International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12 Number 21 (2017) pp. 11205-11212
© Research India Publications. http://www.ripublication.com

Analysis and Prediction of Amazon EC2 Spot Instance Prices

Ashish Kumar Mishra[1] and Dharmendra K. Yadav[2]

[1,2] Department of Computer Science & Engineering, Motilal Nehru National Institute of Technology Allahabad, Allahabad - 211004, Uttar Pradesh, India.

[1]Orcid Id: 0000-0002-7532-5585

Abstract

Amazon Elastic Compute Cloud (EC2) is a web service that provides secure, re-sizable compute capacity in the cloud. It is designed to make web-scale cloud computing for developers. There are three different ways of pricing for Amazon EC2 instances: On-demand, Reserved instances, and Spot instances. Spot instances provide users with access to unused Amazon EC2 capacity at high discounts relative to On-demand and reserved prices. The spot prices fluctuate based on the demand and supply of available unused capacity of EC2. When users request spot instances, they specify the maximum spot price they are willing to pay. Spot instances are launched when the spot price is lower than the price specified by user. It will continue to run until either user chooses to terminate it or the spot price exceeds the maximum price specified by users. The major challenge for researchers in this area is to predict the price for a particular instant of time which assures uninterrupted execution. In this paper, price of spot instances is analyzed and regression algorithm is used to predict useful information for cloud clients and cloud sellers who want to start selling spot instances. Efforts have been made to design a methodology based on regression algorithm to provide useful data to the customers of spot instances

Keywords: Spot instance, AWS, Neural Networks, Prediction, Spot Price.

Introduction

Amazon offers three pricing models, all requiring a fee from a few cents to a few dollars, per hour, per running instance. The models provide different assurances regarding when instances can be launched and terminated. Paying a yearly fee (hundreds to thousands dollars) clients buy the ability to launch one reserved instance whenever they wish. Clients may instead choose to forgo the yearly fee and attempt to purchase an on-demand instance when they need it, at a higher hourly fee and with no guarantee that launching will be possible at any given time. Both reserved and on-demand instances remain active until terminated by the client.

The third, cheapest pricing model is spot instance which provides no guarantee regarding either launch or termination time. While placing a request for a spot instance, clients bid the maximum hourly price they are willing to pay for acquiring it (called declared price or bid). The request is granted if the bid is higher than the spot price; otherwise, it waits. Periodically, Amazon publishes a new spot price and launches all waiting instance requests with a maximum price exceeding this value; the instances will run until clients terminate them or the spot price increases above their maximum price.

By analyzing the spot price history of Amazon's EC2 cloud, it is inferred, how the prices are set. A model is also being constructed to predict the probability of a certain bid price in future. Change in price is analyzed to identify the pattern for future prices as well.

Having knowledge about, how a leading cloud provider (for example Amazon) prices its unused capacity is of considerable interest to cloud providers and cloud clients as well. Understanding how the bidding process works, may allow other cloud providers to better compete and to utilize their own unused capacity more effectively. Clients can likewise exploit this knowledge to optimize their bids and to predict how long their spot instances would be able to run.

Literature Review

A considerable interest has been shown by researchers in the area of spot pricing from the time Amazon has started selling its unused cloud capacities using this scheme. One of the algorithms proposed in paper [1], is an auto-regressive algorithm (AR (1)). In the article, it was said that the spot price of VM in Amazon EC2 are not based on market demands instead the spot instance prices are decided by using a dynamic algorithm without considering client's bid. Authors show that the high prices are based on market while the low prices are mostly based on AR (1) (autoregressive) dynamic algorithm. Amazon advertises the spot price while hiding how the price is decided. Recent price history of spot instances is given by Amazon. Authors analyze the price history to reverse engineer the method of setting the prices and a model is built to compute prices which is coherent with the available pricing traces.

