Deconstructing Amazon EC2 Spot Instance Pricing

Orna Agmon Ben-Yehuda Muli Ben-Yehuda
Assaf Schuster Dan Tsafrir

Department of Computer Science
Technion — Israel Institute of Technology

CloudCom 2011 (and Epilogue)

Amazon EC2 cloud terminology

Amazon rents virtual machines with prices which vary according to:

- Instance types
- Regions
- Operating systems
- Commitment level: reserved, on-demand, spot
- Payment by the hour, except for the last hour fraction of a terminated spot instance.

What are spot instances?

- Clients bid (attach a maximal price to the instance request).
- The provider publishes a uniform spot price every so often, which the user pays.
- As long as the bid exceeds the spot price, the instance can stay.
- An instance is killed if the price goes above the bid.

Why sell spot instances?

Idle machines	Spot Instances
kept for elasticity	easily evacuated
can be sold cheap	must be sold cheap

Amazon EC2 Spot instances declaration

"The Spot Price changes periodically *based on supply and demand*. . ."

- How does Amazon price its spot instances?
- Are spot prices really based on natural supply and demand? Or
- Are they artificially set, raised above the market value?

Who cares?

- Researchers learn about the market from EC2 price histories; they assume (following Amazon's statement) that spot prices reflect real bids [Zhang et al. 2011], or represent market clearing prices [Chen et al. 2011].
- Clients bid and evaluate bidding strategies using price histories.
- Other providers seek information about the market and pricing algorithms.

Who cares?

- Researchers learn about the market from EC2 price histories; they assume (following Amazon's statement) that spot prices reflect real bids [Zhang et al. 2011], or represent market clearing prices [Chen et al. 2011].
  - If prices are artificial, their results are questionable.
- Clients bid and evaluate bidding strategies using price histories.
- Other providers seek information about the market and pricing algorithms.

Who cares?

- Researchers learn about the market from EC2 price histories; they assume (following Amazon's statement) that spot prices reflect real bids [Zhang et al. 2011], or represent market clearing prices [Chen et al. 2011].
- Clients bid and evaluate bidding strategies using price histories.
  - If prices are artificial, an algorithm change may make the past irrelevant to future predictions.
- Other providers seek information about the market and pricing algorithms.

Who cares?

- Researchers learn about the market from EC2 price histories; they assume (following Amazon's statement) that spot prices reflect real bids [Zhang et al. 2011], or represent market clearing prices [Chen et al. 2011].
- Clients bid and evaluate bidding strategies using price histories.
- Other providers seek information about the market and pricing algorithms.
  - If Prices are artificial, they do not supply such information.

Examples of spot instance market-driven mechanisms

- Clients bid secretly.
- The provider sorts the bids (descending order).
- Uniform price for all granted instances.
- The provider grants only the first N bids. N is limited:
  - supply
  - revenue maximization
  - minimal price
  - (hidden) reserve price.
- Pricing according to minimal price or bid $N + 1$.

Price histories

Amazon encourages clients to look at price histories and bid accordingly. A common view:

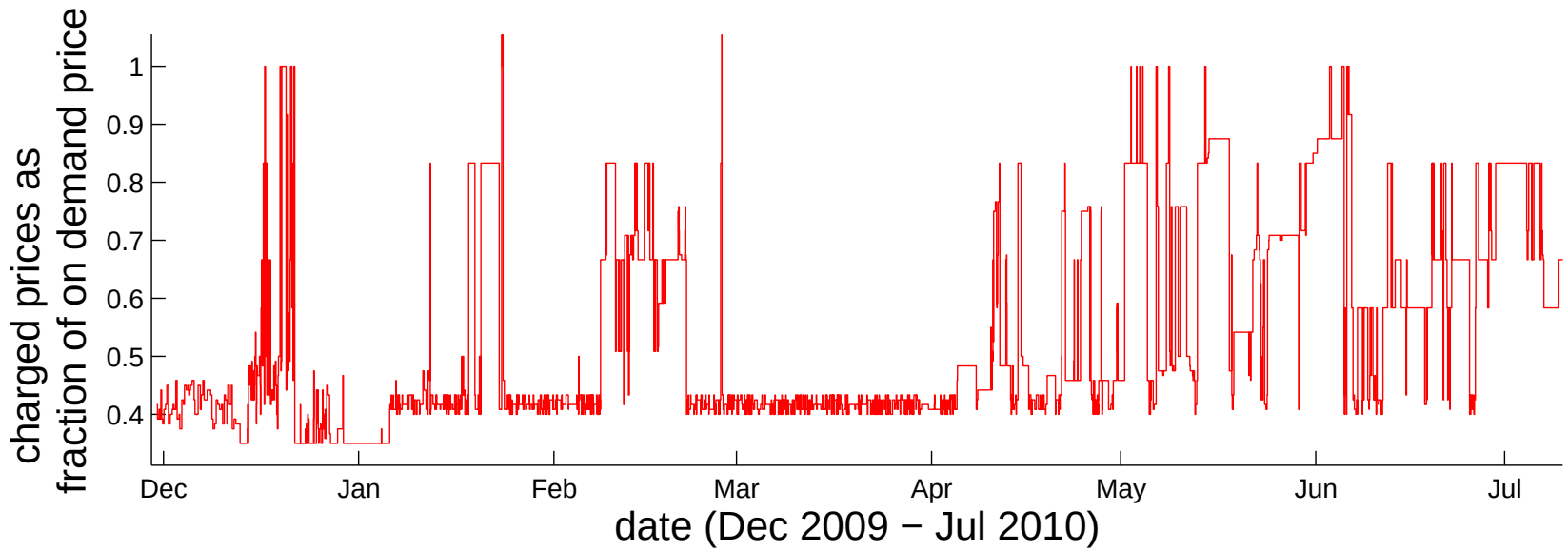
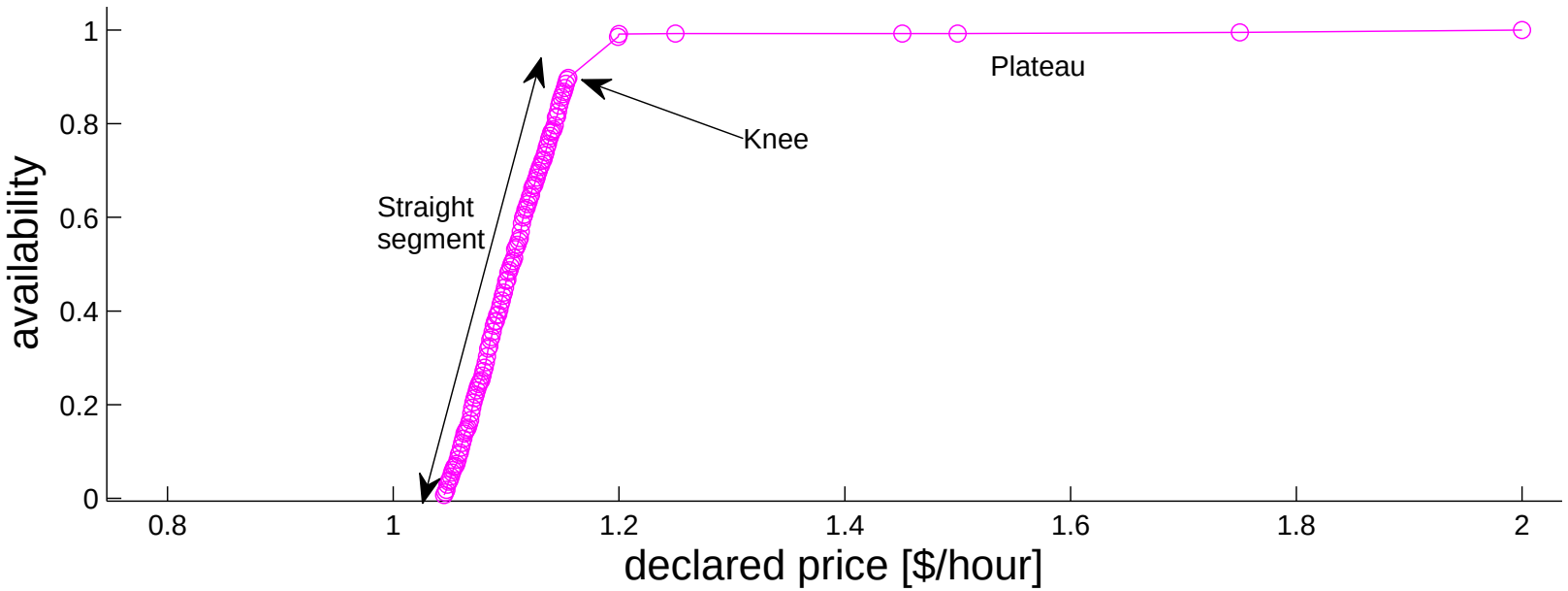


