auction: We categorize bidders into three typical valuation types, and each of them can be specified by a valuation function, where users' valuations change with respect to the allocation they get. By doing so, each type of the valuation function can then be mapped into a corresponding request form which is more concise and regulated.

Second, we present COCA, a truthful (incentive-compatible) online cloud auction mechanism building on top of our proposed bidding language. The main building blocks of COCA include: (1) a payment-function-based payment rule which is uniquely determined by the allocation result and the request submission time, and (2) an allocation rule that tries to maximize bidders' utility. COCA ensures truthfulness by introducing a nondecreasing auxiliary pricing function in terms of the current supply-demand relations. Further our extensive theoretical analysis shows that the worst-case performance of COCA can be well-bounded. Finally, in simulations the performance of COCA is seen to be comparable to the well-known off-line VCG mechanism.

The remainder of the paper proceeds as follows: Section II introduces our auction model and biding language. Section III presents COCA mechanism for online auctions Section IV conducts the competitive analysis of COCA. Preliminary simulation results are illustrated in Section V. Finally Section VI concludes the paper.

II. Auction Model and Bidding Language

A. Online Auction Model for Cloud Resources

The auction procedure is shown in the left part of Fig. 2, cloud user (bidder) i with a specific valuation v_i (defined below) for the cloud resource arrives at an arbitrary time, maps (translates) his valuation into a request (bid) $\mathbf{r_i}$, and then submits $\mathbf{r_i}$ to the resource provider. After receiving the request, the resource provider is committed to determine the allocation γ_i and the payment pay_i immediately according to the adopted auction mechanism.

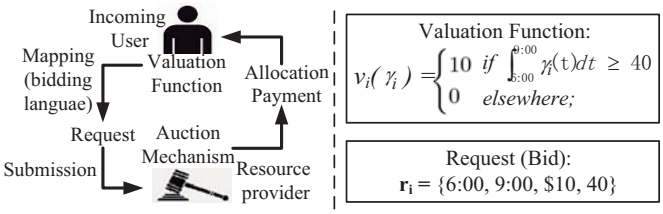


Fig. 2. An illustrative example of the online cloud auction.

Resource—as shown in Fig. 3, we consider one cloud resource provider (e.g., a large data-center) who has a large number of computational resources [1], [10] with a fixed capacity Q in an infinite time interval $[0, \infty]$.

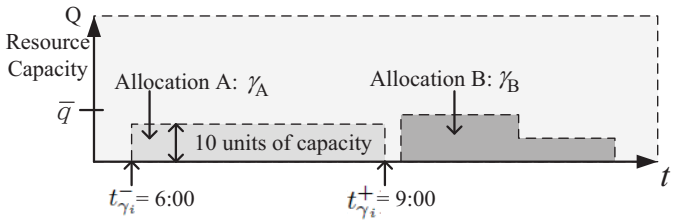


Fig. 3. The cloud resource and the presentation of allocation γ_i.

Allocation & Payment—an allocation γ_i presents how the resources are allocated to a cloud user i. A typical allocation γ_A, as shown in Fig. 3, presents the usage of cloud resource from 6:00 to 9:00 with a fixed 10 units of cloud capacity. As such, if we denote the start time (end time) of an allocation γ_i as $t^-_{\gamma_i}$ ($t^+_{\gamma_i}$) (as is shown in Fig. 3 for allocation A), an allocation $\gamma_i = \bigcup_{t=t^-_{\gamma_i}}^{t=t^+_{\gamma_i}} \gamma_i(t)$ can be regarded as a function of t, where $\gamma_i(t)$ is the instantaneous quantity of resources allocated to the user at time t. Additionally, like γ_B in Fig. 3, the capacity allocated is not necessarily time-invariant, in this paper we assume that $\gamma_i(t)$ can be varying within the range $[0, \overline{q}]$, and we denote all possible allocations to some user i as a set Γ_i. Moreover, we use $\gamma_i^* \in \Gamma_i$ to denote the *allocation decision*: the allocation finally determined for bidder i by the adopted auction mechanism, and we use $pay_i \in \mathbb{R}$ to represent the amount of money user i is required to pay.

Valuation—the valuation of bidder i is a function $v_i : \Gamma_i \to \mathbb{R}$, representing the benefit bidder i obtains from receiving

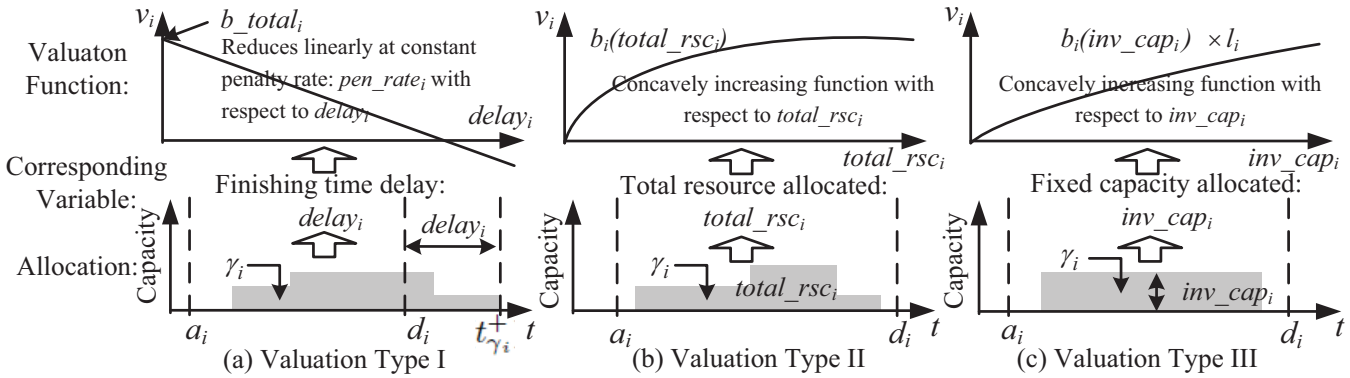


Fig. 4. The valuation function of three types of users.

a certain allocation γ_i of the cloud resource. Note that the valuation is known only to the bidder himself. Consider a cloud user with a job of size 40 (it takes 40 units of resource capacity running for one time unit to finish the job), who has a valuation of \$10 if the job is carried out within [6:00,9:00]. The corresponding valuation function is presented in the right part of Fig. 2.

