Discovering Valuations and Enforcing Truthfulness in a Deadline-Aware Scheduler

Zhe Huang*, S. Matthew Weinberg*, Liang Zheng*, Carlee Joe-Wong†, and Mung Chiang*
*Princeton University, NJ, USA,
†Carnegie Mellon University, Silicon Valley Campus, CA, USA
Emails: {zheh, smweinberg, liangz, chiangm}@princeton.edu, cjoewong@andrew.cmu.edu

***Abstract*—A cloud computing cluster equipped with a deadline-aware job scheduler faces fairness and efficiency challenges when greedy users falsely advertise the urgency of their jobs. Penalizing such untruthfulness without demotivating users from using the cloud service calls for advanced mechanism design techniques that work together with deadline-aware job scheduling. We propose a Bayesian incentive compatible pricing mechanism based on matching by replica-surrogate valuation functions. User valuations can be discovered by the mechanism, even when the users themselves do not fully understand their own valuations. Furthermore, users who are charged a Bayesian incentive compatible price have no reason to lie about the urgency of their jobs. The proposed mechanism achieves multiple desired truthful properties such as Bayesian incentive compatibility and ex-post individual rationality. We implement the proposed pricing mechanism. Through experiments in a Hadoop cluster with real-world datasets, we show that our prototype is capable of suppressing untruthful behavior from users.**

I. Introduction

Cloud computing has become popular by providing inexpensive computing services to users through shared computing clusters. Users can pay more to improve their quality of service (QoS), e.g., lower job completion times. Therefore, it is crucial to automate the prioritization of user jobs with different QoS requirements. Recently, several deadline-aware job schedulers have been proposed for shared computing clusters that allocate resources among jobs according to the jobs' urgency and valuations as a function of delay and deadline [1]–[3]. These schedulers consider not just the total social welfare of the system, but also objectives like fairness across different users. However, all of them suffer from the fundamental problem of greedy users cheating the system by falsely advertising the importance and urgency of their jobs.

Simply enforcing usage budget limits on users (e.g., by forcing users to spend a limited amount of virtual currency to receive cloud resources) is not a complete solution to this user disincentive. Charging users more for better QoS can discourage them from lying about their jobs' importance, but these prices must take into account how much the user values the service received. Otherwise, users will still overstate the urgency of the jobs which have cheaper prices compared to the users' valuations, simply because they can afford it. Therefore, a pricing mechanism that penalizes greedily untruthful users is key to guaranteeing the fairness and efficiency of schedulers, especially for those in cloud computing.

Vickrey-Clarke-Groves (VCG) auctions [4]–[6] have been used to derive many well-known truthful pricing schemes for resource allocations that maximize social welfare [7]–[11]. However, these schemes may not incentivize truthfulness when resources are not allocated so as to maximize social welfare (e.g., scheduling jobs according to their deadlines). In particular, it can be proven that there does not exist a dominant truthful strategy for any pricing scheme for resource allocation based on job completion times [1], [2] as presented in this work. Therefore, to accommodate these resource allocation objectives, we must design an **incentive compatible mechanism for deadline-aware scheduling** that does not directly price the resource allocation but still motivates user truthfulness.

Incentive compatible pricing schemes generally induce QoS requirements from user-submitted valuations that precisely quantify the values of completing jobs at different times. In practice, assistance should be provided to untrained users to guide them in quantifying their valuations. A more desirable pricing scheme should generate multiple service options with clearly defined outcomes (e.g., expected job completion time) and prices from which the users can choose, yet incentive compatible pricing mechanisms do not necessarily yield pricing-service options that can be easily understood. To the best of our knowledge, our paper is the first work that meets these mechanism design and practicality challenges.

The recent advancement of deadline-aware schedulers that can estimate the completion times of submitted jobs (e.g., [2]) has enabled more advanced pricing techniques that meet the above challenges. We propose a universal Bayesian incentive compatible (BIC) pricing mechanism for an *arbitrary* deadline-aware scheduling algorithm. Our pricing scheme precisely determines the price for each job according to its declared urgency and valuation, so that it is in each user's best interest to truthfully declare the valuation of the service received by each job. We accommodate generic deadline-aware schedulers by decoupling the resource allocation scheme from the pricing scheme, introducing a valuation translation process that we call "replica-surrogate matching." Intuitively, replica-surrogate matching replaces users' declared job valuations with surrogate valuations that lead to different scheduling outcomes. Thus, instead of bidding for resource allocation opportunities, users bid for chances to declare alternative valuations that may have

This work was in part supported by NSF CNS-1527513 and ARO W911 NF-11-1-0036.

better scheduling performance and hence increase their QoS. We ensure Bayesian incentive compatibility by extending a well-known truthful pricing scheme on the auction game that models the valuation translation. Our proposed mechanism has the following desired properties:

1) **Truthfulness Guarantee.** [Section II, Experiment IV-A] The proposed pricing mechanism can be implemented as a generic incentive compatible framework that determines the prices for completion-time-aware cloud services for various applications and resource allocation schemes.
2) **Service Option Visualization.** [Sections II-D and III-D, Experiment IV-A] Given some initial parameters, users can choose from multiple alternative QoS levels that clearly show the expected completion times of jobs and the associated prices that users must pay.
3) **Ex-Post Individual Rationality.** [Section II-E] Users are guaranteed to have a positive utility (i.e., the valuation of their jobs minus the price), ensuring that they do not lose interest in utilizing the cloud services.
4) **Light-Weight Implementation.** [Sections III and IV] Computations in the proposed mechanism have very low time complexity. We build a prototype system for an in-house Hadoop v2.6 cluster equipped with a deadline-aware scheduler and show that it can generate pricing information in a real-time manner.

We outline our pricing mechanism in Section II before discussing our implementation in Section III and presenting its performance results in Section IV. Section V gives an overview of related work, and Section VI concludes the paper.