Figure: windows.m1.small.us-east

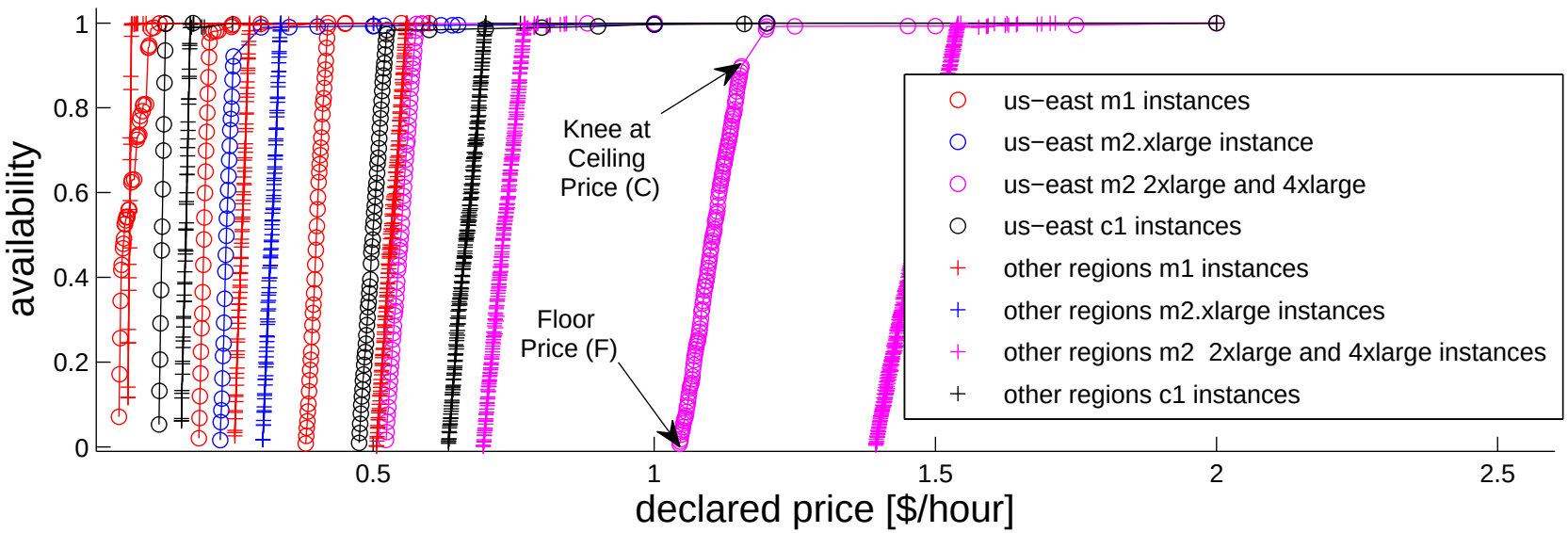
Alternative view—availability of bid price

The time in which the spot price was below the bid price, divided by the total time.



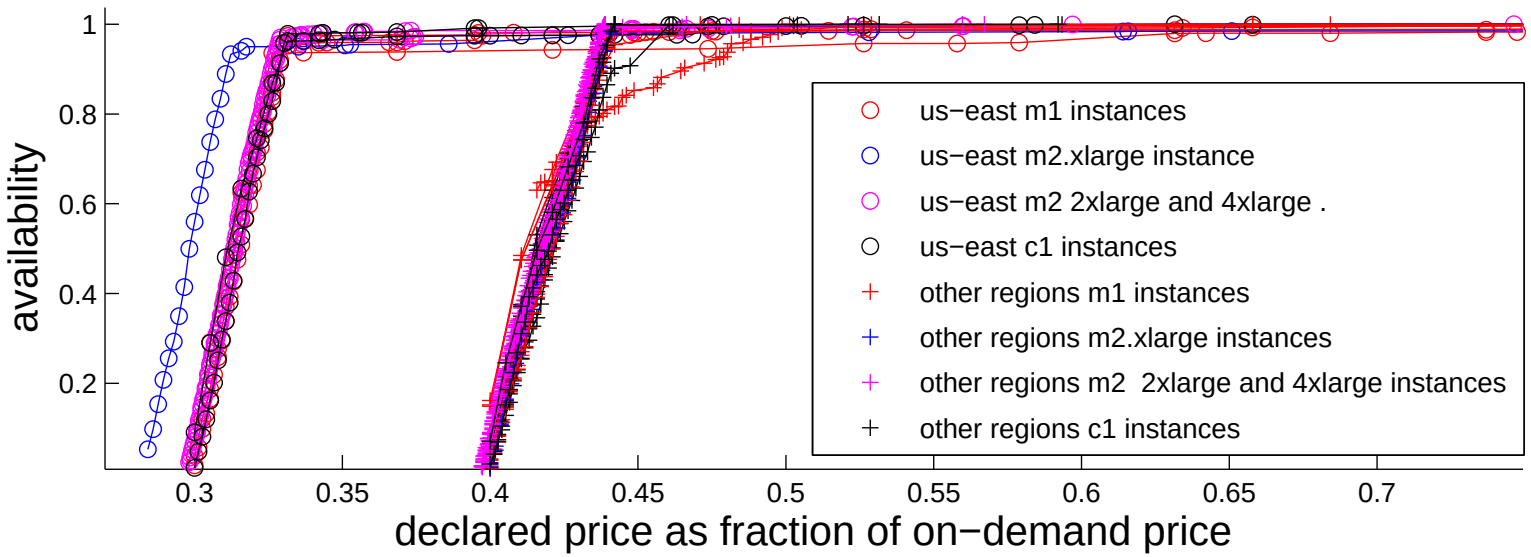
Typical shape: a straight segment and a high knee.

Windows instance availability as a function of price



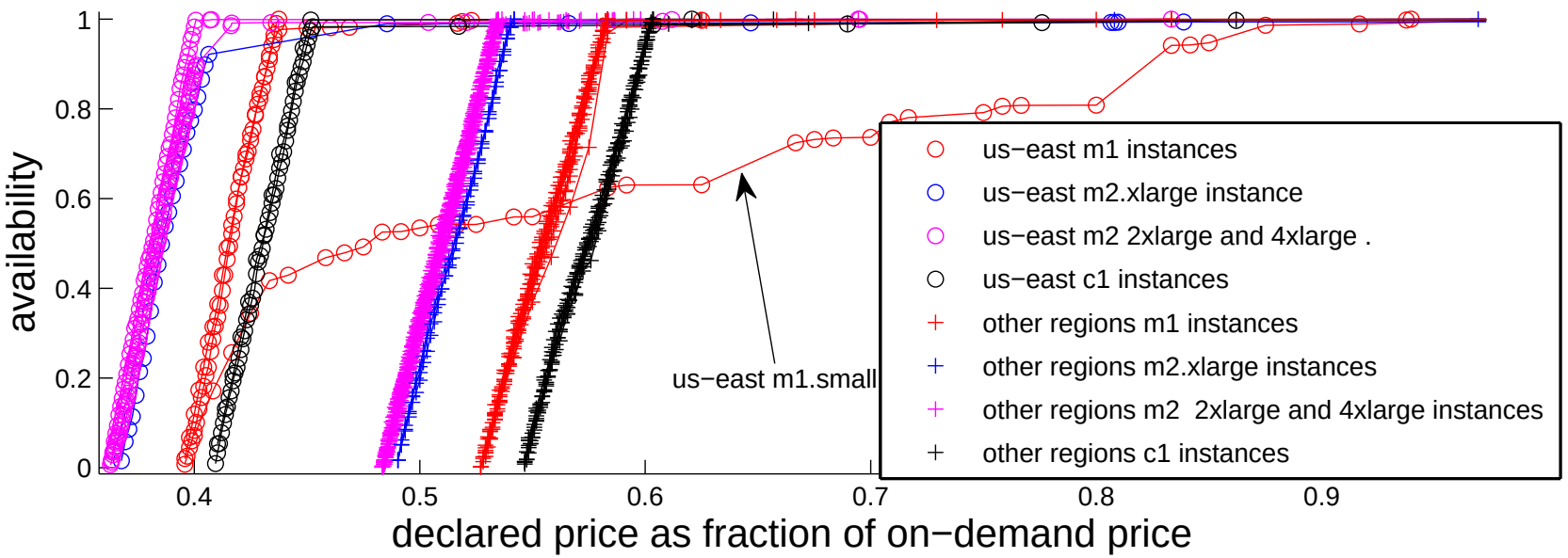
The typical shape at different prices. Looks similar for Linux.

Linux instance availability as a function of normalized price



Two groups of regions (one and the rest). The forest disappears.

Windows instance availability as a function of normalized price



A repeating pattern within the two region groups.
Windows clients differ from Linux clients.

