Predication Instances spot price in EC2

test test

Abstract We analyze G2 spot instance price

1 Introduction

Cloud computing provides various types of compute resources to serve diverse application scenarios. The cloud computing frees the burden of system administration overheads without incurring prohibitive initial hardware purchase cost. From the service provider's perspective, fully utilizing the already established hardware resources and services is crucial to maximize monetary gain. As the users' resource demand can vary from time to time, some cloud computing providers offer services at chaper price than the regular price to maximize hardware/service utilization. For instance, Amazon Web Services (AWS), a leading cloud computing vendor, provides its surplus of EC2 computing resources at a cheaper price in the form of *spot instance*. A user who wants to use spot instance bids for a price that one is willing to pay, and if the bid price is higher than the spot price that is decided by the service provider, one can get the resource allocated and pays for the spot price in the hourly basis. Other than AWS, Google Cloud Engine provides such opportunistic resources in the form of *preemptive instances*, and Microsoft Azure provides its excess compute capacity as *low-priority VM*.

Though users can utilize the opportunistic resources at a cheaper price, sudden service termination can happen at anytime as the demand for the computing resource changes. To mitigate the chance of sudden service interruption, few works were conducted to better predict and model the price change of EC2 spot instance in

Name of First Author
Name, Address of Institute, e-mail: name@email.address

Name of Second Author
Name, Address of Institute e-mail: name@email.address

literature. Ben-Yehuda et al. [1] and Zhao et al. [7] tried to predict the future spot instance price using various predictive analysis algorithms, but they all concluded that the spot price is rather random and hard to make meaningful prediction for future price changes. Since then, most of studies that are related to the utilization of spot instance focus on the handling sudden service interruption [6, 2, 4, 3] or spot instance bid strategy [5, 8].

In this paper, we apply few time-series analysis algorithms to predict future price change pattern of AWS EC2 spot instances. By carefully desiging the period of train datasets, we could uncover that applying seasonal-arima (s-arima) can improve the accuracy of price change prediction error by XY% comparing to the naive method that references most recent price to predict future price [3]. In addition to the contribution of improved price prediction accuracy, we could also discover that the configuration values to get the best prediction accuracy differs significantly across different availability zones (AZs) and instance types. Based on the extensive experiments and promising results, we bring up an opportunity of improving spot instance prediction accuracy that can result in significant cost gain for cloud computing users with increased system stability.

2 Time-Series Analysis for GPU Spot Instances

List basic methods that are widely used in other wors focus to analyze the predication of spot instances price for 7 regions by use different time series models such as ARIMA, SARIMA VAR, DLM, prophet, average and naïve model our work covered 7 regions of spot instances datasets, ap-northeast-1, ap-southeast-1, ap-regions, and 5 types and different period of datasets G2(03-02 to 2016-10-31),I2(2016-05-03 to 2017-02-04),M4,c3,R3(2016-03-02 to 2017-02-04)

In our work, we use many forecasting time series models and parameters to get best fit to data, from sample models as mean method which use mean of history as predation, assume that our historical data sets be x

$$\frac{y_t = x_{t-1} + x_{t-2} + x_{t-3} + ... + x_{t-n}}{X} \tag{1}$$

Naïve method another simple model to forecasting but anested to use mean historical data the model just simple set last value to be as predation result

$$y_t = x_{t-1} \tag{2}$$

seasonal mean method to forecast time series the model use historical data to find the same S previous observed for predation at time T

$$y_t = x_s \tag{3}$$

Where x previous observed from historical data at time s where s equal to t

ARIMA with different parameters stander for Autoregressive Integrated Moving Average, which is the most popular statistical model and widely used to forecasting a time series, the model is a combination of autoregressive Eq. 4 and moving-average model Eq. 5, with three parameters (p,d,q) where p is the number of autoregressive terms, which is depends on past values, d is the degree of differencing and q is the number of lagged forecast errors in the prediction equation, depends only on the random error terms

$$y_t = w_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ...\beta_n y_{t-n} + \varepsilon_t \tag{4}$$

$$y_t = w_0 + \varepsilon_t + \delta_1 \varepsilon_{t-1} + \delta_2 \varepsilon_{t-2} + ... + \delta_n \varepsilon_{t-n} \tag{5}$$

Thus, ARIMA if $d = n$ will be

$$y_t = w_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ...\beta_n y_{t-n} + \delta_1 \varepsilon_{t-1} + \delta_2 \varepsilon_{t-2} + ... + \delta_n \varepsilon_{t-n} + \tag{6}$$

Where the term β_i is, weight applied to prior values in the time series δ_i is autocorrelation coefficients at lags and ε_i is residual error term

3 Evaluation

Compare all the methods of different types of algorithms to different instance types

Availability Zones	m4 instances	c3 instances	r3 instances	i2 instances
ap-northeast-1a	(60,4)	(60,4)	(60,4)	(120,15)
ap-northeast-1c	(60,2)	(60,2)	(120,15)	(120,15)
ap-southeast-1a	(150,1)	(150,1)	(150,1)	(150,1)
ap-southeast-1b	(150,1)	(60,2)	(150,1)	(150,1)
ap-southeast-2a	(60,4)	(60,4)	(150,1)	(150,1)
ap-southeast-2b	(60,4)	(60,2)	(150,1)	(120,15)
eu-west-1a	(60,4)	(90,8)	(30,2)	(150,1)
eu-west-1b	(60,4)	(60,2)	(30,1)	(120,1)
eu-west-1c	(60,2)	(60,4)	(30,1)	(150,1)
us-east-1a	(50,1)	(150,1)	(30,1)	(60,2)
us-east-1c	(150,1)	(150,1)	(30,1)	(150,1)
us-east-1d	(60,2)	(150,1)	(30,2)	(150,1)
us-east-1e	(60,4)	(150,1)	(30,1)	(60,4)
us-west-1a	(60,2)	(150,1)	(30,1)	(120,15)
us-west-1b	(60,4)	(150,1)	(30,1)	(120,15)
us-west-2a	(60,2)	(150,1)	(150,1)	(150,1)
us-west-2b	(60,4)	(150,1)	(30,1)	(150,1)
us-west-2c	(60,2)	(60,2)	(60,2)	(150,1)

