© 2016 IJEDR | Volume 4, Issue 4 | ISSN: 2321-9939

A Survey on various Auto-scaling mechanism in cloud environment

[1]Aayushi Chaudhari, [2]Prof. Gayatri Pandi
[1]M.E. Student, [2]Professor
Computer Engineering Department,
L.J College of Eng. & Tech., Ahmedabad, India

***Abstract*—Cloud Computing is emerging technology nowadays. Scalability is an important aspect in cloud computing especially, when the demands of user changes abruptly. Auto-scaling is the strategy that has ability to adjust the available resources to meet the user demands. As the popularity and usage of cloud computing has increased, it leads to highly unpredictable demands of users. One major issue about providing automatic scalability is that it considers user-defined threshold, which cannot respond to real time internet traffic loads. In this paper a model is proposed which includes scaling of resources by means of infrastructure as a code. This mechanism will dynamically create the high level language code to automate provisioning of resources based on availability of existing resources and user demands. Based on the code generated, the virtual machines will shrink or expand.**

***Index Terms*—Bayesian Network, Auto-Scaling, Cloud Watch, Auto-scaling Demand Index, Hidden Markov Model**

I. INTRODUCTION

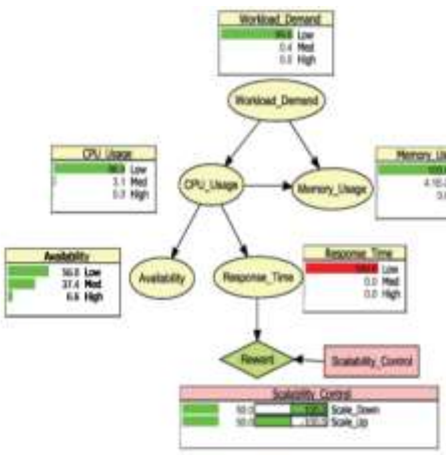
Cloud computing offers small and big organizations, the opportunity to scale their computing resources. It is done by either increasing or decreasing the required resources. Cloud computing provides delivery of resources on demand over the internet. It also provides the users to store and access the data stored by them on cloud. It provides metered service, so that users are asked to pay only for what they use. Cloud provides elasticity by scaling up as computing needs increase and then scaling down again as demands decrease.

Amazon Cloud Watch Monitor is a monitoring service for AWS cloud resources and the applications you run on AWS. Amazon Cloud Watch collects and monitor log files, set alarms, and automatically react to changes in AWS resources. Amazon Cloud Watch Monitor (CWM) is used to monitor resource utilization and application performance.

Infrastructure as Code (IAC) is a type of IT infrastructure that operations teams can automatically manage and provision through code, rather than using a less flexible or manual process. Infrastructure as Code is sometimes referred to as programmable infrastructure. It does configuration through machine-processable definition files, rather than physical hardware configuration or the use of interactive configuration tools.

II. RELATED WORK

[1] In first paper, Abul Bashar proposed Bayesian Networks based predictive modeling framework that captures the historical behavior of the system involving various performance metrics. He used two modules, namely, the Datacenter Management System (DMS) and the Decision Support System (DSS) to create the model to make the decisions and also to predict future behavior by estimating the desired metrics of interest. Then this learned model is transformed to ID by adding decision node to it i.e. Scalability_Control with two actions Scale_Down and Scale_Up and a utility node named Reward. So the ID is responsible to make the decision to scale up or scale down based on the Response_time. So, if the ID makes correct decision then it is rewarded and if it makes wrong decision then it gets the penalty.



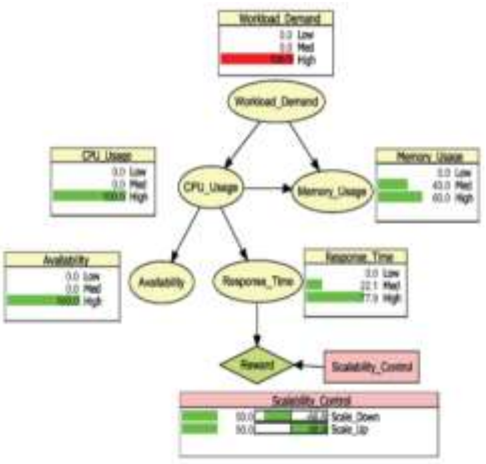


Fig. 1- Workload_Demand node having a High value. ID rightly decides to Scale Up the resources(+88.9) [1]
Fig. 2- Response_Time is Low, ID is now scaling down(+100) [1]

