International Conference on Machine Learning and Computational Intelligence-2017
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 4 | Issue 4 | ISSN : 2456-3307

A Comparative Analysis of Various Auto-Scalers in the Cloud Environment

Dhrub Kumar[1], Naveen Gondhi[2]
[1]Scholar, Department of Computer Science and Engineering, Shri Mata Vaishno Devi University, Katra, Jammu, India
[2]Assistant Professor, Department of Computer Science and Engineering, Shri Mata Vaishno Devi University, Katra, Jammu, India

ABSTRACT

The IaaS service model offers resources to its customers in the form of virtual machines (VMs) on a pay per use basis. These days, large enterprises and even small and medium businesses (SMBs) have started deploying their applications on clouds due to the various advantages it offers. The elastic feature of the clouds lets the deployed applications to scale their resources in accordance with the workload demands. This ensures that the applications provide the guaranteed QoS to its users as specified in the SLAs. To handle the automatic acquiring and releasing of resources as per application workload demands in the cloud environment (auto-scaling), various techniques have been proposed by researchers in the past. This paper performs a comparative analysis of various auto scaling techniques in cloud with respect to a number of factors viz. scaling technique, scaling type, scaling timing, and workload nature.

Keywords : Auto-scaling, Application Provisioning, Cloud Computing

I. INTRODUCTION

Providing on-demand, scalable and virtualized resources to its customers in a pay per use fashion are some of the key features of cloud computing. Many companies are shifting towards clouds for deploying their applications to avoid over-provisioning or under-provisioning of resources and to balance the cost-performance trade off [1]. The elastic feature of clouds is attracting large enterprises and even small and medium businesses (SMBs) to host their web applications on cloud infrastructures so as to handle varying workload demands. This, in turn, leads to improved QoS guarantees and reduced rental costs. For example, Animoto – an image processing web application experienced a sudden increase in workload requests that it has to increase its number of instances from 50 to 4000 in just three days in April, 2008 [2]. This way Animoto scaled up its resources to guarantee performance to its end users and later on scaled down its resources to reduce costs. Application deployment on cloud infrastructures brings many challenges. Ensuring automatic provisioning of sufficient amount of resources to application instances according to the current workload demands taking into account performance and cost constraints is one of the challenges. Fig. 1 demonstrates the fluctuations in requests to the FIFA 1998 Soccer World Cup website. The fluctuations in requests depend on a number of factors like what time of day is it, what day of week is it, and other seasonal factors.

Allocating suitable resources for such a workload is quite challenging. If resources are allocated according to average workload (under-provisioning), then cost

of renting resources from cloud is low but at the same time performance will be affected as the end users may experience long delays or service unavailability. On the other hand, if resources are allocated according to peak workload (over-provisioning), then application QoS requirements will be met but at a higher cost as resources will remain idle most of the times. To tackle these problems of over and under provisioning and under provisioning, auto scaling is employed in cloud environments. From the application provider's perspective, lack of both expert knowledge about application dynamics and modeling expertise complicate the scaling of the cloud hosted applications [4].

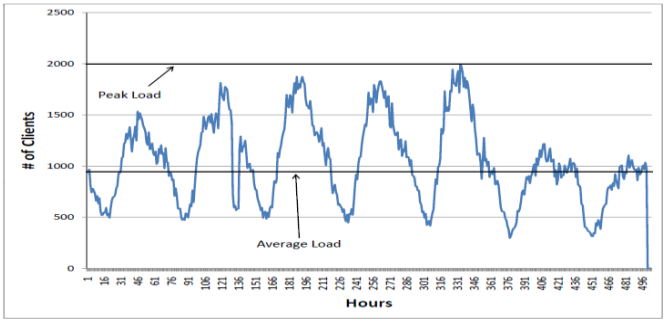

Figure 1. Workload of Soccer World Cup 1998

In section 2, the concept of auto-scaling is explored with respect to cloud environment. The work related to auto-scaling is summarized in section 3. Section 4 performs comparative analysis of various successful auto-scaling models proposed by various researchers in the past. Finally, section 5 concludes the paper.

