Washington University in St. Louis Engineering

Pricing (and Bidding) Strategies for Delay Differentiated Cloud Services[*]

Roch Guérin
Washington University in St. Louis,
December 2019

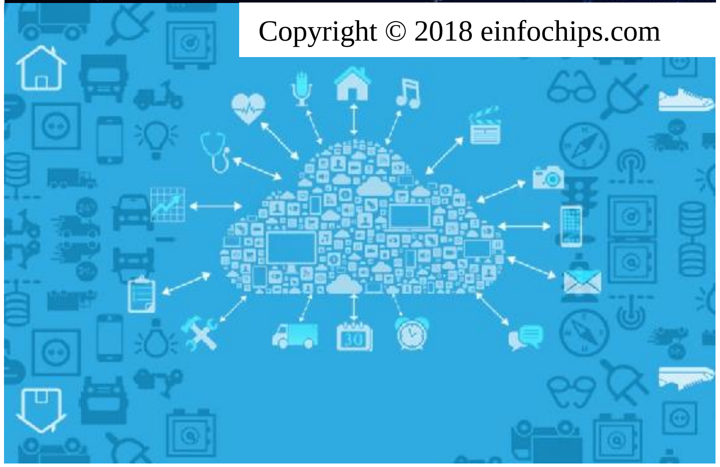
[*] *Joint work with Jiayi Song*

The Rise of the Cloud

- A $100B market in 2016 on target to hit $200B in 2020
- Growth in both volume and diversity
  - Cloud users span a growing range of industry and applications
- Cloud providers have responded with a fast expanding set of offerings
  - IaaS, PaaS, SaaS, etc.
  - Reserved, on-demand, spot instances, preemptible VM, serverless computing, etc.

that offer different trade-offs when it comes to resource and their availability

⇒ **Pricing as a major control knob**



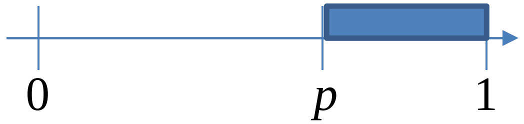
https://aws.amazon.com/pricing/

Our Focus

- The role of pricing in
  - Improving provider’s ability to extract value
    - When and why is it useful to offer differentiated tiers of service?
    - Optimal pricing strategies (maximum revenue)
  - Matching users/jobs to services that best meet their needs (jobs are heterogeneous in what they are worth and the performance they require)
    - We focus on performance = timeliness of access to resources (and consequently job completion time)
    - What is the best service and what to pay for it?

Optimal Pricing (Backup)

- Assume
  - for simplicity a digital product with zero incremental cost
  - A user population with a "willingness to pay" (for the product) uniformly distributed between 0 and 1
- How should we price the product to maximize revenue?
  - Revenue is ~ $(1 - p) \times p$
  - Maximized for $p = 1/2$



Some Motivating Examples

- Service offerings from major cloud providers
  - Amazon: On-demand and spot instances
    - Spot instances offered at a discount but variable pricing and possibility of preemption when price exceeds bid
    - Alibaba and Packet.com offer similar services
  - Google and Microsoft: Preemptible instances
    - Fixed price but possibility of preemption when resources are needed

  In both cases

  Lower price ⇒ Longer (expected) execution time

- A similar trade-off exists in other scenarios
  - Slower but cheaper vs. faster but more expensive processor/instance

Sneak Preview of Main Results

- Correlation between job valuation and sensitivity to delay needs to exceed a certain threshold for delay differentiated services to improve provider's revenue
  - Basically you need enough jobs willing to pay a high price for fast service, and at the same time also enough jobs that are relatively insensitive to delay but unable to afford an expensive service
- In the presence of variable prices (spot instances) a fixed bidding strategy is often optimal or near-optimal for users

Our Focus

- A semi-monopolistic cloud provider like AWS
  - We ignore the impact of competion
- A range of services, but in particular services that trade-off price for timeliness of execution
  - On-demand vs. spot instances or preemptible instances (more on this in a moment)
- Questions we seek to answer
  - When does having both services help the provider improve revenue?
    - Should we offer two services, and if yes, how should prices be set?
  - What are effective bidding strategies for users?
    - Generate highest "utility" when prices vary (spot service)?

