An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing

Zongpeng Li

Department of Computer Science, University of Calgary

June 4 2014, at Institute of Network Coding

Our work on Virtual Machine Auctions

- INFOCOM'14, Dynamic Resource Provisioning in Cloud Computing: A Randomized Auction Approach
- SIGMETRICS'14, An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing
- IEEE Cloud'14, Core-Selecting Auctions for Dynamically Allocating Heterogeneous VMs in Cloud Computing
- IWQoS'14, RSMOA: A Revenue and Social Welfare Maximizing Online Auction for Dynamic Cloud Resource Provisioning

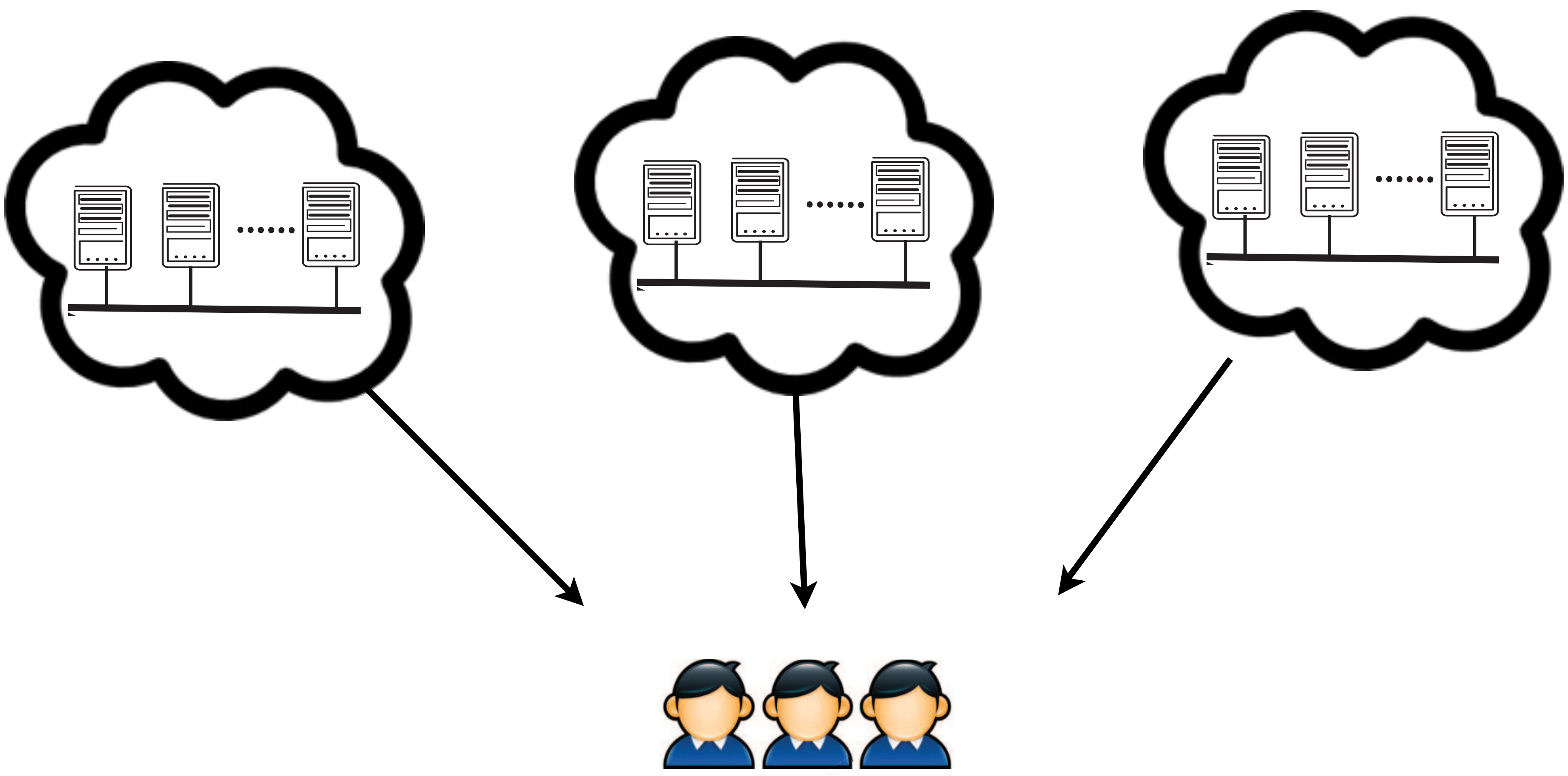
- Ongoing work: Smoothed polynomial-time auctions and FPTAS auctions for cloud computing, demand response through reverse auctions

Our work on Virtual Machine Auctions

- INFOCOM'14, Dynamic Resource Provisioning in Cloud Computing: A Randomized Auction Approach
- **SIGMETRICS'14, An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing**
- IEEE Cloud'14, Core-Selecting Auctions for Dynamically Allocating Heterogeneous VMs in Cloud Computing
- IWQoS'14, RSMOA: A Revenue and Social Welfare Maximizing Online Auction for Dynamic Cloud Resource Provisioning

- Ongoing work: Smoothed polynomial-time auctions and FPTAS auctions for cloud computing, demand response through reverse auctions

User demands different cloud resources in different geo locations



Current practice in resource provisioning

- Amazon EC2
  - fixed instance types
  - fixed prices

VM Type	CPU*	RAM	Disk	Virginia	Ireland	Tokyo
m1.medium	2	3.75GB	410GB	$0.120	$0.130	$0.175
m1.large	4	7.5GB	840GB	$0.240	$0.260	$0.350
m1.xlarge	8	15GB	1.68TB	$0.480	$0.520	$0.700
c1.medium	5	1.7GB	350GB	$0.145	$0.165	$0.185
c1.xlarge	20	7GB	1.68TB	$0.580	$0.660	$0.740
m2.2xlarge	13	34.2GB	850GB	$0.820	$0.920	$1.101

* EC2 compute units

What we want

- Customized VM instances from different datacenters

- A price for the customized VM that caters to the supply-demand relationship at this moment

What we want/what we will do

- Customized VM instances from different datacenters

- A price for the customized VM that caters to the supply-demand relationship at this moment

What we want/what we will do

- Customized VM instances from different datacenters
  - dynamic resource provisioning (i.e., dynamic VM assembly)

- A price for the customized VM that caters to the supply-demand relationship at this moment

What we want/what we will do

- Customized VM instances from different datacenters
  - dynamic resource provisioning (i.e., dynamic VM assembly)

- A price for the customized VM that caters to the supply-demand relationship at this moment
  - a new pricing scheme through an online auction that

    discovers the "right" price

    requires no estimation

    brings more social welfare than fixed pricing

What others have been doing

- Amazon Spot Instances
  - no service guarantees

- "When cloud meets eBay" (Wang et al., INFOCOM 2012)
  - one-round static auction

- COCA (INFOCOM 2013)
  - "A Framework for Truthful Online Auctions in Cloud Computing with Heterogeneous User Demands", Zhang et al.
  - one type of VMs considered

Our Contribution

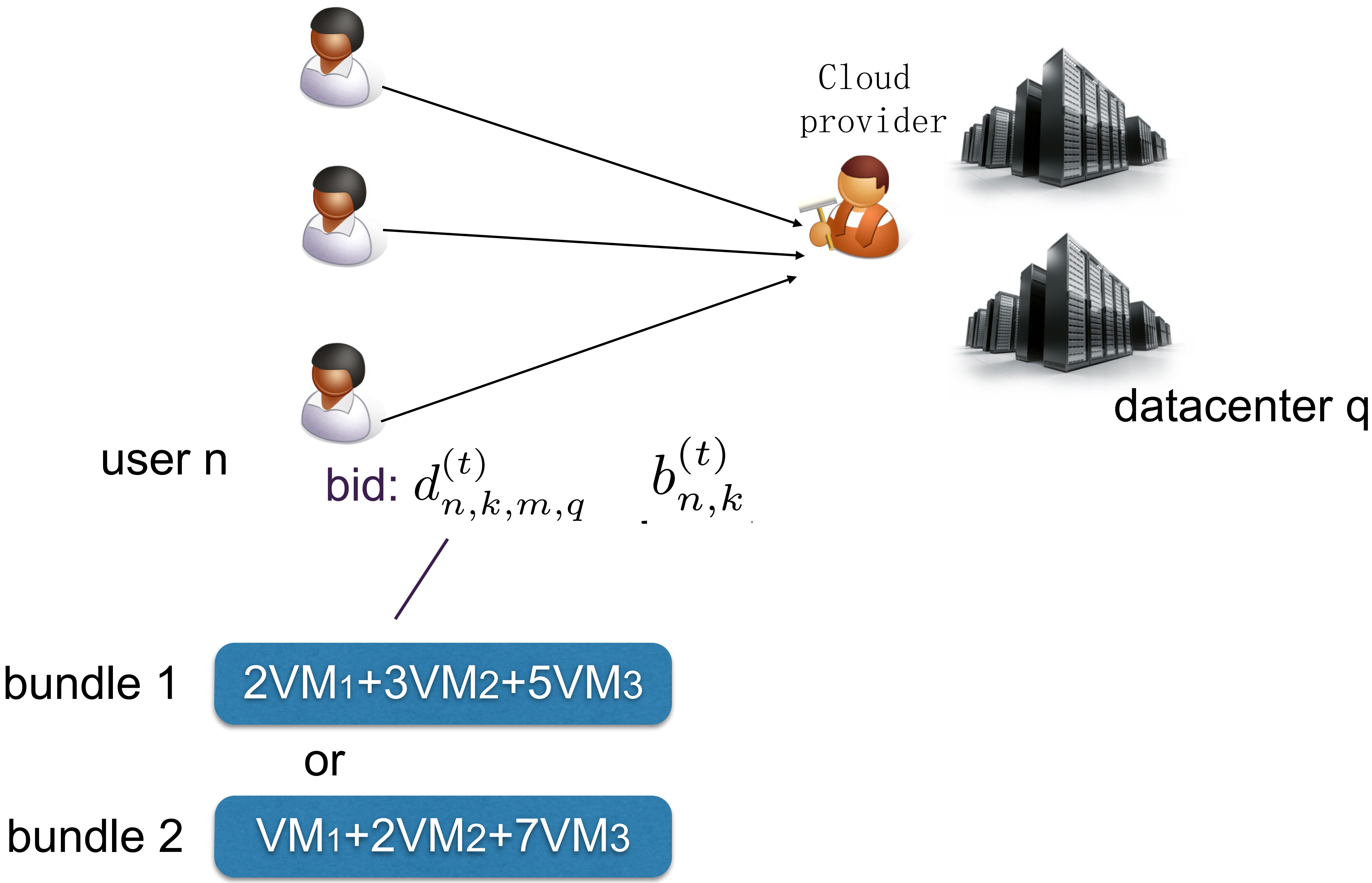
- An online auction mechanism for dynamic resource provisioning
  - users' demands arrive over time; provider responds instantly, without *a priori* information
  - nice properties