II. AUTO-SCALING IN CLOUDS

In this section, we present the various concepts related to auto-scaling in cloud environment. We discuss what auto-scaling means in a cloud environment, the direction of scaling – horizontal or vertical, the timing of scaling – reactive or proactive, and the techniques used for auto-scaling.

Auto-Scaling

The process of acquiring and freeing resources as per application's workload demands in a dynamic, automatic fashion that takes into account resource costs and performance guarantees is called auto-scaling. According to [6], auto-scaling ensures that an application has the correct number of Amazon EC2 instances to handle the application's load. Fig. 2 shows two different ways of provisioning resources to application's workload [7]. The left portion shows the traditional way of provisioning resources where a business gradually increases its in-house infrastructure capacity to meet increasing application demands. On the right side of the graph, the cloud model allows business applications to scale resources up or down in line with the application's workload. This leads to better performance and reduces cost of renting infrastructure.

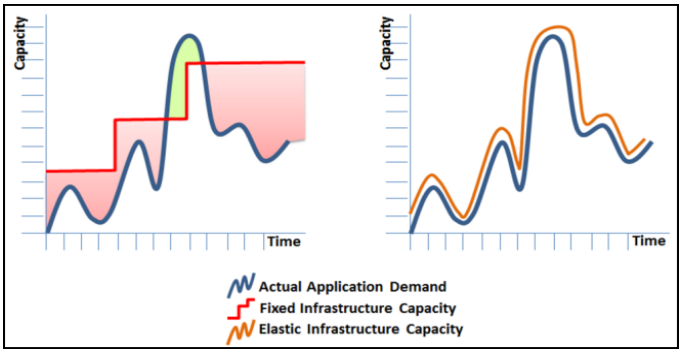

Figure 2. Traditional model versus cloud capacity model

Static versus Dynamic Provisioning

Auto-scaling uses the dynamic provisioning approach. Unlike dynamic provisioning, where resource allocation may be changed during runtime, the static provisioning keeps the resources assigned to an application fixed i.e. adding or removing new VMs is not done even if a change in application workload is detected [3]. Under dynamic provisioning, there are two ways of scaling resources in response to changing workload demands viz. horizontal and vertical. Horizontal scaling deals with adding new VMs or removing the allocated ones. On the other hand,

vertical scaling configures the resources (CPU, Storage etc) assigned to an already allocated VM. Vertical scaling uses a technique called as hot plug, which changes configuration of a VM on the fly without requiring it to shut down. According to [5], vertical scaling is better than horizontal scaling as VM instance acquisition time is shorter in vertical scaling.

Reactive versus Proactive Scaling

This relates to the timing of performing auto-scaling in cloud environments. The auto-scaler can be reactive or proactive. In case of reactive scaling, application is scaled only when certain pre-defined conditions are met, for example- when the CPU utilization stays over 80% for 2 minutes. This scheme is rule-based and often requires setting threshold values on part of the user and when these predefined thresholds reach certain values, some scaling action is triggered [8]. In comparison, proactive scaling relies on making predictions about future workload demands and then provisioning or de-provisioning resources accordingly.

Auto-scaling Techniques

A number of approaches have been tried in past by various researchers for implementing auto-scaling systems. Each implementation has its own scenario viz. the objectives to meet, the application architecture, the scaling parameters, the scaling method etc. In the past, researchers have used the following techniques to build an auto-scaling system:

1. Rule-based approach
2. Time Series analysis
3. Queuing theory
4. Control theory
5. Machine learning

1. Rule-based approach:

This technique is purely reactive, simple and easy to implement. It requires application providers to specify scaling indicators and set threshold values for these. On occurrence of the specified event, scaling action is triggered. Amazon's Auto-scaling Service is also rule-based [9]. Rule-based approaches are less accurate as they take scaling action after the workload changes. Also, deciding selected thresholds for the application is also a challenging task and requires a deep understanding of the nature of workloads. In [11], Dutreilh et al. emphasized the careful tuning of thresholds to avoid oscillations in the system. To tackle this, a cool-down period is set during which no scaling decisions is allowed once a scaling action has been implemented.

