Analyzing AWS Spot Instance Pricing

Gareth George, Rich Wolski, Chandra Krintz
Computer Science Dept.
Univ. of California, Santa Barbara
{gareth,rich,ckrintz}@ucsb.edu

John Brevik
Dept. of Mathematics and Statistics
California State Univ., Long Beach
John.Brevik@csulb.edu

***Abstract*—**
Many cloud computing vendors offer a preemptible class of service for rented virtual machines. In November 2017, Amazon.com changed the pricing mechanism for its preemptible "spot instances" so that prices would change more "smoothly." This paper analyzes the effect of this change on spot instance prices. It examines the prices immediately before and after the mechanism change to determine the extent to which prices themselves changed. It then compares the 90-day period immediately after the change in mechanism to the next 90-day period. Finally, it compares the two most recent 90-day periods (ending on October 15, 2018). Our results indicate that in addition to smoothing prices, the mechanism change introduced generally higher prices which is a trend that continues.

***Index Terms*—Amazon Web Services, AWS, Economics of Cloud**

I. Introduction

One of the fundamental tenets of cloud computing is that resources (*e.g.* machines, network connectivity, storage, etc.) be characterized by their capacity and capability characteristics rather than their physical construction. Programmers and users reason about the use of cloud resources in terms of these characteristics (in principle) without regard to the physical infrastructure that is used to deliver them. Amazon provides no guarantee that any specific processor model will be used to fulfill a specific user's request (or is even available in their cloud).

Instead, users enter into a "Service Level Agreement" (SLA) that quantifies the minimum capability that a particular request will receive. Typically, if a service provider violates the agreement by failing to meet this "quality of service," the user is entitled to some form of financial compensation. Thus the cloud computing model is one in which users can reason about the capabilities that their computations require, and will receive, in terms of SLAs and not the physical capabilities of specific resources.

Cloud computing vendors may offer different SLAs at different price points so that users can control the value transaction at a fine level of granularity. Vendors such as Amazon and Google [12], for example, offer reduced-price preemptible service tiers with no reliability SLAs.

Until recently (November 2017), Amazon offered these instances as "spot instances" [5], for which the reliability was based (in part) on the maximum amount of money a user agreed to pay for them. This "maximum bid price" is specified in a user's resource request and is private to the user and Amazon. To allocate instances to requests Amazon then created a market for each instance type and satisfied the requests of the highest bidders. Periodically, Amazon recalculated the market price and terminated those instances whose maximum bid was below the new market price. Thus, each user had to devise a bidding strategy that met his or her own reliability needs [13], [19]. Amazon also offers the same instance types under a variety of other pricing plans (such as "on-demand" pricing) that are typically more expensive, where the user is paying for guaranteed reliability without the risk of termination.

At the end of November, 2017 Amazon announced a change to the spot instance pricing mechanism. While originally described as "price smoothing," the new mechanism is a retail pricing mechanism in which Amazon sets prices dynamically using an algorithm that it does not disclose. In particular, this new mechanism does not use maximum bids as a market-clearing mechanism: Amazon simply sets the retail price from moment to moment.

Importantly, Amazon also changed the algorithm it uses to select instances for termination when a price change or resource shortfall occurs. No longer are "maximum bids" used to prioritize terminations. Thus, previous work such as [21] can no longer be used to determine maximum bid prices that would provide some statistical bound on minimum execution duration. Furthermore, Amazon now provides a "Termination Notice" [8] when an instance is about to be terminated.

In this paper, we examine the price change that accompanies this product change for the spot-instance product. Because spot instance execution duration is no longer partially determined by the client's "maximum bid" we would expect a drop in prices as we view this as a drop in reliability which is the primary differentiating characteristic between spot and on-demand instances, yet the introduction of new features such as price smoothing and termination notices could explain increased prices to "cover" sharp changes in demand or supply.

By comparing price histories occurring immediately before the change in prices to ones for the same instance types immediately following the switch, this work illustrates the nature of the price change. We find that in general, the switch to smoother retail pricing carried a price increase for many, but not all instance types.

We also compare prices from the time period immediately

This work is supported in part by NSF CNS-1703560, OAC-1541215, CCF-1539586, ACI-1541215

following the switch in November of 2017 to those set in the second quarter of 2018 as well as the prices in the two most recent periods available to us (as of October 15, 2018) to determine whether there is a trend in retail pricing implemented by Amazon. This study shows that Amazon spot instance prices have continued to increase since the switch from market pricing to retail pricing.