    truthful

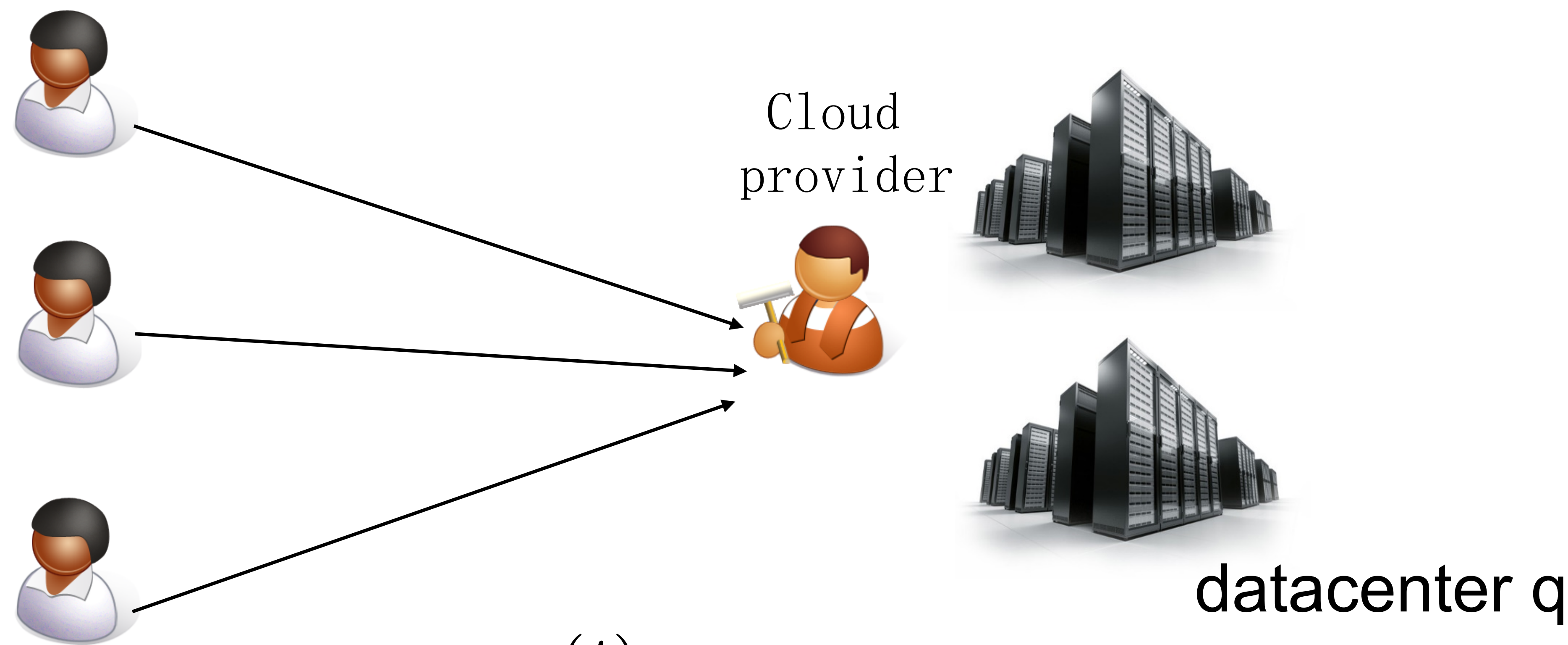
    computationally efficient

    guaranteeing a competitive ratio 3.30 in long-term social welfare in typical scenarios

Model


Cloud provider
datacenter q
user n
bid: $d^{(t)}_{n,k,m,q}$
$b^{(t)}_{n,k}$
bundle 1
2VM1+3VM2+5VM3
or
bundle 2
VM1+2VM2+7VM3