[2] In second paper, Marco A. S. Netto, Carlos Cardonha, Renato L. F. Cunha, Marcos D. Assuncao evaluates various autoscaling strategies uses Auto-scaling Demand Index (ADI) that reports utilization level, if the auto-scaling demand Index is above the target utilization then QoS is penalized and if the auto-scaling demand Index is below the target utilization. So, the main goal is to reduce the gap between actual utilization and the target interval. Here, the two main challenges are executed they are: first one decision on when to trigger auto-scaling operations and second is the step size that is used to expand or shrink the resource pool. It included three strategies for triggering auto-scaling operations that are reactive, conservative and predictive and two strategies for setting auto-scaling step sizes are fixed and adaptive. Reactive strategy had an excellent performance as it always reacts immediately to any deviation from the desired utilisation interval.

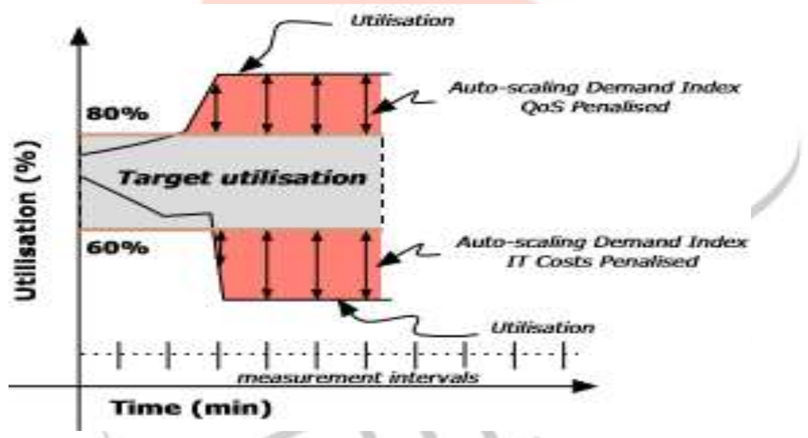


Fig. - 3 Auto-scaling Demand Index (ADI) metric [2]

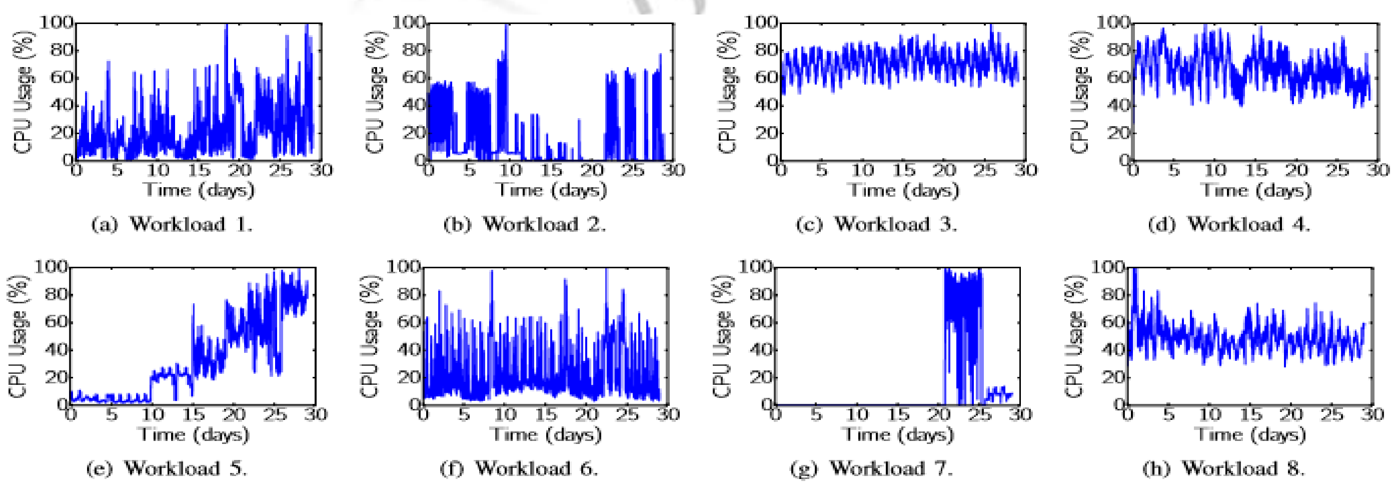


Fig. 4 - Workloads after clustering users by their CPU usage pattern into eight groups [2]

Different trace logs publically available from production data center cluster by Google, this are the records of user jobs submitted to the cluster, their CPU usage and duration.

