Deconstructing the 2017 Changes to AWS Spot Market Pricing

Conference Paper · June 2019

DOI: 10.1145/3322795.3331465

Matthew Baughman
University of Chicago

Christian Haas
University of Nebraska at Omaha

Ryan Chard
Argonne National Laboratory

Kyle Chard
University of Chicago

Deconstructing the 2017 Changes to AWS Spot Market Pricing

Matt Baughman
Minerva Schools at KGI
San Francisco, California

Simon Caton
University College Dublin
Dublin, Ireland

Christian Haas
University of Nebraska at Omaha
Omaha, Nebraska

Ryan Chard
Argonne National Laboratory
Argonne, Illinois

Rich Wolski
University of California, Santa Barbara
Santa Barbara, California

Ian Foster
Argonne Nat Lab & U. Chicago
Chicago, Illinois

Kyle Chard
University of Chicago
Chicago, Illinois

ABSTRACT

The Amazon Web Services spot market sells excess computing capacity at a reduced price and with reduced reliability guarantees. The low cost nature of the spot market has led to widespread adoption in industry and science. However, one of the challenges with using the spot market is that it is intentionally opaque and thus users have little understanding of the underlying dynamics. In late 2017, the mechanisms underlying the spot market were significantly altered—no longer are bid prices used to clear capacity and as a result the pricing is much less volatile. In this paper, we revisit prior work with the aim to analyze the differences in market dynamics between the pre-change and post-change spot instance market. We then use these analyses to highlight possible properties of the current and previous pricing algorithms, including artificial manipulation, dynamic algorithm adjustment, and persistent trends in market supply, demand, and pricing.

ACM Reference Format:
Matt Baughman, Simon Caton, Christian Haas, Ryan Chard, Rich Wolski, Ian Foster, and Kyle Chard. 2019. Deconstructing the 2017 Changes to AWS Spot Market Pricing. In *10th Workshop on Scientific Cloud Computing (ScienceCloud'19), June 25, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3322795.3331465

1 INTRODUCTION

Since its introduction in 2011, Amazon Web Services (AWS) has sold unused Elastic Compute Cloud (EC2) capacity through a preemptable spot market. The principles behind the spot market are simple—rather than paying a fixed price for guaranteed availability, users instead pay for compute instances at a price that is determined by the market, with the caveat that AWS may reclaim those instances at any time, and with little warning. The result is that compute instances may be obtained at a fraction of the price of their on-demand alternatives.

Prior to 2017, spot instance prices were determined solely by market dynamics: that is, by the interplay between supply and demand. Potential consumers would bid a maximum willingness to pay, and the spot price was calculated as the maximum price at which there were just enough bids to successfully clear the market (consume all excess compute capacity). Compute instances were then delivered at that price to any consumer who had bid at or above that price. Likewise, terminations occurred when a specific consumer's bid price was exceeded by the present spot price. Note that the market dynamics were also effected by the constant variation in supply, due to on-demand provisioning, which takes precedence over spot instance provisioning.

In late 2017, AWS overhauled the way spot instances are priced, provisioned, and terminated, with the goals of decreasing price variability, increasing spot instance durability, and regularizing the market [2]. Their new (undisclosed) algorithm updates spot prices over time based on long-term market pressures. Under the new scheme, the user sets a maximum price for a desired instance type in a specified availability zone. Depending on the arbitration of AWS's algorithm, a spot instance is then provisioned if the user's stated maximum exceeds the market price. AWS may then change the market price over time, and users pay the current market price. If the market price exceeds a user's maximum price, the instance will be terminated. AWS may also terminate instances at any time, irrespective of the market price and user bids.

As a result of the 2017 change, the extensive prior work on spot market dynamics [3, 13, 14, 22, 23] is no longer applicable. Thus we undertake here a before-after analysis in which we compare the volatility, price, and dynamics of the spot market in order to determine the effects of the pricing change on the market and with respect to users. We show that the new market is significantly more stable with far fewer price events and longer times between price events. We also show that the seasonal nature of the spot market has changed with events more uniformly distributed throughout the week. Finally, we show that the new system exhibits longer price durations at the expense of providing extremely low prices for short durations. The result of which is that the changes likely benefit users with longer running tasks.

The remainder of this paper is structured as follows: Section 2 describes the data used for analysis. In Section 3, we explore changes with respect to price event-level data and compare the availability

ScienceCloud'19, June 25, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6758-5/19/06…$15.00
https://doi.org/10.1145/3322795.3331465

and duration of spot prices before and after the pricing change. In Section 4, we present our methodology to evaluate the practical effects of the change and discuss what the results of this method mean for spot instance users. In Section 5, we present related work, and in Section 6 we summarize our contributions.

2 SPOT HISTORY DATA

We begin by presenting a summary of the spot history data that we use for our analyses, and describe general statistics of those data.

AWS continuously makes available 90 days of historical spot price data, broken down by operating system, region, availability zone, and instance type. This data are *event-level*, meaning that a price is only recorded when it changes. One benefit of this level of granularity is that it allows us to examine the volatility of the spot market through the lens of the pricing algorithm.

We have aggregated nearly three years of pricing history leading up to the pricing-scheme change and nearly one year of data produced thereafter. In order to focus on the transition period and to minimize irregularities due to the addition of new instance types we restrict our analysis to the three months preceding the change (Jul–Sep 2017), the three months during which the change occurs (Oct–Dec 2017), and three months thereafter (Jan–Mar 2018). At the point of extraction and storage, data have not been altered. However, throughout our analyses, we undertake a number of feature engineering exercises to facilitate analysis.