Client can utilize proposed mechanisms and policies of pricing to decide best bids, predict the uninterrupted execution time of a spot instance and understand when to buy a low-priced or a high-priced capacity. For providers, this information related to pricing can help them to better exploit their free resources effectively. Authors claimed that a dynamic algorithm is used by Amazon for fixing a reserve price of the auction without considering the user's bids. Authors also argued that dynamic algorithm takes as input a minimum price and a maximum price of every instance types. This range is considered as a pricing band. Availability is the linear function of price in the range considered by authors. Change in the reserve price is done randomly by the algorithm. Due to this, there is linear relation between prices in the min-max range and availability of instances. The linear relationship ensures that SI's reserve prices lie in the range of minimum and maximum prices. Different minimum and maximum value of price in the price band is set by Amazon for different regions, types of SIs and operating systems. Thus creating illusion that the prices are based on market demands.

To validate the speculation, authors provide the simulation of the prices and availability. Availability is the result of fixing the auction prices with a reserve price. It is done by setting the number of provided instances which maximizes the provider's profit. Fixing reserve prices randomly will provide several benefits. A random reserve price can hide the low demand and idle price time, by making an illusion of demand and supply changes, and so rising the provider's stock. A wider band of minimum and maximum prices can also hide the condition of high demand and low supply and thus provide a false impression of never ending flexible cloud. On the other hand, in narrower band, the use of dynamic algorithm for deciding price within the band will hide the situation of low demands. Authors demonstrated that different price characteristics like minimum value, change timing and band width could vary every six months as the decision taken by Amazon. It is claimed in the paper that arbitrarily employing Amazon's current traces to model the behavior of clients is not useful 98% of the time on an average.

To determine the cost of Amazon's EC2 spot instances, a model based on bidding or auction is being proposed by authors [2]. They examine the instance pricing and study the mechanism of bidding in the model. Previous spot prices are used by authors for studying hidden aspect of auction strategies and to find potential gain without truthful bidding. They find some strategic aspect regarding communication among cloud providers and users in the setting of AWS.

To analyze various aspect of Amazon EC2 schemes, an experimental set up has been provided by authors. Matlab is used to write script for finding optimal bids for different values starting from lowest to highest prices recorded in data. The experimental results show that there is significant impact of bidding at low price. The demonstrated experiment provides a method that reveal the potential gains with untruthful bidding.

Authors demonstrated a bidding strategy based on Q-learning by focusing on workload, execution time, previous checkpoints and recorded spot price history [3]. They have considered an application provider for executing periodic long running time jobs at instances offered by an IaaS provider. Minimum throughput with QoS restriction is assured by provider. The concept of checkpointing at regular intervals is used to increase the reliability of spot instance. Application provider can get on-demand and spot instance by the IaaS provider. The objective of application provider is to decrease cost with satisfying QoS constraints. So, optimal VMs type (on-demand and spot VM) to be requested and bidding levels for spot instances must be decided by the providers.

The proposed strategy is being evaluated in different environments with real price traces. The conduct of application provider is being examined under various IaaS pricing mechanisms with different checkpointing frequencies. The experimental results show that with Q-learning strategy, application provider restricts its behavior periodically to determine action. This action lowers the cost and increase the profit, with better outcomes in-comparison to MDP and blind mechanisms.

A provisioning algorithm for enhancing the capacity of cluster by the use of spot instance is being proposed by authors [4]. The objective of the algorithm is to decrease the job's waiting time in the queue of cluster where users have reservations to define the time and amount of resources needed by the job. To fulfill the need of high demands of computing clusters, authors devised a dynamic policy for buying the less costly spot instances. For extra provisioning of servers to accommodate workload in a local cluster, a heuristic algorithm is devised by authors. They have considered the economics of buying resources in the spot market to deal with hike of unanticipated workload. The mechanism does not consider the type of workload and failure of resources to decide requests redirection. If a drastic change is done, the models are no longer relevant.

Authors in article [5], claimed that retraction of spot instances occur according to Poisson process. Spot pricing data can be modeled using Poisson distribution. The authors argued that bid failure can be modeled as a rare event and so Poisson distribution can be used.

