DEEP LEARNING ASSISTED AUTOSCALING IN CLOUD COMPUTING

Anushka Kapure,
Student,
Information & Technology Engineering,
Silver Oak Collage of Engineering & Technology, Ahmedabad, India.

Abstract : A goal of Cloud service management is to design Self-Adaptive Auto-scalar to react to workload fluctuation and changing the work load assign. Auto-Scaling is the practice of automatically adding or removing resources for an application deployment to meet performance targets in response to changing Infrastructure as a Service (IaaS) provider's deliver threshold based auto-scaling techniques. However, setting up threshold with right values that minimize cast and achieve service level agreement is not an easy task, especially with variant and sudden work load changes. This thesis focus on full fledged Deep-Learning assisted Auto-scaling mechanism which is one of the successful technique as well as self configuring Deep-Learning techniques like Support Vector Machine (SVM), Convolution Neural Network (CNN), etc.

IndexTerms - Hybrid auto-scaling on clouds, Threshold base Auto-scaling, Scheduling with SLAs, Machine Learning, Deep Learning, Convolution Neural Network, Support Vector Machine, Cost Effectiveness.

I. INTRODUCTION:

Automated elasticity and dynamism, as tow important concepts of cloud computing, are beneficial for application owners. Auto-scaling system is a process that automatically scales the number of resources and maintains an acceptable Quality of Service (QoS) [1]. However, from the perspective of the user, determining when and how to resize the application makes defining a proper auto-scaling process difficult. Threshold-based auto-scaling approaches are proposed for scaling application by monitoring metrics, but setting the corresponding threshold conditions still rests with the user [6]. Recently, automatic decision-making approaches, such as Machine Learning techniques and Deep Learning techniques, have become more popular. The key advantage of these methods is that prior knowledge of the application performance model is not required, but they rather learn it as the application runs [2].

Our motivation here is to compare different auto-scaling services that will automatically and dynamically resize user application to meet QoS requirements cost-effectively.

We have discussed here two Machine Learning techniques Support Vector Machine (SVM) and Logistic regression and Deep Learning Technique Convolution Neural Network (CNN). Support Vector Machine (SVM) is an algorithm used for classification problems similar to Logistic Regression (LR). LR and SVM with linear Kernel generally perform comparably in practice [7].

The objective of the support vector machine algorithm is to find the hyperplane that has the maximum margin in an N-dimensional space (N—the number of features) that distinctly classifies the data points [21].In logistic regression, we take the output of the linear function and squash the value within the range of [0,1] using the sigmoid function(logistic function). The Sigmoid-Function is an S-shaped curve that can take any real-valued number and map it into a value between the range of 0 and 1, but never exactly at those limits. Typically, if the squashed value is greater than a threshold value we assign it a label 1, else we assign it a label 0. This justifies the name 'logistic regression' [22].

Main differences between these two techniques are Logistic loss diverges faster than hinge loss. So, in general, it will be more sensitive to outliers. Logistic loss does not go to zero even if the point is classified sufficiently confidently. This might lead to minor degradation in accuracy. LR is more sensitive to outliers than SVM because the cost function of LR diverges faster than those of SVM [21] [22].

The Deep learning (DL) as the name suggests is about stacking many processing layers one atop the other. The deeper the architecture is the more layers it has. The intuition behind DL comes from the compositional nature of natural stimuli such as speech and vision. Natural signals are highly compositional, that is, simple primitive features combine to form mid-level features while mid-level features combine to form high-level features [24].

So that the main difference between the Machine Learning Technique and the Deep Learning techniques are any structure that processes a signal in an increasingly abstract manner using many layers one atop the other is called a DL structure. The specific processing that occurs in each of the layers doesn't matter, DL is abstract in short. An SVM on the other hand is a binary classifier that learns by margin maximization [21] [22] [24].

II. RELATED WORK:

This section discusses a representative set of existing work on auto-scaling systems in cloud environments. "General auto scaling approaches" section discusses general auto-scaling approaches, "Hybrid auto-scaling approaches" section addresses work that highlights auto scaling performed via hybrid approaches.

Traditional techniques for auto-scaling are easy to deploy and use a mechanism to manage the number of resources assigned to an application hosted on a cloud platform (e.g., [1]). The ease-of-use of these rules make them appealing to cloud users. However, creating the rules for auto-scaling requires an effort from the application manager, who needs to select a suitable performance metric or logical combination of metrics and the values of several additional parameters, mainly thresholds that are used for scaling resources. Thresholds are the key to ensuring that the rules implemented are correct.