This paper makes the following contributions:

- It describes the previous and new pricing mechanisms for spot instances and discusses techniques for analyzing price histories under each regime.
- It quantifies the change in prices that accompanied the switch in pricing mechanism.
- It identifies trends in the new retail pricing mechanism since the switch.

These contributions demonstrate the depth of analysis that is necessary to understand a change in the spot-instance product from Amazon. More generally, virtualized cloud products are opaque to their users, who must rely on vendor documentation to understand the nature of the value proposition with which they are presented. In this case of this change, the spot-instance product is the *same* product with a different pricing mechanism and consequently different reliability characteristics in implementation. However, this work shows that this change generally manifests itself as an increase in price. Thus, this work is a harbinger of the analysis techniques that may be necessary to understand change to current and future cloud resource values.

II. Background

Amazon Web Services (AWS) includes the ability to rent "virtualized" data center components that are hosted in Amazon data centers under the brand name "Elastic Compute Cloud" (EC2). Originally available only "on demand" on an hourly rental basis, EC2 "instances" (virtual machines) are now available under a variety of fixed-price rental options [6]. These fixed-price rentals all share a common Service Level Agreement (SLA) which guarantees a minimum of 99% availability over the period of a month. Instances are instantiated from "instance types," which describe the CPU, memory, and external storage capabilities associated with a virtual machine.

Amazon uses its data centers to host instances, these datacenters do not share infrastructure other than the Simple Storage Service (S3) [7]. Instead, AWS is organized as "regions" each of which operate as an independent service venue. Regions are further subdivided into "availability zones" (AZs), each of which corresponds (roughly) to a single datacenter building.

In 2009, Amazon introduced spot instances as a lower-cost, dynamically priced alternative to its fixed-price EC2 offerings. The original spot-instance product offering advertised that prices were set dynamically in separate "spot markets" based on instantaneous supply and demand. Specifically, each user would enter a sealed "bid" indicating the maximum hourly charge she would agree to incur for an instance in a specific AZ. While on-demand prices are quoted for each region, for spot instances, each AZ within each region formed a separate market. The region-wide spot price published by Amazon was set to the minimum for that instance type across all AZs.

Periodically Amazon would sort the bids in each AZ in descending order of bid value and assign resources to bidders in that order until all resources were exhausted. The bidders who received resources all paid the lowest bid price among those bids that could be satisfied. The remaining bidders either received no resources or had their resources terminated and then transferred to a higher bidder. Amazon did not publish the available supply but did make available (in real time) the current market-clearing price for each instance type in each AZ.

The conception for spot instances was as a way for Amazon to monetize its idle instance capacity. Thus, price fluctuations were caused by two separate factors: newly arriving bids and unannounced changes in idle capacity. Subsequent research [1] hypothesized a third factor, namely the introduction of a hidden reserve price to help obfuscate the supply and demand signals in the market. However, from a user perspective, the "spot market," as it was known colloquially, offered significantly cheaper hourly rental rates for instances (compared to fixed-price instances), with the proviso that these instances could be terminated at any time due to a change in supply or demand.

At the end of November, 2017 Amazon announced a change from this "market-based" price-setting mechanism to one that uses a dynamically changing retail price set by Amazon that is not based on market inputs [4]. Originally, the product announcement indicated that the change simply smoothed the visible price histories to make spot instances more attractive to users. That is, because demand (and more probably supply) could change suddenly and substantially, users viewing real-time pricing data or price histories could be surprised by sharp changes in price.

However, subsequent reports by heavy spot-instance users indicate that Amazon also changed the way in which it determines which instances to terminate when there is a supply shortfall [18]. Instead of basing the termination decision on sealed bids from users, Amazon switched to using a hidden internal algorithm that may or may not take bid value (now called "maximum value") into account.

In this paper, we quantify how the retail prices set by Amazon after the switch to the dynamic retail pricing mechanism, compare to the prices that were set before the switch in response to sealed bids and internal idle capacity (henceforth termed the "market-based" mechanism). This change is also significant because it alters the utility of the spot instance product substantially independent of price. Specifically, because the termination algorithm for the retail pricing mechanism is not public [8] and is based on hidden parameters, terminations cannot be predicted in the way they once were [21]. Thus, the switch to retail pricing changes both the price of spot instances and their reliability characteristics.