Utility— the utility $u_i(\gamma_i)$ refers to the "net profit" bidder i gets from an allocation γ_i [11], that is, $u_i(\gamma_i) = v_i(\gamma_i) - pay_i$. As bidders are assumed to be selfish, they may untruthfully reflect their bidding parameters in their requests in order to maximize their utility.

Social welfare— as a commonly used criterion to evaluate the performance (outcome) of an auction mechanism, social welfare refers to the sum of all the valuations of the allocated resources. Specifically, for any request sequence τ, the social welfare is defined as $E(\tau) = \sum_{i \in \tau} v_i(\gamma_i^*)$.

Request—to apply for cloud resources, a cloud user i submits a request $\mathbf{r_i}$ to the provider representing his valuation for the resource. A request is always represented as a set of bidding parameters. Recall the example shown in the right part of Fig. 2, obviously, in this case a concise request form of $\mathbf{r_i} = \{$6:00,9:00,\$10, 40$\}$ is enough to reflect the entire valuation function. Here we call such mapping from valuations to requests as *bidding language*. Besides, we denote the *submission time* of request i (the time i arrives at the market) as t_{sub_i}. Note that we allow users to *reserve* resources, that is, in the above example user i can submit his request at anytime before 6:00.

B. Bidding Language for Heterogeneous User Demands

In this subsection, we turn to the mapping from valuations to requests. In practice, the user valuations are heterogeneous and often have complicated forms in the cloud market. This leads to a dilemma: *how can we present such heterogeneous valuation in requests with a concise and regulated form?* In response, we put forward in this paper a bidding language, by which the representation of requests captures as many features of the heterogeneous demands as possible, while keeping itself concise and consistent.

One challenge here is that a single request type can not reveal the diversity of users' valuation types. As such, after extensively investigating a great number of user requirements in cloud computing [4], [5], [10], [12], we categorize bidders into three typical valuation types, each with a corresponding valuation function. After that, each of the valuation functions is mapped into a corresponding concise request form respectively. As a result, each bidder can translate his specific valuation into a concise request according to his own type.

Valuation TYPE I: Job-oriented users. Analytic and batch jobs account for a large population in the cloud market [4], [10]. Generally, a job-oriented bidder i has a job with size $size_i$, and the job should be carried out within a time period $[a_i, d_i]$ (a_i denotes the earliest available time, and d_i denotes the deadline). Typically, bidder i has a valuation b_total_i if the job is finished before the deadline d_i, otherwise the longer the delay $delay_i$ is, the less the valuation will be [16]. To model this, we assume each bidder has a specific penalty rate pen_rate_i representing his valuation loss per unit delay time. As shown in Fig. 4(a), we can specify the valuation function $v_i(\gamma_i)$ of job-oriented users as:

$$v_i(\gamma_i) = \begin{cases} b\_total_i - delay_i \cdot pen\_rate_i & \text{if } \int_{a_i}^{d_i + delay_i} \gamma_i(t) \geq size_i \\ 0 & elsewhere \end{cases} \tag{1}$$

where $delay_i = max(t^+_{\gamma_i} - d_i, 0)$ uniquely corresponds to a given allocation γ_i. Recall the example shown in the right part of Fig. 2, obviously it belongs to such valuation type with pen_rate_i equals ∞. Accordingly, a request with the form: $\mathbf{r_i} = \{a_i, d_i, pen\_rate_i, b\_total_i, size_i\}$ is capable of reflecting the TYPE I valuation function.

Valuation TYPE II: Resource-aggressive users. For another typical kind of bidders, the biggest concern is to get sufficient number of resources in a specific time period. Such requirement is widely considered in traditional auction market settings, and it is also quite common in the cloud market. As an example, for an SaaS Provider who wants to provision for his peak-load during the rush hours $[a_i, d_i]$ [4], the more resources he acquires within such period, the more benefit he may get. As is shown in Fig. 4(b), such a valuation function v_i of TYPE II can be specified as a nondecreasing function $b_i(\cdot)$, with

respect to the total quantity of resource $total\_rsc_i$ allocated to bidder i within the preferred time period $[a_i, d_i]$. So we have: $v_i(\gamma_i) = b_i(total\_rsc_i)$, where $total\_rsc_i = \int_{a_i}^{d_i} \gamma_i(t)dt$. Accordingly, the request i of TYPE II can be organized as: $\mathbf{r}_i = \{a_i, d_i, b_i(total\_rsc_i)\}$.

Valuation TYPE III: Resource-aggressive users with time-invariant capacity requirements. Some users may be in need of time-invariant computing power. So for the third type, we consider users that require an invariable capacity of computing power (as the resource model in [3], [10]). Typically, such a cloud user i may request cloud resources of invariable capacity for a time length l_i, within a preferred time duration $[a_i, d_i]$ ($l_i \leq d_i - a_i$). And the valuation can be presented by a concavely increasing function $b_i(\cdot)$ with respect to the invariant capacity inv_cap_i allocated to him. For example, a user may have a valuation of \$10 in total if $inv\_cap_i = 5$ (5 VMs allocated to him), or a valuation of \$8 if $inv\_cap_i = 3$. Accordingly, we have $v_i(\gamma_i) = b_i(inv\_cap_i) \cdot l_i$, where $inv\_cap_i = \gamma_i(t)$ for all $t \in [t^-_{\gamma_i}, t^+_{\gamma_i}]$. Such a request can be organized as: $\mathbf{r}_i = \{a_i, d_i, l_i, b_i(inv\_cap_i)\}$.

Assumptions: First, aiming at a compelling user experience, preemption [7] is not allowed. Second, we do not assume any specific distribution of bidding parameters in the request $\mathbf{r_i}$ — we only apply a very general assumption that the *unit valuation* (the valuation for one unit resource per unit time) is within a known interval $[\underline{p}, \overline{p}]$, and the job length of Type III bidders is within the interval $[\underline{l}, \overline{l}]$.

C. Performance Metrics for the Auction Mechanism

Truthfulness (also called incentive-compatibility) is one of the most critical property of auction mechanism [11]. As mentioned in Section I, an auction could be vulnerable to market manipulation without truthfulness-guaranteed.