II. Incentive Compatible Pricing for Deadline-Aware Scheduling

We develop and implement a BIC mechanism based on a replica-surrogate matching technique [12], [13]. This technique treats the deadline-aware scheduling algorithm as a black box, putting no constraints on the algorithm used but requiring the scheduler to output jobs' expected completion times when they are submitted. Such schedulers have been only proposed recently [2], enabling our novel mechanism design. The replica-surrogate matching technique can achieve incentive compatibility because it avoids modeling the scheduling events directly as auction games of resource allocation. Instead, it observes possible scheduling outcomes from the scheduler and prices them according to the opportunity cost of choosing one. Since the deadline-aware scheduler will trigger scheduling events repeatedly due to the dynamic arrivals and departures of user jobs, we focus our discussion on the scheduling logic in a single scheduling event.

Our incentive compatible mechanism starts with users describing their valuations for their jobs being completed at different times. Such valuations may not necessarily be truthful or accurate. We suppose that the user valuations can be categorized according to valuation templates that we refer to as job profile classes. (Section II-A). The idea of replica-surrogate matching is to replace the submitted valuation of the job by a surrogate valuation that preserves the distribution of users' valuation types in the same job profile class, a process we describe in Section II-B. An incentive compatible price is then calculated for each replica-surrogate match using the famous VCG auction scheme [4]–[6]. The replica-surrogate matching technique essentially translates the submitted valuation type to an appropriate type so that the VCG pricing will return a price that reflects the user's true valuation (Section II-C). We transform these prices into a menu of contracts from which the user chooses a service level and the associated price, which may be distinct from the price and service level corresponding to the declared valuation (Section II-D).

Once the jobs complete, our mechanism computes the final prices that users must pay for the completed jobs (Section II-E). While we show that the final prices ensure incentive compatibility, our mechanism may lead to a lower social surplus (i.e., summation of the values received by the users), since the scheduler uses surrogate instead of users' submitted valuation types. However, we can prove that the loss in social surplus is arbitrarily small (Section II-F). Important symbols used in our formulation are summarized in Table I.

A. Job Valuation

Consider a computing cluster equipped with a deadline-aware job scheduler that attempts to finish N active jobs before their deadlines, indexed by $i = 1, 2, \ldots, N$. To capture the jobs' sensitivity to their completion times, each user i submits a job valuation function $v_i(x_i)$ that describes the sensitivity of their satisfactory level to the job completion time x_i, implicitly defining a "soft" deadline requirement. Jobs can be said to meet their deadlines if they complete before their valuations drop to zero. Instead of requiring the users to draw the curve of the valuation, we assume that users submit vectors of parameters that can completely describe the valuation shape; thus, $v_i\left(x_i\right)$ can be written as $v\left(t_i, x_i\right)$ where t_i is user i's parameter vector. Since the jobs' valuations directly determine their completion times, our goal is to design a pricing mechanism to dis-incentivize greedy users from submitting untruthful valuations.

We refer to the distinct parameter vector t_i as the *type* of a user i's valuation. We assume that the types of the valuations are private information known only to the users and let t_i^* denote user i's true valuation type. The incentive compatible mechanism penalizes the users who submit untruthful valuation types. Due to the private nature of the valuation type information, the only practical way for the incentive compatible mechanism to evaluate the truthfulness of the submitted valuation is through monitoring and learning the statistics of the valuation types. We divide users' job valuation functions into different *job profile classes* and suppose that each class corresponds to a set of valuation types–for instance, one class of jobs might consist of those with sigmoidal valuation functions, with types determined by the sigmoid function parameters. The estimated distribution d_i of the valuation types for each user i's job profile class will be used to calculate user i's price; we discuss estimating d_i in Section III-B. Since the incentive compatible pricing mechanism penalizes valuation types that do not agree with the estimated distribution, users are encouraged to help improve the estimation accuracy by submitting truthful valuations. We suppose that users do not collude with each other to induce a false valuation distribution, since in practice they likely do not know the other users who are submitting jobs.

We assume that every user declares her type t_i when she submits job i. Note that t_i may not be truthful. Later in Section II-D, we show that the user can later amend an incorrectly declared type. Denote the vector of valuation types by $\mathbf{t} = (t_1, ..., t_N)$. Let $\mathbf{t}_{-i} = (t_1, ..., t_{i-1}, t_{i+1}, ..., t_N)$ be the vector of submitted valuation types except the type of job i. We assume that the proposed mechanism has access to the deadline-aware scheduler as a black-box system. By submitting the valuations of all active jobs, the deadline-aware scheduler is able to estimate the completion time of each job. Please note that our mechanism only requires the completion time estimation to be accurate enough in the aggregate. We later show how prices are adjusted to compensate the inaccuracy in completion time estimation in Section II-E. Let $X_i(\mathbf{t})$ denote the expected finishing time of job i estimated by the deadline-aware scheduler given the type vector $\mathbf{t}$. With these definitions, we now discuss the replica-surrogate matching.

B. Replica-Surrogate Matching

The replica-surrogate matching process starts by generating a maximum weighted matching for each job. Without loss of generality, we consider job i in the following. We first generate $M-1$ i.i.d. random replica types and M i.i.d. random surrogate types from distribution d_i. Let r_j^i and s_j^i be the jth replica and surrogate types generated for job i, respectively.[1] The declared valuation type t_i, the replica types, and the surrogate types are then used to construct a bipartite graph. The left-hand side of this graph comprises replica types r_j^i with the declared valuation type t_i inserted uniformly at random. Let i^* be the index of t_i (i.e., $r_{i^*}^i = t_i$). The right-hand side consists of the surrogate types s_j^i. A weight of $v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))$ is assigned to the edge between nodes r_j^i and s_k^i where $X_i(s_k^i, \mathbf{t}_{-i})$ is the expected completion time of job i when s_k^i is submitted to the deadline-aware scheduler instead of t_i. The weight value represents the valuation of job i evaluated using the replica valuation type, but with the completion time achieved if the surrogate type is submitted. With the bipartite graph constructed, we then compute a maximum weighted bipartite matching. This matching locates the type translations that maximize the valuation received by the users. Later in Section II-D, user will be allowed to select one replica-surrogate pair from the maximum weighted matching according to her true valuation. The selected surrogate type will be sent to the deadline-aware scheduling algorithm. Figure 1 shows the constructed bipartite graph and the corresponding maximum weighted matching.