Provider Side

- ***Monopoly***, *no* competition
- ***Unconstrained*** cloud resources (capacity is not a constraint)
  - A reasonable assumption for large cloud providers and supported by recent empirical work*
- ***Two scenarios:*** Variable and fixed prices (spot & preemptible instances)
  - We'll focus on the former as they offer a more general framework
- ***Variable prices are not responsive*** to demand
  - Consistent with empirical findings and recent evolution of Amazon's own spot pricing*
  - Known price distribution (Amazon, Alibaba make historical spot prices available)

⇒Assume ***random*** price variations (drawn from a given distribution)

* [1] O. A. Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, *"Deconstructing Amazon EC2 spot instance pricing,"* ACM Trans. Econ. Comput., vol. 1, no. 3, September 2013.

[2] H. Xu and B. Li. 2013. *"Dynamic Cloud Pricing for Revenue Maximization."* IEEE Trans. Cloud Comp. Vol. 1, No. 2, July 2013.

[3] https://aws.amazon.com/blogs/aws/amazon-ec2-update-streamlined-access-to-spot-capacity-smooth-price-changes-instance-hibernation/

Spot Service Behavior

- Spot price is periodically updated (new value drawn from advertised distribution)
  - Customers register bids ahead of each period
  - Jobs run (stop) whenever their bid exceeds (falls below) the spot price
  - Jobs are charged spot price (not bid) whenever they run
- **Goal**: pick prices and price distribution to maximize expected revenue
  - Note: If answer is to use a *single* price, then spot = on-demand (no need for differentiation)
- Similar behavior when job interruption is caused by exogenous preemption
  - But, no control through price selection over probability of interruption

Customer Side

- Heterogeneous job requirements:
  - Job value per unit of execution time (v)
  - Job timeliness / sensitivity to delay (κ)
  - Job execution time (t)

Job profiles (v,κ,t) are private information, but distribution is known to the service provider

- Customer Decisions:
  1. Whether or not to purchase the (spot) service (knowing price & distribution)
  2. How to bid for the service (bidding strategy), if answer to 1. is Yes

Target Bidding Strategy

- For each job, bidding strategy Γ should maximize the job's *expected utility* (expectation is over possible spot price realizations)

$$U(t, v, \kappa, \Gamma) = V(t, v, \kappa, \Gamma) - P(t, v, \kappa, \Gamma) - D(t, v, \kappa, \Gamma)$$

where

– $V(t, v, \kappa, \Gamma)$: job value (vt) realized at job completion

– $P(t, v, \kappa, \Gamma)$: expected payment (for the spot service)

– $D(t, v, \kappa, \Gamma)$: expected delay penalty (given bidding strategy Γ)

Customers bid if and only if $U(v, t, \kappa, \Gamma) > 0$

$\Longrightarrow$ Two Primary Questions

- How should provider select prices to maximize expected revenue given known distribution of customer/job profiles?
  - Assuming rational users
- How should customers decide whether or not to bid, and if they bid, how to bid to maximize a job's expected utility?
  - Assuming known price distribution and knowledge of job profile

