Revenue Maximization with Dynamic Auctions in IaaS Cloud Markets

Wei Wang, **Ben Liang, Baochun Li**
Department of Electrical and Computer Engineering
University of Toronto
June 3, 2013

Prevalent Pricing Schemes for IaaS Clouds

On-demand (pay-as-you-go)

- Static hourly rate

Reservation

- One-time reservation fee to reserve one instance for a long period
- Free or discount rate during the reservation period

Bid-based (spot market)

- Users bids for computing instances
- A spot price is posted periodically
- No service guarantee
    - Instance terminates when the spot price exceeds the submitted bid

Prevalent Pricing for IaaS Clouds (Cont'd)

Comparison

	Upfront commitment	Service guarantee	Market responsiveness
On-demand (pay-as-you-go)	N	Y	Slow
Reservation	Y	Y	Slow
Bid-based	N	N	Fast

Can we do better?

Desired Properties

	Upfront commitment	Service guarantee	Market responsiveness
On-demand (pay-as-you-go)	N	Y	Slow
Reservation	Y	Y	Slow
Bid-based	N	N	Fast
New design	N	Y	Fast

Dynamic Auctions

A sequence of auctions periodically carried out

- Users bid for a number of computing instances (VMs)
- Each winning user receives a fixed usage fee (hourly rate) throughout its usage

Guaranteed services

- A user's instance will never be terminated against its will

Quick response to market dynamics

- Use the auction to discover the "right price"
- More flexible and profitable than the static pricing

Our Contributions

Near-optimal dynamic auctions with provable performance

The optimal design is NP-hard (0-1 knapsack problem)

Computationally efficient

By taking use of some optimization structures, we significantly reduce the computational complexity

Truthfulness

A user has no incentive to lie on its bids

General model

A cloud provider has allocated a fixed capacity C to host a type of instance

At any time, the number of hosted instances cannot exceed C

A sequence of auctions, indexed by $t=1,2,\ldots$, are periodically carried out

In period t, N_t users arrive, bidding for instances

User model

User *i* arrives at *t* and bids for computing instances

Reported bid = (# of requested instances, maximum price)

True bid: private information

It is possible that the user lies on its bid

No partial fulfilment: A user is either rejected or gets all requests fulfilled

User receives a fixed hourly rate if it wins

User's Problem

User *i* chooses the best bid to maximize its utility

$$u_i(r_i, b_i) = \begin{cases} \sum_{j=1}^{n_i} (v_i - p_i) l_{i,j} - \sum_{j=n_i+1}^{r_i} p_i l_{i,j} \,, & \text{if } r_i \geq n_i; \\ 0 \,, & \text{o.w.} \end{cases} \tag{1}$$

User *i* has no incentive to lie on its bid (truthful) if and only if its true bid maximizes the utility

Cloud Vendor's Problem

Decide how many instances to auction off at each time t

Design the optimal auction mechanism M_t at each time t

Decide the winners and their prices

$$V_t^*(C_t) = \mathbf{E}\Bigg[\max_{\mathcal{M}_t, 0 \le Q_t \le C_t} \{ \Gamma_{\mathcal{M}_t}(Q_t) + \mathbf{E}_K \left[V_{t+1}^*(C_t - Q_t + K) \right] \} \Bigg].$$

Revenue generated at time t

Future revenue

C_t: # of instances available at time t

Q_t: # of instances auctioned off at time t

How many instances should be auctioned off?

NP-Hardness and Relaxations

Directly solving the problem is at least as hard as a 0-1 Knapsack problem

Because no partial fulfillment is allowed

A relaxed problem

Solve the problem *as if* partial fulfillment is allowed

$$\bar{V}_t(C_t) = \mathbf{E}\Big[\max_{\mathcal{M}_t, 0 \le Q_t \le C_t} \{ \bar{\Gamma}_{\mathcal{M}_t}(Q_t) + \mathbf{E}_K [\bar{V}_{t+1}(C_t - Q_t + K)] \} \Big].$$

Auction revenue with partial fulfillment

Saturday, 29 June, 13

Optimization Structure

Directly solving the relaxed problem is inefficient

Dynamic programming takes $O(C^3)$ time, where C is the number of instances that can be hosted (capacity)

Reduce the computational complexity based on some optimization structures

No need to compute from scratch

Reuse previous computation results

$$Q_\tau^*(c+1) - 1 \le Q_\tau^*(c) \le Q_\tau^*(c+1)$$

Reduce the complexity to $O(C^2)$

Truthful auction based on the capacity allocation strategy

Design a truthful auction mechanism

The following auction mechanism is truthful based on the previous capacity allocation strategy