Our hypothesis

Natural supply and demand conditions cause all this? Gee, that's funny.

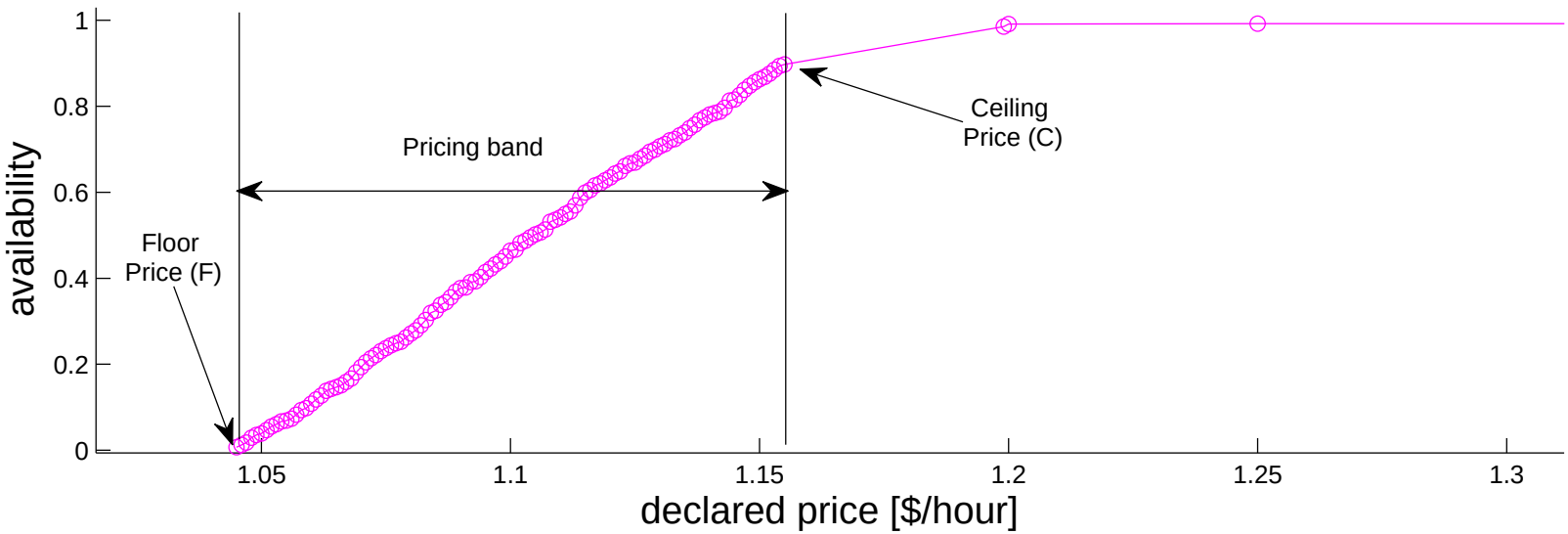
Alternatively...

- Amazon often changes the auction's reserve price, independently of client bids.
- The reserve price's value and its changing frequency are not market driven.
- Usually, the spot price is identical to the reserve price.
- Hence, the spot prices are usually not market-driven.
  - In contradiction to Amazon's statement.

Why dynamic secret reserve price?

- A dynamic reserve price maintains an impression of constant change. Forces clients to
  - Bid higher or
  - Tolerate sudden unavailability.
- A secret dynamic reserve price also masks times of low demand and price inactivity, by giving an illusion of false activity.

Planning the dynamic reserve price algorithm

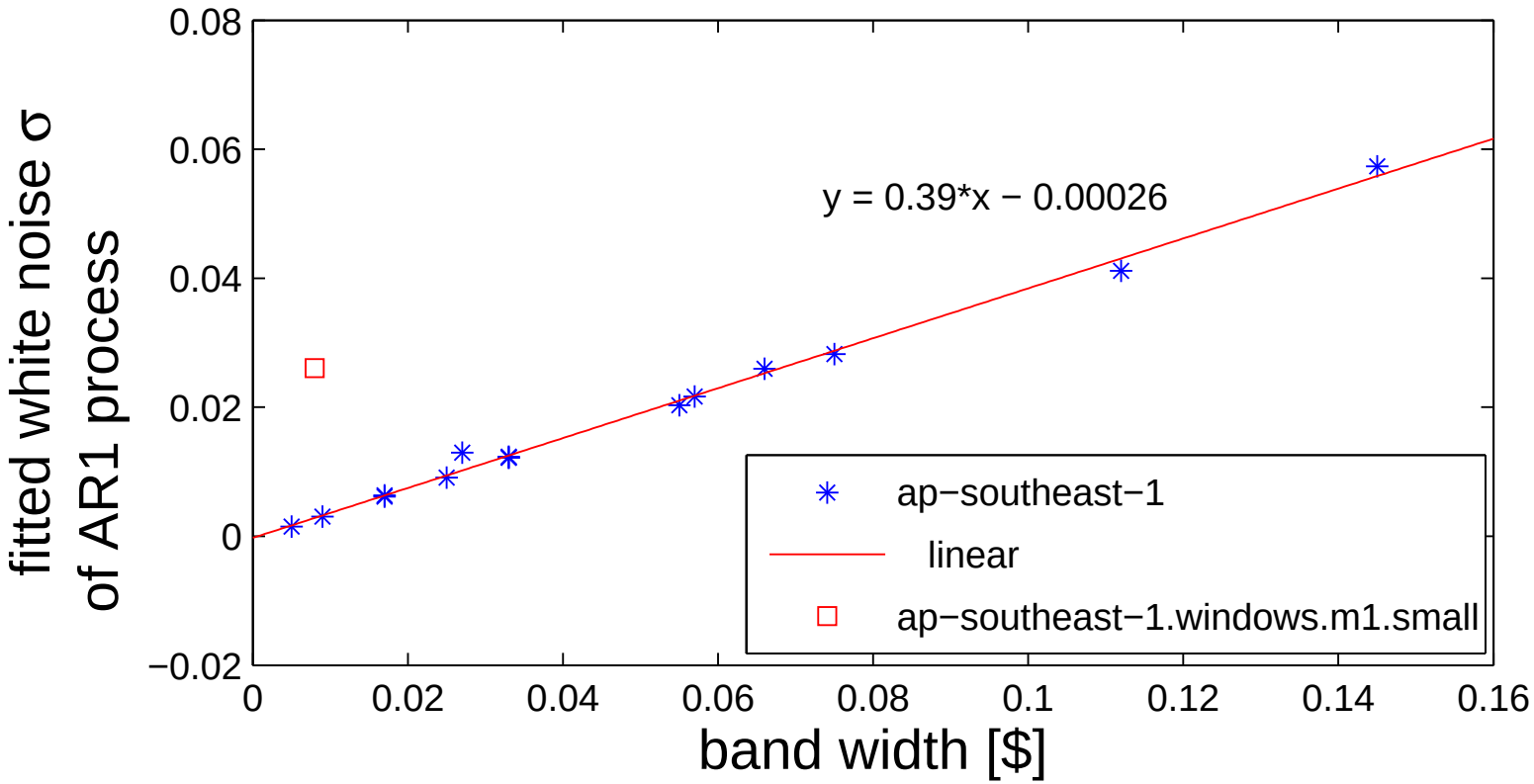


Fitting an auto-regressive process $AR(1)$ for ap-southeast.windows types

$$\Delta_i = -a_1 \Delta_{i-1} + \epsilon(\sigma)$$

- Δ_i is the difference of two consequent prices.
- $a_1 = 0.7$.
- $\epsilon(\sigma)$—white noise with a standard deviation $\sigma = 0.39(C - F)$.
- m1.small matched $a_1 = 0.5, \sigma = 0.5(C - F)$.

Variance of the fitted AR(1) process



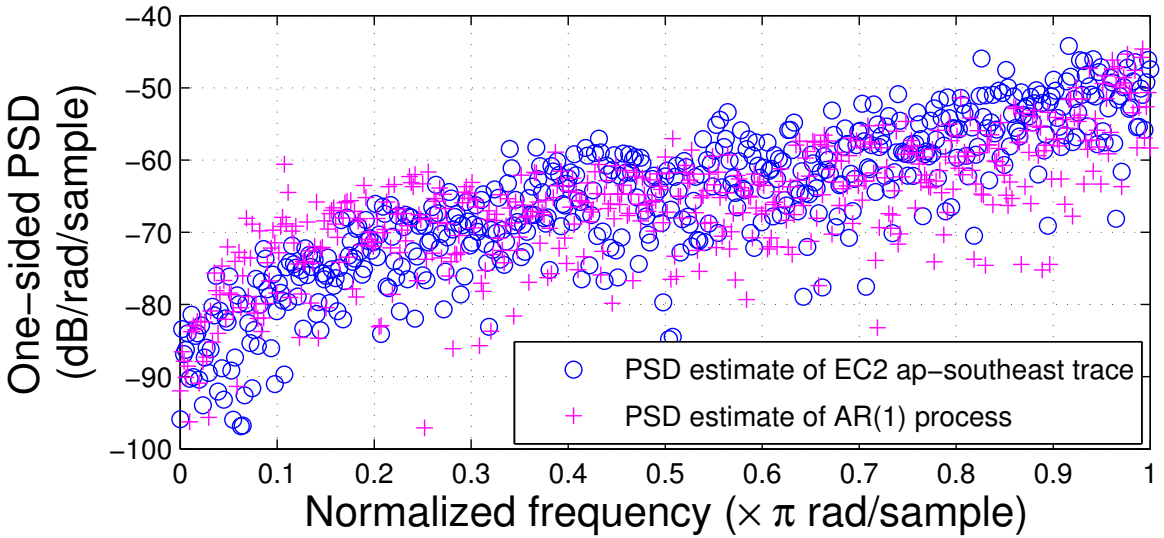
The close fit supports our hypothesis.