Definition 1. *(Truthfulness) An auction mechanism $\mathcal{A}$ is said to be truthful if, for any bidder i, regardless of the behaviors of other bidders, declaring a bid that truthfully reveals his valuation can maximize his utility.*

Existing pricing designs for cloud market [12], [13], [17] typically target at achieving the "optimal" allocation performance on revenue or social welfare. However, the optimal solution generally needs further information of the variation of user demands, which is very hard to estimate. To this end, in this paper we focus on an alternative problem: without assumptions on any specific distribution information on bidders' arrival or valuation, *how can we achieve a good worst-case performance on social welfare?* To evaluate the worst-case performance of an auction mechanism by its social welfare, a commonly adopted way is competitive analysis [8], [9] — to compare the allocation performance against the optimal offline solution — VCG mechanism.

Definition 2. *(Competitive ratio on social welfare) An auction mechanism $\mathcal{A}$ is ζ-competitive with respect to the social welfare if for every bidding sequence τ, $E_{\mathcal{A}}(\tau) \geq E_{VCG}(\tau)/\zeta$. Accordingly, ζ is the competitive ratio of $\mathcal{A}$.*

To that end, next in Section III we propose the online auction mechanism, called COCA, along with the proof for the truthfulness, followed by Section IV where we conduct extensive competitive analysis on COCA.

III. Mechanism Design: Ensuring Truthfulness

In this section, we first present our design methodology on ensuring truthfulness under our auction model, and then propose a truthful (incentive-compatible) online cloud auction mechanism, called COCA, building on top of our proposed bidding language in the previous section.

A. Design Methodology on Ensuring Truthfulness

1) How to determine the payment: Under our proposed auction model, the payment of a bidder i can be generally considered as a function: $pay_i = p_i(\gamma_i, t_{sub_i}, \mathbf{r_i})$. [1] However it may not be the simplest form of the payment function as the parameters are not totally independent. Indeed, the following lemma shows that the payment function can be further simplified as $pay_i = p_i(\gamma_i, t_{sub_i})$.

Lemma 1. *For any truthful auction algorithm $\mathcal{A}$, given γ_i and t_{sub_i}, the payment should be uniquely determined for any bidder i regardless of his request $\mathbf{r_i}$.*

Proof: We prove it by contradiction. Given some γ_i and t_{sub_i}, assume that there are two different requests $\mathbf{r_i}$ and $\mathbf{r'_i}$ with $p_i(\gamma_i, t_{sub_i}, \mathbf{r_i}) > p_i(\gamma_i, t_{sub_i}, \mathbf{r'_i})$. In this case, a bidder with true request $\mathbf{r_i}$ will increase his utility by declaring $\mathbf{r'_i}$. Therefore the auction is no longer truthful, completing the contradiction. ■

Next we discuss how γ_i and t_{sub_i} are correlated to the payment function $p_i(\gamma_i, t_{sub_i})$.

Definition 3. *(Monotonic with allocation) We say $\gamma_i \succeq \gamma'_i$ if $\forall t, \gamma_i(t) \geq \gamma'_i(t)$. A payment function p_i is monotonic with allocation if for any t_{sub_i} and any allocation $\gamma_i \succeq \gamma'_i$, we have $p_i(\gamma_i, t_{sub_i}) \geq p_i(\gamma'_i, t_{sub_i})$.*

Definition 4. *(Monotonic with submission time) A payment function is monotonic with submission time if for any allocation γ_i and any submission time $t_{sub_i} \leq t'_{sub_i}$, we have $p_i(\gamma_i, t'_{sub_i}) \geq p_i(\gamma_i, t_{sub_i})$.*

Given the above definitions, we are now ready to present the following theorem:

Theorem 1. *For any truthful online auction mechanism $\mathcal{A}$, the payment for any bidder i can be determined by a payment function $p_i(\gamma_i, t_{sub_i})$, which should be monotonic with submission time and monotonic with allocation.*

Proof: We prove it by contradiction. First, assume that p_i is not monotonic with allocation, that is, there exists a bidder i with two possible allocation decisions $\gamma'_i \succeq \gamma_i$, such that $p_i(\gamma_i, t_{sub_i}) > p_i(\gamma'_i, t_{sub_i})$. Denote $\mathbf{r'_i}$ and $\mathbf{r_i}$ as the requests

[1] The function can be more formally written as $pay_i = p_i(\gamma_i, t_{sub_i}, \mathbf{r_i}, \Phi)$, where Φ denotes all the parameters which can not be directly affected by the user strategy. Note that it is not necessary to explicitly show Φ in our analysis about truthfulness due to Φ's independence of user strategy.

lead to these two allocations respectively, then bidder i with truthful request $\mathbf{r_i}$ will increase his utility by declaring request $\mathbf{r'_i}$. That's a contradiction with the fact that $\mathcal{A}$ is truthful.

Second, we assume that p_i is not monotonic with submission time. That is, for some request $\mathbf{r_i}$ with submission time t_{sub_i}, $p_i(\gamma_i, t_{sub_i}) \geq p_i(\gamma_i, t'_{sub_i})$ holds for some $t'_{sub_i} > t_{sub_i}$. In this case, user i may increase his utility by declaring the same request $\mathbf{r_i}$ at such a later submission time t'_{sub_i}. That is also a contradiction with the fact that $\mathcal{A}$ is truthful. ■

Theorem 1 provides a payment rule which serves as a necessary condition to ensure truthfulness. Intuitively, a bidder may strategically delay his submission time, or try to manipulate the allocation decision by reporting an untruthful request $\mathbf{r'_i}$, so a later submission time, or a "better" allocation should lead to a higher payment.

2) How to determine the allocation: In response to the above payment function, we resort to a general allocation rule.

Proposition 1. *(N. Nisan et al. [11]) For any truthful auction mechanism $\mathcal{A}$, the auction mechanism optimizes for all bidders, i.e., the allocation decision maximizes the utility gain for each bidder i.*

Proposition 1 provides a generalized necessary condition to ensure truthfulness. Following this route, we present our guideline of how to determine the allocation under our auction model: For any auction mechanism $\mathcal{A}$, denote all possible allocations results to some bidder i as a set Γ_{iA}, then for any bidder i we will try to maximize his utility according to the constraint Γ_{iA} and his request $\mathbf{r_i}$. More formally we have

$$\gamma_i^* = argmax_{\gamma_i \in \Gamma_{iA}}(\widetilde{v}_i(\gamma_i) - p_i(\gamma_i, t_{sub_i})) \tag{2}$$

where $\widetilde{v}_i(\gamma_i)$ is the valuation function learned from request $\mathbf{r_i}$, and γ_i^* is the allocation finally determined for bidder i.

