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

Shi Weijie

March 7, 2013

Model and Bidding Language

Truthfulness and COCA Auction

Competitive analysis

Online Auction

- Off-line auction: wait until there are many bidders (1 hour or 1 day)
- Online auction: user come and get resources immediately (1 minute or 1 second)
- Similarity with online algorithm: make allocation promptly, no future information
- Difference: also needs to decide the payment

Framework

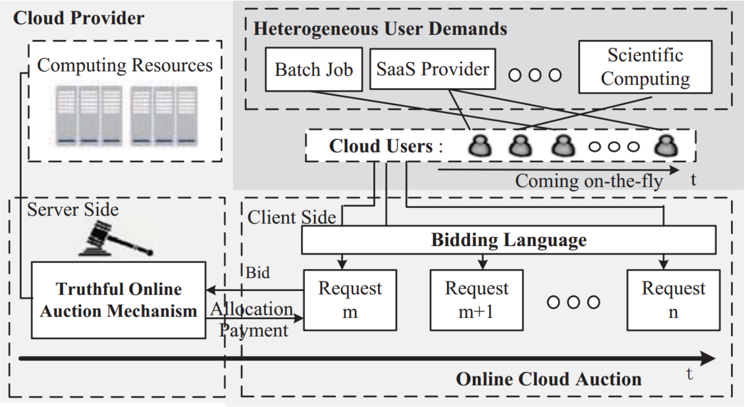
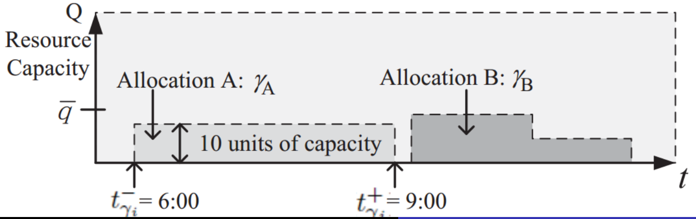


Fig. 1. Infrastructure of the framework for truthful online cloud auctions with heterogeneous user demands

Resource and allocation

- The provider has a fixed capacity of Q VMs.
- Allocation function $\gamma_i(t)$ is the resource user i get. Assume $\gamma_i(t)$ is in range $[0, \overline{q}]$
- In the following example, Allocation A: γ_A gets 10 units from 6:00 to 9:00. Then $\gamma_A(t) = 10$ for $t \in [6:00, 9:00]$, and $\gamma_A(t) = 0$ elsewhere. Denote $t^+_{\gamma_A} = 6:00$, $t^-_{\gamma_A} = 9:00$
- For γ_B, $\gamma_B(t) = 10$ for $t \in [9:10, 10:30]$ and $\gamma_B(t) = 5$ for $t \in [10:30, 11:20]$.



Valuation

- $v_i(\gamma_i)$ is the valuation of the user i. It maps an allocation function to a real number.
- We restrict the valuation to have some special forms.
- Just an example, a user wants to finish a job of size 40 within a time period. Then he will have a valuation like this:

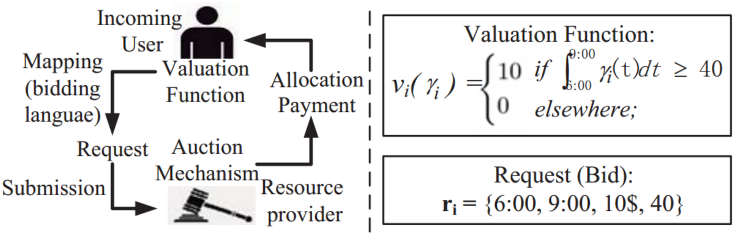


Fig. 2. An illustrative example of the online cloud auction.

Bidding Language

- Type I: job-oriented users
- Type II: Resource-aggressive users
- Type III: Time-invariant capacity users

Type I: job-oriented users

- Similar to the previous example
- $r_i = \{a_i, d_i, pen\_rate_i, b\_total_i, size_i\}$

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

Type II: Resource-aggressive users

- The more resource during the period, the better
- Described by a concavely increasing function $b_i(\cdot)$, maps the amount of total resource to a real number.
- For example, a user is willing to pay \$5 for a job size of 10, and \$8 for a job size of 20. Then $b_i(10) = 5, b_i(20) = 8$.
- $r_i = \{a_i, d_i, b_i(\cdot)\}$

$$v_i(\gamma_i) = b_i(\int_{a_i}^{d_i} \gamma_i(t)dt)$$

Type III: Time-invariant capacity users

- A user wants an invariant number inv_cap_i of units during a length l_i of time.
- For example, a user wants 1 VM for 10 time units, or 2 VM for 10 time units, and is willing to pay \$3 and \$4, respectively. Then $l_i = 10, b_i(1) = 3, b_i(2) = 4$.
- $b_i(\cdot)$ now maps the amount of units per time slot, to a real number.
- $r_i = \{a_i, d_i, l_i, b_i(\cdot)\}$

$$v_i(\gamma_i) = b_i(inv\_cap_i) \cdot l_i$$

- Assume: the valuation for one unit per time slot is within a known interval $[\underline{p}, \overline{p}]$.
- Assume: the job length of Type III bidders is within $[\underline{l}, \overline{l}]$

Utility

- $u_i(\gamma_i)$ is the "profit" bidder i gets from an allocation γ_i
- $u_i(\gamma_i) = v_i(\gamma_i) - pay_i$
- Selfish bidders try to maximize their utility

Social welfare

- The criterion to evaluate the performance of an auction mechanism
- The welfare of the provider is the payment collected from the buyers pay_i
- The welfare of the buyers is their profit (utility) $u_i = v_i - pay_i$
- So the total social welfare is the sum of every buyers' valuation $E_A(\tau) = \sum_{i \in \tau} v_i(\gamma_i)$, here τ is a sequence of requests.