We first provide an overview of our data and summarize important statistical qualities. Table 1 shows the number of price events that correspond to a specific feature of the data. In total, we analyze 30.4 million price change events over the 9 month period of interest.

Geography: Our data set comprises four regions in the US: *us-east-1*, *us-east-2*, *us-west-1*, and *us-west-2*. Each is further divided into a number of availability zones denoted with a letter. In total, this gives 14 distinct *availability zones* to which an instance can belong, as shown in Table 1a.

Instance Types: AWS offers many different virtual machine instance types, each with a certain resource capacity, and each typically named by a one- or two-letter *family*, a *generation*, and a *size* (amount of resources) [1], For example, `m5.xlarge` is a m-family (`m`), fifth-generation (`5`), extra-large (`xlarge`) instance type. The family indicates the general class of application for which the instance is intended; the generation is updated as new hardware replaces old; and the size the number of cores, amount of memory, etc. Our data capture 15 families (see Table 1b), six generations (see Table 1c), and 11 sizes (see Table 1d) for a total of 102 unique instance types during the nine-month period studied. AWS users can also select different operating systems, but we consider only Linux/UNIX instance types here as other operating systems are not well represented in the data.

Price: All prices are captured in US$ charged per hour of availability. The empirical quantiles of price for the 9 months of data across all instance types and geographies are (rounded to four decimal places places) 0% (min) 0.0000, 25% 0.0938, 50% (median) 0.2103, 75% 0.4103, and 100% (max) 266.8800, with a mean of 0.5572 and standard deviation 4.8471. The extent of detail here indicates that AWS have extremely finely grained pricing models that operate to the fraction of a US$ cent. The price of an instance also follows an exponential distribution, which is to be expected—there are many low prices, and a small number of high prices.

Time: All price events are date-time stamped and represented in UTC to the nearest second. To facilitate data analysis, we also added several features, notably the time between events, segmented by the tuple [*zone*, *type*], and the price difference between events. We also decomposed the date-stamp into its constituent parts (day, hour, min, etc.)

3 SPOT MARKET ANALYSIS

We now present a before-after analysis of the spot market. We begin by summarizing general statistics including the mean price and variability, we then analyze the frequency of event changes, before reviewing changes to market dynamics.

3.1 General Statistics

The primary change, both stated by AWS and evident in the spot pricing, is the reduction of variability in the pricing data. Overall, across our 841 instance type and availability zone combinations, the average instance is 11.55% cheaper than it was before the change, though the deltas vary wildly between instance types. While this increase in mean price is significant, the reduction of spot price variability represents a much much more noticeable change. Before the change, the mean standard deviation was 87.9% of the mean spot price; following the change, the mean standard deviation was just 7.9% of the mean spot price. The most notable difference is related to the duration of a given spot price: after the change, spot price persisted, on average, more than 85x longer than they did before the change. This increase is consistent with AWS's claim that spot pricing is more consistent and that the price is regularized to fit more long term trends.

3.2 Event Frequency and Period

The number of price updates and thus price events drops dramatically following the switch to the new policy, as shown in Figure 1. The spike in late November suggests an amount of reorganization at that time, perhaps as AWS further tuned its pricing algorithm.

Some regions have significantly more price updates than others. This is true for both different instance types and families. Therefore, the update to the AWS pricing mechanism appears to increase the time to algorithmically "consider" a price change. Figure 2 illustrates how the duration between events has changed over time. For ease of presentation, time between events (which follows an exponential distribution) has been log transformed. The distribution of intervals shows that there are many small intervals and a small number of extremely large intervals. The extent of these extreme outliers (right tails of the distribution) can be seen in Figures 2a and 2b. Also notable is that the frequency of events (y-axis) is much lower in Figure 2c. Two-sample Kolmogorov-Smirnov tests are used to check if the time between events follow similar distributions, i.e. if price changes occur similarly before and after the change in pricing mechanism. Table 2a indicates that the time between events has changed significantly. However, it is worth noting that based on the size of our data, even subtle changes would be considered significant. Instead, we note that Jan-Mar 18 is further away, i.e., more different to Jul–Sep 17 and Oct–Dec 17 than they are from

Table 1: Features of the dataset used in this study, in numbers of price events.

(a) Price events by zone

Zone	No. of events
us-east-1a	1 295 339
us-east-1b	4 097 936
us-east-1c	3 087 629
us-east-1d	2 860 662
us-east-1e	1 886 269
us-east-1f	489 835
us-east-2a	1 822 934
us-east-2b	1 021 140
us-east-2c	910 951
us-west-1a	1 926 182
us-west-1b	1 609 008
us-west-2a	3 090 387
us-west-2b	3 302 754
us-west-2c	3 041 185

(b) Price events by instance family

Grouping	Family	No. of events
General purpose	m	6 602 502
	t	92 211
Compute-optimised	c	8 544 268
	cc	215 020
	cg	124
	cr	48 693
Memory-optimised	r	8 143 617
	x	221 736
Storage-optimised	d	617 218
	h	8999
	hi	20 967
	i	2 439 870
Accelerated computing	f	215 898
	g	1 996 965
	p	1 274 123

