New Mechanism for Reservation in Cloud Computing

(Extended Abstract)

Changjun Wang[1] Weidong Ma[2], Tao Qin[2], Feidiao Yang[2],
Tie-Yan Liu[2], Xujin Chen[1], Xiaodong Hu[1]
[1]Chinese Academy of Sciences, Beijing, China.
[2]Microsoft Research, Beijing, China.
[1]{wcj, xchen, xdhu}@amss.ac.cn,
[2]{weima, taoqin, feiyan, tyliu}@microsoft.com

ABSTRACT

In this paper, we study the problem of designing new mechanism for selling reserved *instances* in cloud computing. Our goal is to maximize the social welfare. We propose a prompt mechanism in the sense that the acceptance and the payment for a job is determined at the very moment of its arrival. We show that the mechanisms has a competitive ratio of $O(\ln(kT))$ under some mild assumption, where k (res. T) is the maximum ratio between per-instance-hour valuation (res. length) of any two jobs. We then prove that no algorithm can achieve a competitive ratio better than $\ln(2kT)$ under the same assumption. Therefore, our mechanism is optimal within a constant factor.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Design, Economics, Theory

Keywords

Mechanism design, Competitive analysis, Posted price

1. INTRODUCTION

Cloud computing becomes more and more popular nowadays because it can provide scalable and elastic access to IT resources and applications via the Internet [1, 5, 4]. Infrastructure as a service (IaaS), in which computing resources are mainly sold through *instances* (also referred to as *virtual machines*), is one of several fundamental models of cloud computing. There are two main pricing models in today's IaaS market: the pay-as-you-go model (for on-demand instances), and the subscription model (for reserved instances).

In the subscription model, a user should first pay some upfront fee for the instances he/she is going to reserve. Then the user can use the reserved resource whenever he/she wants during the subscription period, under a significantly discounted per-instance-hour usage price.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

However, the options for buying reserved instances through the subscription model are very limited in today's cloud industry. For example, in Amazon EC2 users can only make a 1-year or 3-year reservation, and they cannot adjust their reservation according to their real demand. To address the limitation of the current pricing model for reserved instances, in this work we design new pricing mechanisms which allow users reserve instances depending on their own need and offer them more flexibility.

1.1 Related Work

The online resource allocation problem for cloud computing has been investigated in many papers. A closely related work is [7], in which Zhang et al. propose an online mechanism framework for cloud resource allocation. Different from our work, agents' valuation functions are continuous and concave in their model; while in our model, the valuation function of an agent is binary.

Chakrabarty et al. [3] consider the online version of knapsack problem. Both their work and our work belong to a class of online decision problem: whether to accept an item (reservation) at its arrival. The difference is that once an item is accepted in the online knapsack problem, it will be in the knapsack forever and one does not need to concern whether to pack the item inside the knapsack. In contrast, in our model, once a reservation is accepted, we need to find a time interval for the reservation (and determine its payment), and the resources can be re-used by other jobs when the time interval of the reservation has passed.

The reservation problem of perishable products originates from the airline industry [6], and then draws a lot of attention in hotel industry [2]. Comparing to the airline reservation, the hotel reservation problem is more closely related to ours since the hotel booking requests can occur for different lengths and can therefore overlap. Our work has obvious difference with these work on hotel reservation. All these work focuses on the revenue maximization, while our goal is to maximize the social welfare.

2. PROBLEM FORMULATION

We study the problem of designing pricing mechanisms for instance reservation in IaaS cloud. We formally setup the problem in this section.

Let C denote the capacity of a cloud provider, i.e., the cloud provider has C instances (virtual machines) for reservation. Customers/agents arrive sequentially and submit their reservation requests. For simplicity, we also call a reser-

vation a job, which is characterized by (r_j, d_j, n_j, v_j), where r_j is the earliest start time of the reservation, d_j is the latest end time of the reservation, n_j ($\ll C$) is number of instances to reserve, and v_j is the value of the reservation. We use $l_j (= d_j - r_j)$ to denote the time length of the reservation. For any job set S, we use $v(S)$ to denote $\sum_{j \in S} v_j$.

We consider an online setting, in which jobs arrive online and the provider has no knowledge about future jobs. Once a job j arrives, its attributes r_j, d_j and n_j are revealed to the cloud system, the system immediately calculates the price p_j of the reservation, and then the agent (the owner of the job) decides whether to accept the price. We assume that the agent is *self-interested* and *rational*. Note that the agent does not need to reveal his/her value v_j to the system.

The goal of the system is to maximize the social welfare (the total value of the accepted jobs). Once a job is accepted, it cannot be rejected or interrupted in the future. We use the competitive ratio to evaluate the performance of a mechanism, i.e., to compare the mechanism against the optimal offline solution.