Payment function

- The payment is decided by a function: $pay_i = p_i(\gamma_i, t_{sub_i}, r_i)$. Here t_{sub_i} is the submission time of the request r_i.

Lemma

For any truthful auction algorithm A, for any bidder i, given γ_i and t_{sub_i}, the payment function should be independent of his request r_i.

Proof.

If we have $p_i(\gamma_i, t_{sub_i}, r_i) > p_i(\gamma_i, t_{sub_i}, r_i')$. Then the bidder with true valuation r_i will declare r_i'. □

Monotonic

Definition

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_{subi} and any allocation $\gamma_i \succeq \gamma'_i$, we have $p_i(\gamma_i, t_{subi}) \geq p_i(\gamma'_i, t_{subi})$.

Definition

A payment function p_i is monotonic with submission time if for any $gamma_i$ and any submission time $t_{subi} \leq t'_{subi}$, we have $p_i(\gamma_i, t'_{subi}) \geq p_i(\gamma_i, t_{subi})$.

Truthful necessity

Theorem

For any truthful online auction mechanism A, the payment function should be monotonic with submission time and with allocation.

Proof.

First, assume p_i is not monotonic with allocation. Then a bidder have 2 possible allocation, one is strictly better than another, but charged less. Denote r_i' and r_i will lead to these two allocation respectively. Then the user with true valuation r_i will lie r_i', get more resource and pay less.

For the submission time, it is similar. The monotonic property is the only way to prevent the users from delaying their requests. □

Allocation

Theorem

For any truthful auction A, the allocation decision maximizes the utility of each bidder i.

Proof.

First, notice that the user's utility is determined uniquely by the allocation he receives, and is independent of his request.
If the allocation decision is not the optimal allocation for him, then he will find a request that can lead to the optimal allocation. □

- Denote all possible allocation to some bidder i as a set Γ_i, then for any bidder i, the allocation decision is:

$$\gamma_i^* = argmax_{\gamma_i \in \Gamma_i}(v_i(\gamma_i) - p_i(\gamma_i, t_{subi}))$$

Back to payment

- Intuitively, the price should be higher if there are less available resources (during the peak time).
- Utilization rate $U(t_1, t_2)$: t_2 is the current time, t_1 is the future time.
- $U(t_1, t_2)$ is the rate between the allocated resources in t_1 to the total amount Q
- We count the reserved units at t_1 until now (t_2).

Auxiliary pricing function

- $P(x)$: the marginal price with respect to U. Higher U, higher $P(U)$.
- $P(x)$ can be predetermined by the provider, and can be any type of nondecreasing function.
- The payment can be calculated as:

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

Mechanism conclusion

- Decide $P(x)$, construct the payment function $p_i(\gamma_i, t_{subi})$
- Search all the possible allocations, find the optimal one γ_i^*
- Determine the payment pay_i
- Updating the utilization rate

Some discussions

- Can be extended to arbitrary valuation functions. Can define any other format of bidding language and allow other types of users.
- Computational complexity: $O(L^2 Q \log Q)$, here L is the number of discrete time slots.

Competitive ratio

- c-competitive means, for every request sequence τ, $E_A(\tau) \geq E_{VCG}(\tau)/c$

Define P(x)

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

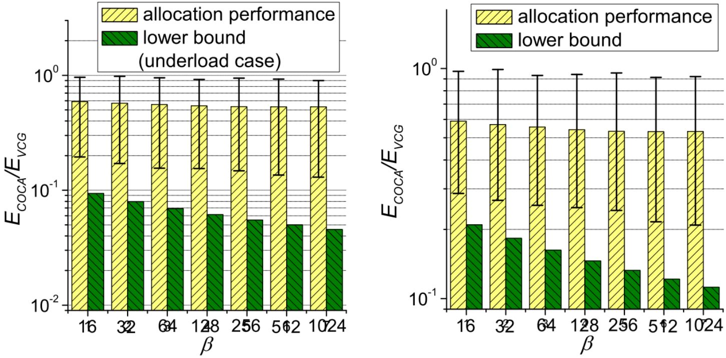
$$P_1(x) = \begin{cases} \overline{p}/e^{(1-x)\cdot r} & 1/r \leq x \leq 1 \\ \underline{p} & 0 \leq x < 1/r \end{cases} \tag{9}$$

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

Competitive ratio

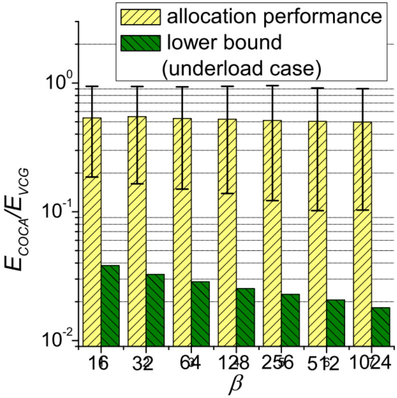
Proposition 2. *For any request sequence τ consisting of bidders of Request Type I, II and III, with auxiliary pricing function $P_1(x)$, COCA is $O(log(\overline{p}/\underline{p}))$-competitive in the underload case, as long as $\overline{q} \leq Q/ln(\overline{p}/\underline{p})$.*

Simulation



(a) Type I Bidders (b) Type II Bidders

Simulation



(d) The mixture arrival case