Table 1 Best fit ARIMA model for different availability

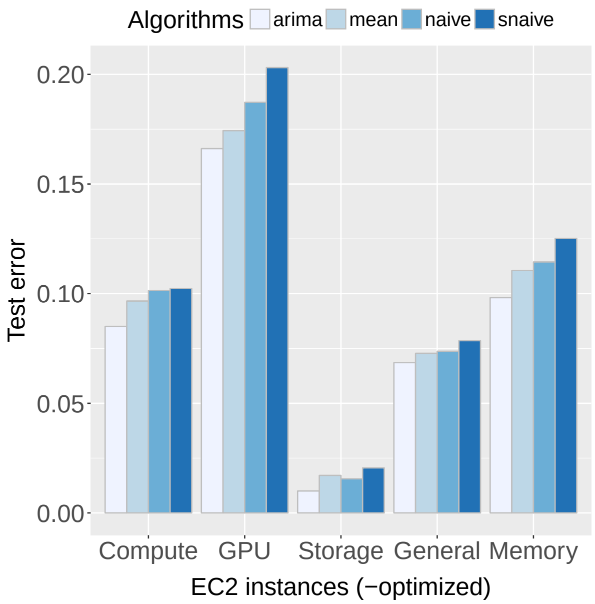

Fig. 1 Algorithms with different instance types

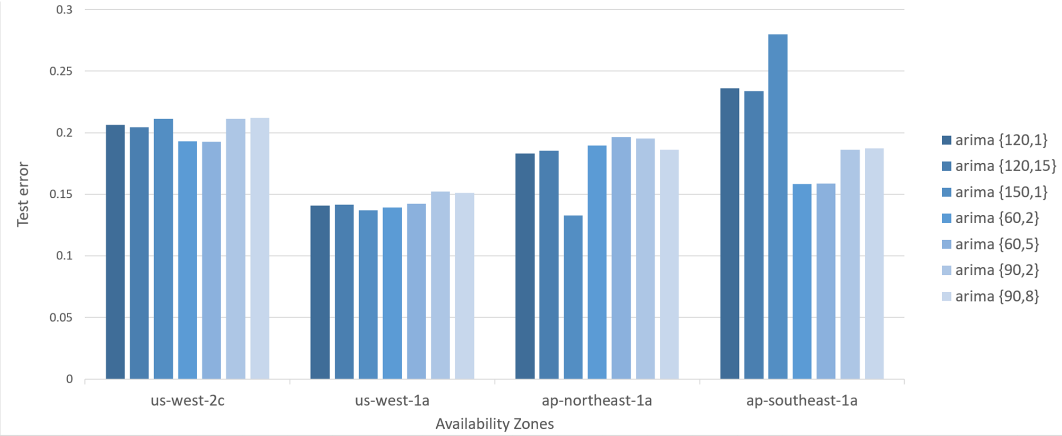

Fig. 2 error for g2.2xlarge for ARIMA modeling

4 Conclusion

Summarize

References

1. Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., Tsafrir, D.: Deconstructing amazon ec2 spot instance pricing. ACM Trans. Econ. Comput. **1**(3), 16:1–16:20 (2013). DOI 10.1145/2509413.2509416. URL http://doi.acm.org/10.1145/2509413.2509416
2. Gong, Y., He, B., Zhou, A.C.: Monetary cost optimizations for mpi-based hpc applications on amazon clouds: Checkpoints and replicated execution. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '15, pp. 32:1–32:12. ACM, New York, NY, USA (2015). DOI 10.1145/2807591.2807612. URL http://doi.acm.org/10.1145/2807591.2807612
3. Lee, K., Son, M.: Deepspotcloud: Leveraging cross-region gpu spot instances for deep learning. In: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD) (2017)
4. Sharma, P., Guo, T., He, X., Irwin, D., Shenoy, P.: Flint: Batch-interactive data-intensive processing on transient servers. In: Proceedings of the Eleventh European Conference on Computer Systems, EuroSys '16, pp. 6:1–6:15. ACM, New York, NY, USA (2016). DOI 10.1145/2901318.2901319
5. Sharma, P., Irwin, D., Shenoy, P.: How not to bid the cloud. In: 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16). USENIX Association, Denver, CO (2016)
6. Yan, Y., Gao, Y., Chen, Y., Guo, Z., Chen, B., Moscibroda, T.: Tr-spark: Transient computing for big data analytics. In: Proceedings of the Seventh ACM Symposium on Cloud Computing, SoCC '16, pp. 484–496. ACM, New York, NY, USA (2016). DOI 10.1145/2987550.2987576. URL http://doi.acm.org/10.1145/2987550.2987576
7. Zhao, H., Pan, M., Liu, X., Li, X., Fang, Y.: Optimal resource rental planning for elastic applications in cloud market. In: Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium, IPDPS '12, pp. 808–819. IEEE Computer Society, Washington, DC, USA (2012). DOI 10.1109/IPDPS.2012.77. URL http://dx.doi.org/10.1109/IPDPS.2012.77
8. Zheng, L., Joe-Wong, C., Tan, C.W., Chiang, M., Wang, X.: How to bid the cloud. In: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM '15, pp. 71–84. ACM, New York, NY, USA (2015). DOI 10.1145/2785956.2787473. URL http://doi.acm.org/10.1145/2785956.2787473