Model

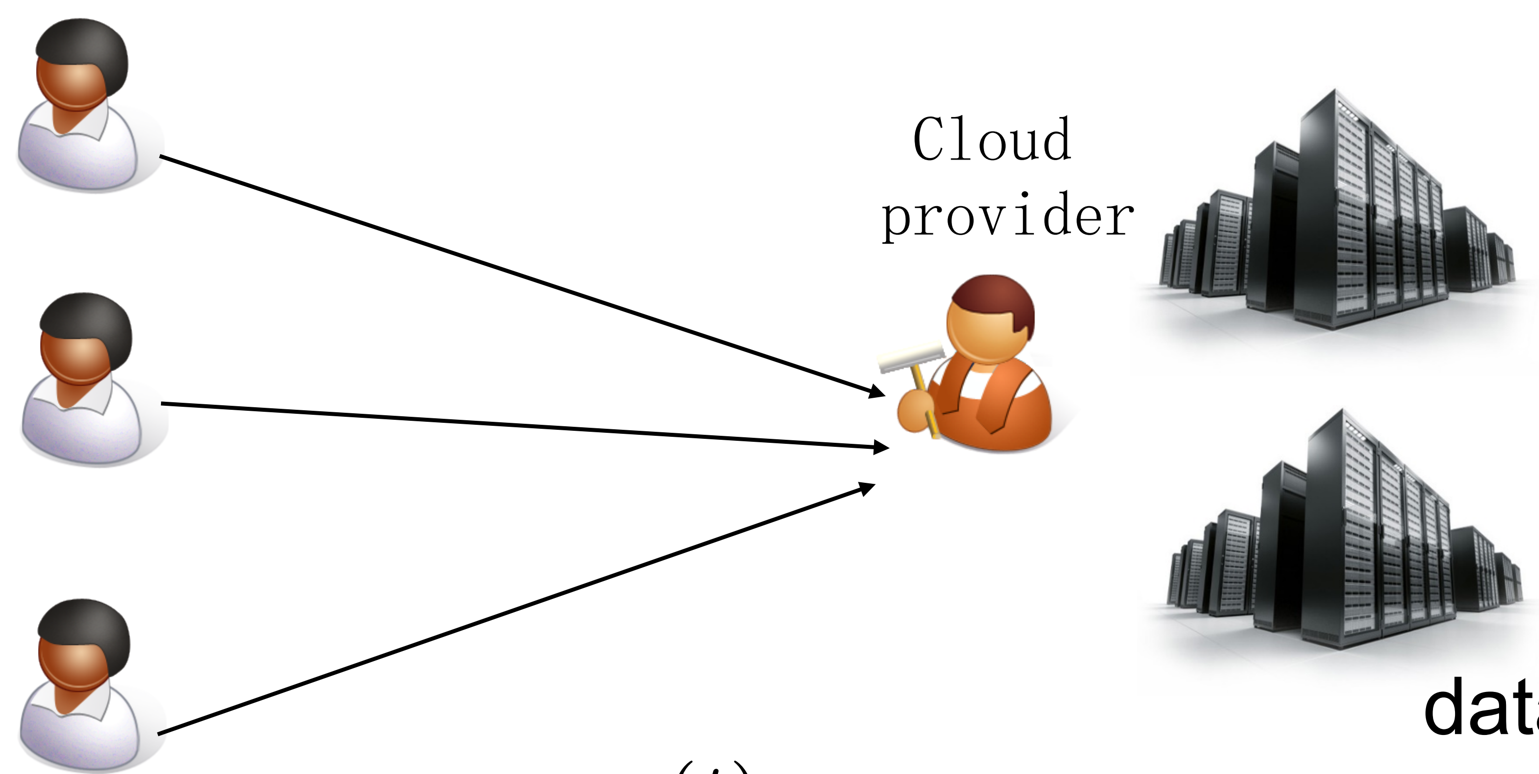


user n

bid: $d_{n,k,m,q}^{(t)}$ $b_{n,k}^{(t)}$

# of type-m VMs in datacenter q in her k-th bundle at t

Model



datacenter q

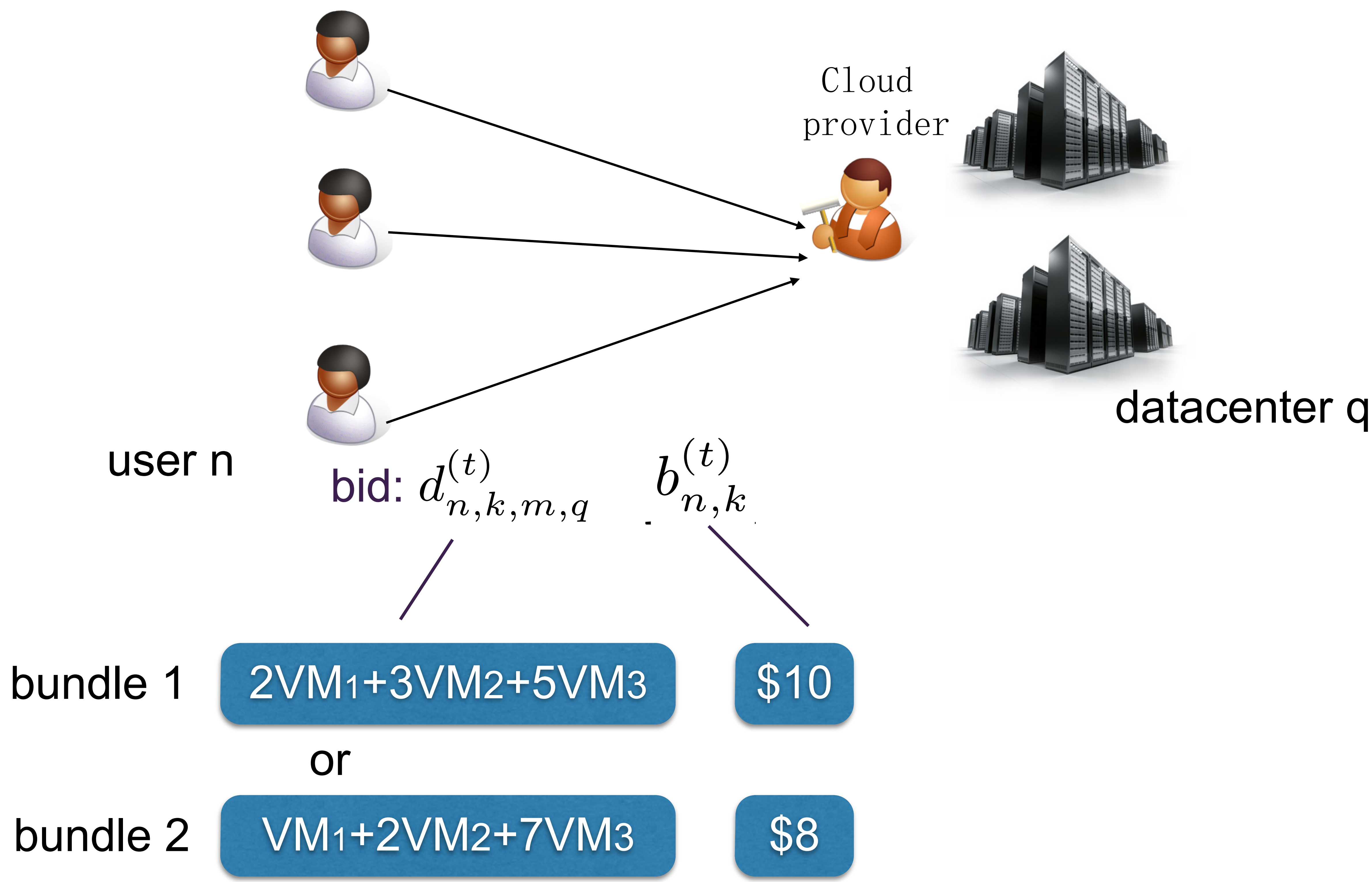
user n

bid: $d_{n,k,m,q}^{(t)}$ $b_{n,k}^{(t)}$

# of type-m VMs in datacenter q in her k-th bundle at t

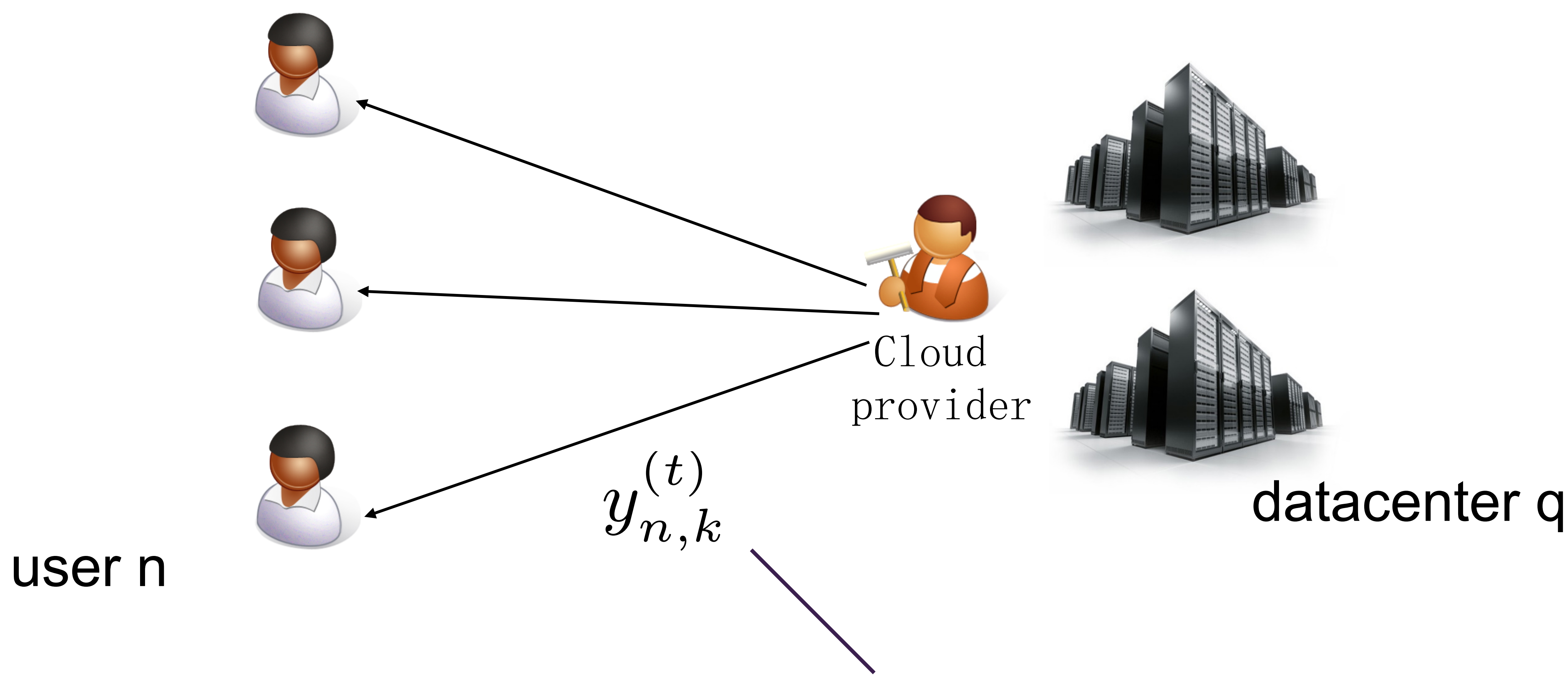
valuation of user n for her k-th bundle at t

Model



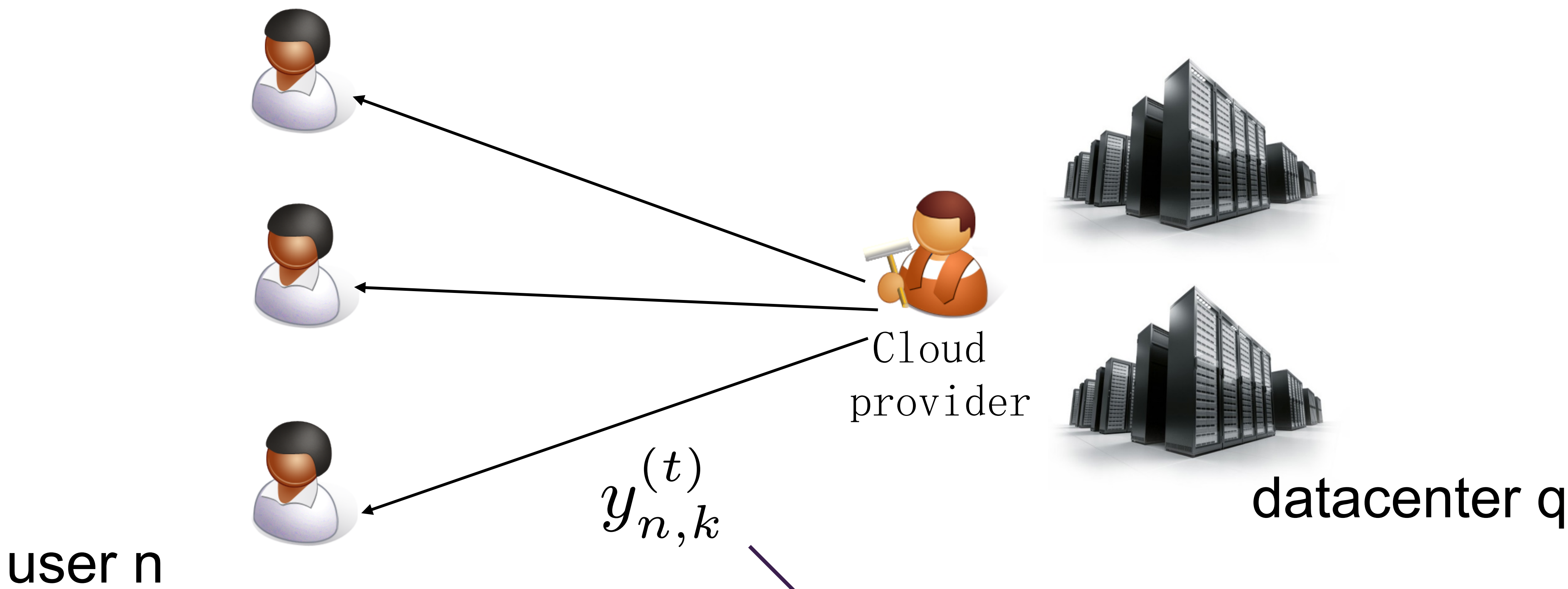
Cloud provider
datacenter q
user n
bid: $d^{(t)}_{n,k,m,q}$
$b^{(t)}_{n,k}$
bundle 1
2VM1+3VM2+5VM3
$10
or
bundle 2
VM1+2VM2+7VM3
$8

Model



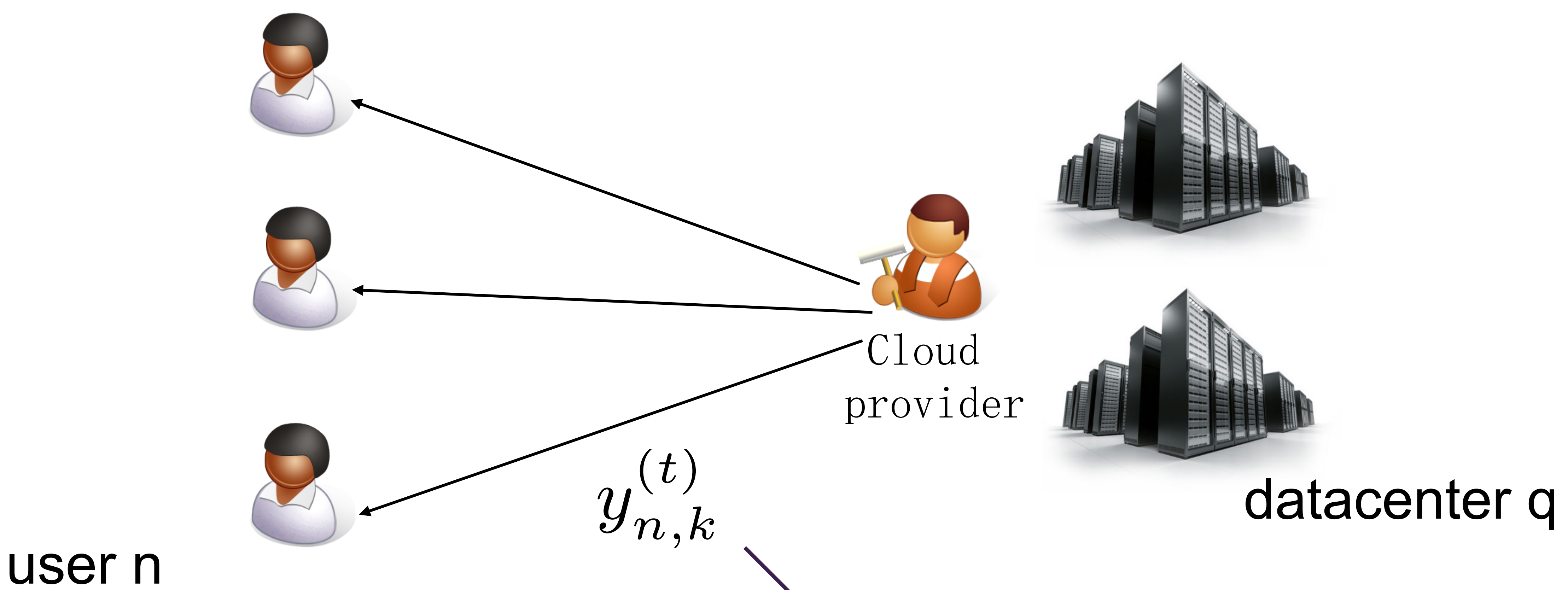
allocation decision:

Model



allocation decision: $y_{n,k}^{(t)} = 1$ user n gets her k-th bid bundle;

Model

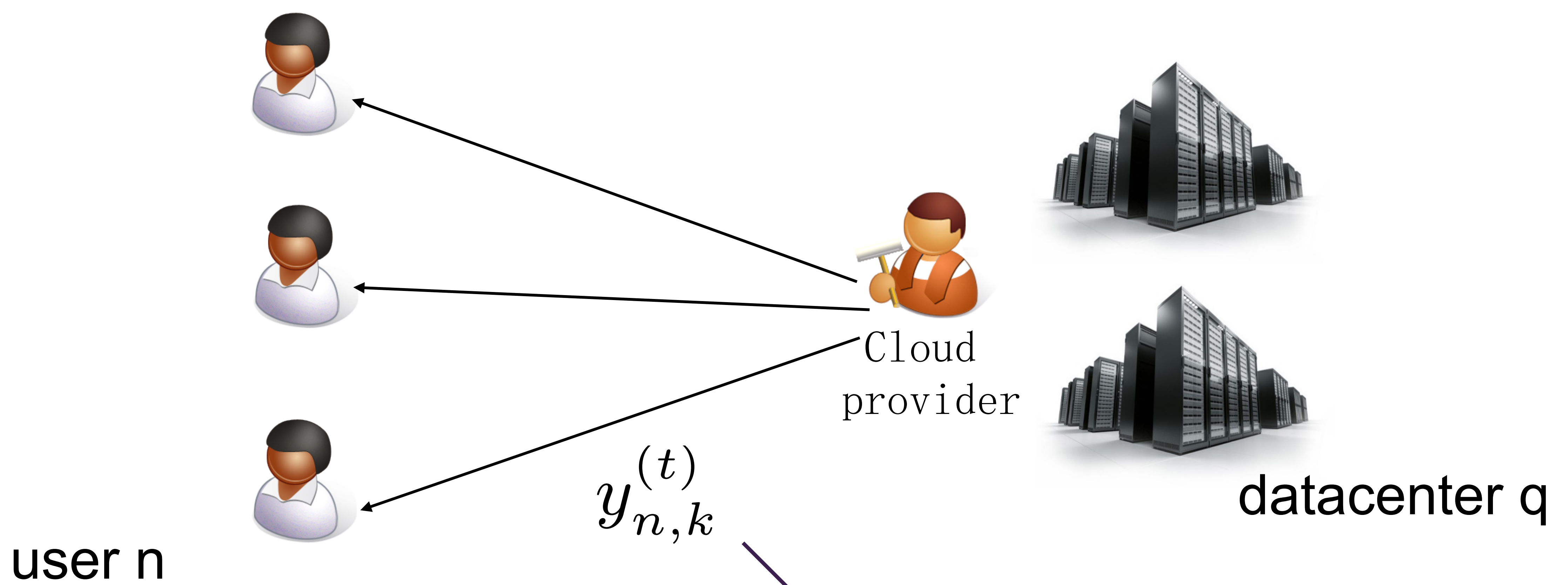


allocation decision:

$y_{n,k}^{(t)} = 1$ user n gets her k-th bid bundle;

$y_{n,k}^{(t)} = 0$ user n does not get her k-th bid bundle

Model



wins at most one bundle in each round

allocation decision: $y_{n,k}^{(t)} = 1$ user n gets her k-th bid bundle;

$y_{n,k}^{(t)} = 0$ user n does not get her k-th bid bundle

Model

user n
has an overall budget B_n

cloud provider
maximizes social welfare
(= total valuation)

Model

user n
has an overall budget B_n

cloud provider
maximizes social welfare
(= total valuation)

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

subject to

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

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

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

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

Model

user n
has an overall budget B_n

cloud provider
maximizes social welfare
(= total valuation)

$$
\begin{aligned}
\text{maximize} \quad & \sum_{t\in[T]} \sum_{n\in[N]} \sum_{k\in[K]} b_{n,k}^{(t)} y_{n,k}^{(t)} \\
\text{subject to} \quad & \\
& \sum_{k\in[K]} y_{n,k}^{(t)} \leq 1, \quad \forall n \in [N], t \in [T], \\
& \sum_{k\in[K]} \sum_{t\in[T]} b_{n,k}^{(t)} y_{n,k}^{(t)} \leq B_n, \quad \forall n \in [N], \\
& \sum_{n\in[N]} \sum_{k\in[K]} c_{n,k,r,q}^{(t)} y_{n,k}^{(t)} \leq A_{q,r}^{(t)}, \quad \forall q \in [Q], r \in [R], t \in [T], \\
& y_{n,k}^{(t)} \in \{0,1\}, \quad \forall n \in [N], k \in [K], t \in [T]
\end{aligned}
$$

Allocation decision

n's valuation
for k-th bundle

Total resource
r at dc q at t

amount of resource r at
dc q in n's k-th bundle

Online Problem

- The budget couples decisions in different rounds of the auction

Example: greedy vs. optimal allocation strategy

User A B_n=\$20

Round 1	\$6
Round 2	\$7
Round 3	\$10

User B B_n=\$20

Round 1	\$3
Round 2	\$6
Round 3	\$2

Online Problem

- The budget couples decisions in different rounds of the auction

Example: greedy vs. optimal allocation strategy

User A

Round 1 $6

Remaining Budget: $14

User B

Round 1 $3

Remaining Budget: $20

Online Problem

- The budget couples decisions in different rounds of the auction

Example: greedy vs. optimal allocation strategy

User A

Round 1 $6
Round 2 $7

Remaining
Budget: $7

User B

Round 1 $3
Round 2 $6

Remaining
Budget: $20

Online Problem

- The budget couples decisions in different rounds of the auction

Example: greedy vs. optimal allocation strategy

User A

Round 1 $6
Round 2 $7
Round 3 $10

Remaining
Budget: $7

User B

Round 1 $3
Round 2 $6
Round 3 $2

Remaining
Budget: $18

Online Problem

- The budget couples decisions in different rounds of the auction

Example: greedy vs. optimal allocation strategy

User A B_n=\$20

Round 1 \$6
Round 2 \$7
Round 3 \$10

User B B_n=\$20

Round 1 \$3
Round 2 \$6
Round 3 \$2

Greedy algorithm: social welfare \$15

Online Problem

- The budget couples decisions in different rounds of the auction

Example: greedy vs. optimal allocation strategy

User A B_n=\$20

Round 1 \$6
Round 2 \$7
Round 3 \$10

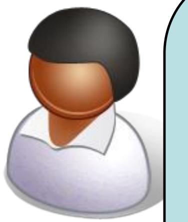
User B B_n=\$20

Round 1 \$3
Round 2 \$6
Round 3 \$2

Greedy algorithm: social welfare \$15
Optimal solution: social welfare \$22

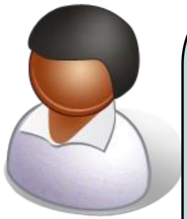
Online Problem

- Lessons learned: do NOT exhaust a user' budget early
  - may lose all the opportunities later on the user
  - but, how to seize the best opportunities to maximize social welfare over long term: classical online optimization dilemma



User A B_n=\$20

Round 1 \$6
Round 2 \$7
Round 3 \$10



User B B_n=\$20

Round 1 \$3
Round 2 \$6
Round 3 \$2

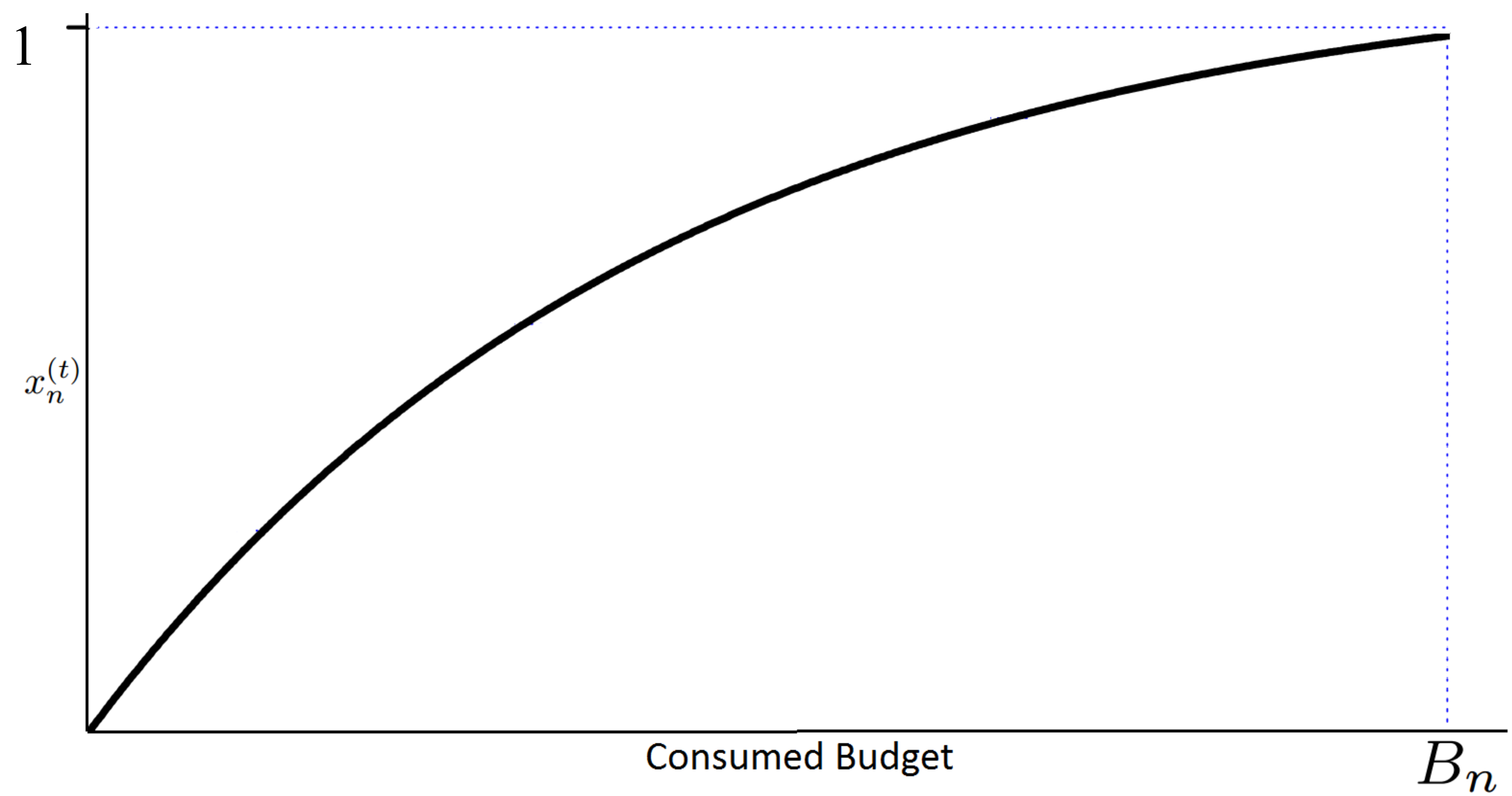
Greedy algorithm: social welfare \$15
Optimal solution: social welfare \$22

