Core-Selecting Auction Design for Dynamically Allocating Heterogeneous VMs in Cloud Computing

Haoming Fu†, Zongpeng Li†, Chuan Wu‡, Xiaowen Chu§

† University of Calgary
‡ The University of Hong Kong
§ Hong Kong Baptist University

June 24, 2014
Cloud markets

- Heterogeneity nature of virtual machine (VM) instances;
- Resource provisioning (resources: CPU, RAM, storage, etc.);
- Auction for efficient VM allocation (e.g. Amazon Spot Instances).

Figure: (a) VM configuration from Spot Instances. (b) Illustration of a cloud market.
The classic VCG mechanism is the only auction mechanism that is both truthful and efficient, but it suffers from two severe economic problems: low revenue and vulnerability to shill bidding.*

Relaxing either truthfulness or efficiency is inevitable to solve these economic problems.

Efficiency leads to NP-hard winner determination problems (0 – 1 integer programs).

Empirical studies show that: integer programs with 5000 variables can be solved in second (using CPLEX on a laptop computer).

Relaxing (economic) efficiency is less justified.

*Shill bidding problem is applicable to combinatorial auctions only and not to simple auctions.
Our VM Auction Design

- We apply the **core-selecting auctions**, which achieve a revenue at least on par with that of VCG mechanisms and are robust against shill bidding. The core-selecting auction is further tailored to minimize CUs’ incentives to deviate from truthful bidding.
- A **three-dimension auction framework** is proposed for dynamic resource provisioning.
- We propose a **combinatorial auction** that is expressive enough to sell bundles of heterogeneous VMs.
**Problem Model**

- **XOR bidding language**: a cloud user (CU) can submit multiple bids, but can win a single bid only.
- $\mathcal{B}_i$ is the set of all VM bundles CU $i$ bids for.

**Utility**

The quasi-linear utility of CU $i$ is:

$$u_i = \begin{cases} v_i(S) - p_i & \text{if CU } i \text{ wins a bundle } S \in \mathcal{B}_i \\ 0 & \text{otherwise} \end{cases}$$

The utility of the cloud provider (CP) is: $u_o = \sum_{i \in N} u_i$.

All CUs are individually rational; $v_i(S)$ is the maximum amount that CU $i$ is willing to pay for $S$. 
winner determination problem (WDP)

\[ w(N) = \max \sum_{i \in N} \sum_{S \subseteq B_i} b_i(S) x_i(S) \]

subject to:

\[ \sum_{j=1}^{m} n_j a_i^j \leq \pi_k \quad \forall 1 \leq k \leq t; \]

\[ \sum_{S \subseteq B_i} x_i(S) \leq 1 \quad \forall i \in N; \]

\[ \sum_{i \in N} \sum_{S \subseteq B_i} x_i(S) r_j \leq n_j \quad \forall 1 \leq j \leq m, \ S = (r_1, r_2, \ldots, r_m); \]

\[ n_j \in \mathbb{N} \quad \forall 1 \leq j \leq m; \]

\[ x_i(S) \in \{0, 1\} \quad \forall i \in N, \ \forall S \subseteq B_i. \]

Theorem 1

Relaxing \( n \) to take fractional values in the WDP does not change the value of \( w(N) \).
The Core of Our VM Auction

An auction outcome is *blocked* by coalition $\mathcal{C} \subseteq \mathcal{N}$ if there is an alternative outcome which generates strictly more revenue for the auctioneer and no less utility for each CU $i \in \mathcal{C}$.