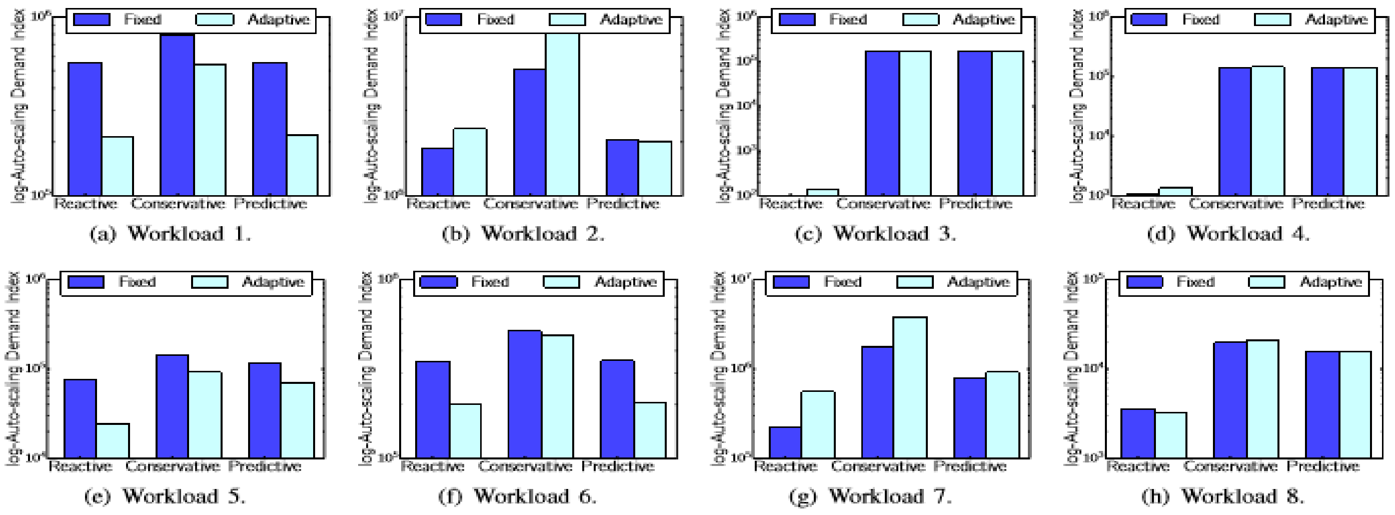


Fig. 5 - Summary of Auto-scaling Demand Index (ADI) [2].

Different trace logs publically available from production data center cluster by Google, this are the records of user jobs submitted to the cluster, their CPU usage and duration. Based on that tree of the strategies are applied along with step sizes to check out the observations. The **Adaptive strategy was better** for Workloads 1, 5, and 6 to their bursty and peaky behaviours. But in certain scenarios we can see that fixed performs better than adaptive.

[3] In third paper, Ali Yadavar, Nikravesh Samuel, A. Ajila, Chung-Horng Lung, conducted experiment on Amazon EC2 infrastructure using Hidden Markov Model (HMM). **CPU utilization, throughput, and response time** are being considered as performance metrics in this experiment. There are two broad categories of auto-scaling systems: **reactive and proactive (predictive**). Reactive auto-scaling approaches react to the system changes but do not anticipate future changes. Proactive scaling approaches try to predict future system behavior and adjust application resources in advance to meet the future needs. To generate historical data, they have used TPC-W benchmark as load generator for 5 hours. After collecting the data it is divided into training and testing data. It considers different metrics such as **Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE).** It compared scaling decisions of cloud watch and Hidden Markov Model.