Poisson regression is an abstracted linear model which is a form of regression analysis that is utilized to model data count and contingency tables [6]. It takes assumption that response variable y has a Poisson distribution and takes the logarithm of its expected value that can be modeled by a linear combination of parameters which are not known. This model is some time also called log-linear model, particularly when it is used to model contingency tables. Poisson distribution can be described by the equation $y = e^{\theta' x}$. An in-depth analysis of spot price data shows that spot price varies frequently, so applying Poisson distribution will not produce any positive results.

Proposed Work

The objective of the article is to analyze the spot instance prices and use regression algorithms to predict prices for cloud users.

In this article, linear regression algorithm is used to forecast the relationship between a given bid price and the time in which it will remain above the spot price. Curves have been plotted to show the relationship between input prices. We have tried to come up with the value of regression parameters using gradient descent algorithm which is best suited for the problem. Estimated price is also computed by analyzing the current change in price using neural networks.

Experimental Setup and Results Analysis

In this section, data preprocessing, linear regression, cost function and gradient descent have been explored for experimental setup and analysis of outcomes.

A. Data Preprocessing

Amazon provides virtual machines called instances on the basis of pay as you go model. Each instance has a type describing its computational resources as follows: m1.small, m1.large and m1.xlarge denote small, large, and extra-large instances respectively; m2.xlarge, m2.2xlarge, and m2.4-xlarge denote extra-large, double extra-large, and quadruple extra-large high memory instances respectively and c1.medium, c1.xlarge denote medium and extra-large high CPU instances respectively.

An instance is leased within a geographical region. In this article, data from four EC2 regions: sa-east, ap-northeast, eu-west and ap-southeast, which correspond to Amazon's data centers in Virginia, California, Ireland, and Singapore respectively, are used. Spot price trace files associated with the 8 different instance types, the 4 different regions, and 2 operating systems (Linux and Windows), has been analyzed.
Availability of a declared price is defined as the fraction of the time in which the spot price is equal to or lower than that declared price. If the request for an instance is persistent type, the instance is immediately re-requested on the occurrence of out-of-bid event (spot price rising above the bid price).

Given a declared price *D*, Availability of *D* is defined to be the time fraction in which a persistently requested instance would run if *D* is its declared price. Formally, let *H* be a spot price trace file, and let T_b and T_e be the beginning and end of a time interval within *H*. The availability of price *D* reflects the probability that spot instances with this bid would be immediately launched when requested at some random time within the given time period.

$$availability^{H}(D) = \frac{t}{Total\ time}$$

where *t* is the time for which the spot price is less than the bid price.
In every trace, it is assumed that the information like timestamp, product description (Linux/Windows), instance type, spot price, availability zone is given. For all data sets in a single input file, only the time and price of instance varies. So, a simple data parsing methodology is designed. This methodology takes input file and generates an output file which contains two specifications only for every data set: one the bid price and other the time for which it was above the spot prices in the data set.

B. Linear Regression

Linear regression is an approach to model the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted as *x*. In the case of one explanatory variable, the regression is called simple linear regression.

In the proposed approach, the bid price is the only independent variable and the corresponding availability is the dependent variable. The following equation is one of the many possible ways to represent the plotted curve.

$$y = \theta_0 + \theta_1 x$$

In this equation, y represents the availability of corresponding price *x*. For improving accuracy, one can plot higher order polynomials instead of using the linear plot.

B.1. Cost Function

The accuracy of the hypothesis function can be measured by using a cost function. This takes an average difference of all the results of the hypothesis with inputs from x and the actual output y. Squared error function is being used as cost function. The mean is halved as a convenience for the computation of the gradient descent, as the derivative term of the square function will cancel out the half term.

The objective is to get the best possible line. The best possible line will be such that the average squared vertical distances of the scattered points from the line will be the least. Ideally, the line should pass through all the points of our training data set. In such a case, the value of cost function will be zero [7]. The cost function represented in Figure 1, can be described as:

$$J(\theta_0, \theta_1) = \frac{\sum_{i=1}^{m}(h_\theta x_i - y_i)^2}{2m}$$

The cost function is thus the mean of the squares of the difference between the predicted value and the actual value. The number of examples in training is represented by *m*.

