Spot Price Characterization: What Are the Right Features to Model?

[Extended Abstract]

Qianlin Liang, Cheng Wang, Bhuvan Urgaonkar
Department of Computer Science and Engineering
The Pennsylvania State University

1. INTRODUCTION

Amazon EC2 has been offering *spot instances* [2] since 2009 and a large segment of its "tenant" workloads has come to embrace these [6]. The appeal of spot instances lies in their low prices - up to one-tenth of the prices of *on-demand* instances of equivalent (advertized) capacities. Unlike an on-demand instance, whose price changes very slowly (over months or years), a spot instance has a highly dynamic price that may change as frequently as once every few minutes. To procure a spot instance, a tenant needs to place a *bid* in the concerned marketplace. Following this, whenever the dynamic price for the requested type of spot instance (simply the spot price henceforth) falls below the bid, an instance is allocated to the tenant. However, when the bid falls below the spot price, a warning is issued to the tenant following which the instance is reclaimed/revoked by EC2. The tenant may choose to use this warning period (2 minutes as of this writing) to save all or some of that instance's state. From a tenant's point of view, a spot instance is a virtual machine (VM) that is cheaper than its on-demand counterpart but appears to have poorer availability.

To benefit from the low prices of spot instances, a tenant must effectively deal with two sources of complexity: (i) it must predict relevant aspects of spot price, and (ii) it must combine these predictions with its application-specific trade-offs [1] to devise online instance procurement algorithms. Both these issues have received a lot of attention recently. In this paper, our focus is on (i). Our findings also have implications for (ii) and will inform our future work as we discuss in Section 4.

A key challenge in using spot instances, as already identified by many recent papers, is the *poor predictability* of spot prices. Researchers have proposed prediction techniques for spot prices with varying degrees of modeling complexity. Among the simpler ones are techniques based on (dynamically updated) empirical probability distributions [3, 4, 7]. More sophisticated techniques based on auto-regressive time-series [1, 8, 10], Markovian models [9, 5], etc., have also been proposed. A key requirement for cost-effective procurement of a spot instance is the ability to predict its *service contiguity*, i.e., the contiguous duration for which it is likely to be available. Regardless of the modeling technique used, all existing approaches (to our knowledge) are based on creating a statistical model for the *raw spot price itself* (with underlying assumptions about its stationarity). We believe this to be an untenable exercise because our extensive analysis of spot price traces indicates that these are best considered non-stationary. For example, Figure 1 shows that the first two moments of spot price for many different marketplaces vary significantly over a 90-day period. Since existing techniques are based on modeling raw spot prices directly (despite their high dynamism), they are either ineffective or computationally non-scalable when used in online control wherein the prediction of some other properties of spot prices (rather than mere raw values) are desired.

What Should be Modeled? Rather than modeling the exact spot price values, we argue that the focus should be on the features we identify below.

- *Feature I*: Since spot prices tend to be significantly smaller than on-demand prices (of equally-sized VMs) during periods when a bid is successful, and since EC2 charges a tenant based on the spot price during such periods (not based on the bid), attempting to predict spot prices very accurately is of little value. A visual inspection of the 90-day long spot price time-series in Figure 1 and how these prices compare with a bid that equals to the on-demand price clarifies this. In particular, it suffices that we predict the average spot price during such periods with reasonable accuracy (since that is what will determine our costs).
- *Feature II*: Tenants want the spot instance to be available for long enough time to maintain the service contiguity, i.e., they are interested in *how long a successful bid is likely to last.* An effective predictor should not overestimate this quantity - doing so may render a control scheme overly optimistic in its estimation of the cost vs. performance trade-off.
- *Feature III*: The *raw* cross-correlations among markets, used by prior works [7, 4] to capture simultaneous bid failures, may not be informative for decision-making. As we will show in Section 2.2 and 3.2, simultaneous bid failures crucially depend on the chosen markets and bids.

The rest of this paper is organized as follows. In Section 2, we present our prediction approach. In Section 3, we evaluate proposed approach with extensive real-world traces. We discuss future directions and conclude in Section 4.