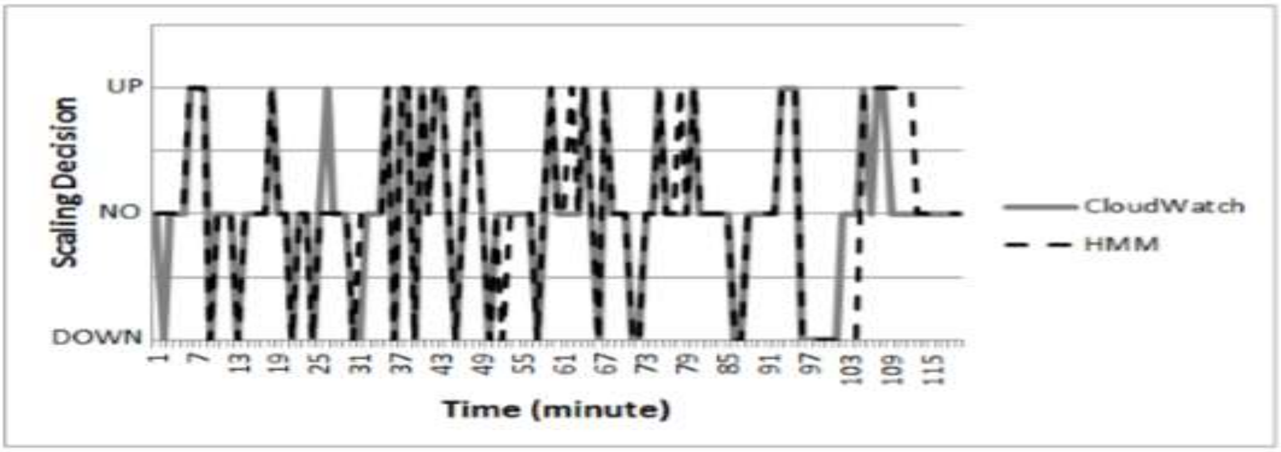


Fig. 6 - HMM and CloudWatch scaling decisions [3]

[4] In fourth paper, Ali Yadavar Nikravesh, Samuel A. Ajila, Chung-Horng Lung proposed a model to increase the prediction accuracy of auto-scaling systems by choosing an appropriate time-series prediction algorithm based on the performance pattern. Workload is considered as the performance metric. Support Vector Machine (SVM) and Neural Networks (NN) were utilized as time-series prediction techniques. Amazon EC2 is used as the experimental infrastructure.

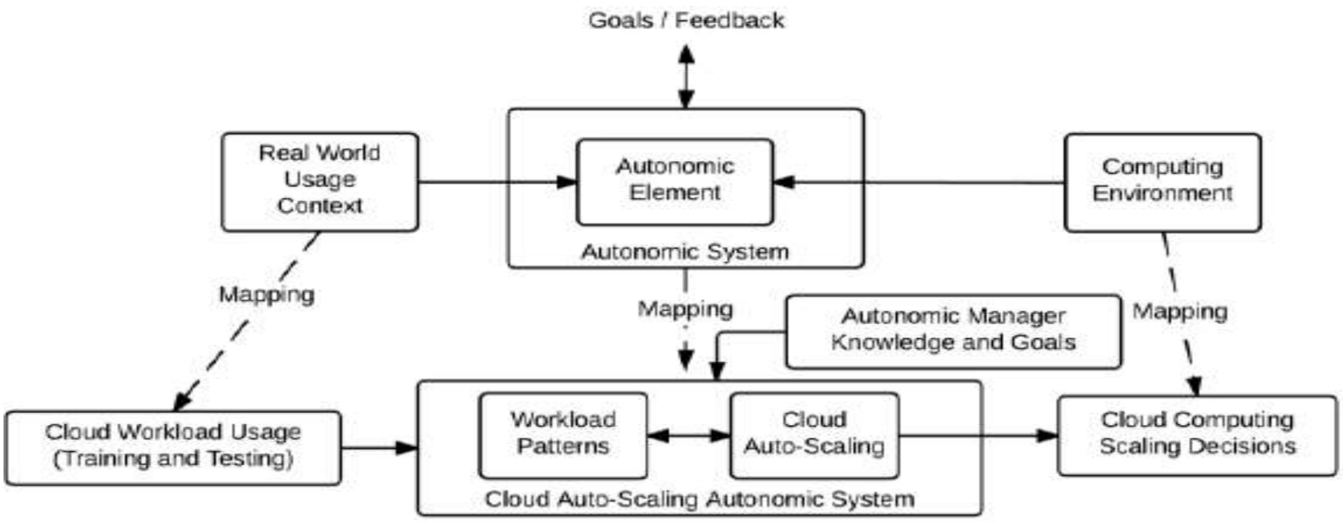


Fig. 7 - Cloud workload scaling autonomic system [4]

Here, an autonomic element will **regularly sense sources of change** by using "**sensors**" or "**reasons**" about the **current system situation** for possible **adaptation** and **action** for the future. **Cloud workload context**; cloud auto scaling autonomic system consisting of two meta-autonomic elements they are workload pattern and cloud auto scaling. Autonomic manager applies the domain specific knowledge linked to the cloud workload pattern and apply the appropriate predictor algorithm. Then, predictor and workload interact to implement the chosen algorithm.