Algorithm 1 The Truthful Mechanism $\mathcal{M}_t$ with Q_t^* Instances Allocated

1. Let k be the index such that $\sum_{j=1}^{k} r_j \leq Q_t^* < \sum_{j=1}^{k+1} r_j$
2. Let $s = \sum_{j=1}^{k} r_j$
3. Let $\hat{b}_s = \phi^{-1}(q\nabla\bar{\mu}_{t+1}(C_t - s + 1))$
4. Top k bidders win, each paying $p = \max\{b_{k+1}, \hat{b}_s\}$

Evaluations

High-Demand Market

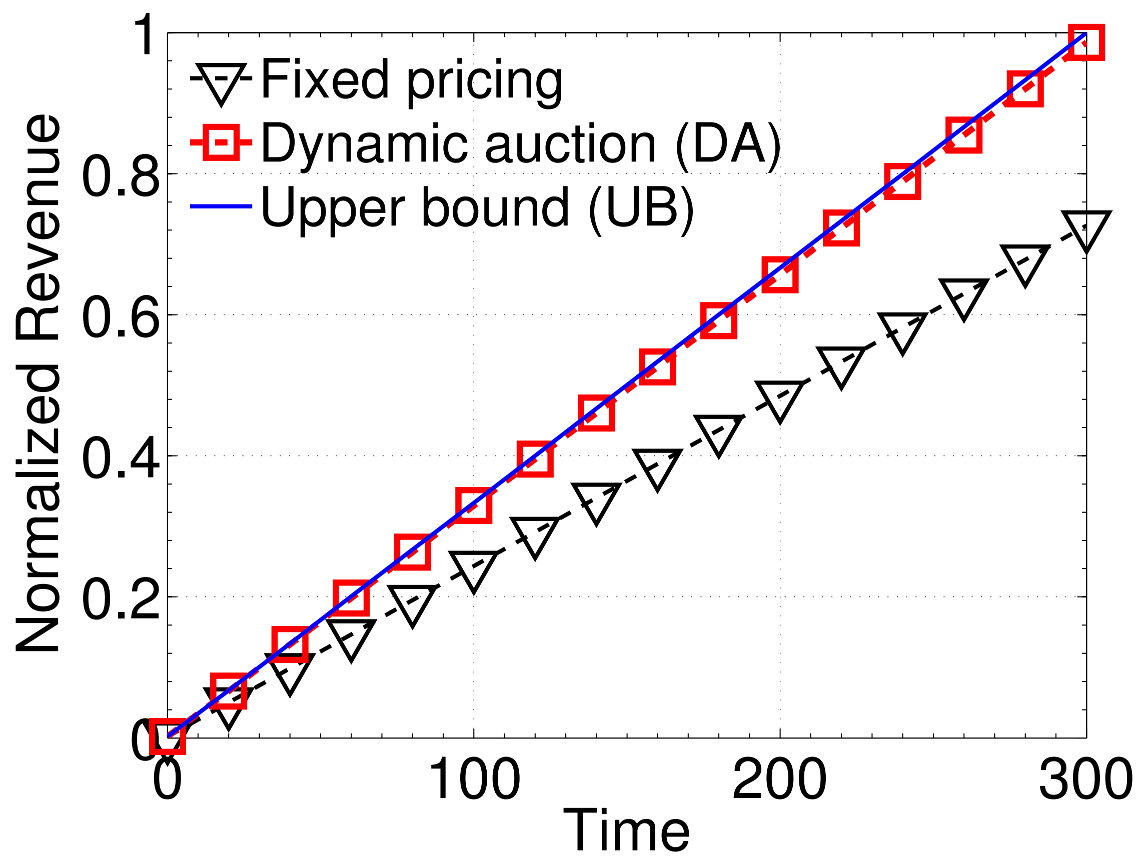
Asymptotical optimality for high-demand market

Proposition 3: The expected revenue $V_t \to V_t^*$ w.p.1 if the user number $N_\tau \to \infty$ for all $\tau = t, \ldots, T$.

Low-Demand Market

Outperform the *fixed pricing* by 30% in terms of the revenue

< 1% revenue loss compared to the optimal design



(a) Normalized revenue vs. time.

Conclusions

Dynamic auctions offer service guarantees while capturing the market dynamics quickly

We have designed near-optimal dynamic auctions

- Truthful
- Asymptotically optimal for high-demand market
- Computationally efficient

Dynamic auctions generate more revenue than the traditional static pricing scheme

Thanks!

http://iqua.ece.toronto.edu/~weiwang/