Constructing the reserve price algorithm

- Initial price is F.
- Initial change is $-0.1(C - F)$.
  - Not all initial conditions are good.
- Compute next price change using the fitted AR(1) process.
- Advance the next price $P_i = P_{i-1} + \Delta_i$.
- Truncate the process to the range $[F, C]$ by regenerating the white noise component while P_i is outside the $[F, C]$ range or identical to P_{i-1}.
- Round all prices to 0.1 cent.

Is the constructed algorithm consistent with reality?

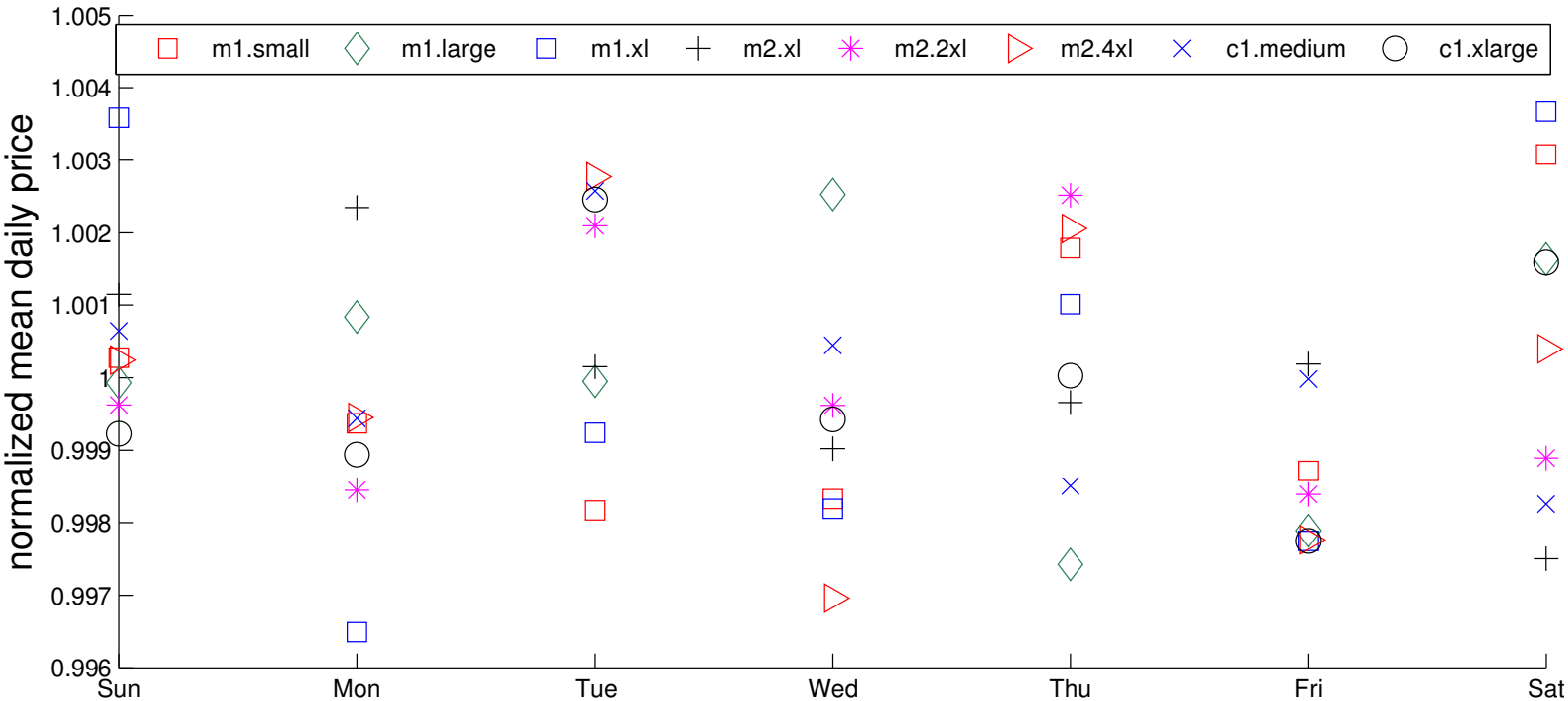
Periodogram (power spectral density): a power-normalized discrete Fourier transform.



The close fit supports our hypothesis.

Is the AR(1) process natural or artificial?

A natural process would have a significant weekly cycle.
The normalized weekly averages of ap-southeast.windows types do not show a weekly cycle:
The day-of-week impact is smaller than the noise (impact of types).



The AR(1) process is inconsistent with a natural process.

Is the AR(1) process partly natural or artificial?

- Partly natural: partly real bids within band above the reserve price, partly reserve prices. Expected to have a mean price above mid-range.
- The mean price is lower than the mid-range (by up to 2%).
- Many clients already noted that bidding inside the band is not cost effective.
- The AR(1) process's average is consistent with an average of an artificial process.

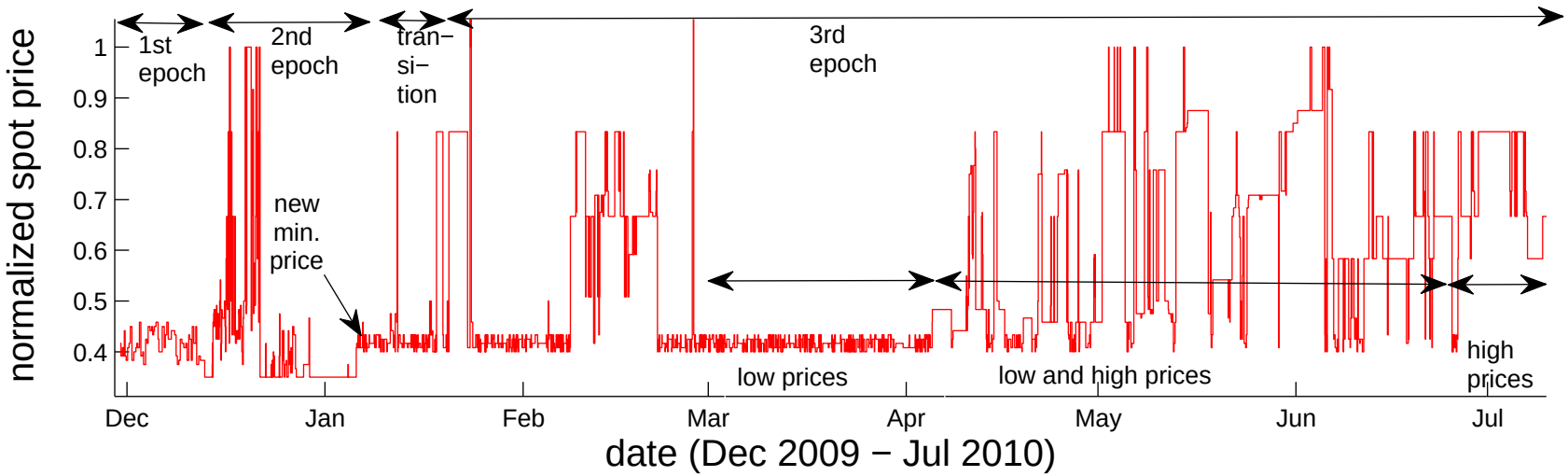
Are traces as a whole natural or artificial?

- 98% of the time, prices are within the band.
- Traces as a whole are consistent with being artificial 98% of the time.