III. Methodology

We compare spot-instance prices in terms of the expected price per hour of usage. For any given hour h, we compute the expected hourly price $E(price)_h$ as

$$E(price)_h = \sum_{i=1}^{D_h} P_i \times d_i, \quad (1)$$

where D_h is the number of prices listed during hour h, P_i is the i^{th} price listed in the hour, and d_i is the fraction of an hour for which price P_i persisted.

Intuitively, $E(price)_h$ is the expected cost a user arriving at a random time during hour h would expect to incur for that hour of usage. However, under the previous market-driven pricing mechanism, it appears that Amazon would sometimes set a reserve price for some instance types that was so high that no instances of that type would run. For example, consider the *left* time series of prices for the *c4.8xlarge* instance type in the *us-west-1b* AZ shown in Figure 1. The *left* time series shows the spot-instance price history at 5 minute intervals from 6:00 AM until 6:50 PM Pacific time on January 26, 2016. The x-axis is time and the y-axis price in US dollars.

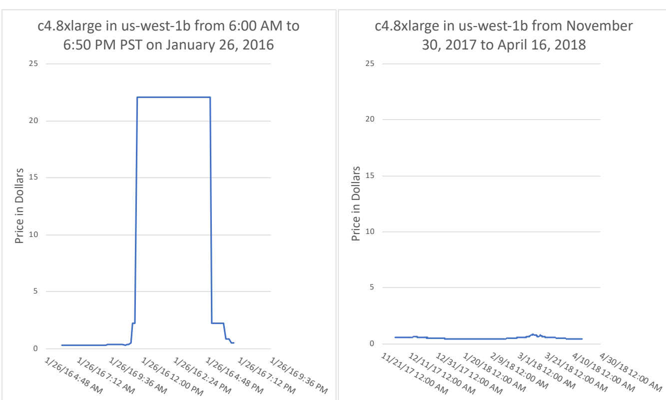

Fig. 1: Both figures show price time series data for the same *c4.8xlarge* instance type in *us-west-1b*. On the *left* we present a short window from 6:00 AM PST 6:50 PM PST on January 26, 2016 before the switch to retail pricing. On the *right* we show the window after the switch to retail pricing from November 30, 2017 and ends at approximately 4:00 PM PST on April 16, 2018

Note that the instance is available for prices fluctuating under $1.00 until 10:10 when the price spikes to 2.028 and then again to $22.08 and remains there with no fluctuation for approximately 6 hours. While conceivable that some customer of AWS bidding for a *C4.8xlarge* spot-instance in *us-west-1b* would be willing to pay $2.028, it is unlikely that any user would pay an hourly rate of $22.80 and that the price would not fluctuate for such a long duration. The on-demand price (at the time of this writing) is $1.19 for this instance. Thus, while a bid of $2.028 might be reasonable in order to prevent an early termination of a spot instance, we (like the authors of [1]) speculate that the $22.08 price is a hidden reserve price. That is, it is a price that Amazon sets to prevent instances from running.

A. Removing Hidden Reserve Prices

Our goal is to have $E(price)_h$ capture the prices that a rational customer would pay. Thus, to compute $E(price)_h$ we first remove hidden reserve prices. To do so, we use a one-dimensional clustering technique based on the Generalized Likelihood Ratio Test (GLRT) [20]. Our implementation of GLRT starts with a sorted list of prices, which it then tries to divide into two sorted lists based on likelihood calculated under the assumption that the original list and each of the two sub-lists is drawn from a pre-specified family of distributions. Once the best split is determined, both of the lists are added to a sorted queue of clusters based on the likelihood associated with the cluster. The process repeats considering subdivision of those clusters with the lowest likelihood statistics until some maximum number of clusters is determined (we use a maximum of 30 in this study). For each clustering it considers, the algorithm computes a Bayesian Information Criterion (BIC) [17] score. It then chooses the clustering among all of those considered with the largest BIC score. For the likelihood computations, as a matter of numerical expediency, we use exponential distributions. We find that the results do not differ substantially from those for which we use normal distributions.

To filter suspected hidden reserve prices, we then remove all of the values in the cluster determined by GLRT that contains the maximum value from the series. From the remaining values, we compute the mean and standard deviation. If the maximum value is more than 5 standard deviations above the mean of the values with this high cluster removed, we remove the high cluster from the series under the assumption that it contains hidden reserve prices.