(c) Price events by generation

Generation	No. of events
1	1 085 330
1e	14 999
2	5 358 067
3	11 282 104
4	12 394 791
5	306 920

(d) Price events by instance size

Size	No. of events
micro	66 746
small	75 431
medium	408 016
large	1 679 234
xlarge	5 113 461
2xlarge	8 108 718
4xlarge	6 888 600
8xlarge	6 061 439
10xlarge	674 428
16xlarge	1 238 291
32xlarge	56 567

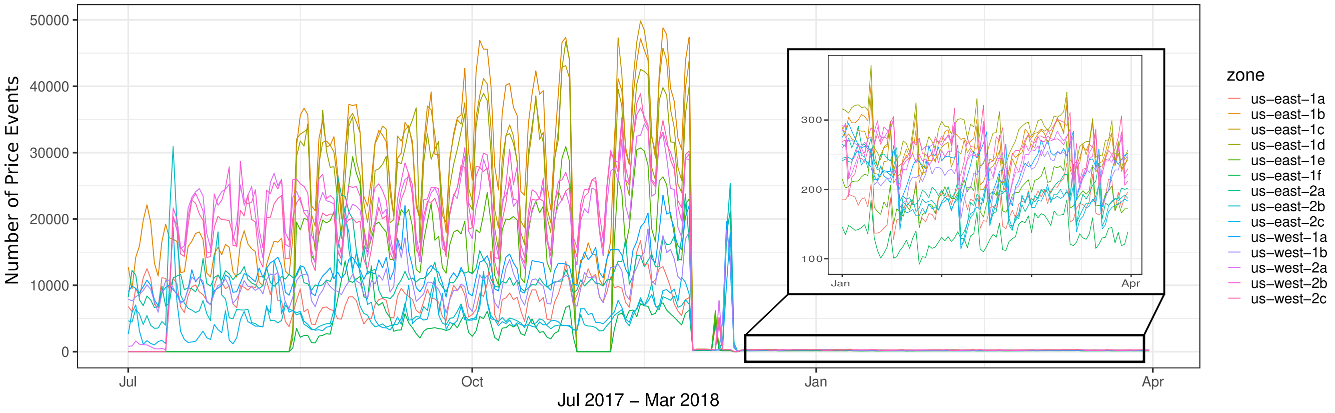
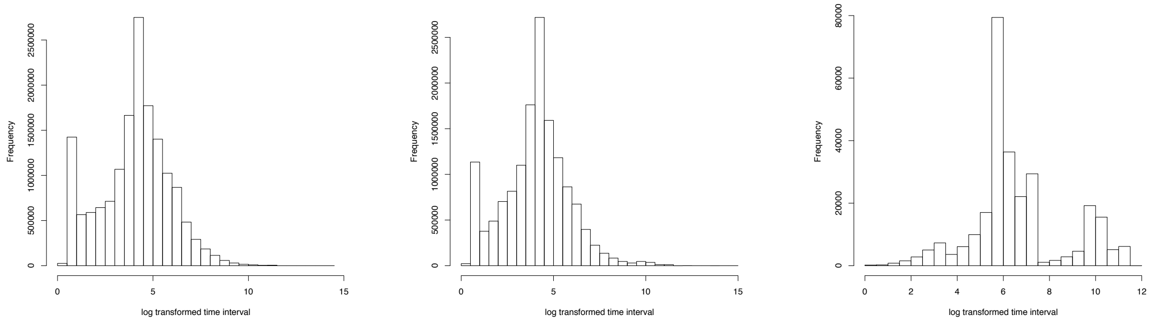


Figure 1: Number of events per day for each region between July 2017 and April 2018 (with Jan–Mar 2018 zoomed in).



(a) Jul–Sep 2017 (b) Oct–Dec 2017 (c) Jan–Mar 2018

Figure 2: Distribution of time between events for each three month period. Here a log transformation is used to spread the frequency distribution of time intervals more uniformly, aiding both visualisation and data interpretation.

each other, as indicated by a higher D statistic. This is corroborated in Figure 2.

The reduction in the number of events prompts the inspection of the effect on price itself. Figure 3 depicts the average daily price (across all instance types, generations, sizes and regions). Here,

Table 2: Kolmogorov-Smirnov tests. *D* is the estimated maximum distance between the two distributions under analysis. Considering a level of significance $\alpha = 0.01$, we reject the null hypothesis of equivalence of the two distributions in all cases.

(a) Kolmogorov-Smirnov tests inspecting the time between events.

	D	p-value
Jul–Sep 17 / Oct-Dec 17	0.03060	< 0.001
Jul–Sep 17 / Jan-Mar 18	0.60879	< 0.001
Oct–Dec 17 / Jan-Mar 18	0.62835	< 0.001

(b) Kolmogorov-Smirnov tests inspecting the change in price between events.