Budget Coefficient

- Higher priority for allocating resource to user with higher remaining budget

in each round:

Original valuation $\times$ Budget coefficient $(1 - x_n^{(t)})$



The Online Algorithm Framework 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

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

The Online Algorithm Framework A_{online}

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

// Loop for each time slot

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

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

adjusted bundle valuation used for resource allocation in this round

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

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

The Online Algorithm Framework A_{online}

$x_n^{(0)} \leftarrow 0, \forall n \in [N]$
// Loop for each time slot
for all $1 \leq t \leq T$ **do**

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

adjusted bundle valuation used for resource allocation in this round

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**

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

Update budget coefficient based on the ratio of consumed budget to the total budget

end for
for all $n \notin \mathcal{N}$ **do**
$x_n^{(t)} \leftarrow x_n^{(t-1)}$
end for
end for
$x_n \leftarrow x_n^{(T)}, \forall n \in [N]$

The Online Algorithm Framework A_{online}

An example run:

only one item; A_{round} simply chooses the user with the larger adjusted valuation as the winner

User A B_n=\$20

Round 1 \$6
Round 2 \$7
Round 3 \$10

User B B_n=\$20

Round 1 \$3
Round 2 \$6
Round 3 \$2

The Online Algorithm Framework A_{online}

An example run:

only one item; A_{round} simply chooses the user with the larger adjusted valuation as the winner

User A B_n=\$20 x_n=0

Round 1 \$6

Adjusted:

\$6*(1-0)=**\$6**

Update: x_n=0.24

User B B_n=\$20 x_n=0

Round 1 \$3

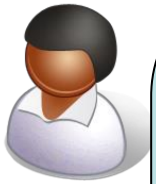
Adjusted:

\$3*(1-0)=**\$3**

The Online Algorithm Framework A_{online}

An example run:

only one item; A_{round} simply chooses the user with the larger adjusted valuation as the winner



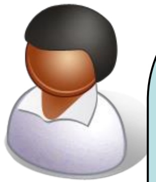
User A B_n=\$20 x_n=0.24

Round 1 \$6

Round 2 \$7

Adjusted:

\$7*(1-0.24)=**\$5.32**



User B B_n=\$20 x_n=0

Round 1 \$3

Round 2 \$6

Adjusted:

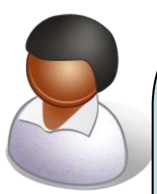
\$6*(1-0)=**\$6**

Update: x_n=0.24

The Online Algorithm Framework A_{online}

An example run:

only one item; A_{round} simply chooses the user with the larger adjusted valuation as the winner



User A B_n=\$20 x_n=0.24

Round 1 \$6
Round 2 \$7
Round 3 \$10
Adjusted:
\$10*(1-0.24)=**\$7.6**

Update: x_n=0.76

User B B_n=\$20 x_n=0.24

Round 1 \$3
Round 2 \$6
Round 3 \$2
Adjusted:
\$2*(1-0.24)=**\$1.52**

Greedy algorithm: social welfare \$15
Optimal solution: social welfare \$22
Online algorithm: social welfare \$22

One-Round Auction Around

- To decide the winners and winning bundles (resource allocation decisions), and the price to charge for each bundle (payment mechanism)

One-Round Auction Around

- To decide the winners and winning bundles (resource allocation decisions), and the price to charge for each bundle (payment mechanism)

- Main design objective: truthfulness

One-Round Auction Around

- To decide the winners and winning bundles (resource allocation decisions), and the price to charge for each bundle (payment mechanism)

- Main design objective: truthfulness

- Payment mechanism is the key to guarantee truthfulness
  - can be very difficult to design

One-Round Auction Around

- To decide the winners and winning bundles (resource allocation decisions), and the price to charge for each bundle (payment mechanism)

- Main design objective: truthfulness

- Payment mechanism is the key to guarantee truthfulness
  - can be very difficult to design
  - VCG auction is a truthful mechanism

    charges bidder the opportunity cost

    needs to compute the exact optimal allocation (cannot be approximate solution)

One-Round Resource Allocation Problem

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

subject to

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

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

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

An NP hard problem!

$w_{n,k}^{(t)}$: adjusted user n's valuation for k-th bundle

$c_{n,k,r,q}^{(t)}$: amount of resource r at dc q in n's k-th bundle

$A_{q,r}^{(t)}$: total resource r at dc q at t

$y_{n,k}^{(t)}$: decision variable, bundle allocated or not

Fractional VCG

- Relax $y_{n,k}^{(t)}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism

Fractional VCG

- Relax $y_{n,k}^{(t)}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism
- But, fractional bundle allocation is not feasible in practice

  we cannot provide 0.3 of a VM instance to a user

Fractional VCG

- Relax $y_{n,k}^{(t)}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism
- But, fractional bundle allocation is not feasible in practice

  we cannot provide 0.3 of a VM instance to a user
- Solution: decompose the fractional solution to a combination of integer solutions, such that the allocation in expectation remains the same

Fractional VCG

- Relax $y_{n,k}^{(t)}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism
- But, fractional bundle allocation is not feasible in practice

  we cannot provide 0.3 of a VM instance to a user
- Solution: decompose the fractional solution to a combination of integer solutions, such that the allocation in expectation remains the same

$$\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{l \in \boldsymbol{L}} \beta_l y_{n,k}^{(t)l} = y_{n,k}^{(t)F} / \lambda, && \forall n \in [N], k \in [K], \\
& \sum_{l \in \boldsymbol{L}} \beta_l \geq 1, && \\
& \beta_l \geq 0, && \forall l \in \boldsymbol{L}.
\end{aligned}$$

Fractional VCG

- Relax $y_{n,k}^{(t)}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism
- But, fractional bundle allocation is not feasible in practice

  we cannot provide 0.3 of a VM instance to a user
- Solution: decompose the fractional solution to a combination of integer solutions, such that the allocation in expectation remains the same

$$\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l$$

probability to choose the respective integer solution

$$\begin{aligned}
\text{s.t.} \quad & \sum_{l \in \boldsymbol{L}} \beta_l y_{n,k}^{(t)l} = y_{n,k}^{(t)F} / \lambda, && \forall n \in [N], k \in [K], \\
& \sum_{l \in \boldsymbol{L}} \beta_l \geq 1, && \\
& \beta_l \geq 0, && \forall l \in \boldsymbol{L}.
\end{aligned}$$

Fractional VCG

- Relax $y_{n,k}^{(t)}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism
- But, fractional bundle allocation is not feasible in practice

  we cannot provide 0.3 of a VM instance to a user
- Solution: decompose the fractional solution to a combination of integer solutions, such that the allocation in expectation remains the same

$$\begin{aligned} \text{minimize} \quad & \sum_{l \in \boldsymbol{L}} \beta_l \\ \text{s.t.} \quad & \sum_{l \in \boldsymbol{L}} \beta_l y_{n,k}^{(t)l} = y_{n,k}^{(t)F} / \lambda, \qquad \forall n \in [N], k \in [K], \\ & \sum_{l \in \boldsymbol{L}} \beta_l \geq 1, \\ & \beta_l \geq 0, \qquad \forall l \in \boldsymbol{L}. \end{aligned}$$

probability to choose the respective integer solution

Fractional VCG

- Relax $y_{n,k}^{(t)}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism
- But, fractional bundle allocation is not feasible in practice

  we cannot provide 0.3 of a VM instance to a user
- Solution: decompose the fractional solution to a combination of integer solutions, such that the allocation in expectation remains the same

$$\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l$$

probability to choose the respective integer solution

$$\text{s.t.} \quad \sum_{l \in \boldsymbol{L}} \beta_l y_{n,k}^{(t)l} = y_{n,k}^{(t)F} / \lambda, \qquad \forall n \in [N], k \in [K],$$

$$\sum_{l \in \boldsymbol{L}} \beta_l \geq 1,$$

$$\beta_l \geq 0, \qquad \forall l \in \boldsymbol{L}.$$

Fractional VCG

- Relax $y^{(t)}_{n,k}$
- Compute optimal fractional allocation: an LP
- Use the same VCG payment mechanism
- But, fractional bundle allocation is not feasible in practice

  we cannot provide 0.3 of a VM instance to a user

- Solution: decompose the fractional solution to a combination of integer solutions, such that the allocation in expectation remains the same

$$\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l$$

$$\text{s.t.} \quad \sum_{l \in \boldsymbol{L}} \beta_l y^{(t)l}_{n,k} = y^{(t)F}_{n,k} / \lambda, \qquad \forall n \in [N], k \in [K],$$

$$\sum_{l \in \boldsymbol{L}} \beta_l \geq 1,$$

$$\beta_l \geq 0, \qquad \forall l \in \boldsymbol{L}.$$

probability to choose the respective integer solution

scale-down factor

Randomized Decomposition

	User A	User B	User C	
Fractional solution:	0.3	0.8	0.5	$y^{(t)F}_{n,k}$
Decomposed Integer solution	1	1	0	Pr = 0.3
	0	1	1	Pr = 0.5
	0	0	0	Pr = 0.2

$y^{(t)l}_{n,k}$ β_l

$$\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l$$

s.t.

$$\sum_{l \in \boldsymbol{L}} \beta_l y^{(t)l}_{n,k} = y^{(t)F}_{n,k} / \lambda, \qquad \forall n \in [N], k \in [K],$$

$$\sum_{l \in \boldsymbol{L}} \beta_l \geq 1,$$

$$\beta_l \geq 0, \qquad \forall l \in \boldsymbol{L}.$$

Randomized Decomposition

- How to decide the scale-down factor λ
- How to find a set of integer solution $y_{n,k}^{(t)l}$ and the corresponding probabilities β_l

$$
\begin{aligned}
&\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l \\
\text{s.t.}\quad & \sum_{l \in \boldsymbol{L}} \beta_l y_{n,k}^{(t)l} = y_{n,k}^{(t)F} / \lambda, \qquad \forall n \in [N], k \in [K], \\
& \sum_{l \in \boldsymbol{L}} \beta_l \geq 1, \\
& \beta_l \geq 0, \qquad \forall l \in \boldsymbol{L}.
\end{aligned}
$$