B.2. Gradient Descent

Gradient descent is a first-order iterative optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point.

So, now with the hypothesis function and the measuring mechanism of how well it fits on the data, estimation of the parameters in it, is required. Parameters are estimated through gradient descent algorithm. Imagine that hypothesis function is plotted based on its fields θ_0 and θ_1 (actually the cost function is graphed as a function of the parameter estimates). Plotting *x* and *y* itself is not done, but the parameter ranges of hypothesis function and the cost resulting from selecting a particular set of parameters. Derivative of the cost function is taken to minimize the cost function. The slope of the tangent is the derivative at that point and it will give a direction to move towards. Cost function is stepped down in the direction of the steepest descent. The size of each step is decided by the parameter *α*, that is called as learning rate.

The gradient descent equation for n features [8]:
repeat until convergence:
{

$$\theta_0 := \theta_0 - \frac{\alpha}{m} \sum\nolimits_{i=1}^{m} (h_\theta x^i - y^i)(x_0^i)$$
$$\theta_1 := \theta_1 - \frac{\alpha}{m} \sum\nolimits_{i=1}^{m} (h_\theta x^i - y^i)(x_1^i)$$
$$\theta_2 := \theta_2 - \frac{\alpha}{m} \sum\nolimits_{i=1}^{m} (h_\theta x^i - y^i)(x_2^i)$$
$$\ldots$$

}

Analysis of linear regression plots

For analysis, bid price prediction vs. availability is plotted. The instance type m1.large is used to plot the graph. In graph. on the *X*-axis bid price in dollars is plotted while on *Y*-axis, the predicted availability out of one is being represented.

A. Neural networks

Artificial neural networks (ANNs) is dependent on a huge collection of elementary neural units (artificial neurons), generally correspondent to the conduct of axons in human's mind. Each unit is connected with several other units, and these connections can enhance the activation state of adjacent units. To add the values of all the inputs collectively, these neural units may have function of summation. Each unit and connection may have a limiting function, so that, it is necessary to surpass the threshold before it can move to other neurons. Such systems are excellent in the fields where the discovery of features is hard to represent in a conventional computer program, because of its auto-learning and trained nature [9]. Architecture of neural networks is shown in Figure 10.

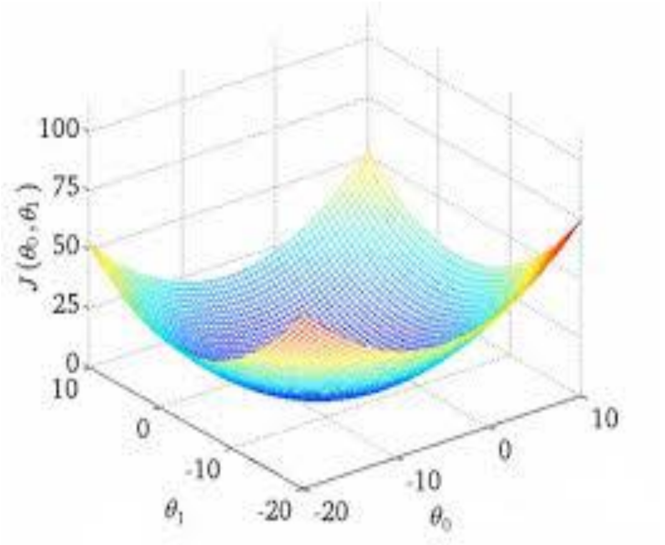

Figure 1. Linear regression cost function

The neural networks solve problems like human brain does. Advanced neural network consists of connection in millions and thousand to million neural units. This is still not as much complex as human brain and nearer to the computing power of an insect. New patterns in neural networks are often generated by some innovative research. One of the mechanism can be utilizing connections that span farther in comparison to use of adjoint neurons always. Another research can be the use of various types of signals in comparison to binary state i.e. on or off only. Neural networks are dependent on real numbers between 0.0 and 1 for the value of the core and axon.