	D	p-value
Jul–Sep 17 / Oct–Dec 17	0.03160	< 0.001
Jul–Sep 17 / Jan–Mar 18	0.26168	< 0.001
Oct–Dec 17 / Jan–Mar 18	0.24311	< 0.001

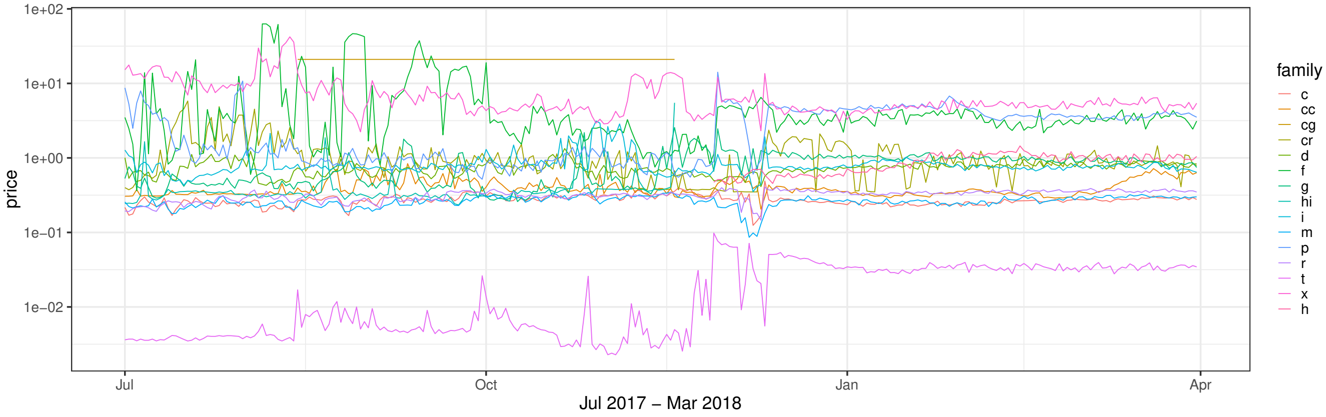


Figure 3: Average daily price by family, in US$, between July 2017 and March 2018.

however, we need to be cautious in the presentation of the data. The price signal has, in general, a high variance, and significant outliers (as already seen) that have high leverage over measures like the mean (plotting the median, however, does not look fundamentally different). We cannot show error bars indicating one standard deviation around the mean as they make Figure 3 unreadable. Similar to the number of events, the mean of the more expensive families has reduced and there is less variance with time. This is to be expected as newer instance families will reduce in price over time.

Similar to the time between events, we see significant differences in the change of price captured by events, see Table 2b, with the same remarks with respect to statistical power, and the difference in distance between the distributions (denoted by the *D* statistic), i.e. the Jan–Mar 2019 distribution is further away (more different).

In summary, the change in pricing mechanism reduced the number of events (Figure 1), increased the time between events (Table 2a and Figure 2), and reduced the degree of price change between events (Table 2b and Table 3). Table 3 also includes statistics after removing occurrences of the artificial spot price cap set by AWS (10x the on-demand price) as these prices may represent situations in which AWS has potentially artificially raised the spot price to terminate running spot instances [10]. In doing so, we eliminate the influence of extreme values and observe reduced overall variation, but still much greater variation than the new pricing model.

We also note that the average change in price represented by an event is negative, which would suggest a general reduction of price over time. Yet more notable are the extremes of price change. The increase in time between events seems to afford AWS the opportunity to be algorithmically less variable with respect to changes in instance costs. This is seen in the reduction in the standard deviation of changes to price between events. As such, it appears that price adjustments now occur less often, are smaller, and have lower variance. These changes indicate a benefit to users requiring instances for prolonged periods of time (reduced variance and increased price durations means more predictable costs), while detrimentally affecting users that opportunistically seek short-term use of cheap instances.

3.3 Event Occurrence

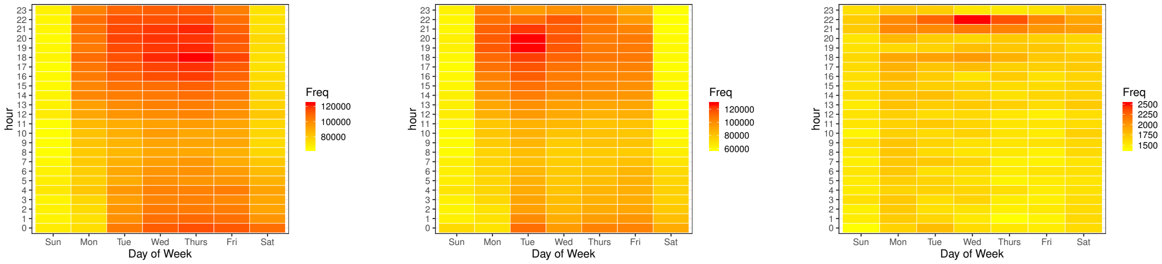
Having observed a change in the number of events, a change in time between events (it increases), and the price difference between events, we can also observe that events occur following different patterns. Figure 4 presents a heat map that corresponds to the frequency of events with respect to the time of day and day of week that events occur. It is evident that there has been a change in the weekly pattern of events. From Jul–Dec 2017 (Figures 4a, 4b) there is a clear seasonality in the event distribution: fewer events occur at weekends, and there is a slight tendency for more events at later hours. Note that UTC is 4 to 8 hours ahead of US time zones.

We observe that in Figure 4c (Jan-Mar 2018) this pattern disappears completely. Aside from a general reduction in events, the distribution of events over the 24 hour period, as well as over the week, appear more normalized. Some of the weekly pattern remains: Sunday appears to have slightly fewer events, and around 10pm UTC midweek also experiences slightly more events.

Figure 5 shows a heatmap of event frequency per day of the week based on the minute of the hour. Here, for readability we have

Table 3: Quantiles, means, and standard deviations of price differences between events, all in US$.