**Core**: Set of outcomes that are not blocked.

\[
\text{Core}(\mathcal{N}) = \{ \mathbf{u} \geq 0 | \sum_{i \in \mathcal{N} \cup \{o\}} u_i = w(\mathcal{N}), \sum_{i \in \mathcal{C} \cup \{o\}} u_i \geq w(\mathcal{C}), \forall \mathcal{C} \subseteq \mathcal{N} \}\]
The Core of Our VM Auction

Example: The cloud provider (auctioneer) has 25 CPUs and 25 GB storage in its resource pool. 7 CUs each submits one bid:

\[
\begin{align*}
    b_1(6, 0, 1)^8_7 &= 4, & b_2(2, 3, 0)^5_{11} &= 5, \\
    b_3(0, 0, 6)^{12}_6 &= 4, & b_4(7, 0, 0)^7_7 &= 27, \\
    b_5(0, 4, 0)^4_{12} &= 25, & b_6(0, 0, 6)^{12}_6 &= 24, \\
    b_7(5, 3, 7)^{24}_{22} &= 33.
\end{align*}
\]

To keep CU 1 from blocking: \( p_4 \geq 4 \). Similarly: \( p_5 \geq 5, p_6 \geq 4 \).

To keep CUs 1, 2, 3 from blocking: \( p_4 + p_5 + p_6 \geq 33 \).

Individual rationality.

Table: VM configuration.

<table>
<thead>
<tr>
<th></th>
<th>VM_1</th>
<th>VM_2</th>
<th>VM_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Storage</td>
<td>1GB</td>
<td>3GB</td>
<td>1GB</td>
</tr>
</tbody>
</table>

Figure: Illustration of the core.
The Core of Our VM Auction

Theorem 2

The payment vector of first price auction is always in the core of our VM auction. (proof by definition)
The VCG mechanism

\[ p_i^{\text{VCG}} = b_i(S_i) - (w(\mathcal{N}) - w(\mathcal{N} \setminus \{i\})) \]

Low revenue: a revenue of $4+5+4=13$ is gleaned in view of the fact that the winners are willing to pay a total up to $27+25+24=76$. 

---

Haoming Fu, Zongpeng Li and Chuan Wu (UofC, HKU)
The Economic problems from VCG Mechanisms

\[ b_1(6, 0, 1)^8 = 4, \]
\[ b_3(0, 0, 6)^{12} = 4, \]
\[ b_5(0, 4, 0)^4 = 25, \]
\[ b_7(5, 3, 7)^{24} = 33. \]

\[ b_2(2, 3, 0)^5 = 5, \]
\[ b_4(7, 0, 0)^7 = 27, \]
\[ b_6(0, 0, 6)^{12} = 24, \]

<table>
<thead>
<tr>
<th>available</th>
<th>VM1</th>
<th>VM2</th>
<th>VM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>CPU</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25GB</td>
<td>storage</td>
<td>1GB</td>
<td>3GB</td>
</tr>
</tbody>
</table>

Vulnerability to shill bidding:
CU 5 impersonates four different CUs, each submitting this bid:
\[ b(0, 1, 0)^1 = 6.25. \] CU 5 still wins 4 instances of VM2 but reduces its payment from 4 to 0!
The Necessity of Core-Selecting Auctions

Theorem 3 (shill bidding proof)

In a VM auction formulated with WDP, no CU can earn more than its VCG utility by bidding with shills if and only if the auction is core-selecting.

\[ u_o + \sum_{i \in N} u_i \] is constant since WDP implies efficiency, then

Corollary (competitive revenue)

The total revenue in a core-selecting VM auction formulated with WDP is at least as high as that in a VCG auction.
The Necessity of Core-Selecting Auctions

Theorem 3 (shill bidding proof)

In a VM auction formulated with WDP, no CU can earn more than its VCG utility by bidding with shills if and only if the auction is core-selecting.

Proof sketch of Theorem 3

∀C ∈ N, their total utility under the VCG mechanism is \( w(N) - w(N \setminus C) \)

Our restriction is: \( \sum_{i \in C} u_i \leq w(N) - w(N \setminus C) \).

Efficiency: \( w(N) = \sum_{i \in N \cup \{o\}} u_i \).

Then \( \sum_{i \in (N \cup \{o\}) \setminus C} u_i \geq w(N \setminus C) \).

Note that:

\( \text{Core}(N) = \{ u \geq 0 | \sum_{i \in N \cup \{o\}} u_i = w(N), \sum_{i \in C \cup \{o\}} u_i \geq w(C), \forall C \subseteq N \} \)
Revenue minimization rule

We pick the payment that minimizes the total revenue over the core.