B. Prediction of spot prices

A simple data parsing approach is designed which takes the input file and generates an output file that contains three specifications for every data set: the current bid price and the previous two bid prices. This is used as training examples for the mechanism. A hypothesis function is being represented using neural networks. Neurons are fundamental calculation units whose inputs are electrical inputs (dendrites) which are routed to outputs (axons). Dendrites are like the input features x_1 ... x_n, and the output is the outcome of the hypothesis function. A sigmoid (logistic) activation function represented as $\frac{1}{1+e^{-\theta^T x}}$, is used in neural networks. Neural network is trained by backpropagation algorithm which is used to minimize cost function. This algorithm is used in the same way as gradient descent is used in linear regression.

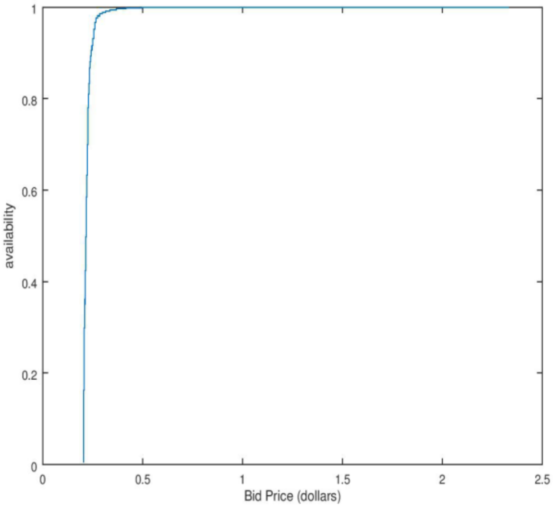

Figure 2. sa-east region data plot

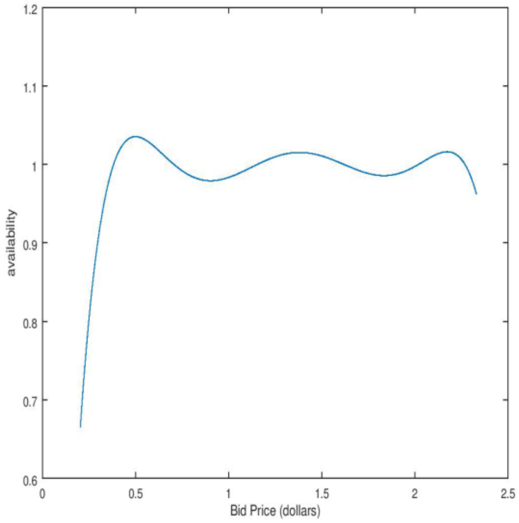

Figure 3. sa-east region prediction plot

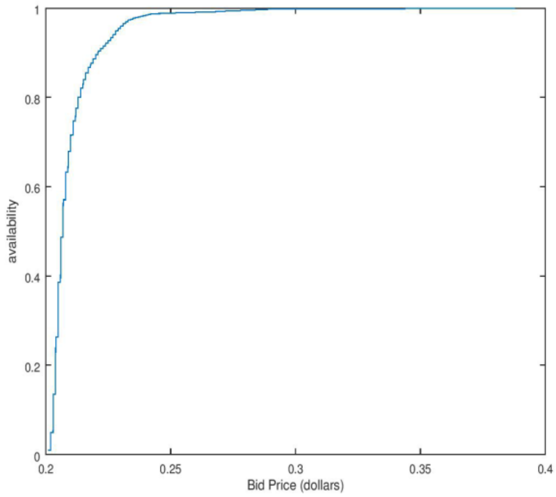

Figure 4. eu-west region data plot

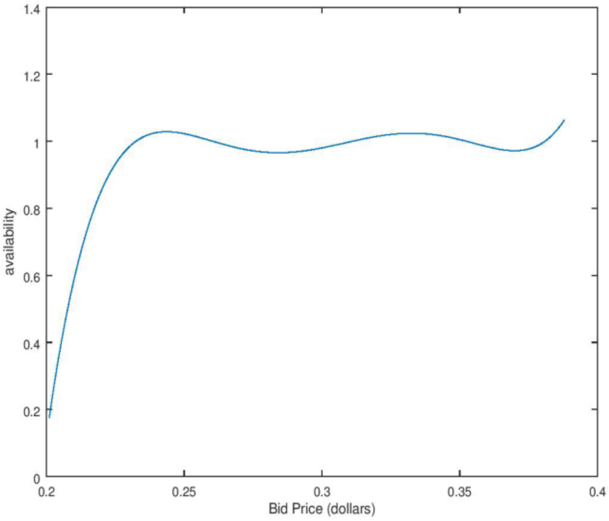