By way of comparison, the *right* time series of prices in Figure 1 shows the price history for the same *c4.8xlarge* spot-instance type in *us-west-1b* for a five-month period immediately following Amazon's switch to retail pricing, and at no time during the period does the price "spike" to a large fixed value. Indeed, for the retail pricing mechanism, we assume there are no hidden reserve prices.

IV. Results

In this study, we use spot instance price history data [2] that we have been gathering from Amazon since January of 2015 as part of the Aristotle federated cloud project [11]. The data is for the "Linux/UNIX" product only in the North American regions (*us-east* and *us-west*).

To determine the degree of change resulting from the switch from a market-based mechanism to a retail pricing mechanism, we compare the average $E(price)_h$ values for the 90 days immediately preceding the switch observed in November 2018 to the 90 days following the switch. We omit the month of November because it appears November was a period of transition in which some instance-AZ combinations were using the retail mechanism while others remained on the market-based mechanism. Specifically, the market-based mechanism

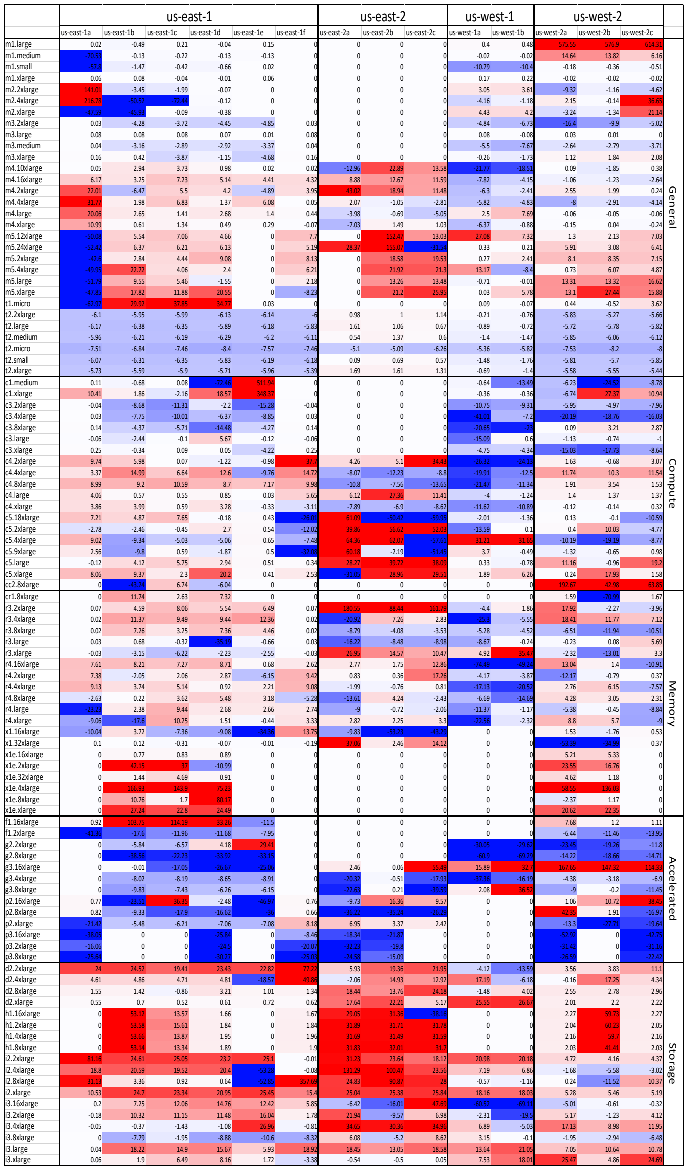

Fig. 2: Percentage price changes for the 90-day period immediately following the switch to retail pricing versus the interim 90-day period beginning on April 2nd, 2018 and ending on July 1, 2018. Blue tones indicate a decrease in price and red tones indicate an increase. Columns correspond to AZs and rows to instance type.

