Dynamic Provisioning of Cloud Resources based on Workload Prediction

Sivasankari Bhagavathiperumal, Madhu Goyal

Centre of Artificial Intelligence, Faculty of Engineering and Information Technology
University of Technology Sydney, PO Box 123, Broadway, NSW 2007, Australia
Sivasankari.bhagavathiperumal@student.uts.edu.au, madhu.goyal-2@uts.edu.au

Abstract. Most of the businesses now-a-days have started using cloud platforms to host their software applications. A Cloud platform is a shared resource that provides various services like software as a service (SAAS), infrastructure as a service (IAAS) or anything as a service (XAAS) that is required to develop and deploy any business application. These cloud services are provided as virtual machines (VM) that can handle the end user's requirements. The cloud providers have to ensure efficient resource handling mechanisms for different time intervals to avoid wastage of resources. Auto-scaling mechanisms would take care of using these resources appropriately along with providing an excellent quality of service. The researchers have used various approaches to perform autoscaling. In this paper, a framework based on dynamic provisioning of cloud resources using workload prediction is discussed.

Keywords: Auto-scaling · Horizontal scaling · Vertical scaling · Virtual machine· Cloud server · Load balancer · Load measurer ·Load predictor · Load detector

1 Introduction

Cloud Computing is the popular technology that provides the distributed infrastructure or services that include environments to host vendor applications, network resources, build and deploy applications. The users like online stores, webmasters tend to prefer the cloud services like Amazon Elastic Compute Cloud (Amazon EC2), Microsoft Azure that offers resources in terms of Virtual Machines(VM) instead of setting up their infrastructure [1]. Scaling is one of the most prominent features which allocates resources dynamically based on the volume of the request to the server. The Cloud servers offer two types of scaling namely horizontal scaling and vertical scaling. Horizontal scaling is connecting more servers at the same time to do the same work increasing the speed or availability of the logical units.

Vertical scaling is the ability to raise the power of the server like increasing the RAM of the computer to improve the speed and performance of the computer.

Auto-scaling is provisioning of resources through virtual machines ensuring the quality of service in accordance with the Service Level Agreements (SLA). For example, consider a situation where X number of machines are serving N number of customers at the peak time of Stock exchange. Suddenly the unanticipated number of customers uses the same application increasing the number of nodes. This is when the performance of the application slows down violating the SLA. At this time an efficient auto-scaling system would be required to manage the load and balances the service [2].

Moreover, auto scaling ensures that the required sources of services are supplied seamlessly when the demand is high and reduce the supply when the demand decreases. The automated solution to this horizontal and vertical scaling would benefit both the cloud providers and the users of this cloud services concerning with utilization of resource wisely and cost effectively along with increasing the performance. However, identifying the required resources would be challenging as the demand fluctuates from time to time [2]. Therefore, it is important to build the frequently occurring events in the system so that the cloud systems predict the requirements of the users [3].

The cost of the service is reduced if less resources is leased resulting in the performance getting affected at the peak hours [4]. As shown in the figure(Fig.1) below [1], the auto scaling mechanism starts with the end user sending a request to the application through a device connected with the internet. The application forwards this request to the virtual machines which is connected to the load balancer. The applications has the auto scaling mechanism when the decision of the virtual machines required are decided and request from the user is served [1].

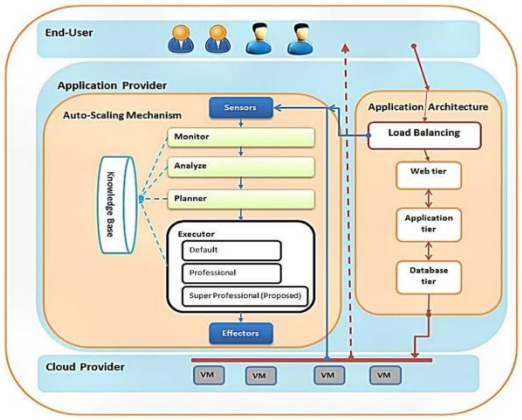

Fig.1 Architecture of cloud based web applications [1]