Hybrid auto-scaling approaches

The authors of [4] propose minimizing the violation in QoS. A novel base approach based on Performance Modeling and Probabilistic Verification. Uses Rule based Auto-Scaling Policies. The research presented in [5] focus on Over/Under VMs provisioning. Based on Self Adaptive Trade-Off Decision Making. Uses self designed method for decision making (MOACO + CD). The authors of [7] propose response time and work load pattern. Reactive approach for auto scaling for different types of work load patterns. Uses Reinforcement Learning Techniques. And in edition in our based paper [8] author has propose a real-time cloud capacity framework, offering a hybrid elasticity controller employing both reactive rule-based and proactive model-based elasticity mechanisms in a coordinated manner using the middle ware Broker system to make more reliable system for both proactive and reactive environment [26]. The hybrid controller examines the scale-up condition which, in a purely reactive auto-scaling environment, is used to acquire new resources, and builds incrementally updateable predictive models to enable a system to proactively scale up before this condition is met.

III. SYSTEM OVERVIEW:

This section presents an overview of the system with the public cloud provider. The section is divided into two subsections. The first subsection explains deep learning technique used for the decision making and the second section makes the predictive decision base on the results derived by the deep learning technique which is Convolution Neural Network.

Scalability is an important characteristics of cloud computing. Most of current Infrastructure as a Services (IaaS) providers deliver threshold based auto scaling techniques. However with variant and sudden workload changes affect the auto scaling. To overcome such a problems we proposed full fledge deep learning assisted self configured auto scaling mechanism based on predicted values for advance auto scaling mechanism and optimal VMs list. So that we can provide cost effective and time effective mechanism.

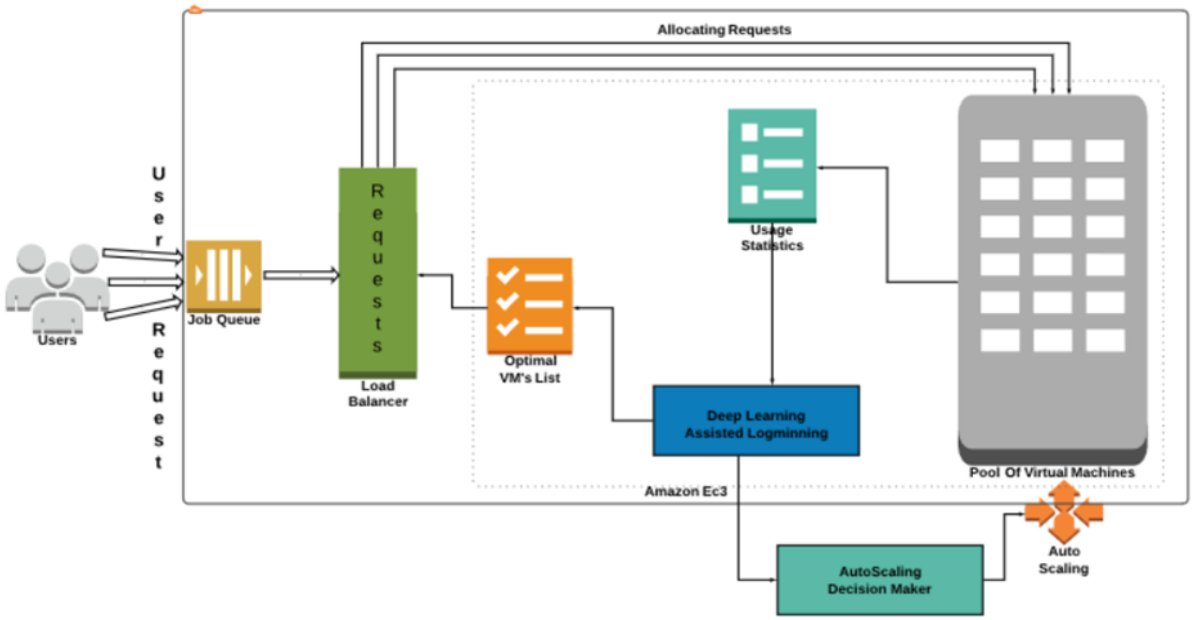

Figure 1 : System Overview

For this proposed architecture we have derived an algorithm. This algorithm shows the functionality and the working of the proposed system depth for reactive and proactive auto scaling system. This algorithm is derived below :

PROPOSED ALGORITHMS

Usage Statistics

Repeat at regular interval for all VMs

usage_statistics[] = [cpu_util, mem_util, latency]

apply de-noise[usage_statistic]

Deep Learning Assisted Logminning

Apply Deep-Learning on Dbase to generate optimal VMs list

if average usage of VMs >= threshold

call for Auto-Scaling

and make optimal VMs list simultaneous.