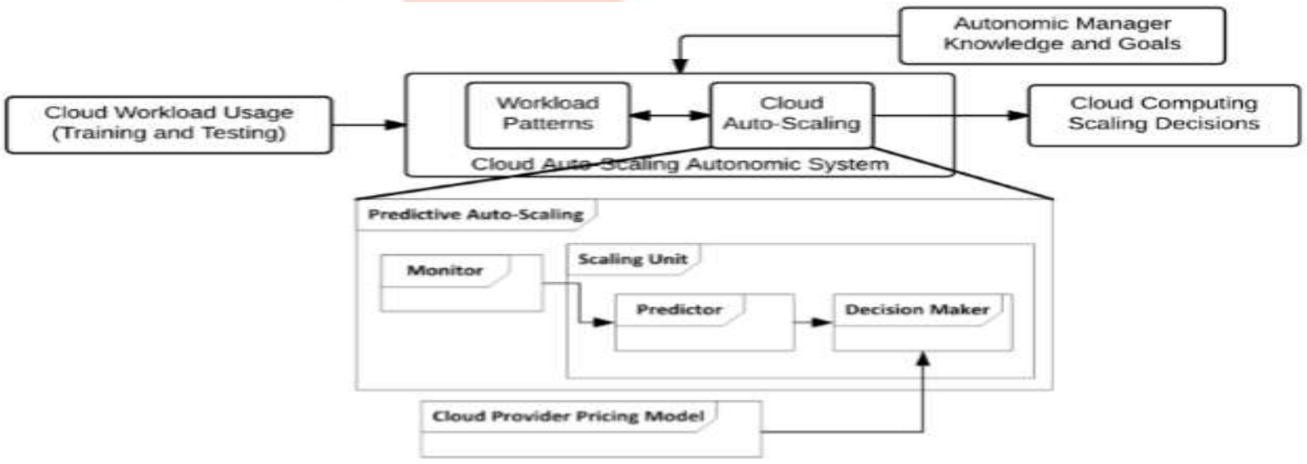

Autonomic Manager Knowledge and Goals
Cloud Workload Usage (Training and Testing)
Workload Patterns
Cloud Auto-Scaling
Cloud Computing Scaling Decisions
Cloud Auto-Scaling Autonomic System
Predictive Auto-Scaling
Monitor
Scaling Unit
Predictor
Decision Maker
Cloud Provider Pricing Model

Fig. 8 - Projection of the cloud auto-scaling autonomic element [2]

[5] In fifth paper, Wen-Hwa Liao, Ssu-Chi Kuai and Yu-Ren Leau, proposed dynamic threshold adjustment strategy that can expedite the creation of virtual machines according to workload demands.Their main goal is to reduce the web application response time and error rate when the system is under a heavy workload. It can expedite the release of virtual machines to reduce virtual machine running time when the system is under a light workload. Firstly, Resource usage and CPU utilization values are monitored using the Cloud Watch service of AWS. Then this values are transferred to a dynamic threshold controller for analysis. The controller transmits a timely policy and threshold to the virtual machines for providing computing resources.

- For auto-scaling of resources, it is necessary to select the upper threshold to increase the number of virtual machines and a lower threshold to reduce the number of virtual machines.
- The **upper threshold** of CPU resource utilization should be set to approximately 50%–75% for divergent web application.
- The dynamic **lower threshold** of CPU utilization was designed in the range 5%–30%. For responding to high bursty workload demand of web applications, method of dynamically minimizing upper threshold is applied.
- This **upper threshold decreases by 0.5%**, when the proportion of virtual machine increases by 1%.

- The objective of dynamically minimizing the lower threshold is to reduce it to a reasonable value when the machines are no longer idle. The **lower threshold increases linearly by 0.5%** when the number of released virtual machines increases by 1%.