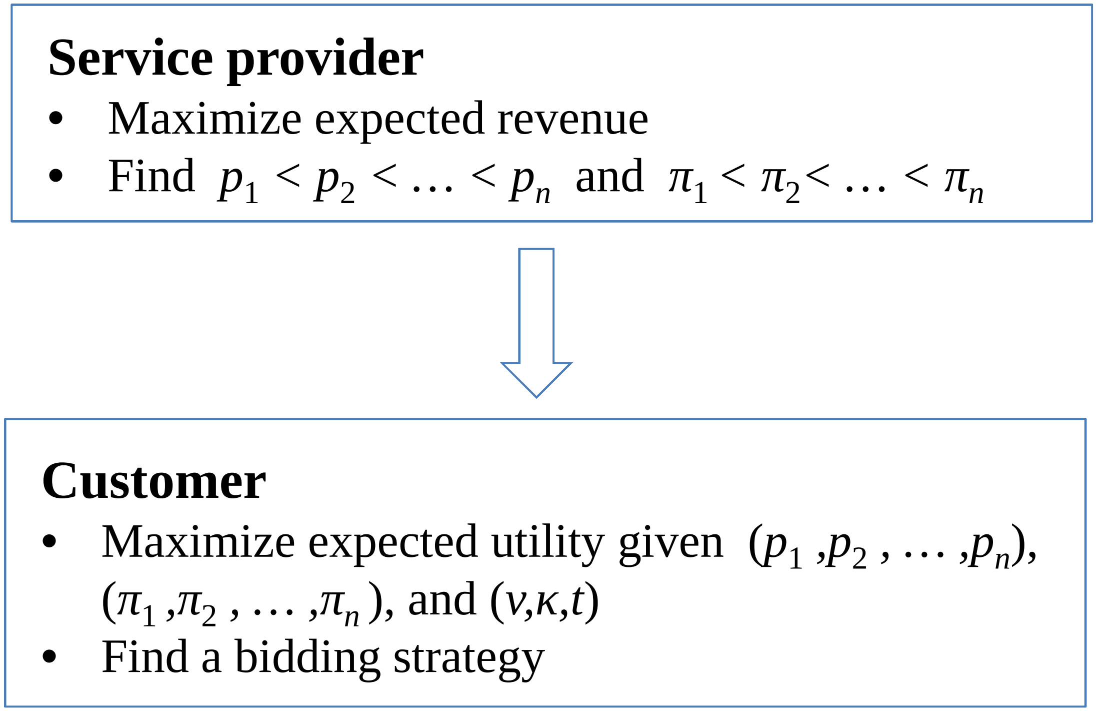
Model Parameters

- Service provider
  - Discrete set of prices $p_1 < p_2 < \ldots < p_n$ from which to choose spot prices (Amazon historical prices)
  - Distribution $\pi_1, \pi_2, \ldots, \pi_n$ for prices (frequency of each price)
- Customers: Job profiles (v,κ,t) and bidding strategy (Γ)
  - v and κ have joint density function $q(v, \kappa)$
    - *Correlation coefficient*, $\rho \in [-1,1]$, $\circ = \frac{E[v©] - E[v]E[©]}{\sqrt{\mathrm{var}(v)\mathrm{var}(©)}}$

    but are independent of t
  - t distributed according to $f(t)$
  - Γ is a function of (v,κ,t) and pricing

Optimization Framework

(Stackelberg Game)



