Efficient Online Mechanisms for Dynamic Cloud Resource Provisioning

Shi Weijie

Why Auction?

- Fixed price (supermarket)

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
m3.medium	1	3	3.75	1 x 4 SSD	$0.113 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	$0.225 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	$0.450 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	$0.900 per Hour

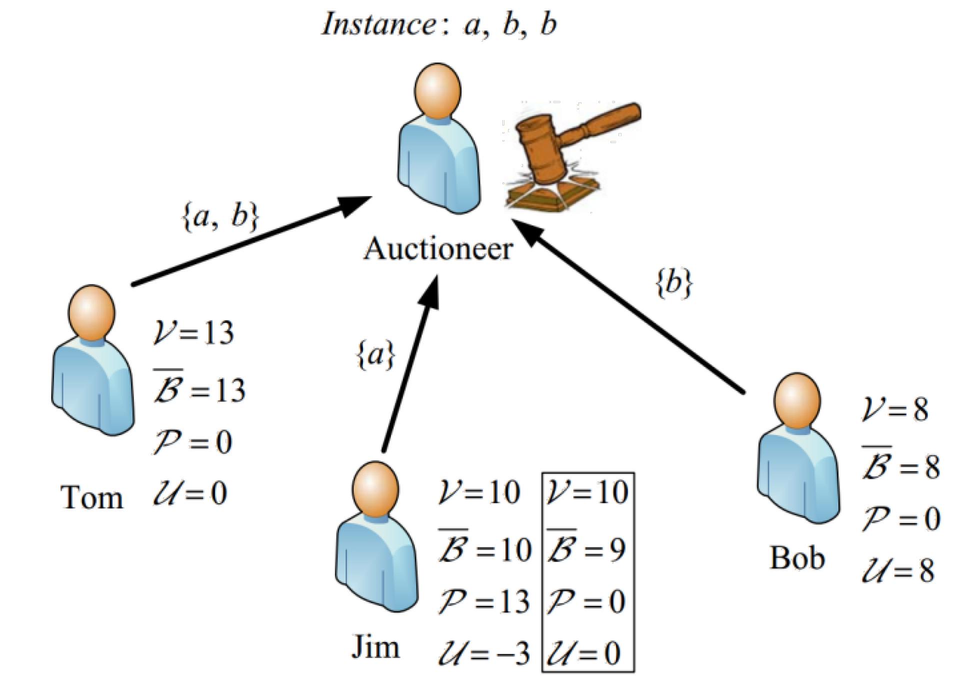
Why Auction?

- In cloud market:
  - Fluctuating demand
  - Real-time supply
- Fixed pricing
  - requires accurate estimation

Existing Mechanisms

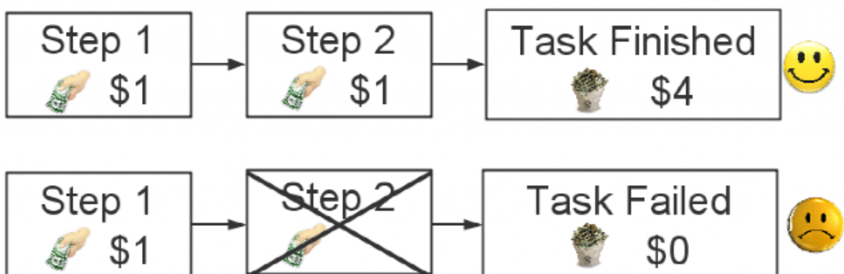
- Amazon Spot Instance
- Wang (Infocom 12)
  - When Cloud Meets eBay: Towards Effective Pricing for Cloud Computing
- Zhang (Infocom 13)
  - (COCA) A Framework for Truthful Online Auctions in Cloud Computing with Heterogeneous User Demands

Existing Mechanisms
Wang (Infocom 12)



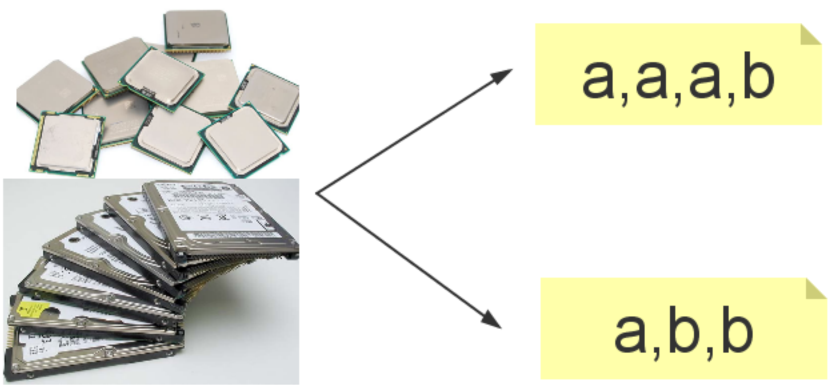
Existing Mechanisms
Wang (Infocom 12)