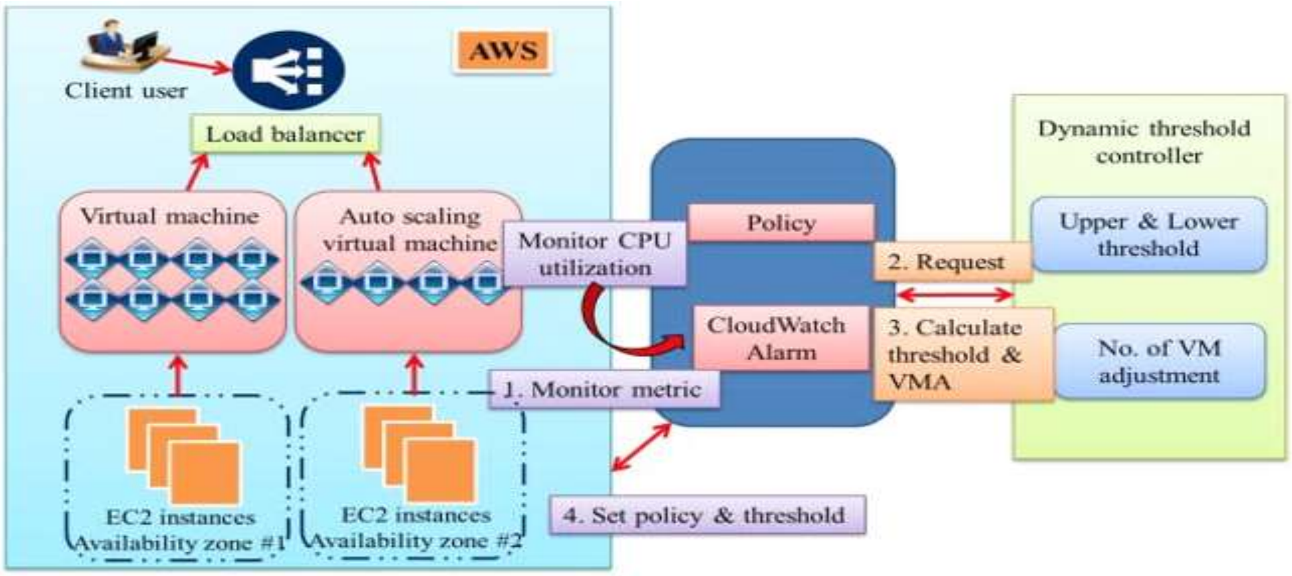

Client user
AWS
Load balancer
Virtual machine
Auto scaling virtual machine
Monitor CPU utilization
Policy
CloudWatch Alarm
2. Request
3. Calculate threshold & VMA
Dynamic threshold controller
Upper & Lower threshold
No. of VM adjustment
1. Monitor metric
EC2 instances Availability zone #1
EC2 instances Availability zone #2
4. Set policy & threshold

Fig. 9 - System Architecture and its flow [5]

III. Conclusion

From one of the conclusions that can be extracted from this survey with Auto Scaling ensure that the number of instances is increasing seamlessly during demand to maintain performance, and decreases automatically during demand reduce to minimize costs. The system should be able to adapt to the customer request so as to increase resources or decrease the resources, so as to maintain the balance between performance and cost effectiveness. Improving the performance and utilization of the cloud systems are gained by the auto-scaling of the applications.

References

[1] Abul Bashar. “Autonomic Scaling of Cloud Computing Resources using BN-based Prediction Models Sensor Network” (2013) IEEE Conference Publications DOI: 10.1109/CloudNet.2013.6710578

[2] Marco A. S. Netto, Carlos Cardonha, Renato L. F. Cunha, Marcos D. Assuncao. “Evaluating Auto-scaling Strategies for Cloud Computing Environments ”. (2014) IEEE Conference Publications DOI: 10.1109/MASCOTS.2014.32

[3] Ali Yadavar, Nikravesh Samuel, A. Ajila, Chung-Horng Lung. “Cloud Resource Autoscaling System based on Hidden Markov Model (HMM)”(2014) DOI: 10.1109/ICSC.2014.43IEEE Conference Publications

[4] Ali Yadavar Nikravesh, Samuel A. Ajila, Chung-Horng Lung. “Towards an Autonomic Auto-Scaling Prediction System for Cloud Resource Provisioning." (2015) 2015 10th International Symposium, DOI: 10.1109/SEAMS.2015.22

[5] Wen-Hwa Liao, Ssu-Chi Kuai and Yu-Ren Leau. " Auto Scaling Strategy for Amazon Web Services in Cloud Computing." (2015) 2015 IEEE International Conference. DOI: 10.1109/SmartCity.2015.209