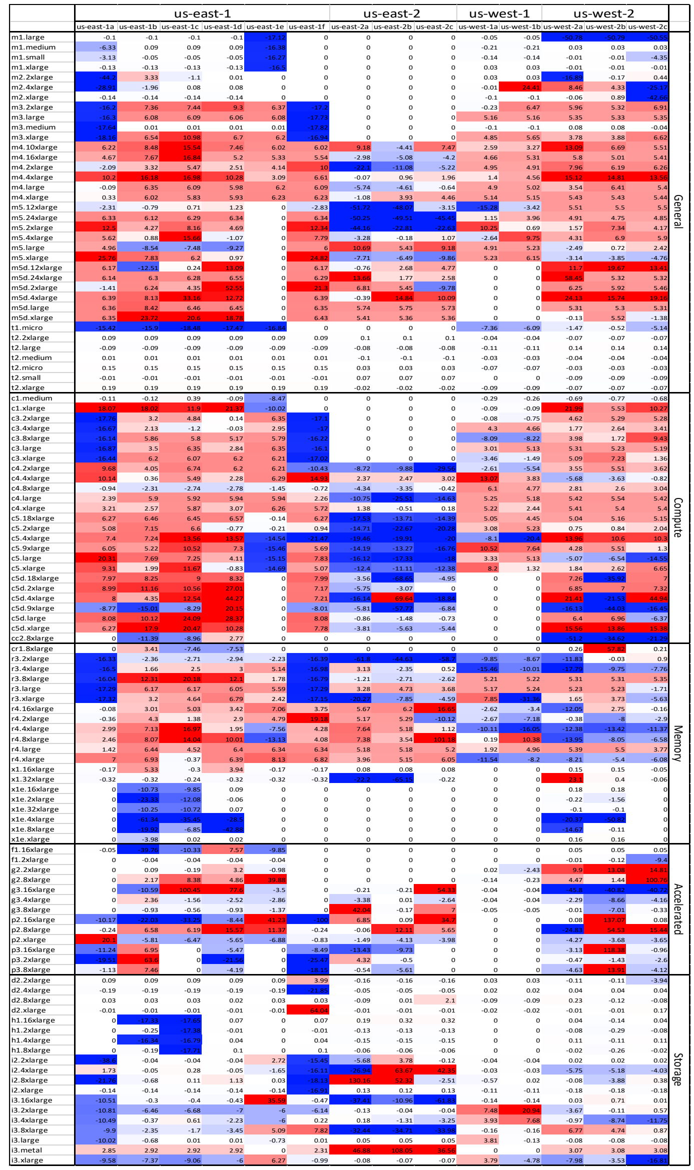

Fig. 3: Percentage price changes for the 90-day period beginning May 1, 2018 and ending on August 1, 2018 compared the the 90-day period beginning August 1, 2018 and ending October 15, 2018. Blue tones indicate a decrease in price and red tones indicate an increase. Columns correspond to AZs and rows to instance type.

period extends from August 2nd, 2017 to October 31st, 2017 and the retail period extends from November 30, 2017 to Febuary 28, 2018.

Table I shows a comparison of average $E(price)_h$ values for the North American regions before and after the pricing mechanism change. The change in price is computed as the difference between $E(price)_h$ after the change minus $E(price)_h$ before the change. Thus, a positive value indicates a price increase and a negative value a decrease. Also, we compute the percentage change as the price change divided by $E(price)_h$ (multiplied by 100) *for each instance type* and then take the average over all instance types in a region to

AWS Region	Price Change	Percentage Change
us-east-1	-1.74	37.28%
us-east-2	-4.42	43.49%
us-west-1	0.38	61.00%
us-west-2	-1.29	59.47%

TABLE I: Comparison of average $E(price)_h$ values for North American AWS regions before and after the pricing mechanism switch. The units of price change are US dollars. Positive values indicate an increase and negative values a decrease.

Region	Quantile				
	0.1	0.25	0.5	0.75	0.9
us-east-1	-89.99%	-29.54%	2.94%	25.11%	84.74%
us-east-2	-25.34%	-1.26%	30.07%	99.07%	132.98%
us-west-1	-29.51%	-6.44%	20.45%	84.48%	159.36%
us-west-2	-31.90%	-16.42%	8.59%	32.95%	122.05%

TABLE II: Quantiles from distribution of average $E(price)_h$ value percentage change for North American AWS regions after the pricing mechanism switch.