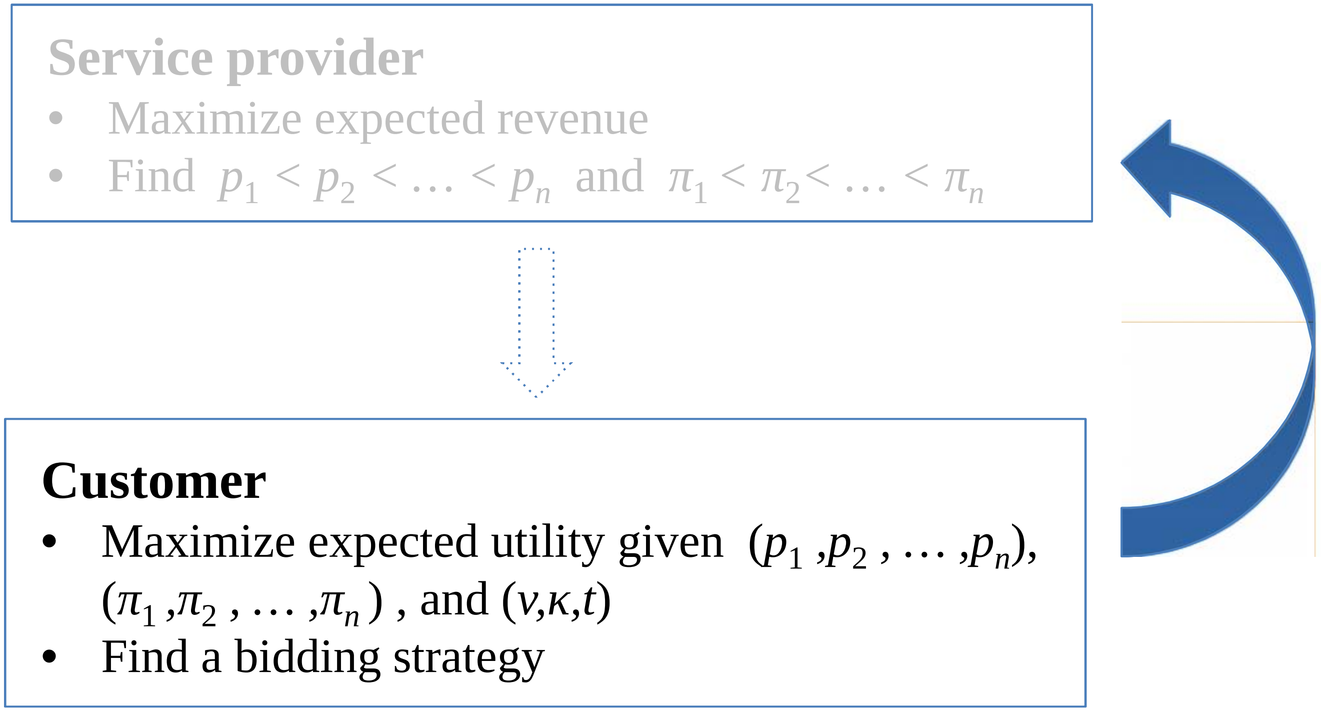
Service provider

- Maximize expected revenue
- Find $p_1 < p_2 < \dots < p_n$ and $\pi_1 < \pi_2 < \dots < \pi_n$

Customer

- Maximize expected utility given $(p_1, p_2, \dots, p_n)$, $(\pi_1, \pi_2, \dots, \pi_n)$, and (v, κ, t)
- Find a bidding strategy

Optimization Framework



Service provider

- Maximize expected revenue
- Find $p_1 < p_2 < \ldots < p_n$ and $\pi_1 < \pi_2 < \ldots < \pi_n$

Customer

- Maximize expected utility given $(p_1, p_2, \ldots, p_n)$, $(\pi_1, \pi_2, \ldots, \pi_n)$, and (v, κ, t)
- Find a bidding strategy

As usual, work backwards

Customer’s Optimization

$$\Gamma^* = \arg\max_{\Gamma} U(v, t, \mathrm{k}, \Gamma)$$

- Two simplifying assumptions (for analytical tractability)

1. Linear delay penalty

$$U(t, v, \kappa,\Gamma) = vt - P(t, v, \kappa,\Gamma) - \kappa T(t, v, \kappa,\Gamma)$$

where $T(t, v, \kappa,\Gamma)$ is the expected execution delay

2. Jobs are not terminated once bidding starts (positive utility in expectation over jobs with the same profile)

- Numerical exploration when relaxing those assumptions

Optimal Bidding Strategy (basically an MDP)

$$\Gamma^* = \arg\max_{\Gamma} U(v, t, \text{k}, \Gamma)$$

- ***<u>Fixed bidding strategy is optimal</u>***
  - Proof starts with jobs of size 1 and shows that a job with profile (1, *v*, *κ*) maps to a static bidding value *b**
  - Extension to a job of arbitrary size by induction
- *b** can be obtained through a simple linear search
  - It belongs to the set of spot prices $[p_1,p_2,\ldots,p_n]$
- As we shall see, the result is, however, fragile to relaxations of our simplifying assumptions, *i.e.*, allowing job termination and non-linear delay penalties