Here are the steps been explained of the system architecture and the algorithm working process.

STEP: 1 Monitor Virtual Machine Pool.

STEP: 2 Calculate usage statistics.

STEP: 3 Apply Preprocessing

STEP: 4 Store structured data in Database.

STEP: 5 Apply predictive methods.

STEP: 6 Generate Optimum VM's list.

STEP: 7 Check threshold and Average usage.

STEP: 8 Make Auto-scaling decisions.

For the same proposed model we have some performance assumptions. So basically we have shown the limitations of the proposed model which is noted below.

- Ignore VM setup latency.
- Fixed VM pool. (42/60 will be maximum VM numbers)
- Fetching only three parameters for decision making. (CPU Utilization, Memory Utilization, Latency)

IV. PERFORMANCE EVOLUTION:

This section describes the proposed model's performance and the results that we have got from the research experiments. This section will help us to choose the best proposed system for the analysis of the model in depth.

Here firstly we have derived some mathematical equations for the proposed algorithm. All the notations and mathematical equations are derived and described below.

Formally, the QoS objective model by the 11th interval is:

$$QoS_k^{ij}(t) = f_k^{ij}(SP_k^{ij}(t), \delta),$$

Where the selective primitive input matrix is $SP^{ij}{}_k(t)$ is :

$$SP_k^{ij}(t) = \begin{pmatrix} CP_a^{xy}(t) & \cdots & EP_b^{mn}(t-1) & \cdots \\ \vdots & \ddots & \vdots & \ddots \\ CP_a^{xy}(t-q+1) & \cdots & EP_b^{mn}(t-q) & \cdots \end{pmatrix}$$

Whereby q is the number of order. In this work, we dynamically create and update $SP^{ij}{}_k$ and the function $f^{ij}{}_k$ using the online QoS modeling approach in our prior work [1], [4], [8].

we update this selective primitive matrix to contain only the most significant cloud primitives that can influence the QoS, including those that cause considerably high level of QoS interference. The required historical data points are determined by a set of Deep learning algorithms [4], [8].

The total cost model for Sij can be represented as:

$$Cost^{ij} = \sum_{a=1}^{n} CP_a^{ij}(t) \times P_a,$$

where n is the total number of control primitive type that used by service-instance Sij to supports its QoS attributes

Following up on the derived mathematical equations we have generated the below results where in the first section we have shown the threshold meeting VMs ratio. In the second section we have given the graphs for the performance matrix and discussed about the Mean Square Error method to find the accurate results for our proposed system.

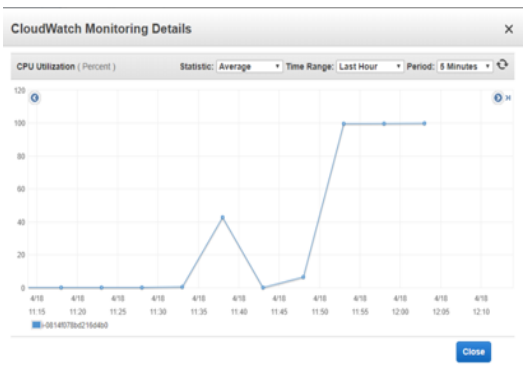

Figure 2: Cloud Watch Monitoring Details

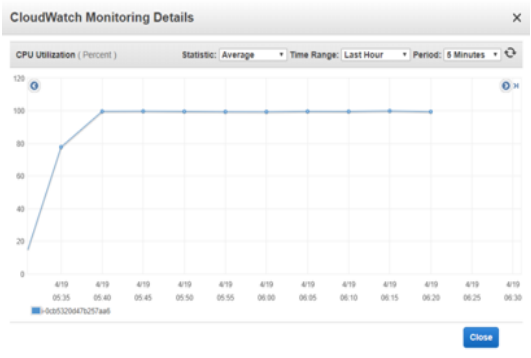

Figure 3: Cloud Watch Monitoring Details

This tow above graphs defines the virtual machine's threshold values according to the RAM/Memory Usage and the CPU Usage of the particular Server over the network. By the time we get the different results for the threshold values. By which we can decide that we need to auto scale the resources or not.