- One round
  - Repeat the one round auction
  - Inconvenient
  - Uncertainty for users



Existing Mechanisms
Wang (Infocom 12)

- Pre-determined VMs
  – Provider can dynamically assemble VMs



Existing Mechanisms
Zhang (Infocom 13)

- Only one VM type
- Users must reveal their departure time

Our Work

- (1) An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing

- (2) RSMOA: A Revenue and Social Welfare Maximizing Online Auction for Dynamic Cloud Resource Provisioning

Model

- Users’ bids: still round-by-round
- XOR bids: several optional bundles
- Example, user n has 2 options:

3 * VM_1 $d_{n,k,m,q}^{(t)}$
2 * VM_2
1 * VM_3

Valuation: $5 $b_{n,k}^{(t)}$

$y_{n,k}^{(t)} = 1$

1 * VM_1
2 * VM_2
3 * VM_3

Valuation: $4

$y_{n,k}^{(t)} = 0$

Model

- Budget: Connects different rounds

  B_n

- Social welfare = Total valuation
- Performance metrics $\sum_{t,n,k} b_{n,k}^{(t)} y_{n,k}^{(t)}$

$$\text{maximize} \quad \sum_{t\in[T]} \sum_{n\in[N]} \sum_{k\in[K]} b_{n,k}^{(t)} y_{n,k}^{(t)} \tag{1}$$

subject to

$$\sum_{k\in[K]} y_{n,k}^{(t)} \leq 1, \quad \forall n \in [N], t \in [T], \tag{1a}$$

$$\sum_{k\in[K]} \sum_{t\in[T]} b_{n,k}^{(t)} y_{n,k}^{(t)} \leq B_n, \quad \forall n \in [N], \tag{1b}$$

$$\sum_{n\in[N]} \sum_{k\in[K]} c_{n,k,r,q}^{(t)} y_{n,k}^{(t)} \leq A_{q,r}^{(t)}, \quad \forall q \in [Q], r \in [R], t \in [T], \tag{1c}$$

$$y_{n,k}^{(t)} \in \{0,1\}, \quad \forall n \in [N], k \in [K], t \in [T]. \tag{1d}$$

Online Problem

- What's the difficulty about budget?

User A	Budget $10	User B	Budget $10
Round 1	$10	Round 1	$9
Round 2	$8	Round 2	$1

Greedy: Round 1 A; Round 2 B. Total valuation $11

Optimal: B; A. Total valuation $17

Lesson Learned

- Do not exhaust users' budgets early!
  - Lose all the opportunities on this user
- But... What if this round is the "Best Opportunity"?
- Classical online problem dilemma

Budget Coefficient

- Original valuation * [Coefficient]
- Remaining budget →0, coefficient →0
- At beginning, coefficient = 1

$$(1 - x_n^{(t)})$$

- Update each round

The Online Framework

Algorithm 1 The Online Algorithm Framework $\mathcal{A}_{online}$

$x_n^{(0)} \leftarrow 0, \forall n \in [N]$

// Loop for each time slot

for all $1 \leq t \leq T$ **do**

$$w_{n,k}^{(t)} = \begin{cases} 0 & \text{if } x_n^{(t-1)} \geq 1 \\ b_{n,k}^{(t)}(1 - x_n^{(t-1)}) & \text{otherwise} \end{cases}, \forall n \in [N], k \in [K].$$

Run $\mathcal{A}_{round}$. Let $\mathcal{N}$ be the set of winning users, and k_n be the index of their corresponding winning bundle, for each winning user $n \in \mathcal{N}$.

for all $n \in \mathcal{N}$ **do**

$$x_n^{(t)} \leftarrow x_n^{(t-1)} \left(1 + \frac{b_{n,k_n}^{(t)}}{B_n}\right) + \frac{b_{n,k_n}^{(t)}}{B_n(\gamma - 1)}$$

end for

for all $n \notin \mathcal{N}$ **do**

$x_n^{(t)} \leftarrow x_n^{(t-1)}$

end for

end for

$x_n \leftarrow x_n^{(T)}, \forall n \in [N]$

One-round & Multi-round

- One round auction $\mathcal{A}_{round}$

- Allocation
  - Combinatorial Optimization
- Payment
  - Incentive compatible (truthful)

Combinatorial Optimization

Algorithm 2 A Primal-Dual Algorithm to Solve One-round Allocation Problem (3)