2. OUR APPROACH

We model as a random variable $L(b)$ the length of a *contiguous* period during which the spot price is less than or equal to a bid b. In other words, $L(b)$ captures the lifetime of a spot instance using bid b. We denote as $\bar{p}(b) = E[p_t|L(b)]$ a random variable for the average spot price p_t during a period when the bid b is successful, which serves to estimate the cost of a spot instance procured by placing a bid b.

[1] These trade-offs would be between costs, on the one hand, and overheads of possible revocations (either in the form of fault-tolerance mechanisms or loss of performance/correctness), on the other.

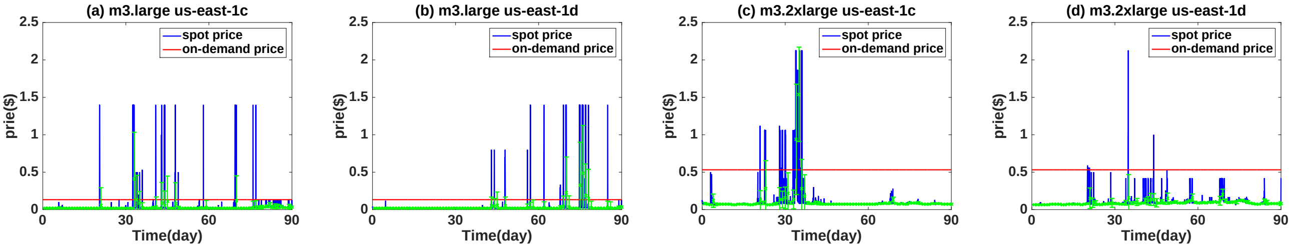

Figure 1: Sample spot price timeseries collected during the 90-day period (2015-07-08 to 2015-10-06) and are chosen due to their very different properties. The green "*" and error bars represent the moving average and standard deviations over each 2-day interval.

2.1 Our prediction technique

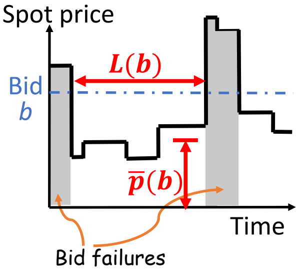

Figure 2: Key ideas underlying our prediction.

Our technique employs empirical probability distributions computed over recent sliding time windows (H most recent time slots, e.g., days) for making predictions of $L(b)$ and $\bar{p}(b)$. H must be chosen such that temporal locality[2] holds for these quantities. Figure 2 clarifies these quantities. Large $L(b)$ and small $\bar{p}(b)$ imply long service continuity and low costs, thereby encouraging the use of spot instances using bid b. We use a small percentile (e.g., 5th) of the recently constructed distribution of $L(b)$ - denoted as $\hat{L}(b)$ - as our prediction in the ongoing horizon. The reasoning behind this choice is that if the statistical properties of $L(b)$ do not change much over H, we expect that with a very high probability, bid b would be successful for at least $\hat{L}(b)$ time units. We use average of $\bar{p}(b)$ during the relevant H as its predictor (denoted as $\hat{\bar{p}}(b)$).

Assessment Metrics: We say that an *over-estimation* of $L(b)$ has occurred when $\hat{L}(b) > L(b)$. This represents a scenario wherein the tenant was likely overly ambitious in using spot instances. We further define $L(b)$ *over-estimation rate* as the fraction of $L(b)$ predictions that result in over-estimation, denoted as $f(b)$. The assessment metric for $\hat{\bar{p}}(b)$ should capture the extent of its deviation from actual values. Therefore, we compute $\xi(b) = (\bar{p}(b) - \hat{\bar{p}}(b))/\bar{p}(b)$ and define as *relative deviation* of $\bar{p}(b)$ the mean value of $\xi(b)$ for all occurrences of $\bar{p}(b)$ in the relevant H. Lower values are better for both. In Section 3, we focus on the evaluating the right choice of history window size (for model training), which turns out to be dependent on both the market and the bid. None of the prior works (to our knowledge) have analyzed these idiosyncracies.

2.2 Simultaneous revocations

When placing a bid, tenants may want to avoid picking spot markets with the likelihood of simultaneous revocations (the spot instances may be terminated simultaneously due to coincident bid failures). Prior works, e.g., [7, 4], suggest bidding across markets where there are no significant statistical correlations among the "raw" history spot price traces.