C. Price Calculation

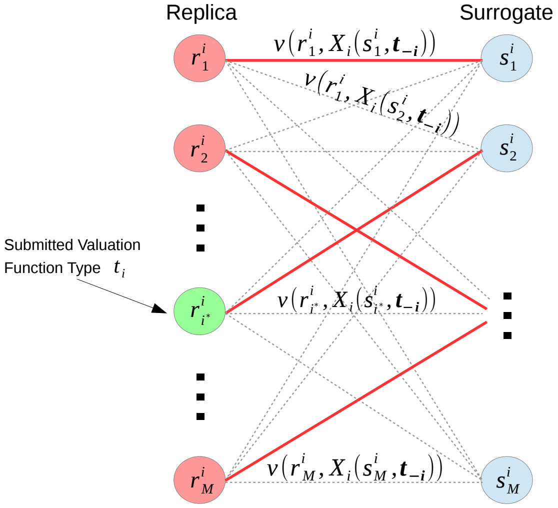
Replica
Surrogate
$v(r_1^i, X_i(s_1^i, \boldsymbol{t}_{-i}))$
$v(r_1^i, X_i(s_2^i, \boldsymbol{t}_{-i}))$
Submitted Valuation Function Type t_i
$v(r_{i^*}^i, X_i(s_{i^*}^i, \boldsymbol{t}_{-i}))$
$v(r_M^i, X_i(s_M^i, \boldsymbol{t}_{-i}))$
r_1^i
r_2^i
$r_{i^*}^i$
r_M^i
s_1^i
s_2^i
s_M^i

Figure 1. The maximum weighted matching is computed upon a bipartite graph constructed with randomly generated replica types and surrogate types. The submitted valuation type t_i is randomly inserted into the replica side.

The replica-surrogate matching is designed so that the function type translation is distribution-preserving: the surrogate type matched to the submitted valuation type t_i also has a distribution of d_i. Therefore, from the other users' points of view, the type translation does not alter the statistical behavior of user i's valuation function declaration. No extra information can be gathered that might be used to produce a more preferable outcome for the other users. This implies that an incentive-compatible pricing for the replica-surrogate matching process can motivate user i to submit a truthful valuation function type, regardless of how the other jobs are priced. As a result, we focus on designing incentive compatible pricing for the replica-surrogate matching process.

We define $p_i(t_i, \mathbf{t}_{-i})$ to be the price that user i pays as a function of user's valuation type t_i. Recall that $X_i(t_i, \mathbf{t}_{-i})$ is the expected completion time of job i given valuation type t_i and other users' valuation types $\mathbf{t}_{-i}$.

Definition 1. *A pricing mechanism is* **Bayesian incentive compatible (BIC)** *if, when all users except i truthfully report their valuation types (i.e., the $\mathbf{t}_{-i}$ are truthful), user i's expected utility is maximized by truthfully reporting as well, i.e.,*

$$\begin{aligned} & v(t_i^*, X_i(t_i^*, \mathbf{t}_{-i})) - p_i(t_i^*, \mathbf{t}_{-i}) \\ & \geq v(t_i^*, X_i(t_i, \mathbf{t}_{-i})) - p_i(t_i, \mathbf{t}_{-i}), \forall t_i. \end{aligned}$$

In this definition, and in the rest of the paper, **we take a user's utility $\mathbf{u}_i$ to be her valuation minus the price paid.** Such utilities are called quasi-linear.

To find a BIC mechanism, we note that the maximum weighted matching in the replica-surrogate matching process can be considered as an auction game. In this game, the players are the replica types, who bid on the "items" represented by the surrogate types. Each player submits a bid equivalent to the edge weight of the bipartite graph and only wants one item. The auction game's social choice function assigns items to the players so that the social surplus (i.e., summation of the bids for the assignment) is maximized. This auction game is a typical example of VCG auction. It is a well-known result that

[1] We discuss the selection of M in Section III-D.

Table I
TABLE OF NOTATION

Symbol	Definition	Symbol	Definition
t_i	valuation type of job i	$\mathbf{t}_{-i}$	vector of valuation type of all jobs except i
t_i^*	truthful valuation type of job i	M	number of replica-surrogate pairs in the matching
$X_i(t_i, \mathbf{t}_{-i})$	expected completion time of job i given $(t_i, \mathbf{t}_{-i})$	s_j^i	jth surrogate valuation type of job i
$v(t_i, X_i(t_i, \mathbf{t}_{-i}))$	value of job i evaluated with t_i at $X_i(t_i, \mathbf{t}_{-i})$	r_j^i	jth replica valuation type of job i
$M_{\mathcal{S}}^{\mathcal{R}}$	total weight sum of matching with replica set $\mathcal{R}$ and surrogate set $\mathcal{S}$	$p_{j,k}^i$	price associated with match (r_j^i, s_k^i)

an incentive compatible pricing scheme exists for the VCG auction. Denote by $\mathcal{R}$ the set of generated replica types and $\mathcal{S}$ the set of surrogate types. We define $M_{\mathcal{S}}^{\mathcal{R}}$ as the summation of the edge weights in the maximum weighted matching. For every replica-surrogate pair (r_j^i, s_k^i), the incentive compatible prices, denoted as $p_{j,k}^i$, can be computed as

$$p_{j,k}^i = M_{\mathcal{S}}^{\mathcal{R}\setminus\{r_j^i\}} - M_{\mathcal{S}\setminus\{s_k^i\}}^{\mathcal{R}\setminus\{r_j^i\}}, \qquad (1)$$

where $\mathcal{A} \setminus \mathcal{B} = \{x \in \mathcal{A} | x \notin \mathcal{B}\}$. The price calculated above represents the marginal harm caused to other participants (i.e., other replica types) by r_j^i. User i is then charged a price $p_{i^*,k}^i$ corresponding to the replica $r_{i^*}^i$ and its matched surrogate.

Theorem 1. *The replica-surrogate matching with pricing calculated using Equation* (1) *forms a BIC mechanism.*

The proof is analogous to that of Theorem 3.1 in [13].