Finding λ

- Use the approximation ratio of an algorithm to solve the one-round allocation problem as the scaling-down factor

  guaranteed to find a feasible solution of the decomposition problem

Finding λ

- A primal-dual approximation algorithm to solve the one-round allocation problem

1: $\mathcal{N} \leftarrow \emptyset$, $z_{base} \leftarrow QR \cdot exp((C_{min}^{(t)} - 1))$
2: $y_{n,k}^{(t)} \leftarrow 0$, $s_n^{(t)} \leftarrow 0$, $z_{q,r}^{(t)} \leftarrow 1/A_{q,r}^{(t)}, \forall n \in [N], k \in [K], r \in [R], q \in [Q]$
3: **while** $\sum_{r\in[R]} \sum_{q\in[Q]} A_{q,r}^{(t)} z_{q,r}^{(t)} < z_{base}$ AND $|\mathcal{N}| \neq N$ **do**
4: **for all** $n \notin \mathcal{N}$ **do**
5: $k(n) = \arg\max_{k\in[K]}\{w_{n,k}^{(t)}\}$
6: **end for**
7: $n^* = \arg\max_{n\in[N]}\{\frac{w_{n,k(n)}^{(t)}}{\sum_{r\in[R]}\sum_{q\in[Q]} c_{n,k(n),r,q}^{(t)} z_{q,r}^{(t)}}\}$
8: $y_{n^*,k(n^*)}^{(t)} \leftarrow 1$, $s_{n^*}^{(t)} \leftarrow w_{n^*,k(n^*)}^{(t)}$, $\mathcal{N} \leftarrow \mathcal{N} \cup \{n^*\}$
9: **for all** $r \in [R], q \in [Q]$ **do**

$$z_{q,r}^{(t)} \leftarrow z_{q,r}^{(t)} \cdot {z_{base}}^{c_{n^*,k(n^*),q,r}^{(t)}/(A_{q,r}^{(t)} - C_{q,r}^{(t)})}$$

10: **end for**
11: **end while**

Dual variable of the resource constraint: the unit price of each type of resources

Evaluate a bundle according to unit prices and required resources; choose users with a higher bid on a lower-valued bundle as the winner

Update the unit price of recourses: higher price if larger amount of the resource consumed

Finding integer solutions and probabilities

- Difficult to solve directly since we need to find the exponentially many feasible integer solutions first

$$\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{l \in \boldsymbol{L}} \beta_l y_{n,k}^{(t)l} = y_{n,k}^{(t)F} / \lambda, \qquad && \forall n \in [N], k \in [K], \\
& \sum_{l \in \boldsymbol{L}} \beta_l \geq 1, && \\
& \beta_l \geq 0, && \forall l \in \boldsymbol{L}.
\end{aligned}$$

Finding integer solutions and probabilities

- Difficult to solve directly since we need to find the exponentially many feasible integer solutions first

- Solution: resort to the dual

  solve the dual using Ellipsoid method in polynomial time, using the primal-dual approximation algorithm as the separation oracle

$$\text{minimize} \quad \sum_{l \in \boldsymbol{L}} \beta_l$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{l \in \boldsymbol{L}} \beta_l y_{n,k}^{(t)l} = y_{n,k}^{(t)F} / \lambda, && \forall n \in [N], k \in [K], \\
& \sum_{l \in \boldsymbol{L}} \beta_l \geq 1, && \\
& \beta_l \geq 0, && \forall l \in \boldsymbol{L}.
\end{aligned}$$

One-Round Random Auction Around

1: Solve LP relaxation of (3), with $w_{n,k}^{(t)} = \max\{0, (1 - x_n^{(t-1)})b_{n,k}^{(t)}\}$. Denote the fractional solution by $y_{n,k}^{(t)F}, \forall n \in [N], k \in [K]$.

2: **for all** $n \in [N]$ **do**

3: $\quad \forall n' \in [N], k \in [K], w_{n',k}^{'(t)} = max\{0, (1 - x_n^{(t-1)})b_{n,k}^{(t)}\}$, if $n' \neq n$. Otherwise $w_{n',k}^{'(t)} = 0$.

4: $\quad$ Solve LP relaxation of (3), with $w_{n',k}^{'(t)}$'s. Denote the optimal objective function value by $\widetilde{V}_{-n}^{(t)}$.

5: $\quad \Pi_n^{(t)F} = \widetilde{V}_{-n}^{(t)} - \sum_{n' \neq n} \sum_k y_{n',k}^{(t)F} w_{n',k}^{(t)}$

6: **end for**

7: Solve the pair of primal-dual decomposition LPs in (6) and (7) using the ellipsoid method, using Alg. 2 as a separation oracle, and derive a polynomial number of integer solutions to (3), $\mathbf{y}^{(t)l}$, $\forall l \in \boldsymbol{L}$, and the corresponding decomposition coefficients, β_l, $\forall l \in \boldsymbol{L}$.

8: Choose $\mathbf{y}^{(t)l}$ with probability β_l, $\forall l \in \boldsymbol{L}$

9: $\forall n \in [N], \Pi_n^{(t)l} = \Pi_n^{(t)F} \cdot \frac{\sum_{k \in [K]} w_{n,k}^{(t)} y_{n,k}^{(t)l}}{\sum_{k \in [K]} w_{n,k}^{(t)} y_{n,k}^{(t)F}}$

One-Round Random Auction Around

1: Solve LP relaxation of (3), with $w_{n,k}^{(t)} = \max\{0, (1 - x_n^{(t-1)})b_{n,k}^{(t)}\}$. Denote the fractional solution by $y_{n,k}^{(t)F}, \forall n \in [N], k \in [K]$.

2: **for all** $n \in [N]$ **do**

3: $\forall n' \in [N], k \in [K], w_{n',k}^{'(t)} = max\{0, (1 - x_n^{(t-1)})b_{n,k}^{(t)}\}$, if $n' \neq n$. Otherwise $w_{n',k}^{'(t)} = 0$.

4: Solve LP relaxation of (3), with $w_{n',k}^{'(t)}$'s. Denote the optimal objective function value by $\widetilde{V}_{-n}^{(t)}$.

5: $\Pi_n^{(t)F} = \widetilde{V}_{-n}^{(t)} - \sum_{n' \neq n} \sum_k y_{n',k}^{(t)F} w_{n',k}^{(t)}$

6: **end for**

7: Solve the pair of primal-dual decomposition LPs in (6) and (7) using the ellipsoid method, using Alg. 2 as a separation oracle, and derive a polynomial number of integer solutions to (3), $\mathbf{y}^{(t)l}$, $\forall l \in \boldsymbol{L}$, and the corresponding decomposition coefficients, β_l, $\forall l \in \boldsymbol{L}$.

8: Choose $\mathbf{y}^{(t)l}$ with probability β_l, $\forall l \in \boldsymbol{L}$

9: $\forall n \in [N], \Pi_n^{(t)l} = \Pi_n^{(t)F} \cdot \frac{\sum_{k \in [K]} w_{n,k}^{(t)} y_{n,k}^{(t)l}}{\sum_{k \in [K]} w_{n,k}^{(t)} y_{n,k}^{(t)F}}$

Allocate VM bundles

One-Round Random Auction Around

1: Solve LP relaxation of (3), with $w_{n,k}^{(t)} = \max\{0, (1 - x_n^{(t-1)})b_{n,k}^{(t)}\}$. Denote the fractional solution by $y_{n,k}^{(t)F}, \forall n \in [N], k \in [K]$.
2: **for all** $n \in [N]$ **do**
3: $\forall n' \in [N], k \in [K], w_{n',k}^{'(t)} = max\{0, (1 - x_n^{(t-1)})b_{n,k}^{(t)}\}$, if $n' \neq n$. Otherwise $w_{n',k}^{'(t)} = 0$.
4: Solve LP relaxation of (3), with $w_{n',k}^{'(t)}$'s. Denote the optimal objective function value by $\widetilde{V}_{-n}^{(t)}$.
5: $\Pi_n^{(t)F} = \widetilde{V}_{-n}^{(t)} - \sum_{n' \neq n} \sum_k y_{n',k}^{(t)F} w_{n',k}^{(t)}$
6: **end for**
7: Solve the pair of primal-dual decomposition LPs in (6) and (7) using the ellipsoid method, using Alg. 2 as a separation oracle, and derive a polynomial number of integer solutions to (3), $\mathbf{y}^{(t)l}$, $\forall l \in \boldsymbol{L}$, and the corresponding decomposition coefficients, β_l, $\forall l \in \boldsymbol{L}$.
8: Choose $\mathbf{y}^{(t)l}$ with probability β_l, $\forall l \in \boldsymbol{L}$
9: $\forall n \in [N], \Pi_n^{(t)l} = \Pi_n^{(t)F} \cdot \frac{\sum_{k \in [K]} w_{n,k}^{(t)} y_{n,k}^{(t)l}}{\sum_{k \in [K]} w_{n,k}^{(t)} y_{n,k}^{(t)F}}$

Allocate VM bundles

Decide payment

What we have done

- A e-approximation algorithm for one-round VM allocaiton
- A_{round}: one-round combinatorial VM auction, truthful, e-approximate social welfare
- A_{online}: online combinatorial VM auction, truthful, $(e + \frac{1}{e-1})$-competitive