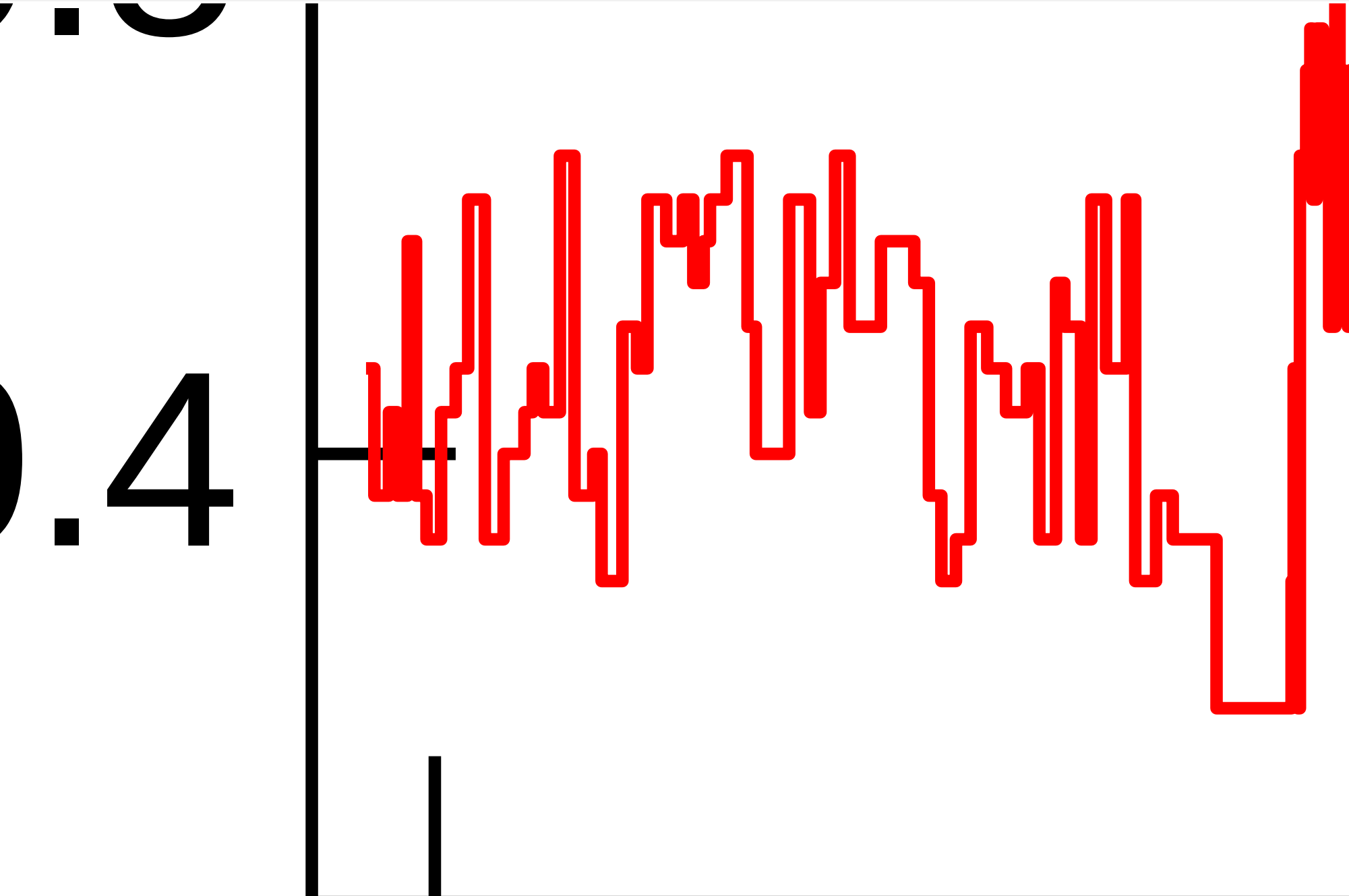
If our hypothesis is correct, then:

- 98% of the time spot prices carry little information about real client bids!
- Researchers *cannot* learn from spot prices about client valuations for products, nor about supply and demand.
- The spot price is *not necessarily* a market clearing price.