DEFINITION 1. *A mechanism $\mathcal{M}$ is* β-competitive *in terms of social welfare if for any job sequence θ, we have $SW(\mathcal{M}, \theta) \geq \frac{1}{\beta} OPT(\theta)$, where $SW(\mathcal{M}, \theta)$ denotes the social welfare achieved by $\mathcal{M}$ over θ and $OPT(\theta)$ denotes the optimal social welfare over θ.*

It is easy to verify that no mechanism that can achieve a constant competitive ratio if there is no restriction on lengths or value density of jobs. Therefore, we make the following assumptions and propose mechanisms under these assumptions.

- We assume the *per-instance-hour valuation* $\rho_j = \frac{v_j}{n_j l_j}$ of every job is supported in $[\rho_{\min}, \rho_{\max}]$.
- The length of every job falls in a known interval $[l_{\min}, l_{\max}]$.

Denote $k := \frac{\rho_{\max}}{\rho_{\min}}$ and $T := \frac{l_{\max}}{l_{\min}}$. Without loss of generality, we further assume $\rho_{\min} = 1$ and $l_{\min} = 1$.

3. MECHANISM

In this section, we design a mechanism for selling reserved instances. In our mechanism, when a job j arrives, we will set a p_j for its request (r_j, d_j, n_j). If the job accepts the price and pays for it for its reservation, then we say our mechanism accepts the job.

In the following, we first present our post-price mechanism and then prove that the competitive ratio of the mechanism is $O(\ln(kT))$.

Let $\gamma_j(t)$ denote the utilization rate of the cloud at time t when job j arrives, i.e.

$$\gamma_j(t) := \sum_{i \in S_j \text{ and } \lfloor r_i \rfloor \leq t < \lceil d_i \rceil} \frac{n_i}{C},$$

where S_j denotes the set of jobs that come earlier than j and are accepted by the cloud.

Our proposed mechanism is shown in Mechanism 1, in which $f(x)$ is an auxiliary function defined as below.

$$f(x) = (3kT + 1)^x - 1$$

The intuition on the auxiliary function is that the price monotonically depends on the cloud utilization rate x: the higher the utilization rate, the higher the price. We show that this specific form can lead to a good competitive ratio.

Mechanism 1

Pricing rule:
When job j arrives,

- If j does not overfill the system (i.e. $\forall t \in \left[\lfloor r_j \rfloor, \lceil d_j \rceil\right)$, $\gamma_j(t) + \frac{n_j}{C} \leq 1$), set price

$$p_j = \int_{\lfloor r_j \rfloor}^{\lceil d_j \rceil} \int_{\gamma_j(t)}^{\gamma_j(t) + \frac{n_j}{C}} \frac{f(x) \cdot \rho_{\min} \cdot C}{3} dx dt;$$

- Otherwise, set price $p_j = +\infty$.

Allocation rule:
If j accepts the price p_j and pays for it, allocate n_j instances to j from r_j to d_j.

THEOREM 1. *The competitive ratio of Mechanism 1 is at most* $1 + 12\ln(3kT + 1)$ *if* $\frac{n_j}{C} \leq \frac{\ln 3/2}{\ln(3kT+1)}, \forall j$.

In the following, we give a lower bound $\Omega(\ln(kT))$ on the competitive ratio of any deterministic algorithm for the case $n_j = 1, \forall j$. This bound shows that our mechanisms are optimal within a constant factor.

THEOREM 2. *The competitive ratio of any deterministic algorithm for this problem is at least* $\ln(2kT)$.

4. CONCLUSION

In this work, we design a mechanism for selling reserved instances in cloud. There are multiple aspects to investigate to make the mechanisms practically useful. First, we have assumed that we have zero knowledge about future jobs. In practice, one can learn from historic jobs and predict future jobs, and design better mechanisms and achieve better social welfare. Second, in this work we have assumed the cloud provider has fixed capacity C. In practice, a cloud provider need to serve on-demand instances, reservation instances, and maybe also spot instances. It is necessary to consider the competition between different kinds of instances while designing mechanisms for reservation.

REFERENCES

[1] Amazon web services. http://aws.amazon.com/.

[2] T. K. Baker and D. A. Collier. A comparative revenue analysis of hotel yield management heuristics. *Decision Sciences*, 30(1):239–263, 1999.

[3] D. Chakrabarty, Y. Zhou, and R. Lukose. Online knapsack problems.

[4] Google cloud platform. https://cloud.google.com/.

[5] Microsoft azure. http://azure.microsoft.com/en-us/.

[6] E. L. Williamson. Airline network seat inventory control: Methodologies and revenue impacts. Technical report, [Cambridge, Mass.: Massachusetts Institute of Technology, Dept. of Aeronautics & Astronautics], Flight Transportation Laboratory,[1992], 1992.

[7] 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 *Proceedings of INFOCOM*, pages 1510–1518. IEEE, 2013.