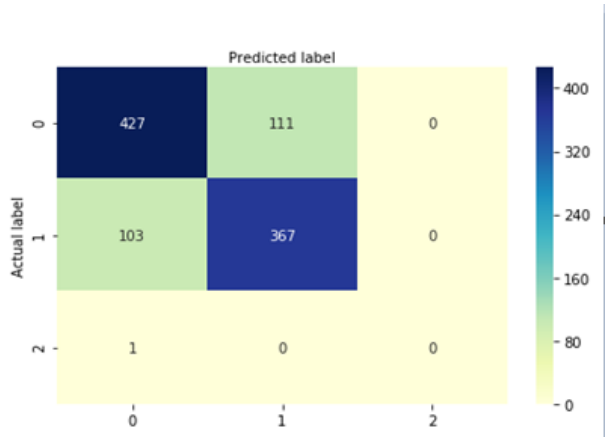

Figure 4 : Confusion Matrix Logistic (CNN)

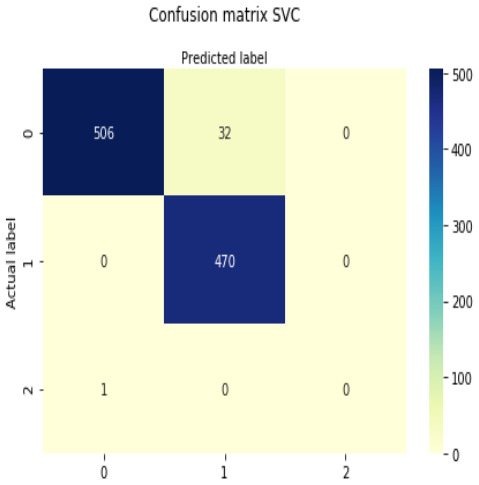

Figure 5 : Confusion Matrix Logistic (SVC)

Above we have generated confusion matrix to summarize the performance of the algorithms. This shows the accuracy of the algorithm by comparing the Prediction Values and the Actual Value logs by the time. So by seeing the results we can conclude that the CNN algorithm gives the more accuracy compared to the SVC algorithm.

Here we arediscussing about the Mean Square Error Equation. This function gives us the estimated error rate of the algorithm. MSE is a risk function, corresponding to the expected value of the squared error loss. The blow equation has been defined for MSE.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

So that we have made the conclusion graphs for the cost effectiveness of the algorithms. This explains the cost effectiveness of the system we have proposed in compare to other machine earning techniques. Here the green line shown for the CNN algorithm and the red line shows the results for the SVM. So that we can clearly say that the cost effectiveness of the CNN is far batter then the SVM algorithm.

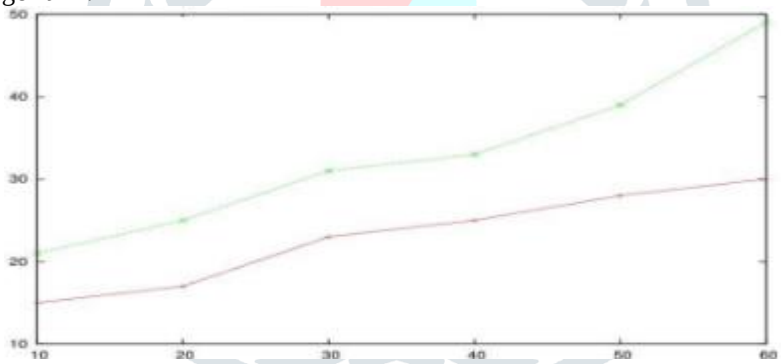

Figure 6: Cost effectiveness of algorithm (SVM and CNN)

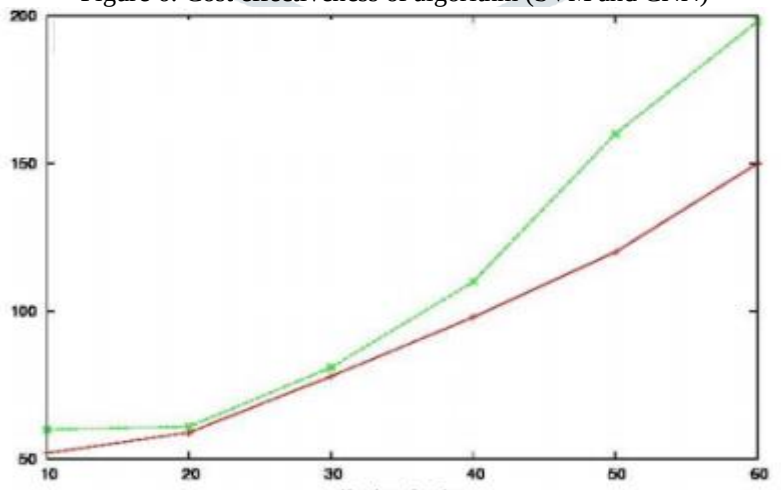

Figure 7: Cost effectiveness of algorithm (SVM and CNN)

V. CONCLUSION:

This technique and the associated algorithms with proposed architectural framework for performing auto-scaling of resources in a cloud. The auto-scaling technique combines Deep-Learning based proactive approach for scaling resources to adapt changes in workload demands. So that we can achieve higher profit by using this hybrid system which may works batter then any proactive or reactive system in the most cases.