2 Related works

The cloud community has proposed various researches to perform the auto scaling. An efficient process of auto-scaling is proposed based by IBM called as MAPE loop[1]. The authors suggested a cost-efficient scale down commands which is more focused towards selecting redundant virtual machines. The authors presented an efficient auto scaling mechanism to adapt to adequate services with lesser operational expenses. The performance of this model was based on four phases of monitoring, analysing, planning and execution. According to the authors, the implementation of the virtual machines will take place after careful monitoring and analysing only. This model suggested an executor named as Suprex that executed the plan of auto scaling that can handle the resources based on the requests from the end users. Another theory proposes the implementation of autonomic computing by the model suggested by IBM called as MAPE-K loop to manage the elements of any software and hardware resources [3].

An approach implemented by Amazon that aimed at selling their unused capacities based on auction like mechanism which were called as spot instances was suggested [5]. These spot instances were suitable mainly for fault-tolerance flexible web applications. The cost of this instance was low compared with on demand price. Moreover, this approach can be used for applications that can be interrupted. Applications like background processing, batch jobs can utilise this approach compared to the critical applications. On the other hand, the spot instances also take more time to boot when compared to the on-demand instances. So, the authors suggested a heterogeneous approach where a mix of both spot and on-demand instances can be used to meet the end users demand.

Another model suggested a repacking approach where they investigated how to repack the virtual machines to provide the required services most efficiently [6]. In addition, another theory proposed a concept of the efficient auto-scaling scheme (EAS) which minimizes the processing time by implementing a huge number of cores in the internet of things [7]. Other studies researched on the work load prediction approach to enhance the service to the users reducing the cost by forecasting the expected load [2]. While the work load prediction model was successful, there was few other approaches that focused more on work load and came up with a linear model [4].

A linear regression model which will predict the work load of the services used in the cloud was proposed [4]. They proposed an algorithm to efficiently scale the cloud services along with developing a cloud scaling architecture. The authors discussed the various cloud service providers and cost of service cloud. They have proposed different algorithms and compared the results with related works and various aspects like REMICS, FP7 project. The authors investigated the problem of auto-scaling based on predicted workloads in service clouds using linear regression model. They proposed approach to scale the service in both real time and pre-scaling.

Although various approaches were proposed the main idea of all the theories was reducing the cost with delivering the resources in an efficient way to benefit both the cloud providers and cloud users. Unlike the traditional method of having a physical server, the cloud servers are reliable, secured and maintainable. Small business to large business relies on cloud computing for their data related activities. When the data is transmitted through the cloud servers the need for the release of the data also increases. Moreover, data is supplied on the virtual machines supplied by the cloud servers. The major benefit of auto scaling is paying as per the use of the resource and using as per the demand of the request. Compared with the traditional approaches of installing an application on their physical servers and then connecting the application through a secured channel the cloud computing provides easier access to the required resources [8]. The elastic nature of the cloud computing provides the resources in proportion to the volume request [9].

3. Proposed framework for auto-scaling based on workload prediction

In this paper, an n-tier architecture is proposed for the auto-scaling framework based on the workload prediction as shown in the below figure (Fig.2)

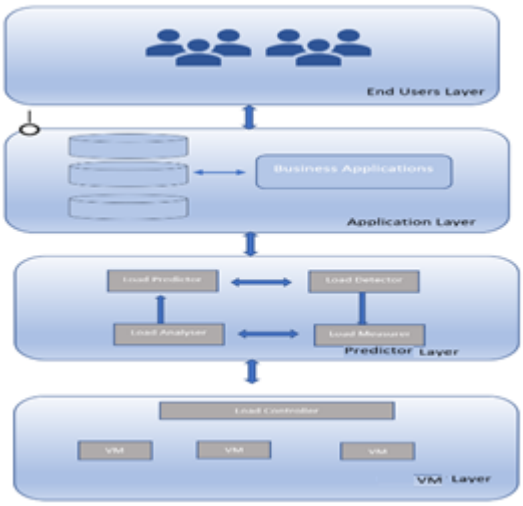

Fig 2: Framework for auto-scaling based on workload prediction

The framework consists of four layers i.e. End-users layer, Application layer, VM-layer, Predictor layer. The predictor layer consists of four different process namely

load detector, load measurer, load analyser and load predictor.