It is worth mentioning that Γ_{iA} may not be equivalent to Γ_i, which is the general set of all possible allocations. On one hand, the auction mechanism $\mathcal{A}$ may restrict the possible allocation results. As an extreme example, an auction mechanism which refuses all the user requests ($\Gamma_{iA} = \emptyset$) is always truthful as it satisfies Equ. 2. On the other hand, in an online auction the possible allocation results at some time can also be restricted by the previous allocation decisions. More specifically, for some bidder i, Γ_{iA} may be restricted for the reason that all the resources in a certain period have already been allocated to bidders with submission time earlier than t_{sub_i}.

B. COCA: A Truthful Online Cloud Auction Design

Motivated by the aforementioned design methodology, let's turn to the design of COCA in detail. We start with introducing how we construct the payment function p_i for every coming bidder i such that it is monotonic with allocation and submission time.

1) Payment function construction: Intuitively, COCA's payment function is committed to reflecting the current supply and demand relationship — the resource should be charged more in "hot" time periods (where there are a greater number of user demands). Similar to [9], COCA exploits an *auxiliary pricing function* $P(x)$ with respect to the current *utilization rate* U to help the resource provider decide the payment function.

Reserved resource utilization rate U: Note that COCA allows a bidder to reserve resources that are not available in the current time period. To that end, we define a variable — the *reserved resource utilization rate* U (we call it utilization rate for short henceforth). Formally, denote all the allocation decisions $\gamma_i^*, i = 1, 2...$ made by time t_2 as a set Γ_{t_2}, then we have $U(t_1, t_2) = \sum_{\gamma_i^* \in \Gamma_{t_2}} \gamma_i^*(t_1)/Q$. Obviously we have $U \in [0, 1]$. With such a definition, U can clearly reflect the status of how the cloud resources are allocated (reserved) *at* time t_1 according to the allocation decisions made *by* time t_2. As an example, Fig.5 shows the utilization rate within one day ($t_1 \in$ [0:00,24:00]) by the time t_2= 10:00. A high

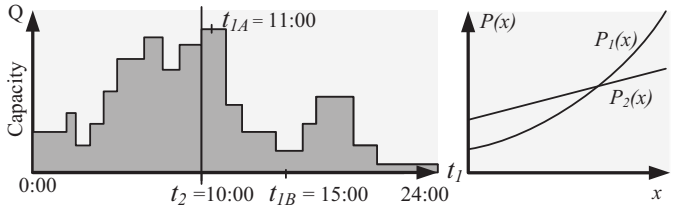


Fig. 5. The utilization rate $U(t_1, t_2)$ with t_2 = 10:00 and the auxiliary pricing function $P(x)$.

utilization rate is observed at U(11:00,10:00), which indicates that the resources at time t_{1A} =11:00 have almost been sold out by 10:00. On the contrary, a low value at U(15:00,10:00) implies that there are still a lot of unallocated resources at time t_{1B} =15:00 by 10:00. Specifically, we denote $U(t_1, t_{sub_i})$ ($U(t_1, t^+_{sub_i})$) as the utilization rate at time t_1 before (after) the allocation of request $\mathbf{r_i}$ submitted at t_{sub_i}. Accordingly we have $U(t_1, t^+_{sub_i}) - U(t_1, t_{sub_i}) = \gamma_i(t_1)/Q$. In addition, it's obvious that $\forall t_1, t_2$, we have $t_2 \leq t'_2 \Rightarrow U(t_1, t_2) \leq U(t_1, t'_2)$.

Auxiliary pricing function $P(x)$: To help the resource provider determine the payment function, in COCA we introduce an auxiliary pricing function $P(x)$ which is predetermined by the resource provider before the auction process. $P(x)$ explicitly presents the "marginal price" with respect to the utilization rate. That is, for any piece of allocation γ_i with

$$\begin{cases} t^+_{\gamma_i} - t^-_{\gamma_i} = \Delta l \to 0 \\ \gamma_i(t) = \Delta q \to 0 \qquad \forall t \in [t^-_{\gamma_i}, t^+_{\gamma_i}] \end{cases} \tag{3}$$

The payment for such γ_i is calculated as:

$$p_i(\gamma_i, t_{sub_i}) = P(U(t^-_{\gamma_i}, t_{sub_i})) \cdot \Delta l \cdot \Delta q. \tag{4}$$

where $U(t^-_{\gamma_i}, t_{sub_i})$ presents the utilization rate reserved at $t^-_{\gamma_i}$ by the time user i submits his request.

In such a way, the total payment for any bidder i can be calculated by summing up all such pieces of allocations, thus our payment function has the following form:

$$p_i(\gamma_i, t_{sub_i}) = \int_{t^-_{\gamma_i}}^{t^+_{\gamma_i}} \int_{U(t, t_{sub_i})}^{U(t, t_{sub_i}) + \gamma_i(t)} P(x) \cdot Q dx dt \tag{5}$$

It is noted that given a nondecreasing $P(x)$, the payment function satisfies all the necessary conditions given in Theorem

1 to ensure truthfulness. Here we just introduce the concept of $P(x)$ without discussing about how we construct it. Later in Section IV we will show that COCA's allocation performance with respect to social welfare greatly depends on the choice of such $P(x)$.

2) Mechanism description: We now describe our design of COCA, shown in Algorithm 1, step by step:

- **Step 1: Lines 4-5 (Constructing payment function)** For any bidder i, the payment function is given as $p_i(\gamma_i, t_{sub_i}) = \int_{t^-_{\gamma_i}}^{t^+_{\gamma_i}} \int_{U(t,t_{sub_i})}^{U(t,t_{sub_i})+\gamma_i(t)} P(x) \cdot Q dx dt$, where $P(x)$ can be any nondecreasing function.
- **Step 2: Lines 6-18 (Allocation rule)** The allocation tries to maximize the utility gain for each bidder i according to his request $\mathbf{r_i}$, that is

$$\gamma_i^* = argmax_{\gamma_i}(\widetilde{v}_i(\gamma_i) - p_i(\gamma_i, t_{sub_i}))$$
$$\text{s.t.} \quad \forall t,\ U(t, t_{sub_i}) + \gamma_i(t)/Q \leq 1$$

  where $\widetilde{v}_i(\gamma_i)$ is the valuation function learned from the request $\mathbf{r_i}$.
- **Step 3: Lines 19-20 (Payment rule)** Determine the payment according to the payment function p_i and the allocation decision γ_i^*: $pay_i = p_i(\gamma_i^*, t_{sub_i})$.
- **Step 4: Lines 21-22 (Updating the utilization rate)** Update the utilization rate U according to the allocation decision γ_i^*.