2. Time Series analysis:

Most auto-scalers are exploiting the timely patterns associated with cloud workloads like day, week or month, for forecasting future workload requests. The methods commonly used for forecasting future workloads include:

a. Moving Average
b. Auto-regression
c. ARMA (auto-regressive moving average)
d. ARIMA (auto-regressive integrated moving average)
e. Exponential Smoothing

In [10], authors have evaluated various forecasting methods using Google cluster data and Intel NetBatch logs for predicting future workloads in cloud environment. Their findings suggest that no method is always accurate and the accuracy of the prediction made by a particular method depends on the frequency and type of the workload.

3. Queuing theory:

Queuing theory has been used in auto-scaling environments to predict future resource requirements by modeling the system. Queueing theory deals with the study of waiting lines or queues in mathematical form. Queueing theory uses probabilistic methods in order to predict queue length or average waiting time of workload requests in a cloud environment. In 1953, Kendall represented

queuing model using the notation A/S/C, where A is the time between the arrivals, S is the time needed to service the job, and C represents the number of servers. Queuing model relies on online monitoring or other different methods in order to estimate parameters such as the input workload or service time [20].

3. Control theory:

Control theory controls an object by treating it as an input/output system, where the input corresponds to the control knobs and the outputs correspond to the metric being monitored [31]. Control theory has been widely used for designing auto-scalers in the cloud environment. Control systems can be classified as: open-loop, feed-back and feed-forward. Out of these, feed-back controllers are mostly used for auto-scalilng. Control theory works by first creating the application model in order to adjust the resources dynamically as per agreed SLAs. Control system should be adaptive to varying workload characteristics or the application itself. Control systems work in both reactive and proactive modes [32].

5. Machine learning:

Machine learning (ML) is closely associated with artificial intelligence, data mining and pattern recognition and has been broadly classified into supervised learning, semi-supervised learning and unsupervised. ML requires creating empirical models in order to understand application dynamics and make precise predictions. Various machine learning techniques like support vector machine, linear regression, neural networks, reinforcement learning etc were used by researchers as a predictive tool to make future workload predictions in cloud environment. In [22], authors observed that SVM provides more accurate results as compared to neural networks and linear regression models in terms of response time and throughput.

In [34], Gong et al. proposed a model called PRESS which uses statistical machine learning to perform resource auto-scaling by predicting future resource demands. In [21], Zhang et al. applied regression-based approximation to estimate the CPU demand, based on the number/type of requests. In [35], Islam et al. applied sliding window to linear regression and correction neural network for performing resource predictions in cloud environment. In [10], Xu et al. found optimal VM configurations in cloud computing environment by applying a unified reinforcement learning approach.

The following table summarizes the various auto-scaling techniques.

Table 1. Comparison of various auto-scaling techniques

Technique	Working	Pros	Cons
Rule based	Works on the principle of setting thresholds and corresponding actions.	Simple and easy to implement	Lacks accuracy and prediction
Time Series	Utilizes a series of historical data values in order to predict future values	Capable of predicting future workloads	Selection of history window is difficult
Queuing Theory	Works by modeling queues to describe the processes behind them and to predict their behavior	Allows the modeling of systems using probabilistic distributions like the	Most of the queuing model are still complex

		Poison and exponential distributions	
Control Theory	Controls the behavior of a dynamic system by comparing the output with a desired value	Use of feedback makes system quickly adapt to varying workload	Difficult to find static control setting so as to make the system stable (static output feedback stabilization problem)
Machine Learning	Deals with training a machine to learn from its past experience so as to improve performance	Automates analytical modeling and enable access to hidden insights	Overhead in learning from a large state space

III. RELATED WORK

This section compares the work done in the area of auto-scaling in the cloud computing environment. The comparison of various works is based on parameters viz. the underlying technique, type of scaling (horizontal or vertical), timing of scaling (Reactive, Proactive or Hybrid, nature of the workload, and the year of publication.