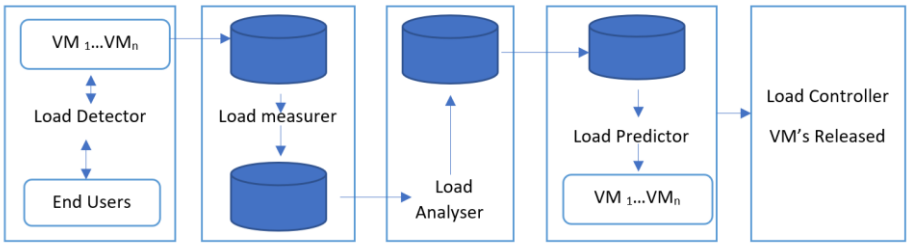

Fig3: Flow chart for the proposed framework

It takes the responsibility of delivering the required VM's making the application run based on the user demands. All these processes contain a storage base that will store the data of load. The processes of predictor layer are described in Fig 3.

3.1 End User Layer

The end user layer is where the business customers use internet to access the business application to perform their required task. The request for accessing the applications varies at different time intervals and thus based on the number of end users request the VM's should be released. At certain times there might be enormous number of users trying to reach the application layer and there might be less or no requests also.

3.2 Application Layer

This is the layer where the business applications are hosted. This layer may consist of numerous services, applications and databases. The end users would send the request to this layer. To maintain a quality of service, this layer should be active always irrespective of the demands.

3.3 Predictor Layer

The prediction layer is the main component of this paper where the scaling process happens. The predictor layer consists of life cycle that detects the signal received from the end user measuring the request, analysing the request and finally predicting the required VM's. This data is sent to the load controller present in the VM layer.

3.3.1 Load detector

This process receives the request from the end user to the application along with detecting the number of virtual machines released to the handle the demands. The load will fluctuate time to time and this gets recorded by the load detector actively. The results will be shared with the load measurer. When we consider S_i as the service provided by cloud, then the VM's utilised is calculated for the t-Δ interval along with calculating the requests[3]. Auto-scaling can be successful only when the metrics of the load is detected with suitable granularity.

In general, there are two possible approaches which are homogeneous approach where the resource pool is of same size and heterogeneous approach in which distinct size of resources is allowed. Most of the cloud providers offer different virtual machines families for diverse types of applications. Currently, there scaling mechanism are provided based on rules or threshold of CPU utilization [10]. The load detector should be capable of detecting such variances to ensure the request always is recorded.

3.3.2 Load measurer

A linear regression model is used to measure the load. Linear regression model calculates the current values of series against the prior values in the series. The general form of linear regression is given as follows:[3]

$$Y_{t+1} = \beta_1 + \beta_2 * X_1$$

where t is indexes, Y_1 is the incoming workload and X is the actual value of the instance at that moment. The load can also be measured using the moving average method or exponential smoothing methods. In moving average, the mean of the n last values is calculated on the other hand the exponential smoothing method decrease values in each value of time series[11].

3.3.3 Load analyser

The load analyser task is to analyse the load that is measured. This will help the controller control the release or ceasing of the VM. The analyser will take the responsibility of analyzing both resource release and time. The end user request is analysed and validated by the load analyser. After care analysis, the data is supplied to the load controller. The load analyser user the auto regressive or linear regression

method to analyse the load so that the data of analysed load helps the load controller predict the future loads.