Following the above steps, we present our detailed design of COCA in Algorithm 1. It is shown that as bidders are categorized to the three valuation types introduced in Section II-B, corresponding allocation rules can be applied accordingly. Meanwhile, such a design spontaneously balances the workload. Since according to the allocation rule, users will be more likely to be allocated in time durations where the current unitization rate is lower, as the payment will be lower according to $p_i(\gamma_i, t_{sub_i})$ (with a nondecreasing $P(x)$).

C. Proof of Truthfulness

Let $\mathbf{r'}_i$ be any untruthful request derived from the truthful request $\mathbf{r}_i$ by making arbitrary changes on the reported valuation function $\widetilde{v}_i(\gamma_i)$ (changing any of its bidding parameters or his request type). In this case we have:

Lemma 2. *When COCA is applied, the utility bidder i can get by submitting $\mathbf{r}_i$ is no less than the utility he can get by submitting any such $\mathbf{r'}_i \neq \mathbf{r}_i$.*

Proof: Denote $u_i(\mathbf{r}_i)$ and $u_i(\mathbf{r'}_i)$ as the utility bidder i gets by submitting request $\mathbf{r}_i$ and $\mathbf{r'}_i$ respectively. By the allocation rule we have

$$u_i(\mathbf{r}_i) = max_{\gamma_i}(v_i(\gamma_i) - p_i(\gamma_i, t_{sub_i}))$$
$$\geq v_i(\gamma_i') - p_i(\gamma_i', t_{sub_i}) = v_i(\gamma_i') - p_i'(\gamma_i', t_{sub_i}) = u_i(\mathbf{r'}_i) \tag{6}$$

where γ_i' is the allocation i gets by submitting request $\mathbf{r'}_i$. Thus, we always have $u_i(\mathbf{r}_i) \geq u_i(\mathbf{r}_i')$, which suffices to complete the proof. ■

Theorem 2. *COCA is a truthful auction mechanism.*

Algorithm 1 COCA Mechanism Design

Input:
1) The request sequence τ: $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, ...\mathbf{r}_\infty\}$, such that $t_{sub_1} < t_{sub_2} < ... < t_{sub_\infty}$;
2) A nondecreasing auxiliary pricing function $P(x)$

Output:
1) The allocation decision γ_i^* for each request $\mathbf{r}_i$;
2) Payment decision pay_i for each request $\mathbf{r}_i$

1: **Initialization of the utilization rate** :
2: $\forall\ t \in [0, \infty], U(t, t_{current}) = 0$ % $t_{current}$ refers to the current time.
3: **for** $i = 1$ *to* ∞ **do**
4: **Constructing payment function** :
5: $p_i(\gamma_i, t_{sub_i}) = \int_{t^-_{\gamma_i}}^{t^+_{\gamma_i}} \int_{U(t,t_{sub_i})}^{U(t,t_{sub_i})+\gamma_i(t)} P(x) \cdot Q dx dt$
6: **Allocation rule** :
7: Check the type $type_i$ of $\mathbf{r}_i$;
8: **switch** $type_i$
9: **case** 1 % Allocation determination for Type I bidder:

$$\gamma_i^* = argmax_{\{\gamma_i\}}(b\_total_i - pen\_rate_i \cdot delay_i - p_i(\gamma_i, t_{sub_i}))$$
$$\text{s.t.} \quad \forall t,\ U(t, t_{sub_i}) + \gamma_i(t)/Q \leq 1$$
$$\int_{a_i}^{d_i + delay_i} \gamma_i(t) \geq size_i$$

where $delay_i = max(t^+_{\gamma_i} - d_i, 0)$
% Find the allocation that maximizes i's utility if the job is accepted.
10: **if** $b_{total_i} - pen\_rate_i \cdot delay_i^* - p_i(\gamma_i^*, t_{sub_i}) < 0$ **then**
11: set $\gamma_i^* = \emptyset$
% If the maximum utility of accepting the job is negative, then reject the job.
12: **end if**
13: **end case**
14: **case** 2 % Allocation determination for Type II bidder:

$$\gamma_i^* = argmax_{\{\gamma_i\}}(b_i(total\_rsc_i) - p_i(\gamma_i, t_{sub_i}))$$
$$\text{s.t.} \quad \forall t,\ U(t, t_{sub_i}) + \gamma_i(t)/Q \leq 1$$

where $total\_rsc_i = \int_{a_i}^{d_i} \gamma_i(t) dt$
% Find the allocation that maximizes i's utility.
15: **end case**
16: **case** 3 % Allocation determination for Type III bidder:

$$\gamma_i^* = argmax_{\{\gamma_i\}}(b_i(inv\_cap_i) \cdot l_i - p_i(\gamma_i, t_{sub_i}))$$
$$\text{s.t.} \quad \forall t,\ U(t, t_{sub_i}) + \gamma_i(t)/Q \leq 1$$
$$\forall t_1, t_2 \in [t^-_{\gamma_i}, t^+_{\gamma_i}], inv\_cap_i = \gamma_i(t_1) = \gamma_i(t_2)$$

% Find the allocation that maximizes i's utility.
17: **end case**
18: **end switch**
19: **Payment rule** :
20: $pay_i = p_i(\gamma_i^*, t_{sub_i})$
21: **Updating the utilization rate** :
22: $\forall t \in [t^{*-}_{\gamma_i}, t^{*+}_{\gamma_i}],\ U(t, t_{current}) = U(t, t_{current}) + \gamma_i^*(t)$
23: **end for**

Proof: We adopt the notations in the above lemma, and additionally denote $\mathbf{r''}_i$ as the request derived from changing the submission time t_{sub_i} to t''_{sub_i} of any request $\mathbf{r'}_i$. First, it is derived from Lemma 2 that γ_i^* maximizes $u_i(\gamma_i)$. Therefore, if we denote the allocation decision for reporting $\mathbf{r}_i$, $\mathbf{r}_i'$ and $\mathbf{r}_i''$

as γ_i^*, γ_i' and γ_i'' respectively, the following inequation holds:

$$u_i(\mathbf{r}_i) = v_i(\gamma_i^*) - p_i(\gamma_i^*, t_{sub_i}) \geq v_i(\gamma_i'') - p_i(\gamma_i'', t_{sub_i}). \tag{7}$$