Table 2. Comparison of work done in auto-scaling in cloud domain

Ref	Underlying Technique	Type of scaling (H/V)	Timing of scaling (R/P/ Hybrid)	Metrics used	Nature of Workload	Year
12	Time Series	H	P	Execution time	Real world (Wikimedia Foundation)	2013
22	Queuing Theory	H	P	Request Rate	Real world (Wikipedia Traces)	2013
23	Time Series (Regression)	H	P	CPU (MIPS)	Synthetic	2015
24	Machine Learning	H	P	Response Time	Real world (NASA, Wikipedia, FIFA 98 world cup traces)	2015
25	Hybrid (Autonomic computing + Reinforcement	H	P	CPU utilization/ Response Time	Real world (ClarkNet and NASA traces)	2017

	Learning)					
26	Hybrid (Threshold based + Heuristic)	H	P	Response Time	Real world (EPA, SDSC and ClarkNet traces)	2014
27	Queueing Theory	V	P	Latency and Throughput	Real world (FIFA98 world cup traces)	2014
28	Machine Learning	H	P	CPU and Memory	Synthetic	2016
29	Time Series	H + V	P	Response Time	Real world (FIFA 98 world cup traces)	2016
30	Control Theory	H + V	P	Response Time	Nginx logs	2015

IV. CONCLUSION

The elastic nature of cloud computing enables the on demand provisioning and deprovisioning of resources in an automatic fashion. However, auto-scaling resources in cloud is a challenging task due to the unpredictable nature of web applications keeping in mind the SLA requirements of the end user. In this paper, we have presented the various aspects of auto-scaling in cloud and performed an exhaustive comparison of recent work done in the field of auto-sclaing in cloud environment.

V. REFERENCES

[1]. JoSEP, A. D., Katz, R., Konwinski, A., Gunho, L., PAttERSon, D., & RABKin, A. (2010). A view of cloud computing. Communications of the ACM, 53(4)

[2]. "Animoto case in rightscale blog," http://blog.rightscale.com/2008/04/23/animoto-facebook-scale-up/

[3]. Shoaib, Y., & Das, O. (2014). Performance-oriented Cloud Provisioning: Taxonomy and Survey.arXiv preprint arXiv:1411.5077

[4]. Gandhi, A., Dube, P., Karve, A., Kochut, A., & Zhang, L. (2014, June). Adaptive, Model-driven Autoscaling for Cloud Applications. In ICAC(Vol. 14, pp. 57-64)

[5]. Yazdanov, L., & Fetzer, C. (2012, November). Vertical scaling for prioritized vms provisioning. In Cloud and Green Computing (CGC), 2012 Second International Conference on(pp. 118-125). IEEE

[6]. http://docs.aws.amazon.com/autoscaling/latest/userguide/WhatIsAutoScaling.html

[7]. https://www.packtpub.com/books/content/elastic-load-balancing

[8]. Loff, J., & Garcia, J. (2014, December). Vadara: Predictive elasticity for cloud applications. In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on(pp. 541-546). IEEE

[9]. Amazon. 2016. Amazon Auto Scaling Service. (2016). http://aws.amazon.com/autoscaling/ Carlos Vazquez- time series

[10]. Xu, C. Z., Rao, J., & Bu, X. (2012). URL: A unified reinforcement learning approach for autonomic cloud management. Journal of Parallel and Distributed Computing, 72(2), 95-105

[11]. Dutreilh, X., Moreau, A., Malenfant, J., Rivierre, N., & Truck, I. (2010, July). From data center resource allocation to control theory and back. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on(pp. 410-417). IEEE

[12]. Calheiros, R. N., Masoumi, E., Ranjan, R., & Buyya, R. (2015). Workload prediction using ARIMA model and its impact on cloud applications' QoS. IEEE Transactions on Cloud Computing, 3(4), 449-458

[13]. Calheiros, R. N., Ranjan, R., & Buyya, R. (2011, September). Virtual machine provisioning based on analytical performance and QoS in cloud computing environments. In Parallel processing (ICPP), 2011 international conference on(pp. 295-304). IEEE

[14]. Ferretti, S., Ghini, V., Panzieri, F., Pellegrini, M., & Turrini, E. (2010, July). Qos–aware clouds. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on(pp. 321-328). IEEE

[15]. Gong, Z., Gu, X., & Wilkes, J. (2010, October). Press: Predictive elastic resource scaling for cloud systems. In Network and Service Management (CNSM), 2010 International Conference on(pp. 9-16). IEEE

[16]. Jiang, J., Lu, J., Zhang, G., & Long, G. (2013, May). Optimal cloud resource auto-scaling for web applications. In Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on(pp. 58-65). IEEE