Properties of Optimal Bidding Strategy

$$b^* = \min \arg\min_{p_i \le p_n} \left\{ \frac{\sum_{p_j \le p_i} \text{®}_j p_j}{\sim(p_i)} + \text{©}\left(\frac{1}{\sim(p_i)} - 1 \right) \right\}$$

Average cost paid

Delay penalty

Fraction of time executing

where $\mathsf{a}(p_i) = \sum_{p_j \le p_i} \mathsf{p}_j$

- **If** a customer decides to bid for job (t, v, κ)
  - – b^* is determined solely by κ (***independent*** of v and t)
  - – b^* increases with κ
- The ***decision to bid***, however, depends on v (a job’s value affects its ability to generate positive utility)

Service Provider’s Optimization

$$\left(p^*,\text{®}\ {}^*\right) = \underset{p,\text{®}}{\arg\max}\, R(p,\text{®})$$

Where $R(p,\pi)$ is expected revenue given pricing (p,π)

Recall:

- t is independent of v and κ
- v and κ are correlated.

$$R_{p,\mathrm{p}} = \iiint_{v,\mathrm{k},t} f(t)q(v,\mathrm{k})P(t,\Gamma^*_{p,\mathrm{p}}(t,v,\mathrm{k}))dvd\mathrm{k}\,dt$$

where

$f(t)$: density function of job length

$q(v,\kappa)$: joint density function of v and κ

Optimal Pricing Strategy

(for discrete distributions)

- For a given density function $q(v, \kappa)$ with fixed marginal and correlation coefficient ρ, there exists ρ^* such that
  - When $\rho \le \rho^*$, a single price strategy is optimal, *i.e.*, introducing delay differentiation does not increase revenue
  - When $\rho > \rho^*$, a two-price strategy is optimal, *i.e.*, delay increases revenue
- The role of ρ
  - Increasing ρ increases the fraction of (v_2,κ_2) jobs that boost revenue, and decreases the fraction of (v_1,κ_2) jobs that lower revenue, and swaps increases in (v_1,κ_1) jobs for decreases in (v_2,κ_1) jobs in a revenue neutral fashion

Basic Discrete Model

- Users belong to four different “categories”

κ_1, v_1: low

κ_2, v_2: high

	κ_1	κ_2	
v_1	q_{11}	q_{12}	r
v_2	q_{21}	q_{22}	$1-r$
	s	$1-s$	

$$\circ = \frac{E[v\copyright]-E[v]E[\copyright]}{\sqrt{\mathrm{var}(v)\mathrm{var}(\copyright)}}$$

high/low value + high/low sensitivity to delay

- Fix marginal
- Vary correlation

⇨ Effect of correlation

Two Extreme Cases

Perfectly negatively correlated

	κ_1 can bid low	κ_2 has to bid high
v_1 can't afford high bid	0	1/2
v_2 can afford high bid	1/2	0

Perfectly positively correlated

	κ_1 can bid low	κ_2 has to bid high
v_1 can't afford high bid	1/2	0
v_2 can afford high bid	0	1/2