D. Menu of Contracts

The replica-surrogate matching scheme assumes that each user declares a valuation type t_i with her job. However, in reality, quantifying a valuation function that accurately describes the users' requirements is not a trivial task. To relieve the user of this burden, we construct a menu of contracts that shows the user the different choices of service and their associated auction prices. Recall that there are M replica-surrogate matchings in the maximum weighted matching of the bipartite graph. For each matching between replica r_j^i and surrogate s_k^i, a contract of $(X_i(s_k^i, \mathbf{t}_{-i}), p_{j,k}^i)$ is added to the menu. If the user selects this contract, the corresponding job will receive an expected completion time of $X_i(s_k^i, \mathbf{t}_{-i})$,a and the user should expect a payment of $p_{j,k}^i$. The user can thus explicitly declare her job requirements by choosing her desired contract from the menu.

The menu of contracts method preserves Theorem 1's Bayesian incentive compatibility because of the well-known *taxation principle* in mechanism design. Since every incentive compatible mechanism results in a unique matching between the possible valuation type declarations (i.e., the replicas) and the algorithm results (i.e., the surrogate), choosing the preferable outcome from the menu is equivalent to a user choosing her true valuation and its associated algorithm outcome from the menu. Asking the user to pick her preferred contract from the menu is then equivalent to declaring her true valuation type.

E. Ex-Post Price Adjustment

The price in Equation (1) is calculated based on the job's expected completion time, as estimated by the deadline-aware scheduler at the time when the job is submitted. However, the actual completion time may deviate from the estimation due to unforeseen uncertainties in the system. Since the completion times of the jobs directly affect the valuations received by the users, users who experience delays suffer from a utility loss because they do not receive the desired service quality promised by the prices they paid. Therefore, it is only fair to the users if their prices are adjusted according to the actual completion times of their jobs. We propose a price adjustment method that preserves Bayesian incentive compatibility (Theorem 1) and ensures that users are motivated to participate in the system, which we call ex-post individual rationality:

Definition 2. *A pricing mechanism satisfies* ex-post individual rationality *if it yields a nonnegative utility for each user with probability 1.*

Let x_i^* be the actual completion time of user i's job, and let r_j^i and s_k^i be the replica and surrogate types in the selected contract, respectively. Let p_i be the price chosen by user i in the menu of contracts. Recalling that $X_i(s_k^i, \mathbf{t}_{-i})$ is the expected completion time returned by the deadline-aware scheduler given the surrogate type s_k^i, we propose an ex-post price adjustment

$$p_i^* = p_i \frac{v(r_j^i, x_i^*)}{v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))}. \qquad (2)$$

We discount the price by the ratio between the actual valuation received and the expected valuation estimated when job is submitted. **This p_i^* is the final price to charge the user given that s_k^i is the surrogate type in the selected contract.** Users must obtain nonnegative utilities with the price adjustment:

Theorem 2. *Equation* (2)*'s ex-post price adjustment achieves ex-post individual rationality as defined in Definition 2.*

Proof. At the completion of the job i the user receives $v(r_j^i, x_i^*)$ value. The utility of user i is then

$$\begin{aligned} u_i &= v(r_j^i, x_i^*) - p_i \frac{v(r_j^i, x_i^*)}{v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))} \\ &= \frac{v(r_j^i, x_i^*)}{v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))} (v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i})) - p_i). \end{aligned}$$

Since $p_{j,k}^i$ is determined by VCG auction, we have $v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i})) \geq p_{j,k}^i$ because VCG achieves ex-post individual rationality. Therefore, we can easily see that $u_i \geq 0$ since $v(r_j^i, x_i^*)/v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i})) \geq 0$ as well. □

Theorem 2 shows that users always value the services received more than the prices they pay.

Theorem 3. *The ex-post price adjustment preserves Bayesian incentive comparability.*

Proof. Note that $v(r_j^i, x_i^*)$ is the value achieved with an actual completion time of x_i^* after surrogate type s_k^i in the selected contract is submitted to the deadline-aware scheduler. On average the expected valuation user i receives is $E[v(r_j^i, x_i^*)]$. Recall that by definition, $v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))$ is the expected valuation received by user i when submitting s_k^i. We then have $E[v(r_j^i, x_i^*)] = v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))$. Therefore,

$$\begin{aligned} E[p_i^*] &= E\left[p_i \frac{v(r_j^i, x_i^*)}{v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))}\right] \\ &= p_i \frac{E[v(r_j^i, x_i^*)]}{v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))} = p_i. \end{aligned}$$

With or without the ex-post price adjustment, the user expects to receive a valuation $v(r_j^i, X_i(s_k^i, \mathbf{t}_{-i}))$ and a price p_i. □

F. Social Surplus Bound

On average, each user suffers from a reduction in her achieved valuation (i.e., the value of $v(t_i, x_i)$ due to the use of surrogate instead of submitted valuation types in the deadline-aware scheduler, we can prove that the value reduction is minimal when we generate a sufficiently large number of replica and surrogate types for the matching process. Denote by $\mathbf{D} = d_1 \times d_2 \times ...d_N$ the joint distribution of valuation function types for all active jobs. We define $E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(\mathbf{t}))]$ to be the expected surplus contributed by job i when submitting type t_i and $E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(s_k^i, \mathbf{t}_{-i}))]$ to be the expected surplus contributed by job i by submitting type t_i and mapping to a surrogate type s_k^i. The difference between these two values represents the cost of introducing the replica-surrogate matching to job i. Let L be the dimension of the valuation function type distribution (i.e., number of parameters submitted by the user to the job profile class). We then bound the reduction in social surplus:

Theorem 4. *For any* $0 < \epsilon < 1$, *the BIC mechanism with* $M = \Omega(\frac{\sqrt{L}^L}{2^L}\epsilon^{-L-2})$ *achieves*

$$E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(s_k^i, \mathbf{t}_{-i}))] \le (1 - O(\epsilon))E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(\mathbf{t}))].$$