[17]. Roy, N., Dubey, A., & Gokhale, A. (2011, July). Efficient autoscaling in the cloud using predictive models for workload forecasting. In Cloud Computing (CLOUD), 2011 IEEE International Conference on(pp. 500-507). IEEE

[18]. Fernandez, H., Pierre, G., & Kielmann, T. (2014, March). Autoscaling web applications in heterogeneous cloud infrastructures. In Cloud Engineering (IC2E), 2014 IEEE International Conference on(pp. 195-204). IEEE

[19]. Nguyen, H., Shen, Z., Gu, X., Subbiah, S., & Wilkes, J. (2013, June). AGILE: Elastic Distributed Resource Scaling for Infrastructure-as-a-Service. In ICAC(Vol. 13, pp. 69-82)

[20]. Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., & Wood, T. (2008). Agile dynamic provisioning of multi-tier internet applications. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 3(1), 1

[21]. Zhang, Q., Cherkasova, L., & Smirni, E. (2007, June). A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In Autonomic Computing, 2007. ICAC'07. Fourth International Conference on(pp. 27-27). IEEE

[22]. Bankole, A. A., & Ajila, S. A. (2013, May). Predicting cloud resource provisioning using machine learning techniques. In Electrical and Computer Engineering (CCECE), 2013 26th Annual IEEE Canadian Conference on(pp. 1-4). IEEE

[23]. Al-Ayyoub, M., Jararweh, Y., Daraghmeh, M., & Althebyan, Q. (2015). Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure. Cluster Computing, 18(2), 919-932

[24]. Liu, J., Zhang, Y., Zhou, Y., Zhang, D., & Liu, H. (2015). Aggressive resource provisioning for ensuring QoS in virtualized environments. IEEE Transactions on Cloud Computing, 3(2), 119-131

[25]. Ghobaei-Arani, M., Jabbehdari, S., & Pourmina, M. A. (2017). An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach. Future Generation Computer Systems

[26]. Tighe, M., & Bauer, M. (2014, May). Integrating cloud application autoscaling with dynamic vm allocation. In Network Operations and

Management Symposium (NOMS), 2014 IEEE(pp. 1-9). IEEE

[27]. Spinner, S., Kounev, S., Zhu, X., Lu, L., Uysal, M., Holler, A., & Griffith, R. (2014, September). Runtime vertical scaling of virtualized applications via online model estimation. In Self-Adaptive and Self-Organizing Systems (SASO), 2014 IEEE Eighth International Conference on(pp. 157-166). IEEE

[28]. Grozev, N., & Buyya, R. (2016). Dynamic Selection of Virtual Machines for Application Servers in Cloud Environments. arXiv preprint arXiv:1602.02339

[29]. Hirashima, Y., Yamasaki, K., & Nagura, M. (2016, July). Proactive-Reactive Auto-Scaling Mechanism for Unpredictable Load Change. In Advanced Applied Informatics (IIAI-AAI), 2016 5th IIAI International Congress on(pp. 861-866). IEEE

[30]. Dupont, S., Lejeune, J., Alvares, F., & Ledoux, T. (2015, September). Experimental analysis on autonomic strategies for cloud elasticity. In Cloud and Autonomic Computing (ICCAC), 2015 International Conference on(pp. 81-92). IEEE

[31]. Zhu, Q., & Agrawal, G. (2010, June). Resource provisioning with budget constraints for adaptive applications in cloud environments. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing(pp. 304-307). ACM

[32]. Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., & Merle, P. (2017). Elasticity in Cloud Computing: State of the Art and Research Challenges. IEEE Transactions on Services Computing

[33]. Bankole, A. A., & Ajila, S. A. (2013, May). Predicting cloud resource provisioning using machine learning techniques. In Electrical and Computer Engineering (CCECE), 2013 26th Annual IEEE Canadian Conference on(pp. 1-4). IEEE

[34]. Gong, Z., Gu, X., & Wilkes, J. (2010, October). Press: Predictive elastic resource scaling for cloud systems. In Network and Service Management (CNSM), 2010 International Conference on(pp. 9-16). IEEE

[35]. Islam, S., Keung, J., Lee, K., & Liu, A. (2012). Empirical prediction models for adaptive resource provisioning in the cloud. Future Generation Computer Systems, 28(1), 155-162