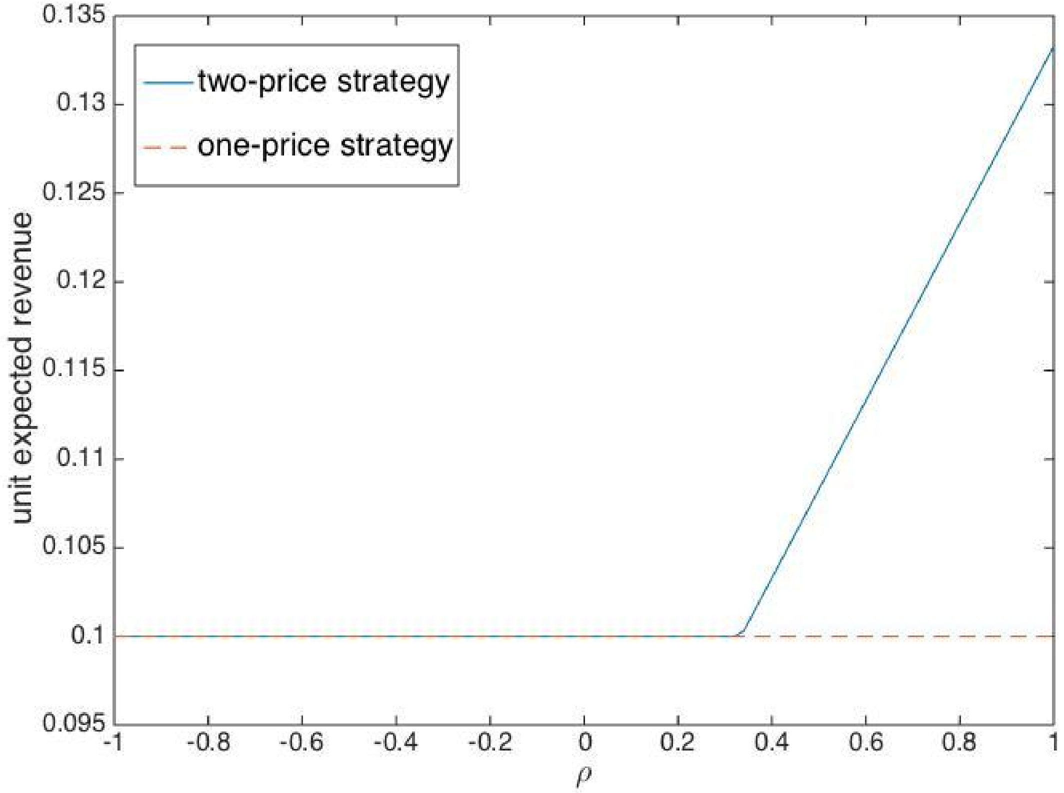
A two-price spot service has a positive impact if jobs with large delay sensitivity pay more. This in turn has the potential to 1) exclude jobs with large delay sensitivity and small valuation, and 2) extract a smaller price from jobs with small delay sensitivity and large valuation. 1) and 2) have to remain small

Properties of Optimal Pricing

- Optimal prices are *independent* of ρ that only affects the magnitude of the provider's revenue and not how to realize it
- The optimal pricing strategy extracts nearly all value from (v_2,κ_2) jobs (bidding at $p^*{}_2$) and (v_1,κ_1) jobs (bidding at $p^*{}_1$)
- The difference in utility between bidding at $p^*{}_2$ and $p^*{}_1$ is very small for (v_2,κ_2) jobs, *i.e.*, the optimal pricing policy is fragile

Expected Revenue

- $v_1 = 0.1$, $v_2 = 0.2$, $\kappa_1 = 0.1$, $\kappa_2 = 0.2$
- 50% of jobs have value v_1, and 50% have delay sensitivity κ_1



Testing for Robustness

- Two main results to test
  1. Optimality of fixed bidding strategy
  2. Presence of a correlation threshold below which spot service is of no benefit
- Two primary assumptions and one secondary
  - 1a. Jobs are never terminated once they start bidding
  - 1b. Delay penalty increases linearly
  - 2. Discrete job profile
- Which results still hold when relaxing those assumptions?
  - ⇒ Allow termination, non-linear delay penalties, continuous job profiles
  - – Optimality of fixed bidding is easily found to be fragile, but the existence of a correlation threshold held across all relaxations
- Approach is numerical in nature
  - – Optimal bidding can be computed as a dynamic program
  - – Test for threshold where single price solution stops being optimal