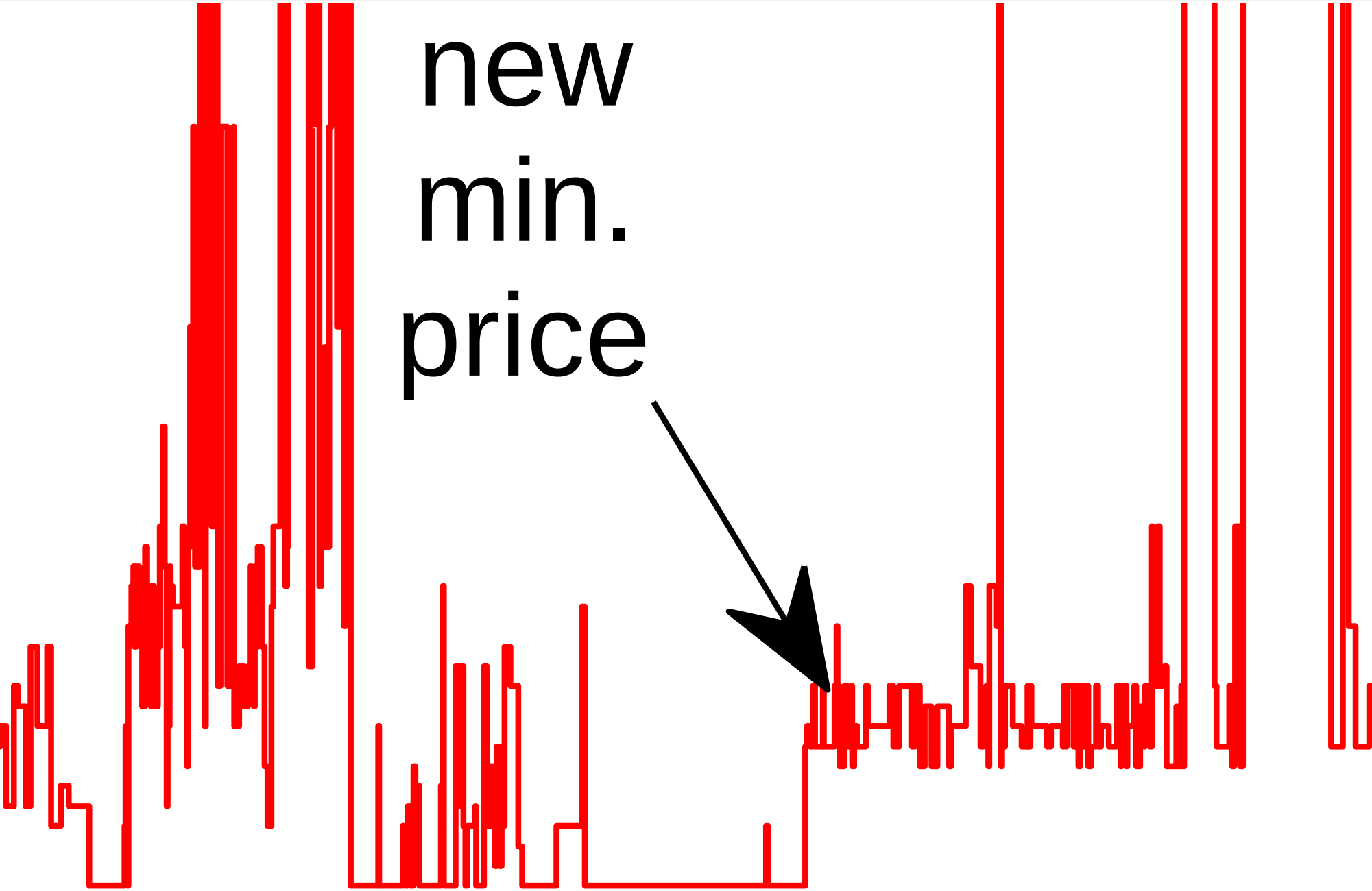
Pricing epochs



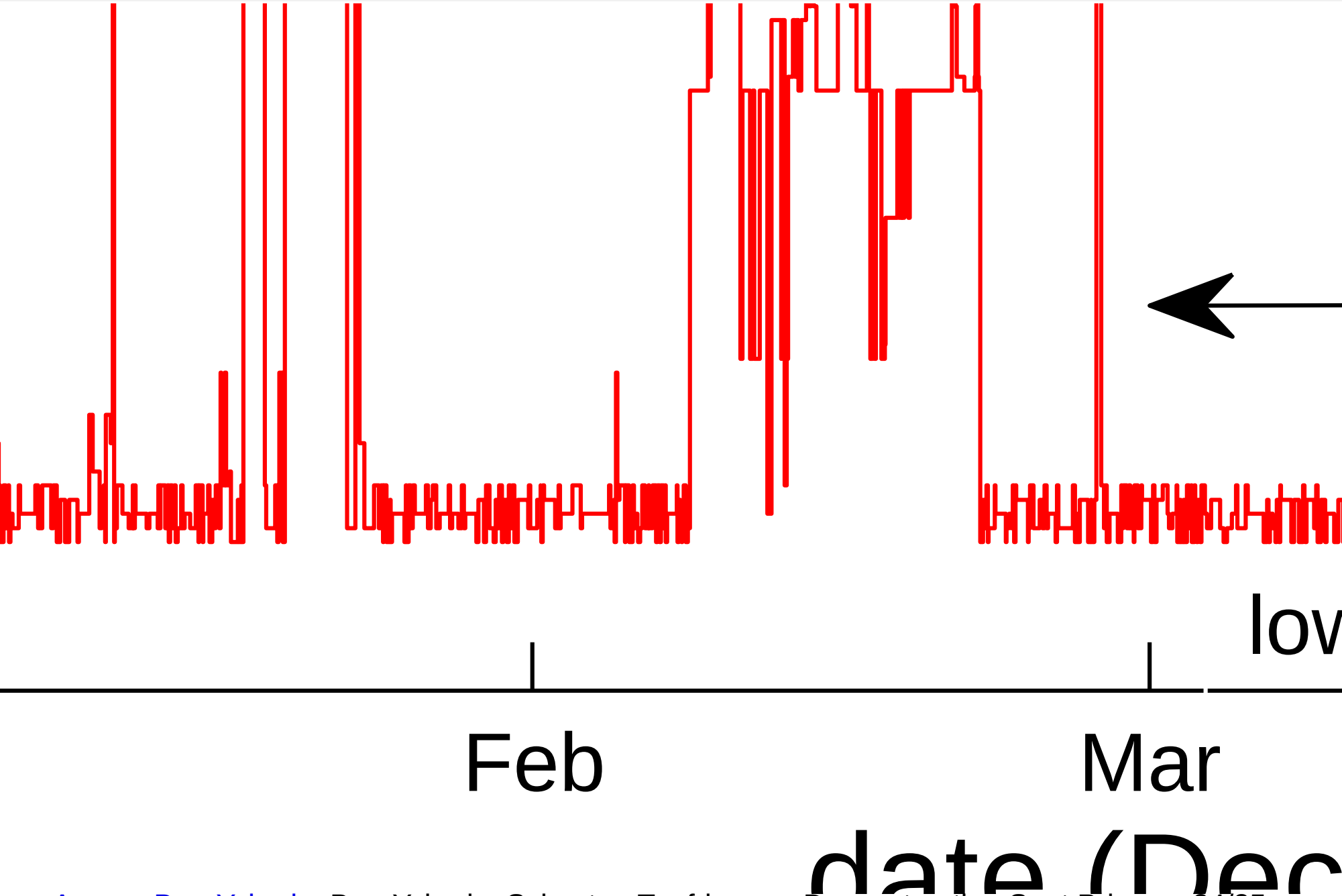
Pricing epochs



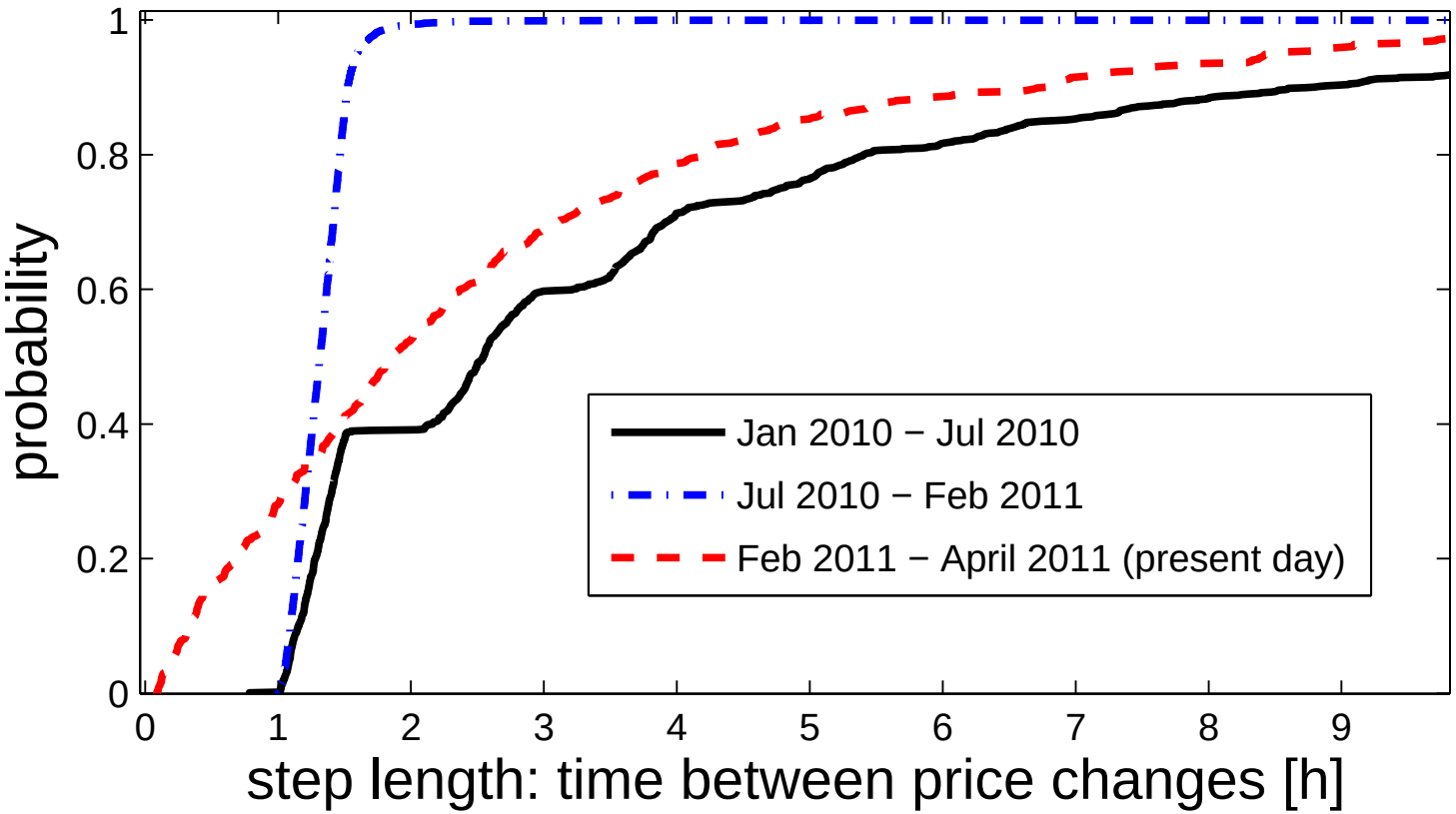
Pricing epochs



Pricing epochs



Price changing timing (us-east)



Workload modeling

- Workload traces of large systems. Truncated to tasks longer than 10 minutes, shorter than 24 hours.
- Grid: LPC-EGEE, a cluster of a large grid.

Customer bid modeling

No data!
We test three models, to show that the qualitative results are insensitive to the model.
Bids are concentrated between a minimal price (0.4) and the on-demand price (1).

- Pareto distribution (minimal value of 0.4, Pareto index of 2).
- $\mathcal{N}(0.7, 0.3^2)$, truncated at 0.4.
- A linear mapping from runtimes to (0.4, 1], which reflects client aversion to having long-running instances terminated.