Second, note that the true submission time is defined as the time that a user arrives at the market, which is also the first time he is aware of his demand. Thus, it's not possible for any user to report a submission time earlier than his true submission time [8]. Since $P(x)$ is nondecreasing, according to Equ. (5), we have $p_i(\gamma_i, t_{sub_i}) \leq p_i(\gamma_i, t''_{sub_i})$ for any γ_i and $t''_{sub_i} \geq t_{sub_i}$. Hence the following inequations holds:

$$v_i(\gamma_i'') - p_i(\gamma_i'', t_{sub_i}) \geq v_i(\gamma_i'') - p_i(\gamma_i'', t''_{sub_i}) = u_i(\mathbf{r}_i'') \tag{8}$$

Equ. (7) and Equ. (8) imply that $u_i(\mathbf{r}_i) \geq u_i(\mathbf{r}_i'')$. This result, together with Lemma 2, can demonstrate that, reporting the true request $\mathbf{r_i}$ always results in a better utility than reporting any untruthful request $\mathbf{r_i'}$ or $\mathbf{r_i''}$. That is, the theorem holds. ■

IV. Mechanism Design: Achieving a Nontrivial Competitive Ratio on Social Welfare

Based on the above mechanism design of COCA, truthfulness can be ensured by the implementation of a nondecreasing auxiliary pricing function $P(x)$. Now we have the following two problems still unsolved — *(i). how to determine the auxiliary pricing function $P(x)$? (ii). how can COCA achieve a nontrivial competitive ratio on social welfare (defined in Section II-B)?* In this section we show that the answers to these two questions are closely correlated — the competitive ratio highly depends on the auxiliary pricing function $P(x)$. Moreover, we show that the competitive ratio of COCA can be well-bounded by appropriately constructing the auxiliary pricing function $P(x)$.

A. Competitive Analysis for a Single Bidder Type

In this subsection, we consider the scenario where the cloud users in a request sequence τ only belong to a *single request type*. In the following analysis, we will show how the competitive ratio of COCA on social welfare is determined by the choice of the auxiliary pricing function $P(x)$.

1) Competitive analysis for Type II bidders: Recall that we apply a general assumption that the *unit valuation* (the valuation for one unit resource per unit time) is within a known interval $[\underline{p}, \overline{p}]$. Under such assumption, denote the social welfare achieved when VCG (COCA) is applied as E_{VCG} (E_{COCA}), we have the following theorem for Type II bidders:

Theorem 3. *For any request sequence τ consisting of Type II bidders, we have $E_{COCA}(\tau) \geq E_{VCG}(\tau)/(1+g_2)$, where $g_2 = max_{\{x\in[P^{-1}(\underline{p}),1]\}}(P(x)/\int_0^x P(u)du)$.* [2]

Theorem 3 indicates that any sequence composed of Type II bidders has a competitive ratio of $1+g_2$, where g_2 is directly determined by the auxiliary pricing function $P(x)$. Thus, we can minimize the competitive ratio by solving the following optimization problem:

$$\begin{aligned} \min_{\{P(x)\}} \quad & g_2 = max_{\{x\in[P^{-1}(\underline{p}),1]\}} P(x)/\int_0^x P(u)du \\ \text{s.t.} \quad & P^{-1}(\overline{p}) = P^{-1}(\underline{p}) + \int_{\underline{p}}^{\overline{p}} (P^{-1}(x))' dx \leq 1 \end{aligned}$$

where the inequality constraint implicitly insures that the quantity of resource allocated at any time should be less than Q (i.e.,utilization rate less than one). To solve this, we use a similar technique in [15], figuring out an auxiliary pricing function $P_1(x)$ such that the competitive ratio can be minimized.

Corollary 1. *For any request sequence τ consisting of Type II bidders, with auxiliary pricing function*

$$P_1(x) = \begin{cases} \overline{p}/e^{(1-x)\cdot r} & 1/r \leq x \leq 1 \\ \underline{p} & 0 \leq x < 1/r \end{cases} \tag{9}$$

COCA is $(1+r)$-competitive where $r = 1 + ln(\overline{p}/\underline{p})$.

2) Competitive analysis for Type I bidders: Achieving a good competitive ratio in the *general case* with no further constrains or assumptions is more difficult for Type I bidders. The reason is that a user of high unit valuation together with a very high penalty rate may be directly rejected if he is blocked by previous allocations. Instead, here we consider a common but less general case in which the resources haven't been fully utilized: we call it the *underload* case if when COCA is applied, for all t we have $U(t,\infty) \leq 1 - \overline{q}/Q$. Note that since the resource provider always has a very large resource capacity Q, $1 - \overline{q}/Q$ will be very close to 1, accordingly the underload case defined above is very likely to happen if there are not so many bidders with high unit valuation asking for a same time period. The following theorem shows that for Type I bidders, we can achieve a competitive ratio comparable to that of Type II bidders in such underload case.

Theorem 4. *For any request sequence τ consisting of Type I bidders, we have $E_{COCA}(\tau) \geq E_{VCG}(\tau)/(1+g_1)$ in the underload case, where $g_1 = max_{\{x\in[P^{-1}(\underline{p}),1]\}}(P(min(1, x+\overline{q}/Q))/\int_0^x P(u)du)$.*

Observe that g_1 is quite similar to g_2. Then according to Theorem 4, in the following corollary we show that auxiliary function $P_1(x)$ can also be applied to achieve a good competitive ratio for Type I bidders in the underload case.

Corollary 2. *For any request sequence τ consisting of Type I bidders, with auxiliary pricing function $P_1(x)$, COCA is $(1+e\cdot r)$-competitive in the underload case if $\overline{q} \leq Q/r$, where $r = 1 + ln(\overline{p}/\underline{p})$.*

3) Competitive analysis for Type III bidders: In the following analysis for Type III bidders, we show the relation between the competitive ratio and the choice of $P(x)$ in both the general case and the underload case.

Theorem 5. *For any request sequence τ consisting of Type III bidders, $E_{COCA}(\tau) \geq E_{VCG}(\tau)/(1+g_1)$ holds in the under-*

[2] As P is not necessarily strictly increasing, here we simply denote $P^{-1}(x)$ as the maximal value y which satisfies $x = P(y)$.

load case. And for the general case, we have $E_{COCA}(\tau) \geq E_{VCG}(\tau)/(1+g_1+g_3)$*, where* $g_3 = \bar{l}\cdot\bar{p}/\underline{l}\cdot\int_0^{(1-\bar{q}/Q)/2} P(u)du$.