Figure 5. eu-west region prediction plot

Table 1. Error obtained in linear regression

Region	Mean Square Error
sa-east region	9.0713e- 04
eu-west region	8.5805e-04
ap-southeast region	5.6608e-03
ap-northeast region	8.5729e-04

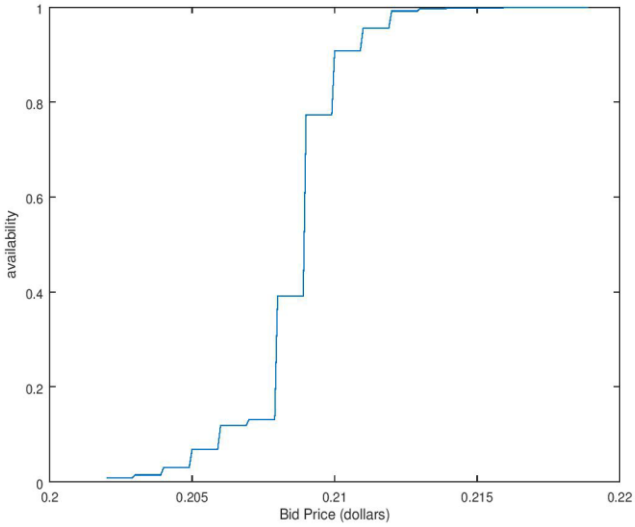

Figure 6. ap-southeast region data plot

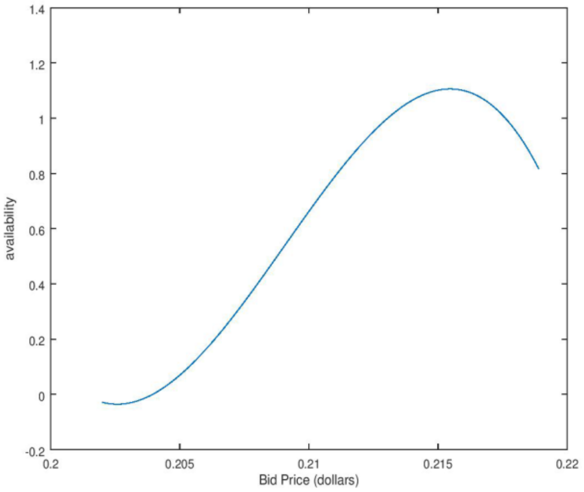

Figure 7. ap-southeast region prediction plot

Conclusion and Future Work

The linear regression model has been applied on one instance namely m1.large. This generates different results for different regions. The mean square error obtained in different regions for selected instance type for the period of time starting from December 18, 2015 to March 16, 2016 are tabulated in Tables 1 and 2.

Table 2. Error obtained in neural networks

Region	Mean Square Error
sa-east region	2.3466e-04
eu-west region	3.0000e-05
ap-southeast region	1.6676e-06
ap-northeast region	1.0892e-05

A. Limitations and Challenges

1. If the change in two consecutive spot prices is large, it is difficult to predict price accurately.

2. Since, we are only able to know about possible availability, due to uncertainty, the client has to take a calculated risk.