Proof. The parameter vectors (i.e., valuation function type) in the valuation function type distribution are drawn from $\tau \subset \mathcal{R}^L$, where $\mathcal{R}^L$ is the L-dimension Euclidean space. Consider an L-dimension hypercube that can be circumscribed by an L-sphere with an arbitrary radius of ϵ. It is trivial to show that the edge of the hypercube has a length of $d = \frac{2\epsilon}{\sqrt{L}}$. Let τ have dimensions of $C_1, \ldots, C_L$, where C_i represents the constant bounded range of parameter i in the valuation function type distribution estimation. Therefore, τ can be covered by $(\prod_{i=1}^L C_i)/d^L$ hypercubes. Let τ' be the subset of τ that contains the centers of the L-spheres that circumscribe the hypercubes. It is clear that τ' is an ϵ-cover of τ with a cardinality of $\frac{\prod_{i=1}^L C_i \sqrt{L}^L}{(2\epsilon)^L}$. From Theorem 3.2 of [13], we have

$$\begin{aligned} &E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(s_k^i, \mathbf{t}_{-i}))] \\ &\le (1 - O(\epsilon + \sqrt{\frac{\frac{\prod_{i=1}^L C_i \sqrt{L}^L}{(2\epsilon)^L}}{M}}))E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(\mathbf{t}))] \\ &= (1 - O(\epsilon + \epsilon\sqrt{\prod_{i=1}^L C_i}))E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(\mathbf{t}))] \\ &= (1 - O(\epsilon))E_{\mathbf{t}\sim\mathbf{D}}[v(t_i, X_i(\mathbf{t}))]. \end{aligned}$$

□

Theorem 4 shows that the gap between the expected surplus value contributed by each user with and without the BIC mechanism can be made arbitrarily small when the number of replica-surrogate pairs M is large enough. Since M determines the size of the bipartite graph and hence the algorithm runtime, we find a trade-off between the computational workload and the social surplus. Theorem 4 provides a guideline on how to select M given a bound ϵ on the social surplus loss. Notably the required value of M increases almost exponentially as L increases, with the percentage loss ϵ serving as the exponent.

III. Architecture and Implementation

In this section, we explain how we implement the proposed BIC mechanism and how the components interact with each other. The BIC mechanism mainly interacts with three components: (i) the owner of the job, (ii) valuation function type estimator, and (iii) deadline-aware job scheduler. We assume that the deadline-aware job scheduler is a black-box system. In the following section, we propose one possible implementation of the valuation function type distribution estimator. However, our implementation allows other distribution estimation methods to be included as plug-in modules.

A. Interactions among the Components

Figure 2 summarizes the interactions of the implementation components. When a new job is submitted to the computing cluster, the user selects a job profile class that best fits the behavior of the job. Users then input the profile parameters (e.g., deadline, completion time sensitivity, weight) using a template that is tailor-made for this class. These parameters will be submitted as training data to the corresponding valuation type distribution estimator associated to the selected job profile class as training data. The BIC mechanism then uses the resulting distribution to generate surrogate types. By submitting these surrogate types to the deadline-aware scheduler, the corresponding expected completion time of each job can be estimated by the deadline-aware scheduler black-box. This expected completion time will be used to prepare a menu of contracts, which will be presented to the user. After the user has selected one of the contracts, the surrogate type of the selected contract is submitted to the deadline-aware scheduler to be scheduled. When the job is finished, its actual completion time will be fed back to the BIC mechanism to perform ex-post price adjustment. The user will be charged the final price when the job is completed. Figure 3 shows the time-line of the message exchanges that realize this process.

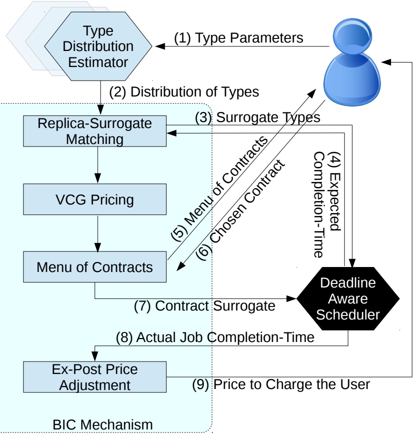

Figure 2. Interactions among the user, the BIC mechanism, the valuation function type distribution estimator and the deadline-aware scheduler.

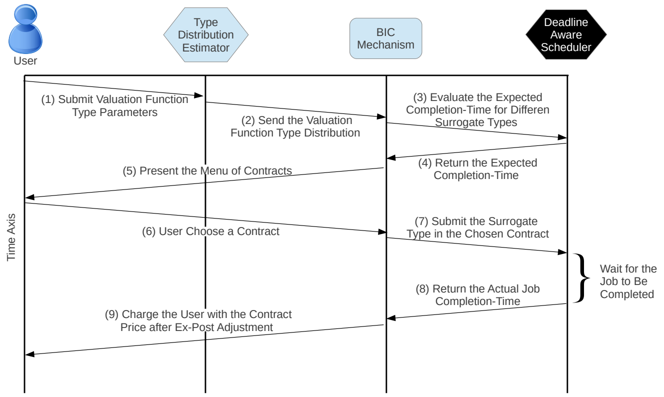
User
Type
Distribution
Estimator
BIC
Mechanism
Deadline
Aware
Scheduler
Time Axis
(1) Submit Valuation Function
Type Parameters
(2) Send the Valuation
Function Type Distribution
(3) Evaluate the Expected
Completion-Time for Differen
Surrogate Types
(4) Return the Expected
Completion-Time
(5) Present the Menu of Contracts
(6) User Choose a Contract
(7) Submit the Surrogate
Type in the Chosen Contract
Wait for the
Job to Be
Completed
(8) Return the Actual Job
Completion-Time
(9) Charge the User with the Contract
Price after Ex-Post Adjustment

Figure 3. Time-line of the protocol message exchanges.

B. Valuation Function Type Estimation

Accurately estimating the distribution of users' valuation types is necessary to guarantee the incentive compatibility of the replica-surrogate matching based pricing mechanism. To improve the estimation accuracy, we propose to group jobs into multiple job profile classes. Jobs in the same class are homogeneous in terms of their valuation function type. One advantage of grouping jobs into separated classes is that we can develop a tailor-made distribution estimation method for each of the classes.