However, such raw correlations might not be informative for tenants' decision making. Let us consider an illustrative example in Figure 3. We generate synthetic spot prices for two markets wherein the cross-correlation between the two markets' spot prices is low and the tenant might be tempted to use both markets if its decision is only based on the raw correlation. However, it is obvious that the bid failures from the two markets are highly correlated under bid 1 but not correlated under bid 2. Therefore, it may be imprudent for the tenants to make decisions solely based on the raw correlations without considering the actual bids. More specifically, what the tenant really needs is measurements of simultaneous revocations, *conditioned on bids.* Furthermore, the statistical correlation of bid failures across markets may not be very informative for decision-making regarding bid placement. Instead, a tenant might find it more beneficial to learn the absolute time durations of simultaneous revocations, i.e., the total amount of time that a bid fails in two candidate markets within the history window.

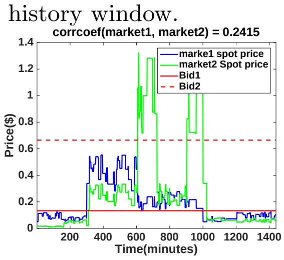

Figure 3: A synthetic example with simultaneous revocations related to bids.

A more informative metric that we propose is based on characterizing simultaneous revocations conditioned on pre-specified bids. Under a given bid, we denote as A and B the sets of time periods when the bid fails in two spot markets under comparison, respectively. Denote as $T(A)$ and $T(B)$ the corresponding lengths/sizes of A and B. $T(A \cap B)$ and $T(A \cup B)$ represent the time durations of coincident bid failures and total bid failures, respectively. $\frac{T(A \cap B)}{T(A \cup B)}$ reflects the probability that the bid fails in both markets when the bid already fails at one market. It is informative to look at both the durations of bid failures (e.g., $T(A)$) and $\frac{T(A \cap B)}{T(A \cup B)}$ when comparing markets. E.g., even if $T(A)$ and/or $T(B)$ are relatively small compared with the history window size, if $\frac{T(A \cap B)}{T(A \cup B)}$ is high, which implies markets (A,B) almost always fail together under the given bid, it may be better not to place bids in markets (A,B) simultaneously. On the other hand, even if both $\frac{T(A \cap B)}{T(A \cup B)}$ and $T(A \cap B)$ are small, we may use neither A nor B if $L(b)$ is also small in both markets. Therefore, tenants can use such metrics, together with predicted $L(b)$ and $\bar{p}(b)$, to get a better understanding of the properties of simultaneous revocations and carry out cost analysis. We show initial results and insights in Section 3.

3. EVALUATION AND LESSONS

Experiment Setup: We evaluate our technique with dozens of spot price traces of which we show four 90-day spot price traces for VM types of `m3.large` and `c3.large` in availability zones `us-east-1c` and `us-east-1d` in Figures 1. We denote as `m3.large-c` the spot market for `m3.large` in `us-east-1c` (same notation rule for other markets). For each trace, we pick bid b from $\{0.5d, d, 2d, 5d, 10d\}$, where d is the corre-

[2] By temporal locality, we mean that over relatively short time-scales (a day to a few days), the key features tend to change little, whereas over longer time-scales (weeks to months), they might undergo more substantial changes.

sponding on-demand price.

3.1 Evaluation of Our Prediction Technique

We vary VM types, markets, bids, history window size H and show the assessment metrics f ($L(b)$ over-estimation rate) and ξ ($\bar{p}(b)$ relative deviation) in Table 1.

Validation of prediction. Under most of (market, bid) pairs, the optimal (lowest) f and ξ are below 10%, which demonstrates the good predictive power of our technique. For `c3.large-c`, f and ξ are much higher than those from other markets; therefore, it might be better not to use this market temporarily until its predictability gets improved.

What is the the right choice for history window size? We observe that (i) $H = 7$ days seem to be the best choice (minimized f and ξ) cases we examine, (ii) the optimal H varies across markets and bids, implying the necessity for considering different markets separately when determining history window size, instead of blindly choosing a window size that has to work for all markets, and (iii) changing bid values may not affect f and ξ much (e.g., m3.large-c), possibly due to the fact that the $L(b)$ and $\bar{p}(b)$ do not vary much when spot price exceeds bid price.