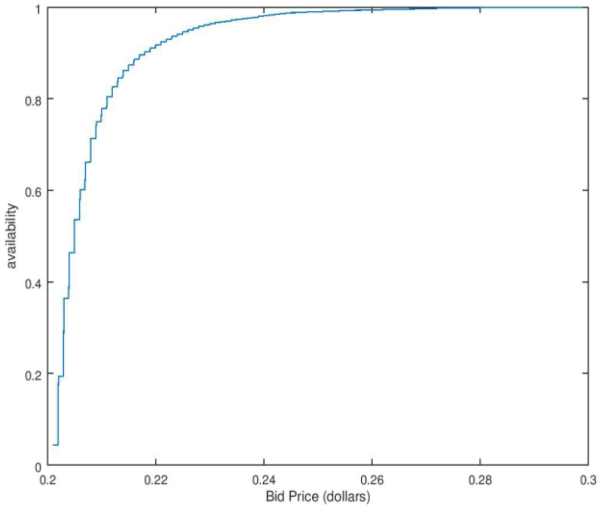

Figure 8. ap-northeast region data plot

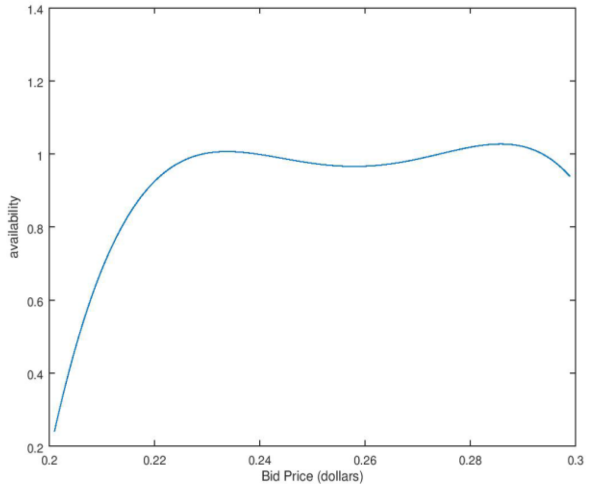

Figure 9. ap-northeast region prediction plot

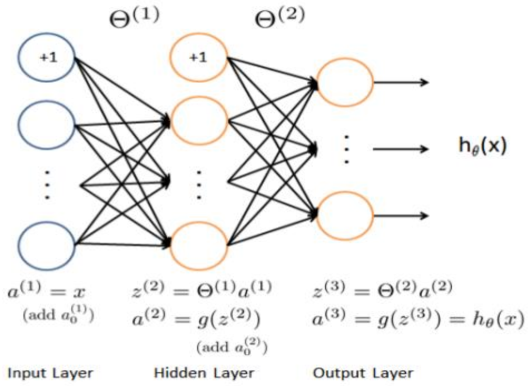

Figure 10. Architecture of Artificial Neural Networks

B. Future Work

The prediction has been done for a particular price to be launched in a specific duration of time. By applying neural network concepts, effort will be in the direction of designing a more specific method which uses the past price pattern and tries to find out the next price from the price history. New prices can be obtained from analysis of the past prices.

Graph of instance availability vs bid price for a single instance in a single region can be plotted. Further, the plot of instance vs availability can be plotted. Region vs instance availability for a single price and a single instance can also be plotted. This will helpful for analyzing the price in a better way.

Work migration allows to change the instance type in case of failures. Such application should be flexible enough to handle varying number of cores or processing units; however, in the case of divisible workloads this capability can be assumed. The idea is to bid for another instance type (than recently used one), if this new instance type can be acquired at a per-core price comparable to user's last bid price. Through migration, waiting time until recovery can be almost eliminated. This elimination can effectively reduce the job completion time without significant increase in cost.

REFERENCES

[1] Orna Agmon Ben-Yehuda, Muli Ben-Yehuda, Assaf Schuster, and Dan Tsafrir. Deconstructing amazon EC2 spot instance pricing. ACM Trans. Economics and Comput., 1(3):16, 2013. doi: 10.1145/2509413.2509416. URL http://doi.acm.org/10.1145/2509413.2509416.