$\mathcal{N} \leftarrow \emptyset$, $z_{base} \leftarrow QR \cdot e^{(C^{(t)}_{min}-1)}$
$y^{(t)}_{n,k} \leftarrow 0, s^{(t)}_{n} \leftarrow 0, z^{(t)}_{q,r} \leftarrow 1/A^{(t)}_{q,r}, \forall n \in [N], k \in [K], r \in [R], q \in [Q]$
while $\sum_{r\in[R]} \sum_{q\in[Q]} A^{(t)}_{q,r} z^{(t)}_{q,r} < z_{base}$ AND $|\mathcal{N}| \neq N$ **do**
    for all $n \notin \mathcal{N}$ **do**
        $k(n) = \arg\max_{k\in[K]}\{w^{(t)}_{n,k}\}$
    end for
    $n^* = \arg\max_{n\in[N]}\{\frac{w^{(t)}_{n,k(n)}}{\sum_{r\in[R]}\sum_{q\in[Q]} c^{(t)}_{n,k(n),r,q} z^{(t)}_{q,r}}\}$
    $y^{(t)}_{n^*,k(n^*)} \leftarrow 1, s^{(t)}_{n^*} \leftarrow w^{(t)}_{n^*,k(n^*)}, \mathcal{N} \leftarrow \mathcal{N} \cup \{n^*\}$
    for all $r \in [R], q \in [Q]$ **do**

$$z^{(t)}_{q,r} \leftarrow z^{(t)}_{q,r} \cdot {z_{base}}^{c^{(t)}_{n^*,k(n^*),q,r}/(A^{(t)}_{q,r} - C^{(t)}_{q,r})}$$

    end for
end while

VCG Auction

- Calculate the optimal allocation
- Payment rule: opportunity cost
- Guarantees truthfulness

- But, allocation: NP-Hard

Fractional VCG

- Relax on $y_{n,k}^{(t)}$
- Allocation: Linear Programming
- Payment: the same rule

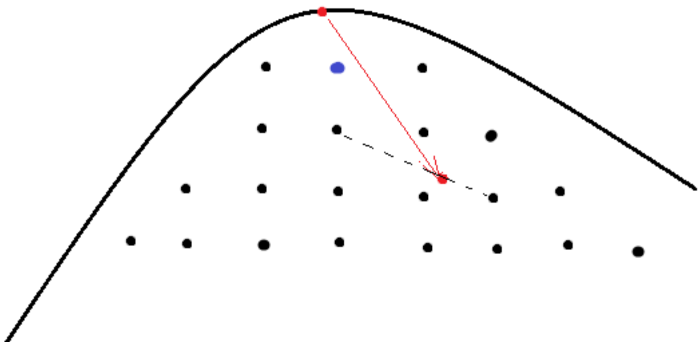
- But, we cannot offer 0.6 instance of VM!

Randomized Decomposition

- User A User B User C Pr
- 0.3 0.8 0.5
- 1 1 0 0.3
- +
- 0 1 1 0.5
- +
- 0 0 0 0.2

Linear Decomposition

- Scale down by c



- Dynamic Provisioning
- Online setting (budgets)
- Truthful, etc.
- Efficient (competitive ratio)

Model

- Time-invariant valuation
- Continuous time interval
- Terminate at any time

Properties

- Efficient   Social welfare & Provider Revenue
- Truthful   Time & valuation
- Non-decreasing user utility

	Time t_1	t_2
Val	9	12
Pay	6	7
U	3	5

Opposite Approach

- First work:
  - Allocation User A gets bundle #2
  - Payment User A pays $10 for #2

- This work:
  - Payment If #1, pays $3
    #2, pays $10
    #3, pays $6
  - Allocation Gets #2

Payment Function

- Independent of his bid

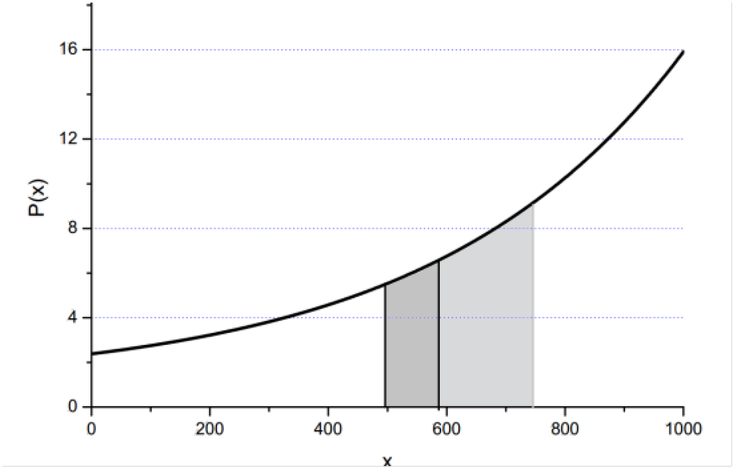
- Depends on (1) other users' bids
- (2) allocation result

$\pi_n(\boldsymbol{d}_n, t)$

Payment Function

- Higher demand, higher price

$$\pi_n(\boldsymbol{d}_n, t) = \int_{x_0}^{x_0 + x(\boldsymbol{d}_n)} P(y)dy$$



Allocation

- Customer-first principle
  - Maximize utility for the user
  - The only way to achieve truthfulness

- (1) Calculate payment for all options
- (2) Pick the best one as allocation decision
- Repeat (1)(2) for each user

Price Curve

- Threat-based strategy
- Competitive ratio O(ln p)

Future Work

- Allocation of bandwidth
- Different structure of problem