Recall that a valuation function type is represented by a vector of parameters. Thus, estimating the distribution of the valuation function types is equivalent to estimating the probability of a user submitting each parameter vector. We propose to use multivariate kernel density estimation to learn this distribution. We assume that users are rational and submit truthful valuation function types, since the BIC mechanism will penalize untruthful declarations. For each job profile class r, a parameter template is provided to the user as an input form to enter the parameters. We suppose that the parameter vector for job profile class r has a dimension L, and that there are in total n parameter vectors submitted to job profile class r. Let $\mathbf{d}_k^r$ be the kth parameter vector inputted. The multivariate kernel density estimator for job profile class r can be written as $P_r = \frac{1}{n}\sum_{k=1}^{n} |\mathbf{H}|^{-\frac{1}{2}} K(|\mathbf{H}|^{-\frac{1}{2}}(\mathbf{x}-\mathbf{d}_k^r))$ where $K(\mathbf{x}) = (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2}\mathbf{x}^T\mathbf{x}}$ is the standard multivariate normal kernel and $|\mathbf{H}|$ is the bandwidth matrix. The estimation accuracy of the multivariate kernel density estimation method heavily depends on the choice of the bandwidth matrix. Previous works have proposed to select the optimal bandwidth matrix by minimizing the asymptotic approximated mean integrated squared error. The details of the multivariate kernel density estimation method are out of the scope of this paper. Please refer to [14] for the asymptotic approximated mean integrated squared error analysis.

In our proposed system, the valuation function type estimator is implemented as a plug-in module that attaches to each job profile class. The following are some typical job classes that we consider in this work.

- **Hard Constant Deadline Class**: Jobs belonging to this class have a hard constant deadline requirement that reduces the valuation function value to zero once the jobs' completion times exceed their deadlines. The parameter vector of this job profile class contains only the deadlines of the submitted jobs.
- **Recurrent Job Class**: Jobs belonging to this class are generated from the same job template but with different workloads (e.g., raw data size). The deadlines of the jobs are proportional to their workloads. The parameter vector of this job profile class contains only the workload of the submitted job.
- **Sigmoidal Class**: The valuation functions of jobs in this class are defined by sigmoid functions that take triples of $(max\ value,\ dropping\ point,\ zero\ point)$ as parameter input. The max value is the maximum value of the sigmoid function. The valuation function starts to decline when the jobs' competion-times pass the dropping point and it becomes zero when the jobs' completion times exceed the zero point. The multivariate kernel density estimator learns the distribution of the triples $(max\ value,\ dropping\ point,\ zero\ point)$.

C. Implementation of the BIC Mechanism

Inside the proposed BIC mechanism, the price calculation starts with constructing the replica-surrogate matching. We randomly generate M replica and surrogate types from the valuation function type distribution learned for the selected job profile class. Among the M replica types, one of them is randomly selected and replaced by the valuation function type submitted by the user, so that the menu of contracts will contain a contract associated with the valuation function parameters submitted by the user. The M surrogate types are submitted to the deadline-aware scheduler in order for the scheduler to estimate the expected completion times when the valuation functions of the jobs vary. As described in Section II, the edge weight between replica j and surrogate k is calculated by $v(r^i_j, X_i(s^i_k, \mathbf{t}_{-i}))$, using the valuation function shape of the replica type r^i_j but a job completion time achieved by submitting the surrogate type s^i_k to the deadline-aware scheduler. Finally, a maximum weighted matching is generated upon the bipartite graph using for example the Hungarian algorithm, which has a time-complexity of $O(M^3)$.

With the maximum weighted matching constructed, the VCG pricing mechanism calculates a price for each matched replica-surrogate pair. For the price between replica type j and surrogate type k, two new maximum weighted matchings will be calculated (Equation (1)). The first is computed by removing replica node j from the bipartite graph. The second maximum weighted matching is computed by removing both replica node j and surrogate node k from the bipartite graph. The price is calculated as the difference between the weight-sums of these two matchings. Since there are M replica-surrogate pairs in the maximum weighted matching, the time-complexity of calculating all the prices is $O(M^4)$.

In the menu of contracts, there are M contracts that are associated with each of the M surrogate types. For each surrogate type, the expected job completion time evaluated and the price calculated together form a contract. The menu of contracts will then be sent to the user so that the user can select the one that she prefers. Let surrogate k be the surrogate type associated with the user's selected contract. The BIC mechanism then records the replica type j that matched to surrogate type k, as well as the weight of the edge between replica j and surrogate k. Next, the job will be scheduled using surrogate type k. If the job is completed by the cloud computing cluster after the expected completion time in the contract, the BIC mechanism adjusts the contract price by multiplying it with the ratio between the actual valuation the user received at the end (i.e., $v(r^i_j, x^*_i)$, where x^*_i is the actual completion time of the job) and the recorded edge weight.

D. Scale of the Computations and the Menu of Contracts

Theorem 4 shows that the percentage of valuation loss of users is $O(\epsilon)$ given that the number of replica and surrogate types in the replica-surrogate matching is $M = \Omega(\frac{\sqrt{L}^L}{2^L}\epsilon^{-L-2})$ where L is the number of user submitted parameters. For example, when the job belongs to a job profile class that only requires one parameter, if we wish to achieve 95% of the user valuation, around 4000 replica and surrogate types should be generated for the matching. However, please note that the theoretical bound provided in Theorem 4 is conservative. A more practical rule of thumb is to select $M = \Omega(\epsilon^{-L-1})$, as we do in Section IV-A. Note that M grows exponentially along with the number of valuation function type parameters. It is a good idea to focus on job profile classes with valuation function types that can be described with a few parameters (e.g., less than five, as we consider in Section III-B).

Any modern computer can easily handle the workload of processing the replica-surrogate matching problem. However, human users may have difficulty in selecting service contracts from a long menu of contracts that contains M entries. A better visual presentation can be introduced to assist the selection. For example, we can sort the contracts according to the expected job completion time, price, or marginal cost. Furthermore, we can dramatically shorten the list by showing users a uniform sample of the contracts and still maintain Bayesian incentive compatibility because uniform sampling preserves the valuation function type distribution.

IV. Evaluation

We implemented a prototype BIC pricing mechanism module for a Hadoop v2.6 cluster equipped with the RUSH deadline-aware scheduler [2]. Heterogeneous jobs with different Hadoop job templates and dataset sizes are generated from the PUMA benchmark suite [15]. In the following experiments, jobs are assigned to sigmoidal valuation functions $v(t) = 100/(1 + e^{a(t-b)})$ that are modeled using only two parameters: (1) μ, the deadline after which the value of the job starts to decline, (2) $\mu + \delta$, the time at which the value becomes zero. From μ and δ we can calculate a and b. The truthful valuation types of these jobs are generated as pairs of (μ, δ) from a joint normal distributions with mean (2000, 1000) and a covariance matrix of $[1000^2\ 0; 0\ 500^2]$. We set $M = 100$ according to the guidelines in Section III-D.