	0% (min)	25%	50% (median)	75%	100% (max)	mean	stdev
Jul–Sep 17	-255.35	-0.0002	0.0001	0.0002	255.1	-0.0000	7.936363
Oct–Dec 17	-262.671	-0.0002	0.0000	0.0002	263.2583	-0.0004	3.2520
Jan–Mar 18	-5.0264	-0.0001	0.0000	0.0001	2.8491	-0.0002	0.0310
Jul–Sep 17 (price caps dropped)	-71.7526	-0.0003	0.0001	0.0004	72.2574	0.0000	0.2622
Oct–Dec 17 (price caps dropped)	-70.1337	-0.0003	0.0001	0.0004	72.1944	0.0000	0.3003
Jan–Mar 18 (price caps dropped)	-5.0264	-0.0001	0.0000	0.0001	2.8491	-0.0002	0.0310

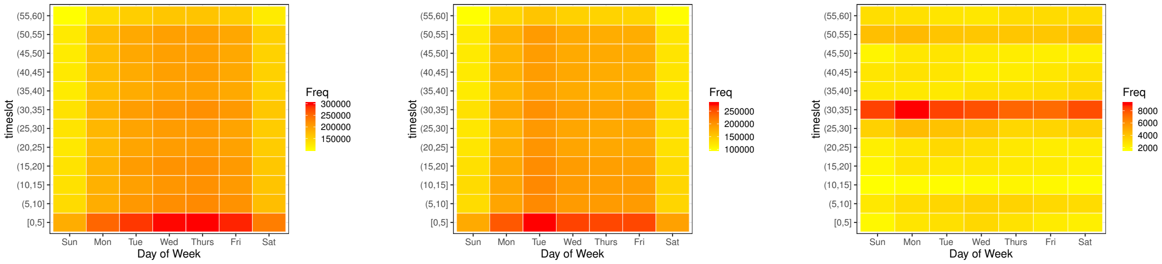


(a) Jul–Sep 2017

(b) Oct–Dec 2017

(c) Jan–Mar 2018

Figure 4: Frequency of event occurrences per time of day by day of week.



(a) Jul-Sep 2017

(b) Oct-Dec 2017

(c) Jan–Mar 2018

Figure 5: Frequency of event occurrences per timeslot (minute of hour) by day of week.

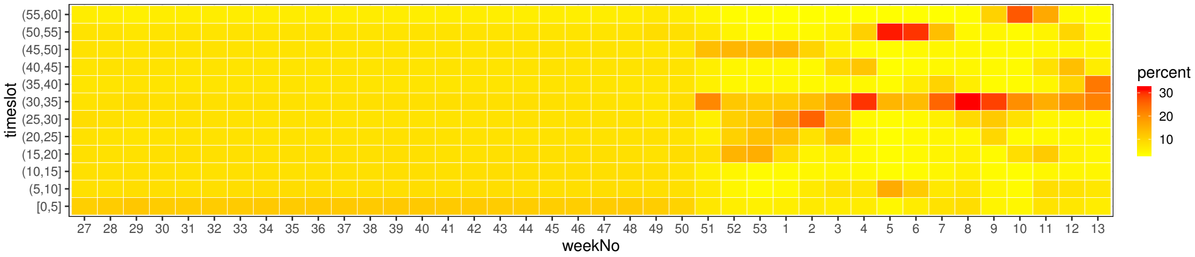


Figure 6: Frequency of event occurrences per calendar week by minute of hour.

binned (discretized) minute into five-minute portions. The figure shows a shift in when events occur. Initially, the majority of events occur at the start of the hour in Jul–Dec 2017 (Figures 5a, 5b), which aligns well with AWS's per-hour billing. The week seasonality is also visible here too. Post change, in Figure 5c (Jan–Mar 2018), the seasonal effect disappears and the majority of events now occur at the half hour.

We need to be a little cautious with the exact interpretation of the heat maps as we are comparing different frequency ranges: there are significantly more events per day in Jul-Dec 2017 than Jan-March 2018. To provide a little context and assist in the interpretation of these figures, Figure 6 presents the frequency of events as a normalized weekly percentage, i.e. each column sums to 100% and each cell represents the percentage of events that occurred in that

5 minute interval for the corresponding week. What we see is that while more events occurred at the start of the hour as a proportion of the total weekly events this is less pronounced. Interesting is that Figure 6 clearly delineates exactly when AWS rolled out the new pricing mechanism in US regions. We also note that the effect of Figure 5c is also visible in Figure 6 but is less pronounced.

3.4 Changes in Market Dynamics

Having explored the changes in event occurrences we now explore how the spot market changes have affected market dynamics.

3.4.1 Spot Prices Over Time. In general, the spot market before the change was highly variable, often exhibiting a high frequency of large price changes. However, the new system seems to have some small sporadic changes but these are much less pronounced and not clearly discernible at the resolution shown in Figure 3.

3.4.2 Availability of Spot Instances. Figure 7 shows the availability of the `r3.4xlarge` instance type across regions and availability zones as the price increases. (Distributions for other instance types exhibit similar trends.) We observe two features that are particularly interesting. The first, which is readily observed in Figure 7 and well expected given AWS's own statements, is the definite upper limits of availability in the new data (which has not been cropped at all). Additionally, the relative linearity seems to indicate the artificial setting of spot prices, as well as defined upper and lower bounds. The second, which appears both before and after the change, is the appearance of multiple discernible inflection points along the availability curve. This could show one of two things: 1) AWS has imposed pressure on pricing or hard upper and lower price bounds for both the old and new pricing schemes, or 2) bids are generally made near round pricing figures, so this would indicate the new spot pricing is still influenced by bid pricing (indirectly by way of demand at a given price).