Consider the example of World Cup Soccer 1998 workload[2] shown below

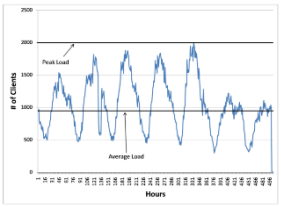

Fig4: World Cup Soccer 1998 workload[2]

In the above figure, the workload is calculated to granularity of hours versus number of clients. The graph shows the fluctuation of the resources while at peak time the load is huge and at some normal hours the load is low, the load analyser should impose these variations and save the data to be provided for the load controller.

3.3.4 Load predictor

The load predictor uses the ARIMA (Auto regressive integrated moving average) to release the VM's based on the prediction calculated by ARIMA method. ARIMA pattern follow a very popular time series model which is a sequence of measurements over time, usually obtained at equally spaced intervals which can be – Daily – Monthly – Quarterly – Yearly. This can also be calculated based on seasonal trends. Seasonal trends are the patterns in time series algorithm where the sequence of measurement is done seasonally. For example, a year could have a seasonal trend of four seasons which typically repeats for every season.

Given a time series of data X_t where t is an integer index of the time interval and X_t are real numbers which represents the number of end-users an *ARMA(p^{l},q)* is given by

$$X_t - \alpha_1 X_{t-1} - \cdots - \alpha_{p'} X_{t-p'} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

3.4 VM Layer

The VM layer is where the virtual machines are ready to auto-scale based on the requirement. This layer consists of the n number of virtual machines and the load controller. The load controller receives data from the predictor layer and ensures the required VM's are released to the meet the end users demand.

4. Conclusion and Future Works

In this paper, a framework for dynamically provisioning of cloud resources is proposed. This would help in handling the increasing volume of data throughput of enterprises and businesses which are moving towards the cloud computing services. It will also be helpful with analysis of configuration issues on provisioning these resources dynamically especially when the applications are running during the peak hours. The future studies will specifically focus on prediction of provisioning vertical or parallel processing of resources using data mining techniques specially to meet the requirements of big data.

References

1. Aslanpour, M.S., M. Ghobaei-Arani, and A. Nadjaran Toosi, *Auto-scaling web applications in clouds: A cost-aware approach.* Journal of Network and Computer Applications, 2017. **95**: p. 26-41.
2. Roy, N., A. Dubey, and A. Gokhale. *Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting*. in *2011 IEEE 4th International Conference on Cloud Computing*. 2011.
3. Ghobaei-Arani, M., S. Jabbehdari, and M.A. Pourmina, *An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach.* Future Generation Computer Systems, 2016.
4. Yang, J., et al., *A cost-aware auto-scaling approach using the workload prediction in service clouds.* Information Systems Frontiers, 2014. **16**(1): p. 7-18.
5. Qu, C., R.N. Calheiros, and R. Buyya, *A reliable and cost-efficient auto-scaling system for web applications using heterogeneous spot instances.* Journal of Network and Computer Applications, 2016. **65**: p. 167-180.
6. Sedaghat, M., F. Hernandez-Rodriguez, and E. Elmroth. *A virtual machine re-packing approach to the horizontal vs. vertical elasticity trade-off for cloud autoscaling*. in *ACM International Conference Proceeding Series*. 2013.
7. Kim, H.W. and Y.S. Jeong, *Efficient auto-scaling scheme for rapid storage service using many-core of desktop storage virtualization based on IoT.* Neurocomputing, 2016. **209**: p. 67-74.
8. Dutreilh, X., et al. *From data center resource allocation to control theory and back*. in *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*. 2010.
9. Mogouie, K., A. Mostafa Ghobaei, and M. Shamsi, *A Novel Approach for Optimization Auto-Scaling in Cloud Computing Environment.* International Journal of Modern Education and Computer Science, 2015. **7**(8): p. 9-16.
10. Sahni, J. and D.P. Vidyarthi, *Heterogeneity-aware adaptive auto-scaling heuristic for improved QoS and resource usage in cloud environments.* Computing. Archives for Informatics and Numerical Computation, 2017. **99**(4): p. 351-381.
11. Lorido-Botrán, T., J. Miguel-Alonso, and J. A Lozano, *Comparison of Auto-scaling Techniques for Cloud Environments*. 2013.