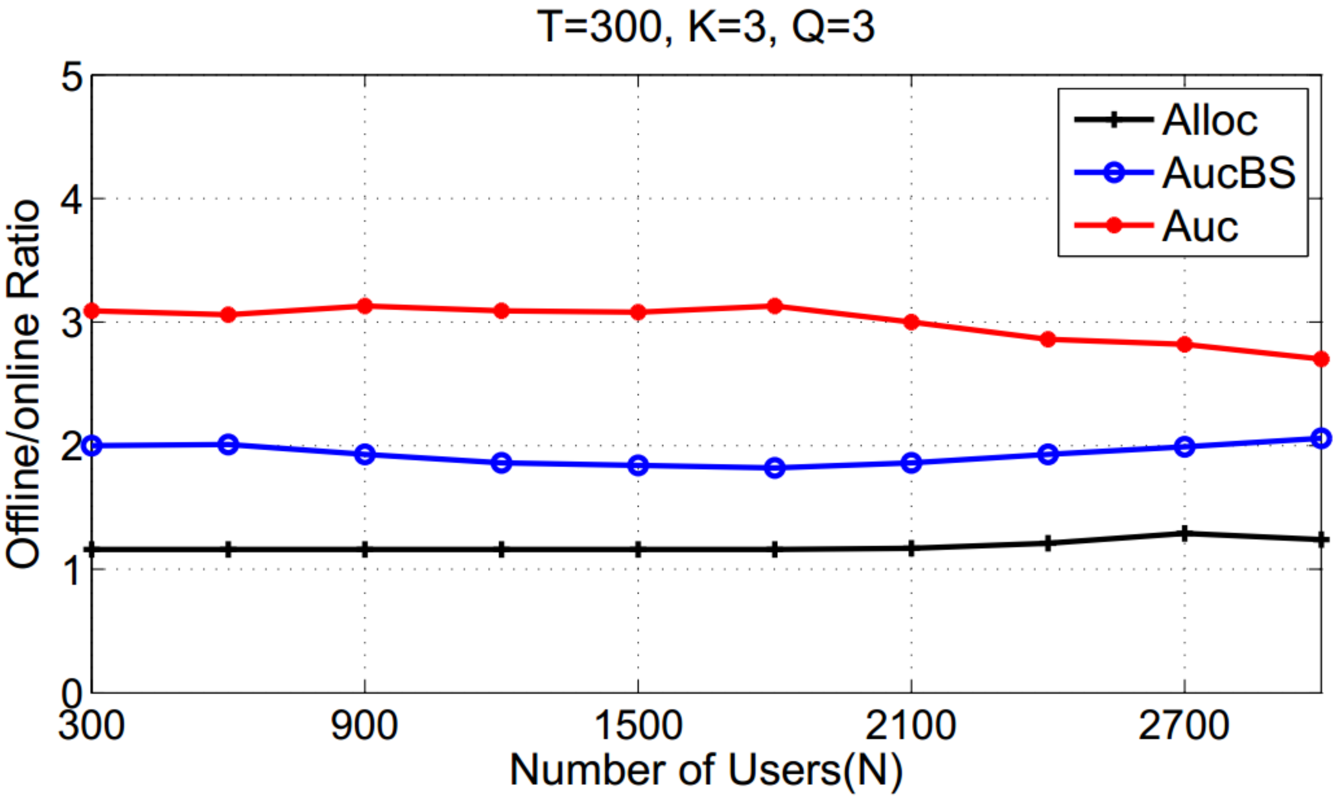
Trace-deriven evaluation

Setup: Google cluster trace

- 6 types of VMs, 3 types of resources
- 3 datacenters
- 3 bundles per user
- 300 ~ 3000 users
- 300 ~ 3000 rounds

Comparison among:

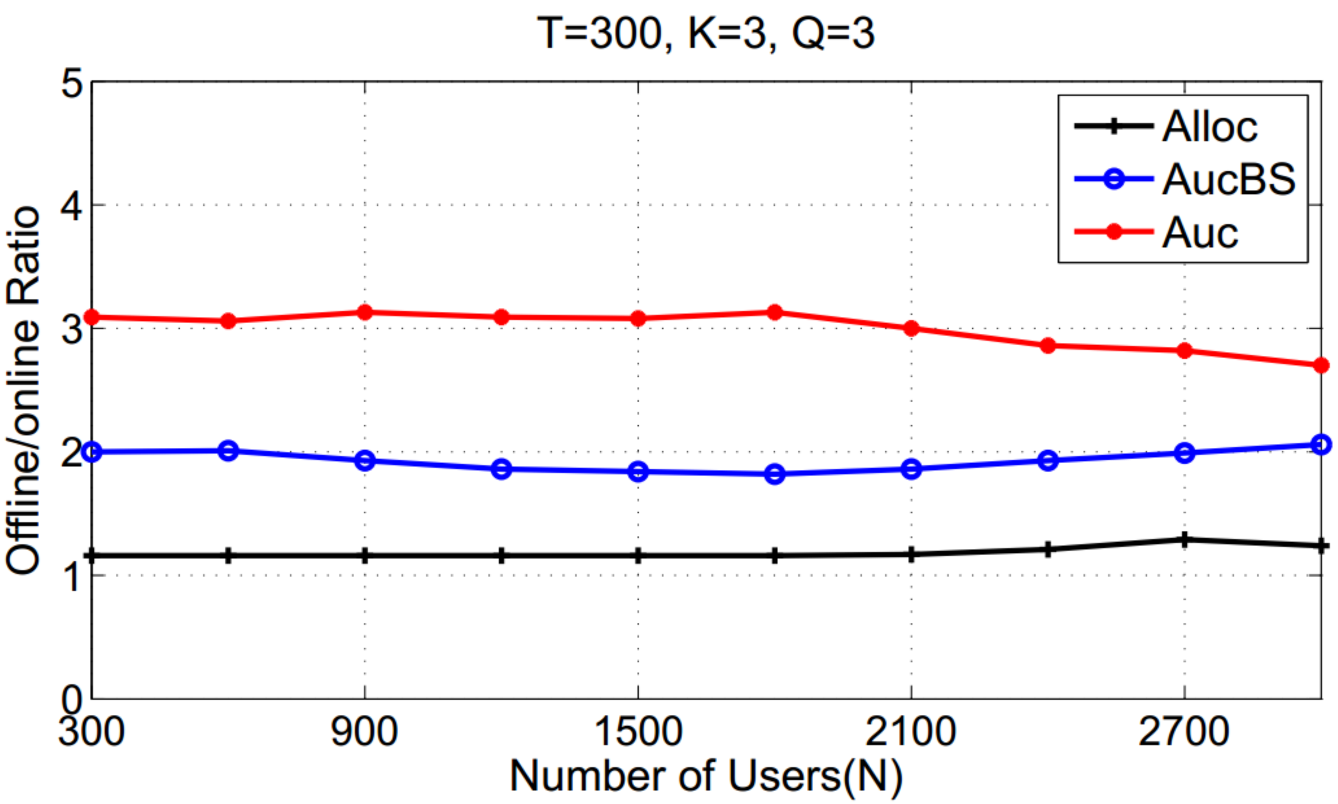
- Alloc: online allocation algorithm
- Auc: online auction with original decomposition method
- AucBS: online auction with binary search improvement