	BID	$f(b)$				$\xi(b)$			
H		7	14	21	28	7	14	21	28
m3.large-c	0.5d	0.1154	0.1237	0.1333	0.1446	0.0796	0.0814	0.0849	0.0902
	1d	0.0698	0.0759	0.0833	0.0923	0.0689	0.0698	0.0988	0.1066
	2d	0.0241	0.0263	0.0290	0.0323	0.0882	0.0960	0.1032	0.1119
	5d	0.0241	0.0263	0.0290	0.0323	0.0882	0.0960	0.1032	0.1119
	10d	0.0241	0.0263	0.0290	0.0323	0.0882	0.0960	0.1032	0.1119
m3.large-d	0.5d	0.1018	0.0692	0.0724	0.0759	0.0619	0.0576	0.0756	0.0665
	1d	0.1037	0.0625	0.0661	0.0702	0.0727	0.0742	0.0908	0.0900
	2d	0.0984	0.0783	0.0926	0.0990	0.0989	0.1040	0.1099	0.1218
	5d	0.1316	0.1215	0.1400	0.1613	0.0907	0.1230	0.1195	0.1282
	10d	0.0727	0.0777	0.0833	0.0899	0.1921	0.1862	0.1698	0.1696
c3.large-c	0.5d	0.1007	0.0909	0.0783	0.0841	0.0586	0.0696	0.1320	0.0000
	1d	0.1226	0.1717	0.1868	0.2024	0.1120	0.1576	0.1856	0.2097
	2d	0.1442	0.1753	0.1910	0.2073	0.1097	0.1576	0.1854	0.2092
	5OD	0.1359	0.1667	0.1818	0.1975	0.1272	0.1621	0.1758	0.1942
	10OD	0.0000	0.0000	0.0000	0.0000	0.8728	0.7562	0.7294	0.7009
c3.large-d	0.5d	0.0989	0.0833	0.0909	0.1000	0.0553	0.0629	0.0687	0.0677
	1d	0.0581	0.0633	0.0694	0.0923	0.0725	0.1081	0.1070	0.1013
	2d	0.0588	0.0641	0.0704	0.0781	0.0725	0.1031	0.1020	0.0993
	5d	0.0588	0.0641	0.0704	0.0781	0.0729	0.1041	0.1029	0.1003
	10d	0.0000	0.0000	0.0000	0.0000	0.3374	0.2172	0.1772	0.1642

Table 1: The assessment metrics $f(b)$ and $\xi(b)$ under different bid values and history window sizes (H in days). The shaded cells represent the optimal window size that minimizes $f(b)$ and $\xi(b)$. "-c" and "-d" represent the markets.

3.2 Simultaneous Revocations

We show some initial results on simultaneous revocations in Table 2. We have several insights: (i) Increasing bid may de-correlate bid failures: even if spot prices of two markets always jump simultaneously, they don't usually reach the same high spot price. When the bid increases, one of the markets may experience less bid failures whereas the other remains unaffected (possibly because the bid is not high enough). (ii) Increasing bid may also increase the the extent to which the simultaneous revocation occurs, e.g., as the total failure time $T(A \cup B)$ decreases in markets (b,d) of `m3.xlarge`, the fraction of time that concurrent bid failure occurs becomes less. (iii) Since the properties of simultaneous revocations highly depend on markets and bids (and possibly also history window size), simply comparing the raw statistical correlations of multiple markets' spot prices may not suffice and might even lead to *faulty* decision making.

4. DISCUSSIONS AND FUTURE DIRECTION

What is the right percentile for prediction? Recall that we use a percentile of $L(b)$ in the history window as our prediction. As this percentile decreases, we may be more conservative in using spot instances and therefore the cost might be much higher than the optimal costs (with perfect knowledge of $L(b)$ and $\bar{p}(b)$). As the percentile increases we are tempted to use spot instances more aggressively, whereas the application performance might degrade beyond the acceptable range due to bid failures. Solving this problem requires a good understanding of the tradeoffs of performance vs. resource allocation vs. costs. How to tune the parameters cost-effectively with acceptable performance remains an open problem and is worth exploring.