Category	Region				Avg. Pct Change
	us-east-1	*us-east-2*	*us-west-1*	*us-west-2*	
General Purpose	-26.03%	21.29%	-9.92%	-4.16%	-4.71%
Compute Optimized	-14.75%	14.32%	28.22%	-10.57%	4.30%
Memory Optimized	1.64%	17.64%	25.85%	17.03%	15.54%
Accelerated Compute	72.94%	32.51%	108.61%	42.30%	64.09%
Storage Optimized	23.52%	91.89%	98.17%	48.62%	65.55%

TABLE III: Comparison of average percentage change in $E(price)_h$ values for North American AWS instance type category for each region after the pricing mechanism switch. The last column is the average percentage change across all regions.

obtain the average overall percentage change.

Note that in *us-east-2* and *us-west-2* the average $E(price)_h$ value decreased, but in all regions the average percentage change is substantially positive. That is, some expensive instance types became cheaper in absolute terms, but overall the average price change is an increase of between 37% and 61%. To effect a retail pricing curve that is smoother than the market-driven curve, Amazon raised prices substantially.

Table II shows the 0.1, 0.25, 0.5, 0.75, and 0.9 quantiles for the percentage change in average $E(price)_h$ value broken out by region. Note that the median change for all regions is positive, with increases ranging from approximately 3% for *us-east-1* to 30% for *us-east-2*. Table III compares average $E(price)_h$ values by category for prices immediately before and after the November switch to retail pricing

When performing this analysis we frequently found that instances would occasionally spike to high prices that no reasonable buyer would pay. For example, *x1e.32xlarge* which was, for some time periods, unavailable on the spot market before the price change and thus was priced at approximately $245.00 which we interpret as a hidden reserve price [1], yet it would occasionally become available at prices as low as $1.5 suggesting that this was the real price for which it was available. To avoid skewing our results due to hidden reserve prices, we employ GLRT clustering to remove them.

A. Price Change Since the Switch to Retail Pricing

We also examine the changes in price for spot instances since the switch to retail pricing. To do so, we first compare the 90-day period immediately following the switch (beginning November 30, 2017 and ending Feb 28, 2018) to the next most recent 90-day period (beginning April 2nd, 2018 and ending July 1, 2018) which we will call the interim period. Finally, we compare the most recent price data available to us – the 90 day periods on either side of Aug 1, 2018 (May 1, 2018 through Aug 1, 2018 compared with Aug 1, 2018 through Oct 15, 2018) to confirm that the trends we have so far observed continue to hold true.

Category	Region				Avg. Pct Change
	us-east-1	*us-east-2*	*us-west-1*	*us-west-2*	
General	-1.13%	11.94%	-1.33%	20.47%	7.49%
Compute	6.46%	9.46%	-6.58%	4.10%	3.36%
Memory	8.19%	9.94%	-10.91%	2.66%	2.47%
Accelerated	-6.00%	-10.14%	-15.62%	2.22%	-7.38%
Storage	13.32%	22.85%	1.84%	7.83%	11.46%

TABLE IV: Comparison of average percentage change in $E(price)_h$ values for each AWS instance category for the 90-day period immediately following the switch to retail pricing verses the interim 90-day perid. The last column is the average percentage change across all regions

Category	Quantile				
	0.1	0.25	0.5	0.75	0.9
General	-6.73%	-5.10%	0.00%	3.95%	13.26%
Compute	-14.48%	-6.37%	0.07%	5.10%	18.57%
Memory	-10.99%	-2.55%	1.51%	7.27%	18.41%
Accelerated	-33.92%	-22.42%	-9.82%	0.76%	32.70%
Storage	-2.94%	0.62%	6.89%	20.95%	31.59%

TABLE V: Quantiles from distribution of average $E(price)_h$ value percentage change for instance categories in all North American regions regions for the 90-day period immediately following the switch to retail pricing versus the interim 90-day period beginning on April 2nd, 2018 and ending on July 1, 2018.

Table IV shows the average percentage change in price for each instance category broken out by AWS region. These numbers indicate an increase in *average* price for all instance categories except “Accelerated Compute.” However, the quantiles for the distributions of average change shown in Table V indicate that the *median* prices, and indeed the middle half of prices, have remained relatively stable for “General Purpose,” “Compute Optimized,” and “Memory Optimized”; in terms of these statistics, “Accelerated Compute” shows a marked decrease and “Storage Optimized” a pronounced increase.

Note that the quantiles on either side of the median are more or less balanced for the first three instance categories. Figure 2 shows the distribution of percentage change in prices when comparing the 90-day period immediately following the switch to retail pricing to the interim 90-day period.