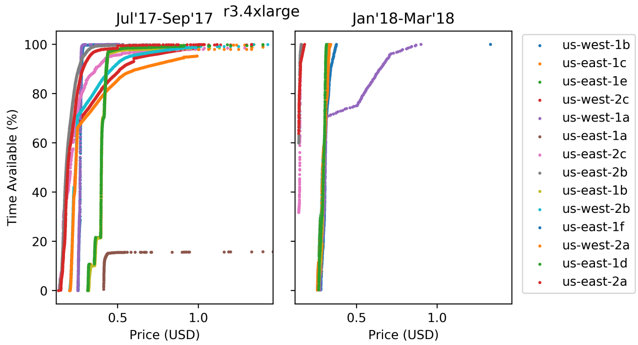


Figure 7: Cumulative availability of the `r3.4xlarge` instance type vs. bid price for both the old and new pricing schemes.

3.4.3 Spot Price Duration. Figure 8 shows the cumulative price duration—the time that a bid price remains above the spot price. The primary observation is the orders-of-magnitude difference between price duration under the old versus new pricing schemes. However, more interesting is the prevalence of "stair-stepping" that we observe where the duration jumps at given points, while still growing in between. This possibly indicates that AWS adjusts (both before and after the scheme change) the spot price both at regular intervals and when demand exceeds a certain threshold. We can also identify that, at least this specific instance type, `us-east-1f` has a cumulative duration curve nearly identical both pre- and post-change. Finally, we see a more logistic shape in the data, whereas the new data (at least for shorter durations) exhibits much more linear or even exponential behavior.

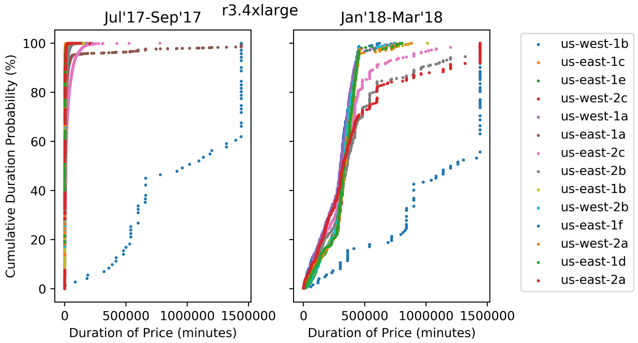


Figure 8: Cumulative duration of the `r3.4xlarge` instance type for both the old and new pricing schemes.

4 PRACTICAL IMPLICATIONS OF CHANGES

We next investigate the implications of the spot pricing change for end users. To this end, we use historical spot pricing data to simulate bidding scenarios, and investigate the outcomes of those scenarios under the old and new pricing schemes.

4.1 Methodology

For each availability zone-instance type combination, we calculated the expected cost of an arbitrary workload that would take X seconds to run on that specific instance type. To do this, we measured, for each possible bid price, the distribution of durations given that specific max bid. If all occurrences of that given max bid are durable for longer than the simulated workload duration, the expected cost is simply the duration-weighted mean spot price for the simulated duration of the workload. If no durations possess the necessary durability (i.e., if the spot instance does not stay under the simulated max bid price for the entire duration of the workload at any point in the time series), then we record this as a "did not finish," automatically forcing a higher maximum bid and, therefore, higher expected cost.

In most cases, we obtain duration distributions where the simulated workload would finish *sometimes* but not *every time*. For example, if we consider a bid of $0.20 and a simulated 6-hour workload. We observe that half of our sampled durations last for more than six hours and half for less than six hours (i.e., somewhere in those six hours the spot price exceeds the bid price). In this case we calculate the expected cost as two times the duration-weighted mean of the time-series within those windows. This is because, on average, if we have a 50–50 chance of instance durability, then we will have to run the workload an equivalent of two full times in order to achieve completion without a preemption.

Using these expected costs, we can then compare the incurred costs of different workload durations on different instance types between on-demand and spot instances under both the old and new pricing schemes. When simulating on-demand instances, the cost is simply the duration multiplied by the cost per time unit.

4.2 Assumptions

While we believe that this analysis provides necessary and accurate insights into the dynamics and cost effectiveness of spot instances and spot pricing methodologies, we do rely on three obvious, though largely unavoidable assumptions in these calculations. The first is that we use the same provisioning and termination algorithm for the post-change simulation as the pre-change simulation. While we know for a fact that the pricing algorithm has changed, the new method used by AWS is not public and, therefore, we could not easily recreate it for our own purposes. However, we do know that the new pricing is based on "long-term pressures" which indicate some consideration of the relative values of supply and demand, and as such may still roughly approximate the old behavior though on a much more long-term and artificially smoothed scale. The second assumption we use is that these simulated workloads of arbitrary duration are not checkpointed, though one could view these simulations as the minimum required time between checkpoints rather than a workload as a whole. The third and final assumption we make is the independence of pricing and market pressures with respect to our own simulated workloads. In other words, we do not factor in the effect, albeit small, that our provisioning of a new instance would have on the overall market supply and demand.