A. Pricing and Truthfulness

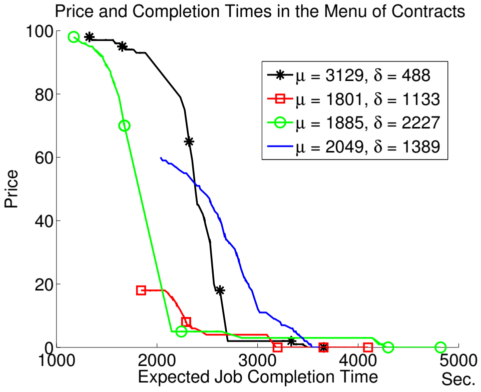

Figure 4. Prices in the menu of contracts offered to users. The non-increasing prices mean that **the better the service, the higher the payment.** Jobs are **always offered with free contracts.** Low prices after passing the jobs' deadlines (i.e., μ) indicate that **the menu of contracts is tailored for each job according to their truthful valuation function types and resource competitions in the deadline-aware scheduling logic.**

In this section, we describe how to offer users a menu of contracts that helps them truthfully select the service levels they want. To demonstrate that the proposed BIC mechanism is able to tailor a menu of contracts to different jobs' truthful valuations, we sample four menus of contracts that belong to jobs with different truthful valuation types generated from the same distribution. We generate 500 random valuation types from this distribution and submit them to the valuation function type estimators as training data (we consider the estimator accuracy with and without training data in the next section). Figure 4 shows the trade-offs between prices and the expected job completion times that are offered to each of the four jobs. Jobs receive different menus of contracts that reflect their truthful deadlines. As we would expect, **(1) users must pay more to get better service**, and **(2) each user is offered a contract with the earliest job completion time that is free.** The distinctive pricing range and magnitude indicate that **(3) the users' pricing options are tailored according to both the truthful valuation types and the resource competition with other users**. The menu of contracts represents the cost of users' QoS options determined by RUSH's scheduling logic.

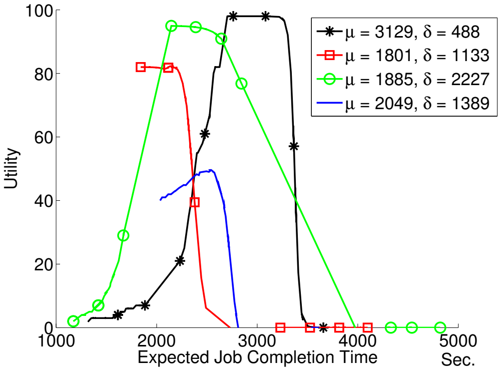

Figure 5. User utilities defined as valuation minus price. **The optimal strategy for the users is to select a contract that agrees with their truthful valuation types (i.e., select a completion time near the jobs' truthful deadline).**

To show that the pricing options are truthful, we show that the best strategy (i.e., utility-maximizing) for the user is to select the contract that best matches his or her truthful valuation type, or in other words, the contract that has an expected job completion time similar to the user's true deadline. Define user utility to be the difference between the user's truthful valuation and the price. Figure 5 shows that the utilities of the four jobs are maximized at the contracts with an expected job completion time around μ of the corresponding jobs. **Untruthfully selecting a contract with completion time $< \mu$ will thus cost users more, while untruthfully selecting a contract with completion time $> \mu$ will lead to a lower valuation that cannot be overcome by its lower price.** Please note that there exist small levels of slackness between the utility peaks and the deadlines of the jobs (e.g., the optimal utility of the blue curve appears slightly after the jobs' completion time passing the deadline). This is likely caused by two factors: (1) statistical randomness and valuation type estimation errors in the BIC model, and (2) resource competition among jobs creating extra costs to complete jobs exactly before the deadline.

B. Valuation Type Estimation Accuracy

In this section, we show the importance of an accurate valuation function type estimation. In the following experiments, 1000 jobs belonging to the same sigmoidal job profile class are submitted to the Hadoop cluster according to a Poisson arrival process with mean arrival time of 130 seconds. We repeat the experiments for three sets of training data to the valuation type estimator (500 truthful type samples, 20 truthful type samples, and 500 untruthful type samples), and compare the distributions of the user utility values assuming that the users always select the contracts that are truthful. The 500 untruthful types are generated using a joint normal distributions with mean (1000, 500) and a covariance matrix of $[500^2\ 0; 0\ 250^2]$. Figure 6 shows the empirical cumulative distribution function of the 1000 jobs' utilities for each training dataset. The well-trained valuation type estimator has a higher probability of achieving higher user utilities. More importantly, an undertrained type estimator achieves an utility distribution even worse than the utility distribution achieved by the untruthful estimator in the extreme case. It shows that untruthful training data is less damaging than insufficient training data, suggesting a degree of robustness to inaccurate training data. **Therefore, at the beginning, some inaccurate training data (e.g., historical valuation types) is better than no training data as reference.**

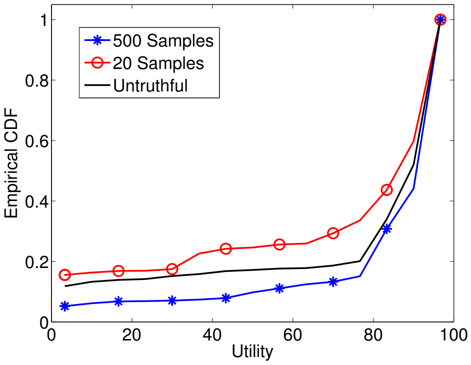

Figure 6. The cumulative distribution functions show that a truthful user will receive a higher utility when the valuation function type estimator is trained with more sample data. The result also shows that **some inaccurate training data is better than insufficient training data.**