Following the switch to retail pricing, Amazon generally raised prices modestly with a few exceptions. The *t2* instances types in the *us-east-1* and *us-west-2* regions show a reduction in price as do most of the instance types in the “Accelerated Computing” category. In contrast, on the whole, the “Storage Optimized” instance types became more expensive across all regions.

Figure 3 shows the most recent distribution of percentage change in prices. It compares the 90-day periods before and after Aug 1, 2018. Particularly, May 1, 2018 through Aug 1, 2018 and Aug 1, 2018 through Oct 15, 2018. This reflects the most recent price data.

As we discussed previously, Table III shows substantial increases in price for the “Memory Optimized”, “Accelerated Compute”, and “Storage Optimized” instance types. The heatmap in this figure however, now shows substantial price increases to to the “General” and “Compute” categories as well as some increase in the “Memory” category. Thus, all instance categories have experienced some price increase. For the previously mentioned categories, the price increase occurred immediately after the switch, while the “General” and “Compute” categories appear to have lagged behind but ultimately have also risen in price.

V. Related Work

Amazon discloses recent spot price traces but provides only limited information [3] about how spot prices are determined. Furthermore, though much work has been done to understand the characteristics of EC2 spot-instance pricing, this price algorithm is subject to immediate change as Amazon demonstrated in November 2017 [9] when it announced a new pricing model promising smoother market prices. Unfortunately, much of the existing analysis on spot prices is now no longer relevant to the new reduced-volatility spot market.

Based on analysis prior to the November 2017 pricing update, the authors of [1] found that spot price changes under the ‘market driven’ mechanism were not entirely market-driven but rather are generated at random from a dynamic hidden reserve price mechanism meaning that bids below an undisclosed and changing reserve price are ignored. In addition, they found that some observed price changes were artificial as opposed to resulting from user bids.

However, it is important to note that this does not mean one should assume that Amazon’s reserve price model is designed to maximize profit, rather the authors of [14] find that Amazon appears to set prices to maximize utilization of their cloud capacity, furthermore they prove that Amazon’s algorithm does not appear to be profit maximizing.

With the November 2017 update, Amazon introduced price smoothing to avoid sudden spikes in price, but it also changed its termination policy such that in addition to termination due to the market price exceeding the user’s bid, an instance may also be terminated when “demand for Spot Instances increases” [8] or “supply of Spot Instances decreases” [8]. This means that instances with bid prices above the market price may be terminated. Amazon seemingly addresses this by providing a new termination notice API [10], perhaps to allow users to checkpoint before shutdown, however it is no longer possible for a user to control their termination through their bid price.

For example, in [21], we use probabilistic techniques such as QBETS [15], [16] to set bid prices in an attempt to create probabilistic reliability guarantees. Doing so may no longer be possible because according to [8], instances above the current market price are still subject to termination. In [22] and [22], the authors explore various strategies for dynamic checkpointing with goals such as cost optimization by minimizing work lost. Similarly, these techniques may no longer be necessary in the new spotmarket due to the addition of termination notices.

VI. Conclusions

In this work, we study Amazon’s switch to retail pricing for its interruptible “spot instance” class of service. In November of 2017, Amazon changed the mechanism for pricing spot instances from one based on user-submitted “maximum bids” to a retail pricing mechanism in which the retail prices would change more smoothly (versus under the market-based mechanism).

Because the new pricing policy changes the mechanism for determining which spot instances will be terminated during a resource shortfall, and that algorithm is private to Amazon, the switch to retail pricing also changes the reliability of spot instances. Specifically, spot instance reliability is degraded since this new mechanism excludes the use of prediction techniques such as those described in [21] that achieve predictable reliability. This reduction in predictable reliability through bid price would seem to militate for a decrease in spot-instance price.

On the other hand, price smoothing might require an increase in price so that Amazon can cover sudden supply shortfalls that arise from changes in internal demand for the machine hosting spot instances. In addition, users might also find spot instances with smoother price histories more attractive, thereby increasing demand.

In this work, we analyze spot instance price histories for the North American regions immediately before and at various times after the switch to retail pricing. We find that the switch to retail pricing (and a decrease in reliability) was generally accompanied by a price increase except for “General Purpose” instances types in the *us-east-1* region.