Allowing Job Termination

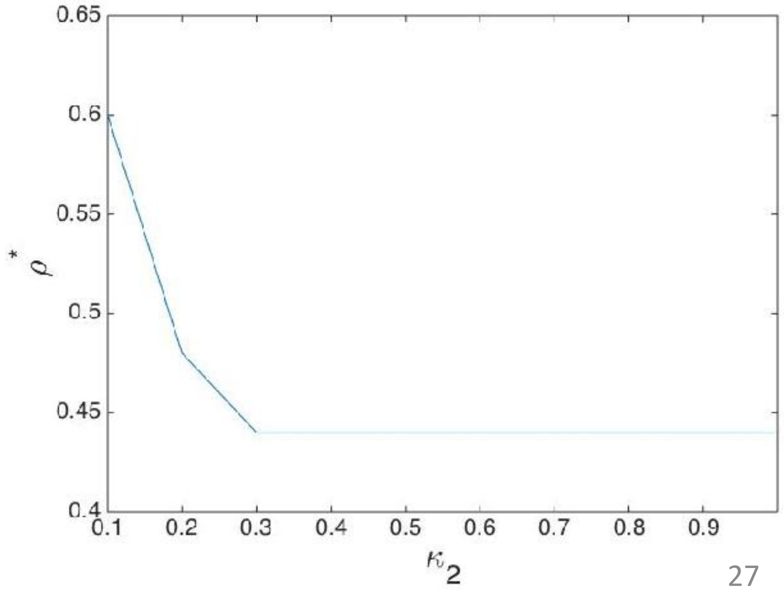
- Jobs are terminated when their expected residual utility becomes negative

  terminate if $vt - p(p_i)(t - t_0) - \text{©}\left(\text{²} - t + \frac{t - t_0}{\sim(p_i)} \right) \le 0$

  for linear penalty and fixed bidding (t_0 is execution time so far, and ² is elapsed time)
- Tested for different binary job profiles and combinations of job sizes (termination depends on job size)

Impact of Job Termination

- Jobs terminate after enough unlucky bids, but more interestingly they can switch to a higher price after enough successful bids
  - The ability to terminate limits the initial risk of bidding at a low price when the job size is large
- Termination also allows jobs with low value and high sensitivity to delay to bid (hoping to be lucky)
- Correlation threshold still exists, but is higher than when terminations were not allowed
  - Termination lowers revenue
- As before ρ^* decreases with κ_2



Nonlinear Delay Penalty Functions

- Piecewise-linear delay penalty function
  - “Convex” delay function
    - $D_1(\kappa, t) = \kappa \max\{0, T(t) - \theta\}$
  - “Concave” delay function
    - $D_2(\kappa, t) = \kappa \min\{T(t) , \theta\}$

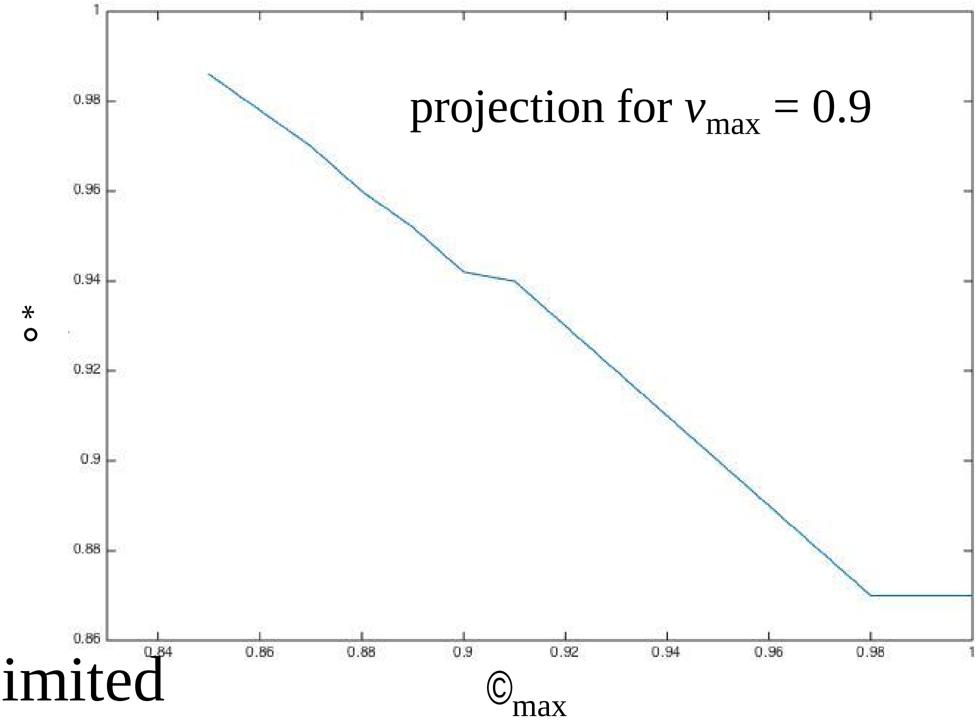
  θ is a threshold, t is the job’s execution time, and $T(t)$ is the job’s execution delay
- For simplicity, we preclude termination