Theorem 4

A core-selecting VM auction formulated with WDP and employing a revenue-minimization payment rule minimizes CUs’ incentives to deviate from truthful bidding.
The core constraint $\sum_{i \in C \cup \{o\}} u_i \geq w(C)$, $\forall C \subseteq N$ is equivalent to:

$$\sum_{i \in W \setminus \tilde{C}} p_i \geq w(\tilde{C} \cup (N \setminus W)) - \sum_{i \in \tilde{C}} b_i(S_i), \forall \tilde{C} \subseteq W$$

Setting $\beta_{\tilde{C}} = w(\tilde{C} \cup (N \setminus W)) - \sum_{i \in \tilde{C}} b_i(S_i)$, the core constraint can be compactly written as $Ap \geq \beta$.

$$\delta = \min_1^T \cdot p$$

(revenue minimization)

subject to: $Ap \geq \beta$ (core constraint)

$p \leq b$ (individual rationality)

Drawback: revenue minimization rule lacks of precision, since the points minimizing revenue are not unique.
Among the in-core payment points minimizing the revenue, we pick the one that is closest to some pre-determined point $p'$.

$$\min(p - p')^T(p - p')$$

(subject to: $Ap \geq \beta$ (core constraint))

$p \leq b$ (individual rationality)

$1^T \cdot p = \delta$ (revenue minimization)

VCG-nearest rule and constant $p'$ reference rule

$p'$ can be set to be the VCG payment vector or some constant vector independent of CUs' bids.

Theorem 5

Under the VCG-nearest rule, the set of constraints $p \leq b$ is unnecessary.
Under the constant $p'$ reference rule, the winner with high valuation relative to the auctioneer's expectation shares less of the burden to conquer a coalitional blocking.

Origin-nearest rule: $p' = 0$.

<table>
<thead>
<tr>
<th>available</th>
<th>VM1</th>
<th>VM2</th>
<th>VM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 CPU</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>18GB storage</td>
<td>1GB</td>
<td>3GB</td>
<td>1GB</td>
</tr>
</tbody>
</table>

\[
b_1(0, 0, 6)^{12}_6 = 100 \quad b_2(0, 4, 0)^{4}_{12} = 20
\]
\[
b_3(0, 4, 6)^{16}_{18} = 60 \quad b_4(0, 0, 6)^{12}_6 = 50
\]

Winners are CUs 1 and 2.

$p^{VCG} = (50, 0)$

$p^{VCG-nearest} = (55, 5)$

$p^{origin-nearest} = (60, 0)$
Dynamic resource provisioning achieves higher social welfare, stably higher resource utilization but lower user satisfaction.
Figure: The monetary burden shouldered by the winner(s) under (a) the VCG-nearest rule, and (b) the origin-nearest rule.

\[ \mu_i = \frac{p_i^* - p_i^{\text{VCG}}}{\sum_{i \in N} (p_i^* - p_i^{\text{VCG}})} \] . Under the origin-nearest rule, the winner with high valuation relative to the auctioneer’s expectation shares less of the burden to conquer a coalitional blocking.
Figure: The monetary burden shouldered by the winner(s) under (a) the VCG-nearest rule, and (b) the origin-nearest rule.

Core-selecting auction achieves 8% higher revenue over the VCG mechanism.
Simulations

**Figure:** The monetary burden shouldered by the winner(s) under (a) the VCG-nearest rule, and (b) the origin-nearest rule.

Our core-selecting VM auction has a stable performance; revenues do not deteriorate when bidding instances are from Google Cluster Data or are randomly generated in different ways.
Core-selecting combinatorial VM auctions are novelly designed for the cloud computing market. They are proved to be proof to shill bidding and generate higher revenues than VCG auctions do.

Payment rules that minimize CUs’ incentives to deviate from truthful bidding are extensively discussed.

We advance the VM auction design by generalizing static resource provisioning to dynamic resource provisioning, and from homogeneous VM instances to heterogeneous VM instances.
Main References


Thank You!