[2] Matthew Burgess and Bryce Wiedenbeck. Strategic bidding on amazon ec2. 2010.

[3] Marco Abundo, Valerio Di Valerio, Valeria Cardellini, and Francesco Lo Presti. Qos-aware bidding strategies for VM spot instances: A reinforcement learning approach applied to periodic long running jobs. In IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Ottawa, ON, Canada, 11-15 May, 2015, pages 53–61, 2015. doi: 10.1109/INM.2015.7140276. URL http://dx.doi.org/ 10.1109/INM.2015.7140276.

[4] M. Mattess, C. Vecchiola, and R. Buyya. Managing peak loads by leasing cloud infrastructure services from a spot market. In 2010 IEEE 12th International Conference on High Performance Computing and Communications (HPCC), pages 180– 188, Sept 2010. doi: 10.1109/HPCC.2010.77.

[5] Sangho Yi, Junyoung Heo, Yookun Cho, and Jiman Hong. Taking point decision mechanism for page-level incremental checkpointing based on cost analysis of process execution time. J. Inf. Sci. Eng., 23(5):1325–1337, 2007. URLhttp://www.iis.sinica.edu.tw/page/jise/ 2007 /200709_01.html.

[6] Poisson regression, 2017.URL https://en.wikipedia.org/ wiki/Poisson_regression. Online; accessed 16 Septembe-r, 2017.

[7] Introduction, Regression Analysis, and Gradient Descent, 2017. URL http://www.holehouse.org/ mlclass/01_02_Introduction_regression_analysis_ and_ gr.html. Online; accessed 14 September, 2017.

[8] Linear Regression with Multiple Variables, 2017. URL http://www.iequa.com/2016/10/08/ml-courserang-w2-01-Linear-Regression/. Online; accessed 20 September, 2017.

[9] Artificial Neural Networks and the Future, 2017.URL http://ravg.org/analysis/ann-and-the-future/. Online; accessed 20 August, 2017.

[10] S. Di, Y. Robert, F. Vivien, D. Kondo, C. L. Wang, and F. Cappello. Optimization of cloud task processing with checkpoint-restart mechanism. In 2013 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (SC), pages 1–12, Nov 2013. doi: 10.1145/2503210.2503217.

[11] Itthichok Jangjaimon and Nian-Feng Tzeng. Adaptive incremental checkpointing via delta compression for networked multicore systems. In 27th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2013, Cambridge, MA, USA, May 20-24, 2013, pages 7–18, 2013. doi: 10.1109/IPDPS.2013.33. URL http://dx.doi.org/10.1109/IPDPS.2013.33.

[12] Sangho Yi, Derrick Kondo, and Artur Andrzejak. Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud. In IEEE International Conference on Cloud Computing, CLOUD 2010, Miami, FL, USA, 5-10 July, 2010, pages 236–243, 2010. doi: 10.1109/CLOUD.2010.35. URL http://dx.doi.org/ 10.1109/CLOUD.2010.35.

[13] Deepak Poola, Kotagiri Ramamohanarao, and Rajkumar Buyya. Enhancing reliability of workflow execution using task replication and spot instances. TAAS, 10 (4):30:1–30:21, 2016. doi: 10.1145/2815624. URL http://doi.acm.org/10.1145/2815624.

[14] Shaojie Tang, Jing Yuan, Cheng Wang, and Xiang-Yang Li. A framework for amazon EC2 bidding strategy under SLA constraints. IEEE Trans. Parallel Distrib. Syst., 25(1):2–11, 2014. doi: 10.1109/TPDS.2013.15. URL http://dx.doi.org/10.1109/TPDS.2013.15.

[15] S. Yi, A. Andrzejak, and D. Kondo. Monetary cost-aware checkpointing and migration on amazon cloud spot instances. IEEE Transactions on Services Computing, 5(4):512–524, Fourth 2012. ISSN 1939-1374. doi: 10.1109/TSC.2011.44.