Theorem 5 indicates that the competitive ratio has a more complex form in the general case, where both g_1 and g_3 have to be considered in order to achieve a good competitive ratio. The following corollary shows that we can obtain a non-trivial competitive ratio for both the underload case and the general case, by using auxiliary function $P_1(x)$ and $P_3(x)$ respectively.

Corollary 3. *For any request sequence* τ *consisting of bidders of Request Type III, with auxiliary pricing function* $P_1(x)$*, COCA is* $(1 + e \cdot r_1)$*-competitive in the underload case if* $\bar{q} \leq Q/r_1$*, where* $r_1 = 1 + ln(\bar{p}/\underline{p})$*. And for the general case, with auxiliary pricing function*

$$P_3(x) = \begin{cases} (\bar{p}\cdot\bar{l}/\underline{l})/e^{(0.5\cdot(1+\bar{q}/Q)-x)\cdot r_3} & 1/r_3 \leq x \leq 1 \\ \underline{p} & 0 \leq x < 1/r_3 \end{cases} \tag{10}$$

COCA is $(1 + 2e \cdot r_3)$*-competitive if* $\bar{q} \leq Q/r_3$*, where* $r_3 = (1 + ln(\bar{p}\cdot\bar{l}/\underline{p}\cdot\underline{l}))/(0.5\cdot(1+\bar{q}/Q))$.

B. Competitive Analysis for the Mixture Arrival Case

In Section IV-A we assume users in a request sequence to be of a *single type*. Since we have three request types, *what competitive ratio can we achieve if bidders of different types come in a mixed manner?* To answer this question, we conduct competitive analysis for the mixture arrival case as follows.

Theorem 6. *For any request sequence* τ *consisting of bidders of Type I, II, and III,* $E_{COCA}(\tau) \geq E_{VCG}(\tau)/(3+2\cdot g_1+g_2)$ *holds in the underload case.*

Theorem 6 tells us how the competitive ratio in the mixture case is determined by the competitive ratio under the scenario where only a single bidder type is considered. Finally, according to Theorem 3 to Theorem 6, a non-trivial bound on competitive ratio is given in the following proposition.

Proposition 2. *For any request sequence* τ *consisting of bidders of Request Type I, II and III, with auxiliary pricing function* $P_1(x)$*, COCA is* $O(log(\bar{p}/\underline{p}))$*-competitive in the underload case as long as* $\bar{q} \leq Q/ln(\bar{p}/\underline{p})$.

The above proposition shows that the competitive ratio on social welfare can be well-bounded by appropriately constructing the auxiliary function $P(x)$, as for any fixed pricing mechanism, the competitive ratio is at least $O(\bar{p}/\underline{p})$.

V. Simulation Results

In this section we propose simple simulations to evaluate COCA under illustrative bid distributions and arrival models. We focus on examining the allocation performance of COCA on social welfare compared with the off-line VCG mechanism. We haven't compared COCA with existing online auction mechanisms because no prior solutions have achieved the generalized truthfulness in our online cloud auction setting.

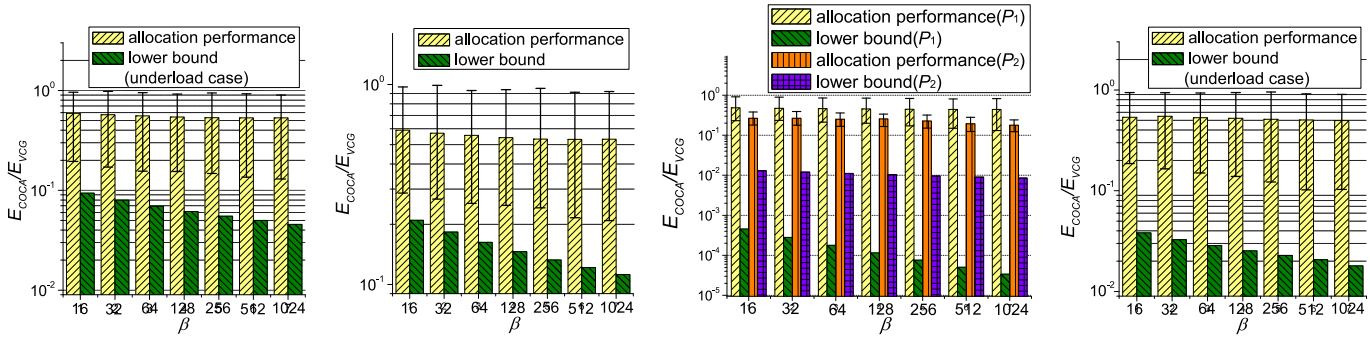
We consider a cloud resource provider of a fixed capacity $Q = 10^4$ (i.e., the provider is able to host up to 10^4 VMs simultaneously), and here a simple simulation model is used: the bidding parameters are assumed to be uniformly distributed (detailed settings refer to Table I), and the (marginal) unit valuation is a fixed number $\rho \in [1, \beta]$, where $\beta = \bar{p}/\underline{p}$, which refers to the ratio between the highest and lowest unit valuation (mentioned in the assumptions in Section II). Moreover, we assume a penalty rate of ∞ for Type I bidders. We run each online auction for 500 time units, and the number of requests generated in a bidding sequence is modeled by a variable uniformly distributed from 100 to 2,000. Each of the following simulations has been carried out for 3,000 runs.

TABLE I
IMPLEMENTATION CONFIGURATION

t_{sub_i}	a_i		d_i	$\bar{q}$	β
$[1, 500]$	$[t_{sub_i}, min(t_{sub_i} + 100, 500)]$		$[a_i, 500]$	100	$\bar{p}/\underline{p}$
Type	l_i	$size_i$	$b_i(\cdot)$		
I	-	$[10^3, 10^5]$	$b\_total_i = \rho_i \cdot size_i; \rho_i \in [1, \beta]$		
II	-	-	$b_i(total\_rsc_i) = \rho_i \cdot total\_rsc_i$		
III	$[5, 200]$	-	$b_i(inv\_cap_i) = \rho_i \cdot inv\_cap_i \cdot l_i$		

Fig. 6 shows the allocation performance of COCA on social welfare compared with the optimal solution (VCG mechanism) over 3,000 runs. We first run the simulations for each of the three request types respectively. As is discussed in Section IV, we use auxiliary pricing function $P_1(x)$ for Type I and Type II users. It can be observed in Fig. 6(a) and Fig. 6(b) that the worst performance in 3,000 runs is always better than the performance lower bound (1 over the competitive ratio) calculated in Section IV-C, and the average performance is always over 50% compared with the optimal allocation. In Fig. 6(c), we show the allocation performance for Type III users when P_1 and P_3 are used respectively. It can be observed that P_1 outperforms P_3 in both average performance and worst performance. Briefly speaking, the reason is that there exist some extreme "bad" cases for Request Type III when we analyze the performance lower bound shown in Fig. 6(c). To achieve a better lower bound, P_3 has to be constructed in such a way where the allocation performance in most cases becomes less satisfactory. However the possibility that the extreme case appears is too small, so it hardly happens in 3,000 runs under our simulation model. In Fig. 6(d), we plot the performance of COCA for the mixture arrival case, where bidders of different types come in a mixed manner. It is observed that the performance is very similar to the single-arrival case shown above.