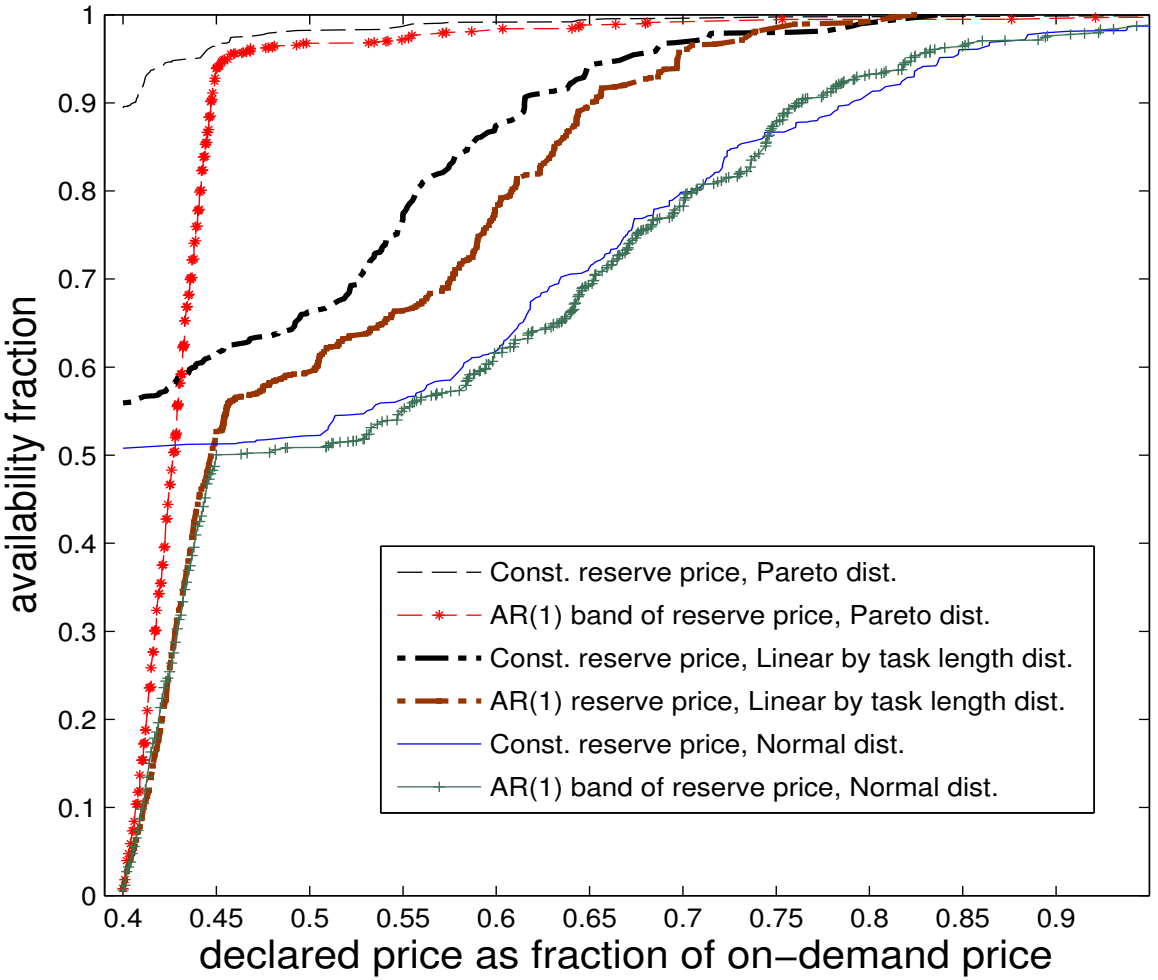
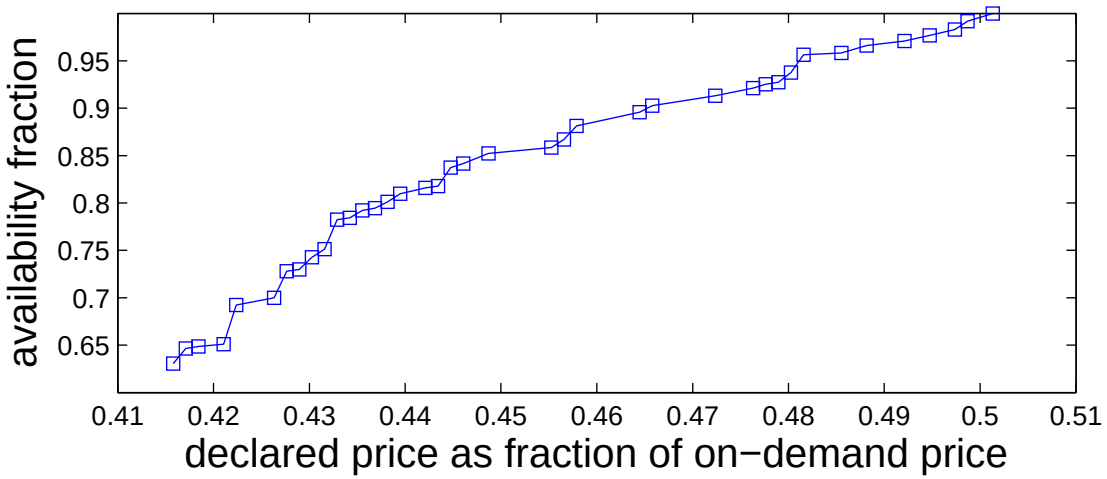
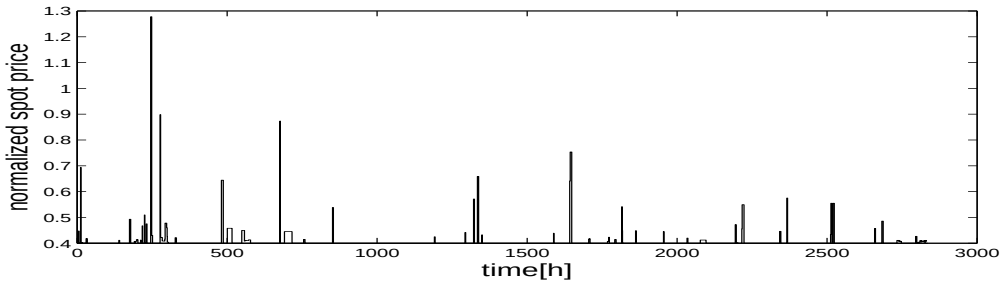


Figure: Linear segment and knee iff simulating with AR1 dynamic reserve price, insensitive to client bidding. Consistent with traces.

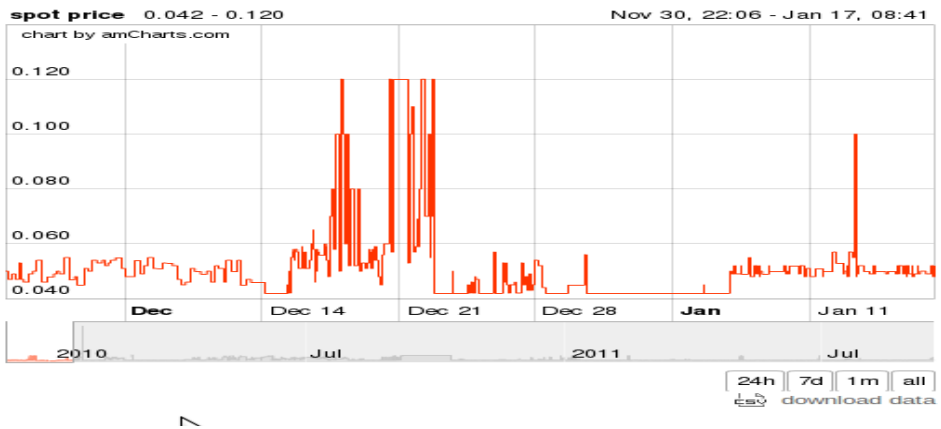
Epoch 2


availability fraction
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.41
0.42
0.43
0.44
0.45
0.46
0.47
0.48
0.49
0.5
0.51
declared price as fraction of on−demand price