4.3 Results

Using this simulation framework, we can evaluate the cost effectiveness of spot instances under new and old pricing. Further, we can use this expected cost information to highlight cases where one pricing scheme may be more beneficial to certain use cases.

First, when comparing spot instance cost effectiveness with on-demand usage, we see that, on average, the old pricing system allowed spot instances to be more cost-effective than on-demand instances for any workload shorter than about 340 hours, or roughly two weeks. However, there were availability zone-instance type combinations that would become more expensive if they had to avoid preemption for even as little as half an hour. Looking at the new pricing scheme, we observe that in all combinations, spot instances are cheaper than on-demand instances for at least one month, and in many cases indefinitely. (This is simply due to the fact that the new time-stabilized spot prices effectively never exceed the on-demand price. However, this is where our assumption of the old termination algorithm begins to fail as this shows that user-facing pricing is insufficient to predict terminations. Otherwise, in such a case where a max bid of the on-demand price would never be met, all rational users would use the spot instances instead.)

Next, comparing the expected costs of the pricing systems to each other, we see a benefit on the side of the new pricing scheme in that, for workloads of uniformly distributed durations up to one month, it is generally much cheaper than under the old pricing system. However, the generality begins to break down when we look at the lower end of the workload duration as, for durations under an hour, the old pricing system is more cost-effective due to the presence of lower "dips" in pricing, albeit for short durations. Therefore, we can see that the end-user cost-effectiveness with respect to on-demand pricing has generally increased but, for users interested in securing spot instances for only a short time, the costs have increased as there are no longer the pricing "valleys" that we used to see within the spot market (though, conversely, there are also no pricing "peaks").

4.4 Suitability for Scientific Computing

The spot market presents opportunities for acquiring low-cost resources to meet the sporadic and non-deadline-constrained needs of scientific applications. We have previously investigated the use of spot resources to cost-effectively [7] and reliably [22] provision computing resources for various scientific applications, including medical imaging [9], genomics [16], and workflows [4]. Here we discuss the implications of the market changes for such workloads.

We showed above that the new pricing algorithm generally increases spot instance prices, while decreasing price variance. While these changes may make short-duration tasks more expensive (e.g., those that can be completed within a pricing valley), for longer running jobs the new market dynamics may benefit scientific computing. Most notably, in previous work we showed that naïve use of the spot market may result in significant costs for scientific computing, upwards of an order of magnitude more than when using on-demand instances [8]. This increase was primarily as a result of pricing "peaks" and an inability to adapt accordingly. The reduction in price variance alleviates these concerns and ultimately reduces the associated risk of using the spot market. Furthermore, more stable prices reduce the need for elaborate provisioning tools designed to dynamically select suitable, yet low-cost instances and to exchange risk with reliability via bidding strategies.

Conversely, the fact that termination decisions are no longer associated with bid price creates new challenges. Without understanding termination dynamics, it is difficult for users to predict total cost when using spot instances or to have any certainty regarding instance durability. Under the old model, researchers could trade off risk (potential total cost) against durability by bidding well above the market price. Such strategies are no longer applicable.

5 RELATED WORK

In order to devise strategies how to interact with the spot market, a thorough understanding of the market dynamics and potential drivers is necessary, which has led to research into potential drivers of the observed spot prices. Prior to the change in pricing and termination policy it was often believed that the spot market prices were mostly demand-driven. However, researchers were able to show that market demand does not account for the entirety of price changes that can be observed, and that observed prices were driven by (hidden) externalities [3, 23]. While a purely demand-driven model was not empirically observable, the introduction of the new pricing policy moves spot market pricing even further away from demand-driven fluctuations. While the change to the spot market is relatively recent, there are already some published studies describing the effects of the new model. Like our study, researchers have applied various statistical measures and equality indices to analyze the spot market. These studies confirm that the frequency of price changes as well as previously observed spikes in prices have decrease, and adjustments to the spot prices have become more smooth after the policy change [5]. Interesting, others have shown that while the average spot price has decreased, the real cost has increased as artificial peaks included before the change were likely not market driven [10].

With the frequent price changes prior to the policy change, the design and analysis of bidding strategies and frameworks that help users interact with the spot market has been another common research topic. Due to potential termination of spot instances when the bidding price falls below the current spot market price, both price estimation and availability estimations were crucial criteria. Examples of bidding strategies and frameworks that have been explired include minimum bid estimations under price and availability constraints [15], revenue maximization models [23], bidding strategies taking into account if jobs can be interrupted [24], and integration of fault-tolerance policies [21]. In addition, to account for uncertainty in the availability and runtime of spot instances, we have previously proposed a method to provide probabilistic guarantees of execution duration [22].

An important aspect of bidding strategies is the forecasting of the expected spot price in the future, and various approaches have been explored in the context of spot prices. Forecasting methods include parametric models maximizing price expectations [13, 14], cost minimization models using Constrained Markov Decision Processes [11, 19], and prediction models that account for additional aspects such as application migration and potential service downtime [12, 18]. Recently, the use of neural networks for price forecasts in spot markets has shown promise [6].