Impact of Convex/Concave Delay Penalty

- Convex delay penalty results in low bids followed by high bids once the number of failed bids exceeds θ
  - Takes advantage of initial zero penalty
- Concave delay penalty can encourage starting with low bids before switching to high bids after enough lucky bids
  - Unsuccessful bids only reinforce the benefits of low bids, while enough lucky bids can tilt the balance (the relative weight of future unlucky bids goes up)
  - But the benefit is marginal under optimal pricing
- Presence of correlation coefficient holds under both convex and concave delay penalty
- Under convex delay penalty ρ^* increases with θ
  - A larger θ lowers the revenue extracted from $(v_2,©_2)$ users
- Under concave delay penalty ρ^* decreases with θ
  - A large θ brings the delay penalty closer to a linear function and makes it harder for jobs to consider initial low bids

More General User Profiles

- $v_{\min} = 0$, $\kappa_{\min} = 0$.
- $v_{\max} \in [0.1, 1.5]$,
- $©_{\max} \in [0.1, 1.5]$.


projection for $v_{\max} = 0.9$

- Gaussian copula
  - marginals: uniform distributions
- Optimal pricing search limited to one and two prices

For all ($v_{\max,}$ $\kappa_{\max}$) pairs, the result still holds, *i.e.*, optimality of one vs. more than one price depends on $\rho > \rho^*$

Some Related Works

- A long history of using pricing and delay differentiation in computing systems
  - Limited initial use as use mostly as internal mechanism (a queueing system formulation)
  - The advent of the cloud changed that
- Many recent works on bidding and pricing strategies to optimize various objective functions
- Most closely related works:

[1] *Fixed and Market Pricing for Cloud Services*

V. Abhishek, I.A. Kash, and P. Key, NetEcon 12 (updated and expanded 2017 version)

[2] *Revenue Maximization for Cloud Computing Services*

C. Kilcioglu and C. Maglaras, SIGMETRICS 15

Both assume

- (mostly) infinite capacity
- Jobs are heterogeneous in valuation, delay sensitivity
- Result: spot service is useful under some conditions

Neither explicitly studies the role of correlation.

Summary and Extensions

- Summary
  - A spot service needs a sufficiently high correlation between job value and sensitivity to delay to be competitive compared to a one-price service (on-demand)
  - Fixed bidding is often adequate though dynamic bidding can offer benefits in some cases
    - When jobs can be terminated, small, low value, high delay sensitivity job can use it to take their chance
    - Under convex delay penalty jobs can bid low as long as the delay penalty remains small

    In practice though, the benefit of dynamic bidding is small and the (computational) cost is non-trivial
- Extensions:
  - Relax assumption of infinite capacity
  - Allow demand-sensitive pricing
  - Explore other pricing mechanisms, *e.g.*, auctions when supporting opportunistic jobs

Thank You!

Questions?

See: J. Song & R. Guerin, "*Pricing (and Bidding) Strategies for Delay Differentiated Cloud Services*," for details

Accepted for publication in ACM Transactions on Economics and Computation