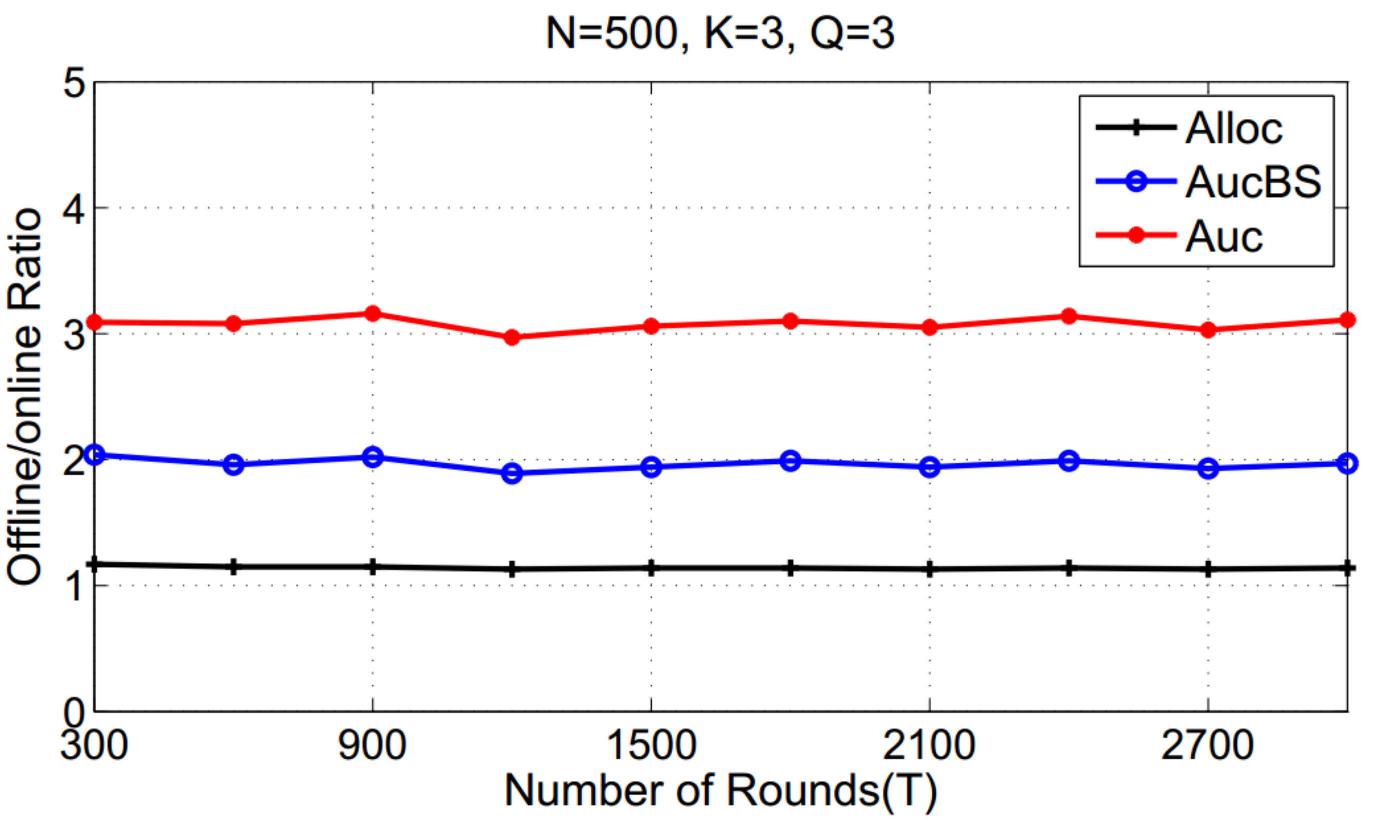


Alloc: online allocation algorithm

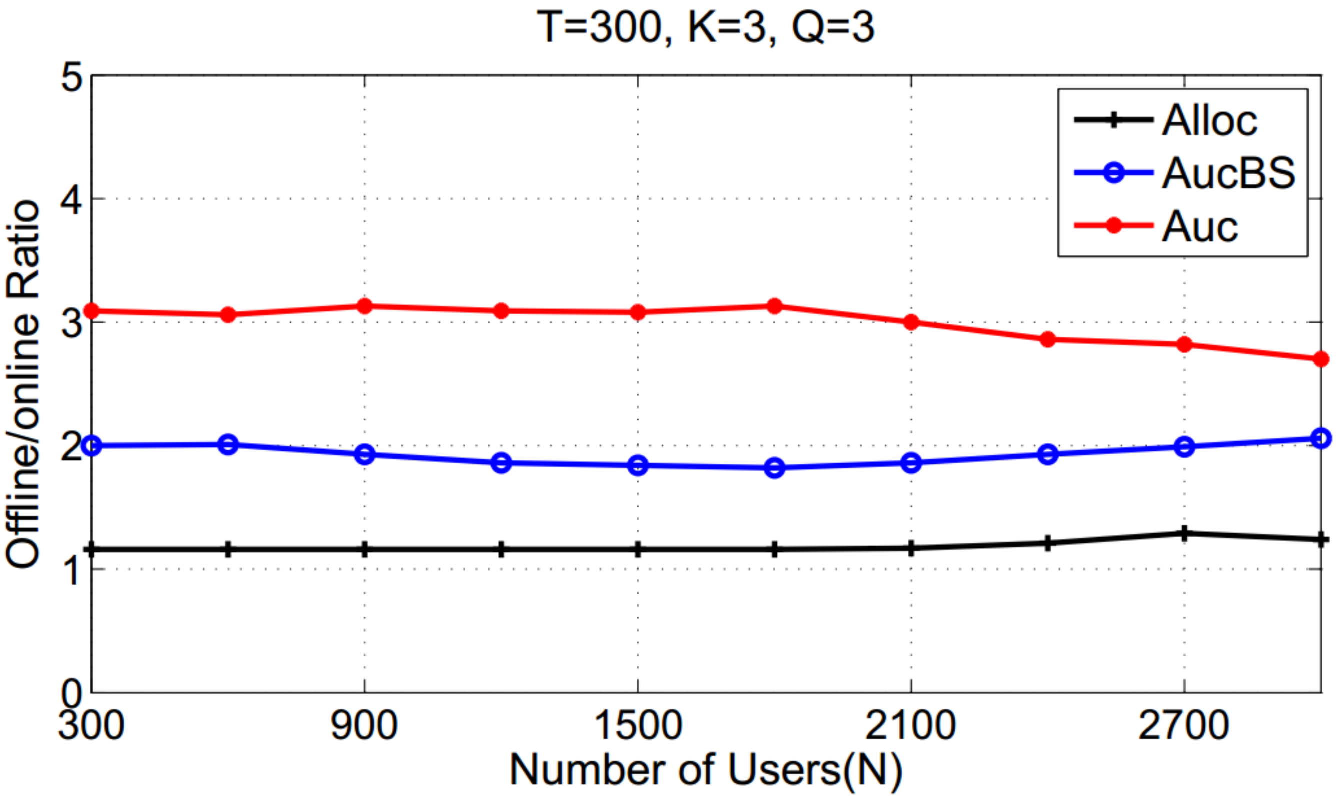
Auc: online auction with original decomposition method

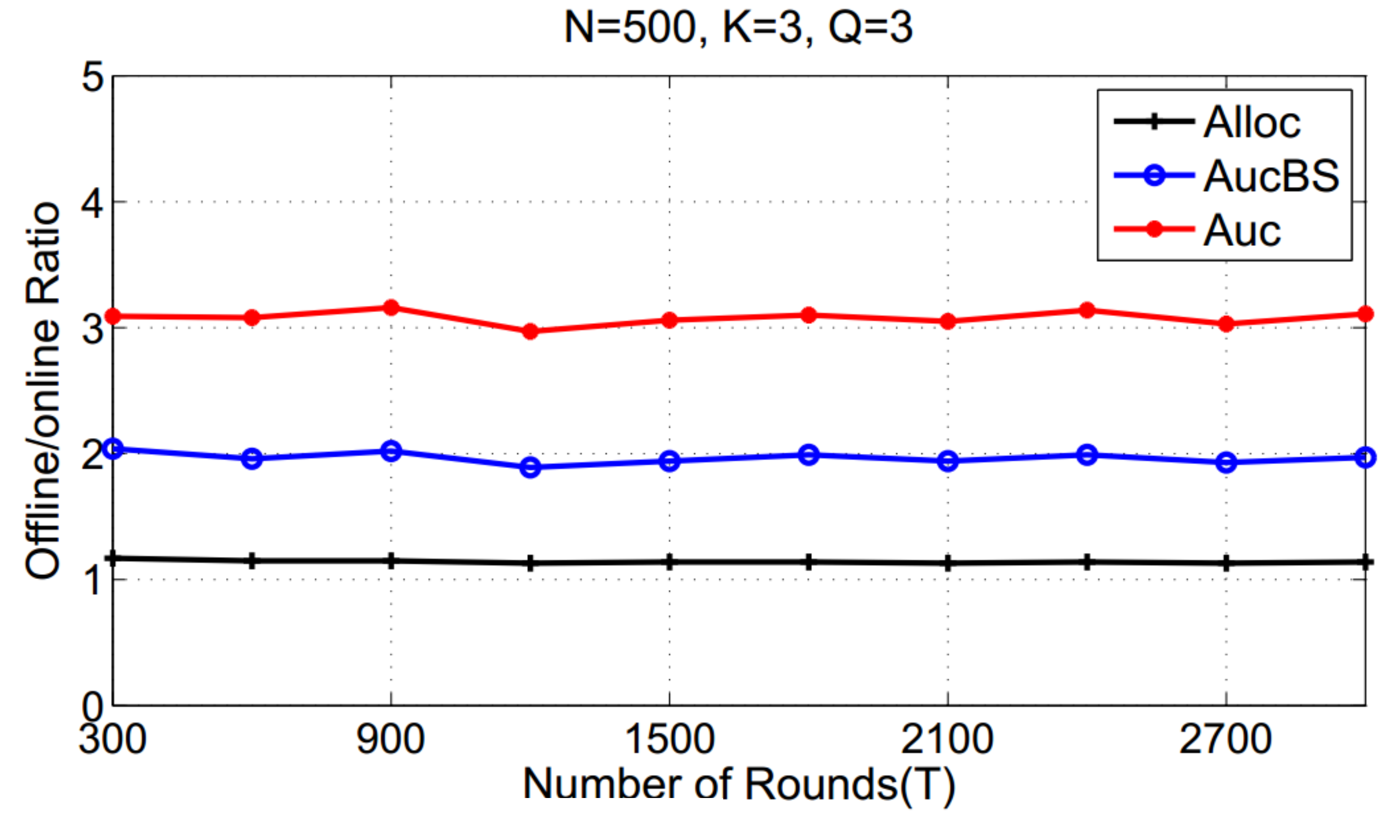
AucBS: online auction with binary search improvement

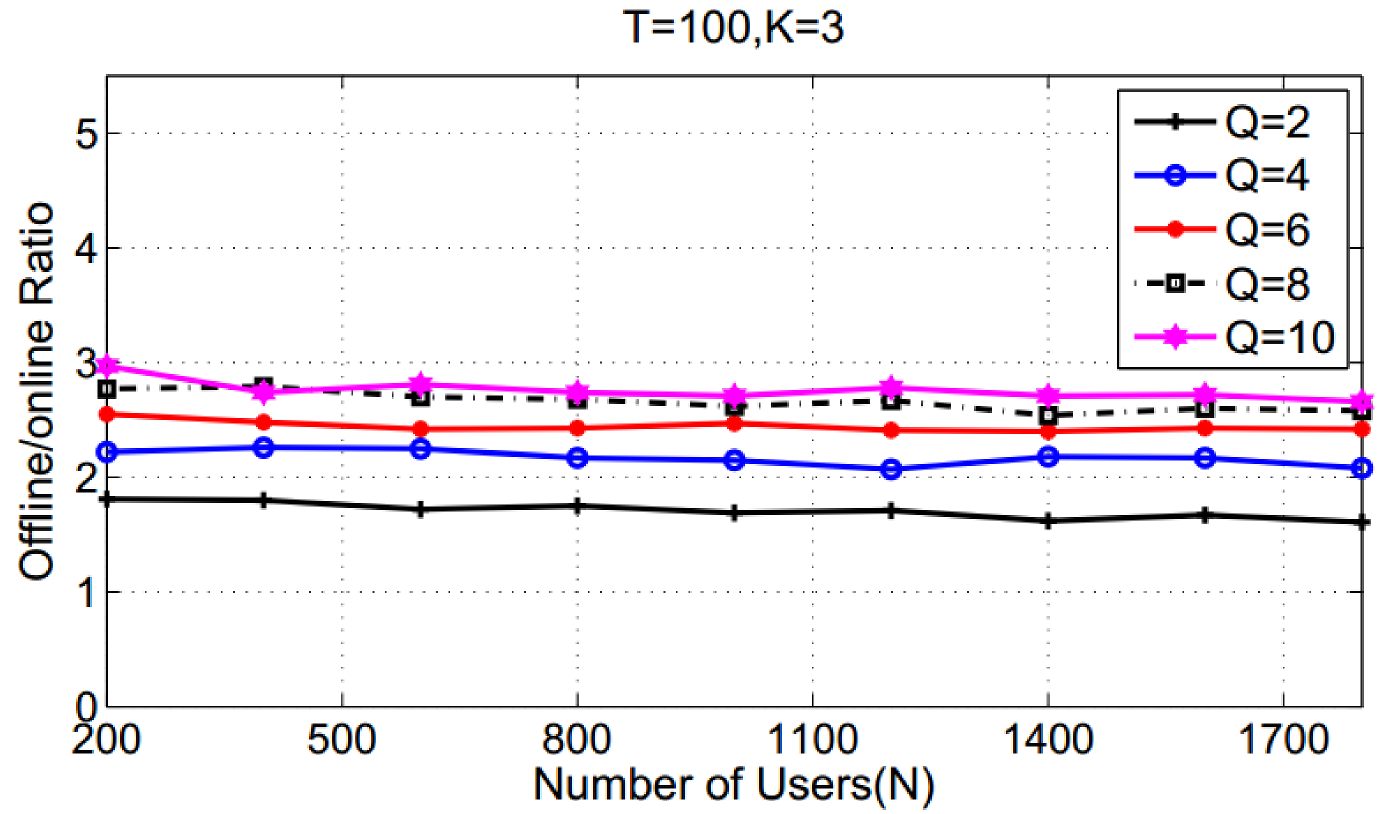




Alloc: online allocation algorithm
Auc: online auction with original decomposition method
AucBS: online auction with binary search improvement







Alloc: online allocation algorithm

Auc: online auction with original decomposition method

AucBS: online auction with binary search improvement

Conclusion

- The first online combination auction for dynamic VM market
  - translates the online social welfare maximization problem into a series of one-round resource allocation problems
  - translates a cooperative approximation algorithm into a truthful auction
  - a theoretical competitive ratio ≈ 3.30 in typical scenarios
- Promising to apply this algorithmic framework in other related settings

Weijie Shi, Linquan Zhang, Chuan Wu, Zongpeng Li, Francis C.M. Lau. "An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing," to appear in the Proceedings of ACM SIGMETRICS 2014, Austin, Texas, USA, June 16-20, 2014.