6 SUMMARY

We have seen through our analyses that there are substantial changes in terms of the frequency of price events, the time between events, the price change represented by an event, and the price distribution (in terms of mean and variance) between the old and new pricing schemes. There is also some evidence that price events now take place in different parts of the hourly billing cycle, previously at the start of the hour, and now around the half hour point. Seasonal effects (in terms of the day of the week) have changed, previously there were fewer price events at weekends, and outside business hours. Now this seems to have been normalized across the week, yet still with some amount of reduced activity at weekends. We have also identified qualities of the pricing data that seem to indicate that AWS readjusted the algorithm (or has set it to automatically readjust) during the transition. We have also found that the new system offers significantly increased duration (thereby benefiting users with longer running tasks) at the expense of providing extremely low prices for short durations.

In future work, we will continue to collect data regarding spot instance terminations so as to explore how the new market algorithms affect termination behavior. We also will inspect the effects of the change in pricing mechanism on the ability to forecast the AWS spot price, which will be of interest to users seeking to estimate costs for specific applications.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grants 1550588 and 1816611 and by the U.S. Department of Energy under Contract DE-AC02-06CH11357. This work also used compute resources provided by Jetstream [17] through the Extreme Science and Engineering Discovery Environment (XSEDE) [20], which is supported by National Science Foundation grant ACI-1548562.

REFERENCES

[1] AWS Instance Types. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html. Accessed May 2019.

[2] New Amazon EC2 Spot pricing model. https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing/. Accessed May 2019.

[3] O Agmon Ben-Yehuda, M Ben-Yehuda, et al. 2013. Deconstructing Amazon EC2 spot instance pricing. *ACM Transactions on Economics and Computation* 1, 3 (2013), 1–20.

[4] Y Babuji, A Woodard, et al. 2019. Parsl: Pervasive parallel programming in python. In *ACM International Symposium on High-Performance Parallel and Distributed Computing*.

[5] M Baruwal Chhetri, M Lumpe, et al. 2018. To bid or not to bid in streamlined EC2 spot markets. In *International Conference on Services Computing*. IEEE, 129–136.

[6] M Baughman, C Haas, et al. 2018. Predicting Amazon spot prices with LSTM networks. In *9th Workshop on Scientific Cloud Computing*. ACM Press, 1–7.

[7] R Chard, K Chard, et al. 2015. Cost-aware cloud provisioning. In *11th International Conference on e-Science*. IEEE, 136–144.

[8] R Chard, K Chard, et al. 2016. An automated tool profiling service for the cloud. In *16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 223–232.

[9] R Chard, R Madduri, et al. 2018. Scalable pCT image reconstruction delivered as a cloud service. *IEEE Transactions on Cloud Computing* 6, 1 (2018), 182–195.

[10] G George, R Wolski, et al. 2019. Analyzing AWS spot instance pricing. In *35th IEEE International Conference on Data Engineering*.

[11] W Guo, K Chen, et al. 2015. Bidding for highly available services with low price in spot instance market. In *24th International Symposium on High-Performance Parallel and Distributed Computing*. ACM Press, 191–202.

[12] X He, P Shenoy, et al. 2015. Cutting the cost of hosting online services using cloud spot markets. In *24th International Symposium on High-Performance Parallel and Distributed Computing*. ACM Press, 207–218.

[13] B Javadi, R. K Thulasiramy, et al. 2011. Statistical modeling of spot instance prices in public cloud environments. In *4th IEEE International Conference on Utility and Cloud Computing*. IEEE, 219–228.

[14] B Javadi, R. K Thulasiramy, et al. 2013. Characterizing spot price dynamics in public cloud environments. *Future Generation Computer Systems* 29, 4 (2013), 988–999.

[15] M Lumpe, M. B Chhetri, et al. 2017. On estimating minimum bids for Amazon EC2 spot instances. In *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 391–400.

[16] R Madduri, K Chard, et al. 2015. The Globus Galaxies platform: Delivering science gateways as a service. *Concurrency and Computation: Practice and Experience* 27, 16 (2015), 4344–4360.

[17] C. A Stewart, T. M Cockerill, et al. 2015. Jetstream: A self-provisioned, scalable science and engineering cloud environment. In *XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*. ACM, Article 29, 8 pages. http://doi.acm.org/10.1145/2792745.2792774

[18] S Subramanya, T Guo, et al. 2015. SpotOn: A batch computing service for the spot market. In *6th ACM Symposium on Cloud Computing*. ACM Press, 329–341.

[19] S Tang, J Yuan, et al. 2012. Towards optimal bidding strategy for Amazon EC2 cloud spot instance. In *5th International Conference on Cloud Computing*. IEEE, 91–98.

[20] J Towns, T Cockerill, et al. 2014. XSEDE: Accelerating scientific discovery. *Computing in Science & Engineering* 16, 5 (2014), 62–74.

[21] W Voorsluys and R Buyya. 2012. Reliable provisioning of spot instances for compute-intensive applications. In *26th International Conference on Advanced Information Networking and Applications*. IEEE, 542–549.

[22] R Wolski, J Brevik, et al. 2017. Probabilistic guarantees of execution duration for Amazon spot instances. In *International Conference for High Performance Computing, Networking, Storage, and Analysis*. ACM Press, 1–11.

[23] H Xu and B Li. 2013. Dynamic cloud pricing for revenue maximization. *IEEE Transactions on Cloud Computing* 1, 2 (2013), 158–171.

[24] L Zheng, C Joe-Wong, et al. 2015. How to bid the cloud. In *SIGCOMM*, Vol. 45. ACM Press, 71–84.