	Bid	d				$5d$			
		A	B	$A \cap B$	$\frac{A \cap B}{A \cup B}$	A	B	$A \cap B$	$\frac{A \cap B}{A \cup B}$
m3.large	b,c	1386	35	6	0.0042	900	6	6	0.0067
	b,d	1386	1730	218	0.0752	900	1489	161	0.0723
	b,e	1386	4517	51	0.0087	900	0	0	0.0000
	c,d	35	1730	24	0.0138	6	1489	0	0.0000
	c,e	35	4517	0	0.0000	6	0	0	0.0000
	d,e	1730	4517	329	0.0556	1489	0	0	0.0000
m3.xlarge	b,c	7050	0	0	0.0000	376	0	0	0.0000
	b,d	7050	1070	377	0.0487	376	774	160	0.1616
	b,e	7050	9	0	0.0000	376	9	0	0.0000
	c,d	0	1070	0	0.0000	0	774	0	0.0000
	c,e	0	9	0	0.0000	0	9	0	0.0000
	d,e	1070	9	0	0.0000	774	9	0	0.0000
c3.large	b,c	1103	2340	976	0.3956	1059	2274	921	0.3818
	b,d	1103	1200	910	0.6533	1059	1194	875	0.6350
	b,e	1103	284	229	0.1978	1059	279	224	0.2011
	c,d	2340	1200	1189	0.5057	2274	1194	1184	0.5184
	c,e	2340	284	238	0.0997	2274	279	228	0.0981
	d,e	1200	284	238	0.1910	1194	279	233	0.1879
c3.2xlarge	b,c	1589	3917	946	0.2075	511	1152	18	0.0109
	b,d	1589	14	3	0.0019	511	0	0	0.0000
	b,e	1589	756	0	0.0000	511	0	0	0.0000
	c,d	3917	14	0	0.0000	1152	0	0	0.0000
	c,e	3917	756	0	0.0000	1152	0	0	0.0000
	d,e	14	756	0	0.0000	0	0	0	0.0000

Table 2: Simultaneous revocations across different pairs of markets (b,c,d,e) and under different bids. The measurements are in minutes. The $T(.)$ operator is omitted for space.

How to use our prediction approach effectively? Our prediction model can be incorporated with online optimal control or heuristics for tenants' cost-effective resource procurement. As a concrete example, a tenant can use our model of simultaneous revocations to choose spot markets with less/un-correlated failures under pre-specified bids. An optimization-based (or heuristic-based) algorithm with performance overhead due to bid failures (as a function of $L(b)$) and estimated costs (as a function of $\bar{p}(b)$) can be used online to exploit the tradeoff between performance and costs.

5. REFERENCES

[1] O. A. Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir. Deconstructing amazon ec2 spot instance pricing. In *Proc. of CloudCom'11*, 2011.

[2] EC2 spot, 2016. http://aws.amazon.com/ec2/spot-instances/.

[3] B. Javadi, R. Thulasiramy, and R. Buyya. Statistical modeling of spot instance prices in public cloud environments. In *Proc. of UCC'11*, 2011.

[4] P. Sharma, S. Lee, T. Guo, D. Irwin, and P. Shenoy. Spotcheck: Designing a derivative iaas cloud on the spot market. In *Proc. of EuroSys'15*, 2015.

[5] Y. Song, M. Zafer, and K. Lee. Optimal bidding in spot instance market. In *INFOCOM'12*, 2012.

[6] Spot instance: featured customer testimonials, 2015. https://aws.amazon.com/ec2/spot/testimonials/.

[7] S. Subramanya, T. Guo, P. Sharma, D. Irwin, and P. Shenoy. Spoton: A batch computing service for the spot market. In *Proc. of SoCC'15*, 2015.

[8] R. M. Wallace, V. Turchenko, M. Sheikhalishahi, I. Turchenko, V. Shults, J. L. Vazquez-Poletti, and L. Grandinetti. Applications of neural-based spot market prediction for cloud computing. In *IDAACS'13*, 2013.

[9] M. Zafer, Y. Song, and K.-W. Lee. Optimal bids for spot vms in a cloud for deadline constrained jobs. In *Cloud'12*, 2012.

[10] H. Zhao, M. Pan, X. Liu, X. Li, and Y. Fang. Optimal resource rental planning for elastic applications in cloud market. In *Proc. of IPDPS'12*, 2012.