Overall, from the simulation results we can conclude that: *first*, it is clearly observed that the ratios of COCA over VCG in terms of social welfare are quite close to 1 in all cases, indicating that COCA is comparable to the off-line VCG mechanism (i.e., the optimal solution) under our simulation model; *second*, with $\beta = \bar{p}/\underline{p}$ increases exponentially, the worst performance of COCA (in 3,000 runs) decreases very slowly in all cases. Such results are in good agreement with the theoretical analysis in Section IV, that COCA achieves a competitive ratio of $O(log(\bar{p}/\underline{p}))$ rather than $O(\bar{p}/\underline{p})$.



(a) Type I Bidders (b) Type II Bidders (c) Type III Bidders (d) The mixture arrival case

Fig. 6. Allocation performance of COCA compared with VCG mechanism (with 5th and 95th percentiles).

VI. Conclusion

This paper conducts the first work on truthful online auction design in cloud computing where users with heterogeneous demands could come and leave on the fly. First, for cloud consumers with heterogeneous demands we propose a novel bidding language, by which user-specific demands can be revealed in a concise and regulated request form. Second we propose the first truthful online cloud auction mechanism, COCA, which is composed of the design of a payment function, an allocation rule and a payment rule. We also implement competitive analysis on COCA in terms of social welfare, which shows that the worst-case performance of COCA can be well-bounded.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61073147, Grant 61173120, and Grant 61271226; by the National Natural Science Foundation of China and Microsoft Research Asia under Grant 60933012; by the Fundamental Research Funds for the Central Universities under Grant 2012QN078; by the CHUTIAN Scholar Project of Hubei Province; by the Youth Chenguang Project of Wuhan City under Grant 201050231080; by the Scientific Research Foundation for the Returned Overseas Chinese Scholars (State Education Ministry); by the National Natural Science Foundation of Hubei Province under Grant 2011CDB044; by the Fok Ying Tung Education Foundation under Grant 132036; by the Hong Kong Scholars Program; and by the Program for New Century Excellent Talents in University under Grant NCET-10-408 (State Education Ministry). Dr. Bo Li's work was supported in part by a grant from NSFC/RGC under the contract N_HKUST610/11, by grants from HKUST under the contract RPC11EG29, SRFI11EG17-C and SBI09/10.EG01-C, by a grant from Guangdong Bureau of Science and Technology under the contract GDST11EG06, by a grant from China Cache Corp. under the contract CCNT12EG01. Dr. Fangming Liu's work was supported in part by The National Natural Science Foundation of China (NSFC) under grant No.61103176 and No.61133006. The corresponding author of this paper is Hongbo Jiang.

References

[1] AmazonEC2SpotInstances. http://aws.amazon.com/ec2/spot-instances/.
[2] V. Abhishek, I. A. Kash, and P. Key. Fixed and market pricing for cloud services. In *Proceedings of 7th Workshop on the Economics of Networks, Systems, and Computation*, 2012.
[3] E. Angel, E. Bampis, and F. Pascual. Truthful algorithms for scheduling selfish tasks on parallel machines. *Theoretical Computer Science*, 369(1):157–168, December 2006.
[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, Electrical Engineering and Computer Sciences, University of California at Berkeley, February 2009.
[5] R. Buyya, C. S. Yeo, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *Proceedings of 10th IEEE International Conference on High Performance Computing and Communications*, 2008.
[6] L. Deek, X. Zhou, K. Almeroth, and H. Zheng. To preempt or not: Tackling bid and time-based cheating in online spectrum auctions. In *Proceedings of IEEE INFOCOM*, 2011.
[7] J. Goossens, P. Richard, and P. Richard. Socially optimal pricing of cloud computing resources. In *Proceedings of Euro Workshop on Project Management and Scheduling*, 2004.
[8] M. T. Hajiaghayi, R. D. Kleinberg, M. Mahdian, and D. C. Parkes. Online auctions with reusable goods. In *Proceedings of the 6th ACM conference on Electronic Commerce*, 2005.
[9] R. Lavi and N. Nisan. Competitive analysis of incentive compatible online auctions. In *Proceedings of the 2nd ACM conference on Electronic Commerce*, 2000.
[10] I. Menache, A. Ozdaglar, and N. Shimkin. Socially optimal pricing of cloud computing resources. In *International Conference on Performance Evaluation Methodologies and Tools*, 2011.
[11] N. Noam, R. Tim, T. Eva, and V. Vijay. *Algorithmic Game Theory*. Cambridge University Press, 2007.
[12] Q. Wang, K. Ren, and X. Meng. When cloud meets ebay: Towards effective pricing for cloud computing. In *Proceedings of IEEE INFOCOM*, 2012.
[13] W. Wang, B. Li, and B. Liang. Towards optimal capacity segmentation with hybrid cloud pricing. In *Proceedings of 32nd International Conference on Distributed Computing Systems*, 2012.
[14] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowstron. Better never than late: Meeting deadlines in datacenter networks. In *Proceedings of ACM SIGCOMM*, 2011.
[15] R. E. Yaniv, A. Fiat, R. Karp, and G. Turpin. Competitive analysis of financial games. In *Proceedings of the 33rd Symposium on Foundations of Computer Science (FOCS)*, 1992.
[16] C. S. Yeo and R. Buyya. Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. In *Proceedings of Cluster Computing*, 2005.
[17] Q. Zhang, E. Grses, R. Boutaba, and J. Xiao. Dynamic resource allocation for spot markets in clouds. In *Proceedings of the 11th USENIX conference on Hot topics Hot-ICE*, 2011.