It is important to propose a guideline on training the type estimator. We do so next by quantifying the accuracy of our prototype's multivariate kernel density estimator with different numbers of truthful samples. For each number of samples, we repeat the estimator training 1000 times. Figure 7 shows the mean and the variance values of the KL (Kullback-Leibler)-divergence between the truthful type distribution and the estimated distribution. The variance of the KL-divergence converges to nearly zero quickly after 20 samples are provided. This explains why 20 samples may not be sufficient to provide a good distribution estimation for Figure 6. The mean KL-divergence value decreases and converges after 50 samples are available, showing that 50 samples are sufficient to construct a good estimation. **It suggests that after accumulating 50 samples, we should screen out the inaccurate training samples.**

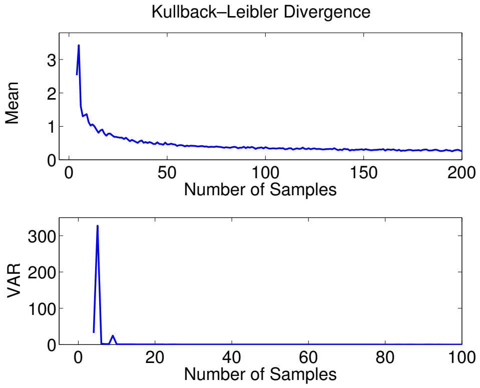

Figure 7. Mean and variance of the Kullback-Leibler-divergence between the truthful type distribution and the estimated distribution. **It suggests that the estimation becomes accurate after accumulating 50 samples, and that we should screen out the inaccurate training samples after this point.**

V. Related Work

The authors in [1] and [2] propose the deadline-aware cloud job scheduling that guarantees the fairness of achieved utilities across all users. Other works have considered the role of pricing as a way to shape user demands for cloud resources as well as to make the cloud resources more efficient and profitable [16]. The authors of [17] leverage the tradeoff between the time-elasticity of job completion and resource utilization to meet users' basic demands, constructing a dynamic auction where users can bid for spare available resources. Users can benefit from such a shared cloud environment by understanding their own demands and optimizing their bidding strategies for cloud resources [18]. However, greedy users could cheat on their demands in order to prioritize themselves in grabbing more resources. Our work allows the cloud provider to dis-incentivize these actions.

Previous works [7]–[11] have applied VCG [4]–[6] to enable truthfulness in cloud resource provisioning. All of these works attempt to ensure truthfulness in a mechanism that maximizes social welfare. However, their solutions rely on either a relaxation or an approximation of the resource allocation problem, and then integrate the VCG mechanism into the pricing of the resulting suboptimal resource allocation. Moreover, these mechanisms assume that the resource allocation maximizes social welfare, while we consider a general resource allocation that can incorporate other objectives.

Other work [12], [13], [19] have treated resource bidding as a game among users, and users reach a Bayesian-Nash Equilibrium with Bayesian incentive compatibility. Some works [20], [21] have charged the successful bidder at the price submitted by the next unsuccessful bidder, while [22] requires the price to be nondecreasing in terms of bid submission time.

Most of the above work merely consider truthfulness in requesting fixed amounts of resources, with a few [9], [22] additionally considering job completion times.

VI. Conclusion

We proposed a BIC pricing mechanism that determines how much to charge cloud users when they submit jobs with completion time requirements. The proposed mechanism achieves desired truthfulness properties such as Bayesian incentive compatibility and ex-post individual rationality. We implement the proposed pricing mechanism. Through experiments in a Hadoop cluster with real-world datasets, we show that our prototype successfully dis-incentivizes untruthful user behaviors.

References

[1] Z. Huang, B. Balasubramanian, M. Wang, T. Lan, M. Chiang, and D. H. Tsang, "Need for speed: Cora scheduler for optimizing completion-times in the cloud," in *Proc. of IEEE INFOCOM*, 2015.

[2] Z. Huang, B. Balasubramanian, M. Wang, T. Lan, M. Chiang, and D. H. Tsang, "RUSH: A robust scheduler to manage uncertain completion-times in shared clouds," in *Proc. of IEEE ICDCS*, 2016.

[3] D. E. Irwin, L. E. Grit, and J. S. Chase, "Balancing risk and reward in a market-based task service," in *Proc. of IEEE HPDC*, 2004.

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

[5] E. H. Clarke, "Multipart pricing of public goods," *Public choice*, vol. 11, no. 1, pp. 17–33, 1971.

[6] T. Groves, "Incentives in teams," *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.

[7] M. M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," *IEEE Trans. on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 594–603, 2015.

[8] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," *Journal of the ACM*, vol. 58, no. 6, p. 25, 2011.

[9] N. Jain, I. Menache, J. S. Naor, and J. Yaniv, "A truthful mechanism for value-based scheduling in cloud computing," *Theory of Computing Systems*, vol. 54, no. 3, pp. 388–406, 2014.

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

[11] X. Zhang, C. Wu, Z. Li, and F. C. M. Lau, "A truthful (1-ε)-optimal mechanism for on-demand cloud resource provisioning," in *Proc. of IEEE INFOCOM*, 2015.

[12] J. D. Hartline and B. Lucier, "Bayesian algorithmic mechanism design," in *Proc. of ACM symposium on Theory of computing*, 2010.

[13] J. D. Hartline, R. Kleinberg, and A. Malekian, "Bayesian incentive compatibility via matchings," in *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, 2011.

[14] M. Wand and M. Jones, *Kernel Smoothing*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Taylor & Francis, 1994.

[15] "Puma: Purdue Mapreduce benchmarks suite." https://sites.google.com/site/farazahmad/pumabenchmarks.

[16] L. Zheng, C. Joe-Wong, C. G. Brinton, C. W. Tan, S. Ha, and M. Chiang, "On the viability of a cloud virtual service provider," in *Proc. of ACM SIGMETRICS*, 2016.

[17] X. Yi, F. Liu, Z. Li, and H. Jin, "Flexible instance: Meeting deadlines of delay tolerant jobs in the cloud with dynamic pricing," in *Proc. of IEEE ICDCS*, 2016.

[18] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang, "How to bid the cloud," in *Proc. of ACM SIGCOMM*, 2015.

[19] X. Bei and Z. Huang, "Bayesian incentive compatibility via fractional assignments," in *Proc. of ACM SODA*, 2011.

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

[21] S. Zaman and D. Grosu, "A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds," *IEEE Trans. on Cloud Computing*, vol. 1, no. 2, pp. 129–141, 2013.

[22] 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*, 2016.