VI. FUTURE DIRECTION:

Approaches the best Deep-Learning technique so that we can create reliable hybrid system. An output of the Deep Learning technique has to integrate with SDK of AWS and has to integrate it with the continuous data prediction input to JSON as the final product which is another interesting topic for future research.

REFERENCES:

[1] Evangelidis A, Parker D, Bahsoon R. Performance modelling and verification of cloud-based auto-scaling policies. Future Generation Computer Systems. 2018 Jan 10.

[2] Chen T, Bahsoon R. Self-adaptive trade-off decision making for autoscaling cloud-based services. arXiv preprint arXiv:1608.05917. 2016 Aug 21.

[3] Arabnejad H, Pahl C, Jamshidi P, Estrada G. A comparison of reinforcement learning techniques for fuzzy cloud auto-scaling. InProceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing 2017 May 14 (pp. 64-73). IEEE Press.

[4] Persico V, Grimaldi D, Pescape A, Salvi A, Santini S. A Fuzzy Approach Based on Heterogeneous Metrics for Scaling Out Public Clouds. IEEE Transactions on Parallel and Distributed Systems. 2017 Aug 1;28(8):2117-30.

[5] Hu Y, Deng B, Peng F. Autoscaling prediction models for cloud resource provisioning. InComputer and Communications (ICCC), 2016 2nd IEEE International Conference on 2016 Oct 14 (pp. 1364-1369). IEEE.

[6] Shariffdeen RS, Munasinghe DT, Bhathiya HS, Bandara UK, Bandara HD. Adaptive workload prediction for proactive auto scaling in PaaS systems. InCloud Computing Technologies and Applications (CloudTech), 2016 2nd International Conference on 2016 May 24 (pp. 22-29). IEEE.

[7] Wajahat M, Gandhi A, Karve A, Kochut A. Using machine learning for black-box autoscaling. InGreen and Sustainable Computing Conference (IGSC0< 2016 Seventh International 2016 Nov 7 (pp. 1-8). IEEE.

[8] Biswas A, Majumdar S, Nandy B, El-Haraki A. A hybrid auto-scaling technique for clouds processing applications with service level agreements. Journal of Cloud Computing. 2017 Dec;6(1):29.

[9] Zhang Q, Cheng L, Boutaba R. Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications. 2010 May 1;1(1):7-18.

[10] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," in High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on, 2008, pp. 5–13.

[11]Wikipedia [online]. Available : https://en.wikipedia.org/wiki/Cloud_computing

[12]Amazon, "What is Cloud Computing by Amazon Web Services | AWS." [Online]. Available: http://aws.amazon.com/what-is-cloud-computing/

[13]US Department of Commerce, "Final Version of NIST Cloud Computing Definition Published." [Online]. Available: http://www.nist.gov/itl/csd/cloud-102511.cfm.

[14]Cisco Systems, "Cloud Computing - Overview." [Online]. Available: http://www.cisco.com/web/solutions/trends/cloud/index.html.

[15]What is cloud.com [online] available: http://www.whatiscloud.com/cloud_deployment_models/index

[17] cheesecakelabs.com [online]. Available:https://cheesecakelabs.com/blog/cloud-scaling-1k-to-1b-users/

[18] kdnuggets.com[online]. Available : https://www.kdnuggets.com/2016/04/deep-learning-neural-networks-overview.html

[19] www.analyticsvidhya.com . [online]. Available : https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[20] scholarpedia.org : [online]. Available : http://scholarpedia.org/article/Support_vector_clustering

[21] Wikipedia.org : [online]. Available : https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[22] Wikipedia.org : [online]. Available : https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[23]Logistic Regression Algorithm : https://en.wikipedia.org/wiki/Logistic_regression

[24] CNN : https://en.wikipedia.org/wiki/Convolutional_neural_network

[25] Research Approach: https://www.idi.ntnu.no/grupper/su/publ/html/totland/ch013.htm

[26] Virtualization : http://www.mirazon.com/high-availability-series-virtualization-configuration/traditionalvsvirtual/

[27] Virtualization : https://en.wikipedia.org/wiki/Virtualization