Price trace comparison



(a) ***LPC-EGEE, constant reserve price***



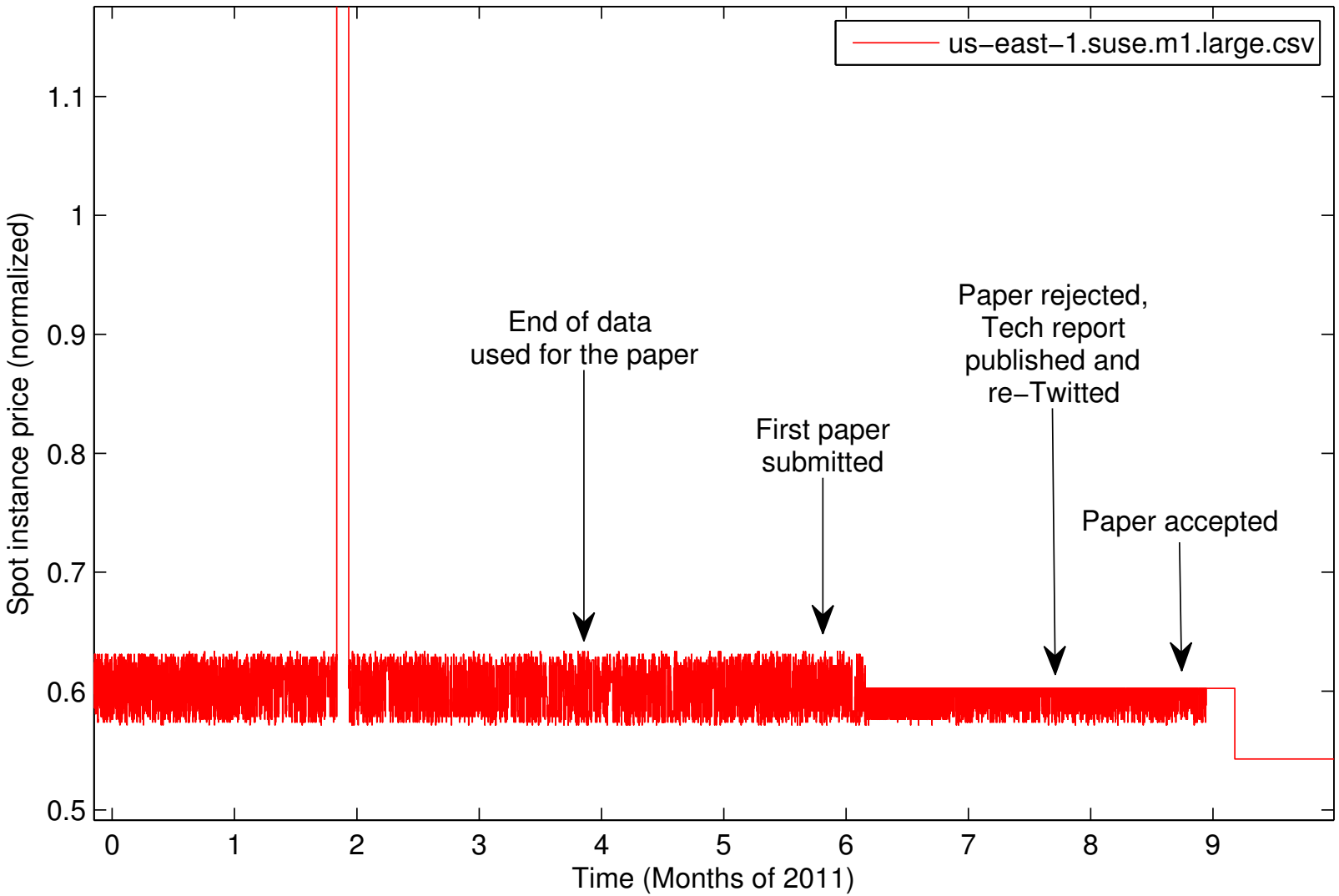
(b) ***Second Epoch***

The second epoch is consistent with a constant reserve price.

Conclusions

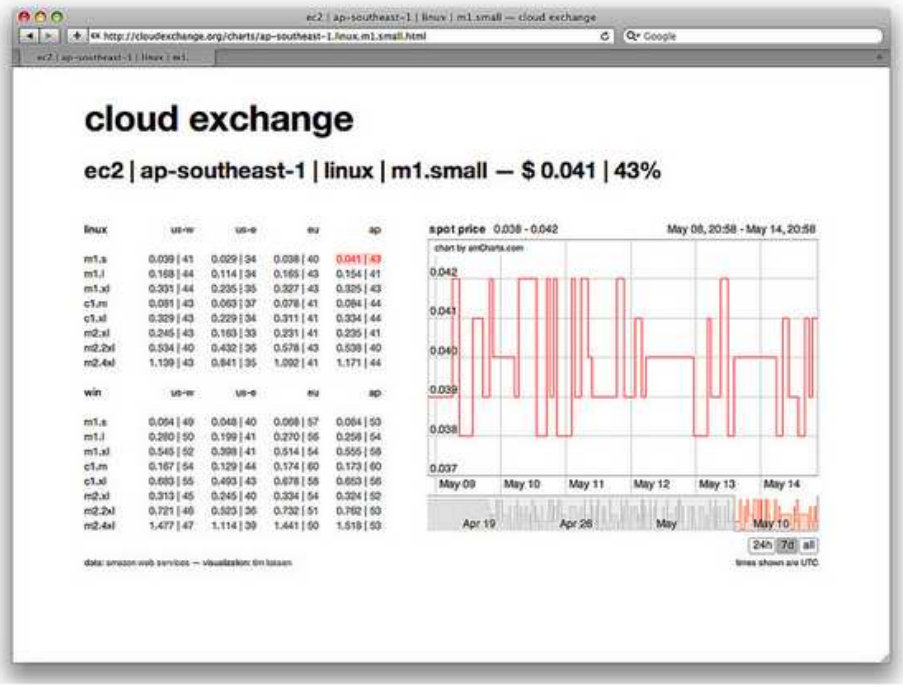
- It is likely that Amazon sets spot prices using an AR(1) (hidden) reserve price.
- 98% of the time:
  - The spot price is probably just the reserve price.
  - EC2 traces do not necessarily represent clearing prices or real bids.
- Many features (minimal price, band width, change timing) are artificial, have changed and may suddenly change again.

Post mortem


us−east−1.suse.m1.large.csv
Spot instance price (normalized)
Time (Months of 2011)
End of data
used for the paper
First paper
submitted
Paper rejected,
Tech report
published and
re−Twitted
Paper accepted
1.1
1
0.9
0.8
0.7
0.6
0.5
0
1
2
3
4
5
6
7
8
9

cloud exchange

this website used to display interactive charts of amazon spot instance prices:

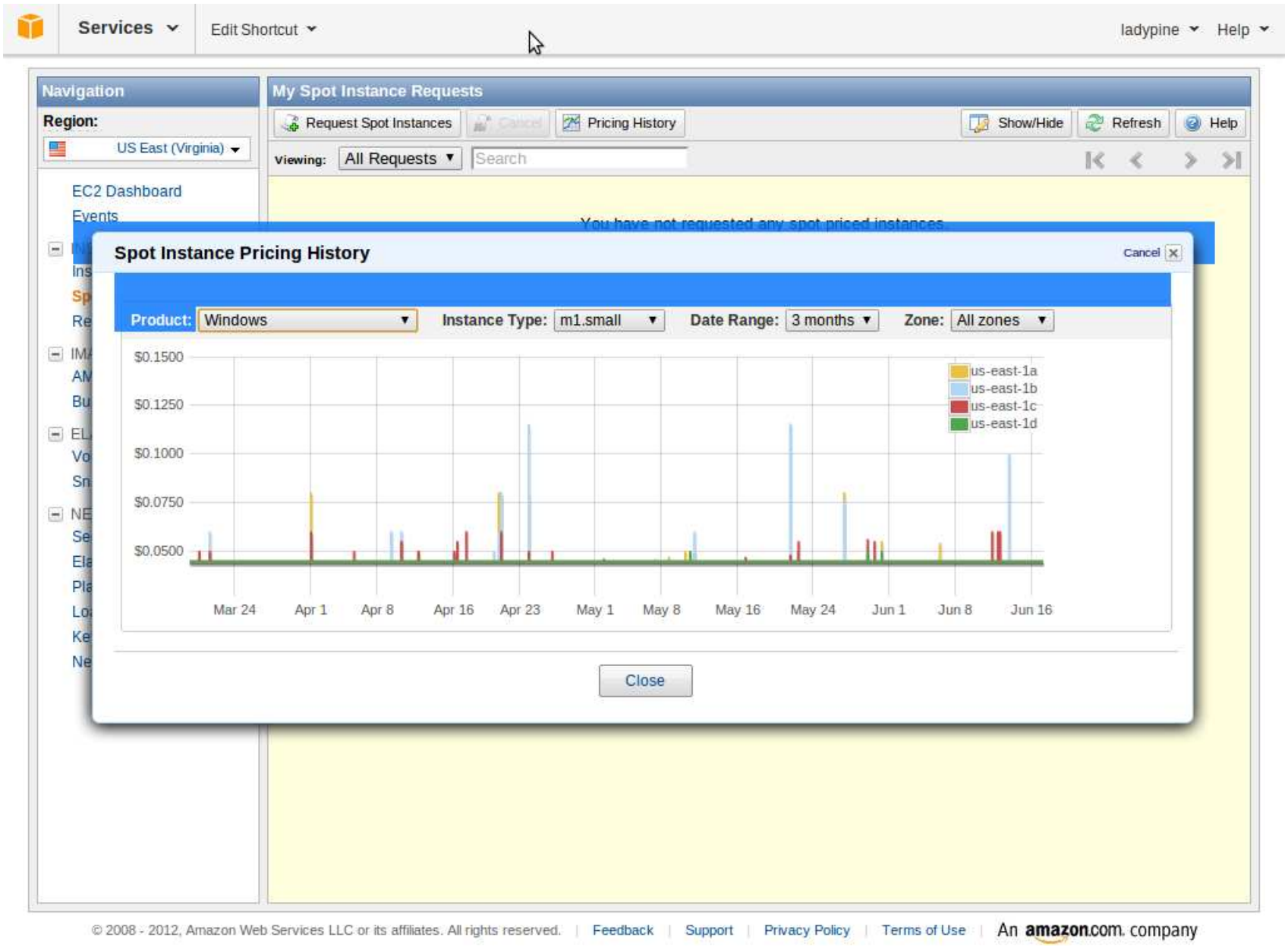


unfortunately, due to time constraints i cannot maintain this service any longer. if you want to build something similar, feel free to use my code as a starting point.

you might also want to check out the paper by orna ben-yehuda et. al., which tries to reverse-engineer the pricing algorithm used by amazon, and concludes that prices *"are usually not market-driven as sometimes previously assumed. Rather, they are typically generated **at random** from within a tight price interval"*.

tim, january 2012

Current View on Amazon



Questions?

Contact us at:
{ladypine, muli, assaf, dan } at cs.technion.ac.il
Thank You!