More generally, this study demonstrates a challenge faced by users of virtualized data center infrastructure products. The change in pricing mechanism was accompanied by an unannounced set of price increases and reductions in reliability. If the infrastructure had been purchased and cited in a data center, the data center operators would have been able to inform users of upcoming changes so that users could have made appropriate preparations. The spot instance class of service is assuredly Amazon’s least expensive class of service, which might argue for immediate and unannounced changes in the product’s function and price. However, for cloud computing use cases (such as scientific research) where fixed-budget accounting is the norm, such changes represent a challenge to adoption.

References

[1] Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., and Tsafrir, D. Deconstructing amazon ec2 spot instance pricing. *ACM Transactions on Economics and Computation 1*, 3 (2013), 16.

[2] Amazon Web Services. AWS EC2 Spot Instance Price Histories. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances-history.html Accessed Jul-2018.

[3] Amazon Web Services. AWS EC2 Spot Instance Prices. https://aws.amazon.com/ec2/spot/pricing/ Accessed Jul-2018.

[4] Amazon Web Services. New AWS EC2 Spot Instance Pricing. https://aws.amazon.com/about-aws/whats-new/2017/11/\\amazon-ec2-spot-introduces-new-pricing-model-and-the-ability-to-launch-new-spot-instances-via-runinstances-api/ Accessed Jul-2018.

[5] AMAZON WEB SERVICES. Amazon ec2 spot instances, 2018. https://aws.amazon.com/ec2/spot/ accessed July 2018.

[6] AMAZON WEB SERVICES. Amazon instance pricing, 2018. https://aws.amazon.com/ec2/pricing/on-demand/ accessed July 2018.

[7] AMAZON WEB SERVICES. Amazon simple storage service, 2018. https://aws.amazon.com/s3 accessed July 2018.

[8] AMAZON WEB SERVICES. How spot instances work, July 2018. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/how-spot-instances-work.html accessed July 2018.

[9] AMAZON WEB SERVICES. New amazon ec2 spot pricing model, July 2018. https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing/ accessed July 2018.

[10] AMAZON WEB SERVICES. New âĂŞ ec2 spot instance termination notices, July 2018. https://aws.amazon.com/blogs/aws/new-ec2-spot-instance-termination-notices/ accessed July 2018.

[11] Aristotle Cloud Federation. https://federatedcloud.org [Online; accessed Aug-2018].

[12] GOOGLE CLOUD PLATFORM. Google preemptable virtual machines, July 2018. https://cloud.google.com/preemptible-vms/ accessed July 2018.

[13] JAVADI, B., THULASIRAM, R. K., AND BUYYA, R. Characterizing spot price dynamics in public cloud environments. *Future Generation Computer Systems 29*, 4 (2013), 988–999.

[14] KHODAK, M., ZHENG, L., LAN, A. S., JOE-WONG, C., AND CHIANG, M. Learning cloud dynamics to optimize spot instance bidding strategies.

[15] NURMI, D., BREVIK, J., AND WOLSKI, R. Qbets: Queue bounds estimation from time series. In *Job Scheduling Strategies for Parallel Processing* (2008), Springer, pp. 76–101.

[16] NURMI, D., WOLSKI, R., AND BREVIK, J. Probabilistic advanced reservations for batch-scheduled parallel machines. In *Proceedings of the 13th ACM SIGPLAN symposium on principles and practice of parallel programming* (2008), ACM, pp. 289–290.

[17] SCHWARZ, G. Estimating the dimension of a model. *Annals of Statistics 6*, 2 (1978).

[18] STEVE FOX. New aws spot pricing model: The good, the bad, and the ugly, July 2018. http://autoscalr.com/2018/01/04/new-aws-spot-pricing-model-good-bad-ugly/ accessed July 2018.

[19] WALLACE, R. M., TURCHENKO, V., SHEIKHALISHAHI, M., TURCHENKO, I., SHULTS, V., VAZQUEZ-POLETTI, J. L., AND GRANDINETTI, L. Applications of neural-based spot market prediction for cloud computing. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on* (2013), vol. 2, IEEE, pp. 710–716.

[20] WILLSKY, A. S., AND JONES, H. L. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control 21*, 1 (1976).

[21] WOLSKI, R., BREVIK, J., CHARD, R., AND CHARD, K. Probabilistic guarantees of execution duration for amazon spot instances. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (2017), ACM, p. 18.

[22] YI, S., KONDO, D., AND ANDRZEJAK, A. Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud. In *2010 IEEE 3rd International Conference on Cloud Computing* (July 2010), pp. 236–243.