Rajeshwari et al. / IJAIR *Vol. 2 Issue 5* *ISSN: 2278-7844*

Dynamic Load and Bill Prediction using ARIMA Model in Cloud Computing

P.Rajeshwari[1]
PG Scholar
Department of CSE
Angel College of Engineering and Technology
Tirupur, India
rajiponraj@gmail.com

P.Sundaramoorthy[2]
Assistant Professor
Department of CSE
Angel College of Engineering and Technology
Tirupur, India
sundarme08@gmail.com

Dr.T.Rajendran[3]
Dean
Department of CSE
Angel College of Engineering and Technology
Tirupur, India
rajendran_tm@yahoo.com

***Abstract*- The Cloud Computing is a technology that uses the internet and central remote servers to maintain data and applications. It is charged for variable rate of resource consumed. The problem in the prediction process is that, it is difficult to predict if suddenenly load increases. In this paper, proactive load prediction method using Auto linear regression for predicting load and price in the future is proposed. Monitoring the cloud resource utilization like Capacity, bandwidth, memory, RAM, network, disk I/O, etc and store in the database. Based on historical data and current usage, the load on the cloud infrastructure is predicted using auto-regressive integrated moving average (ARIMA) Model. The function of load prediction and pricing scheme is used to generate bill and it is displayed to the customer. The pricing for every time period is published and it is the consumer's decision to continue using the service or to suspend usage.**

***Keywords*: cloud computing, load prediction, ARIMA Model, Bill Calculation, Decision making.**

I. INTRODUCTION

Cloud computing is the latest evolution of computing, where IT resources are offered as services. The hardware and software systems that manage these services are referred to as Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), while the actual applications managed and delivered by IaaS and PaaS are referred to as Software as a Service (SaaS).

The purpose of having a cloud-optimized billing component is to be able to provide an interface for generating usage bills. Cloud billing is the process of generating bills from the resource usage data using a set of predefined billing policies. The bill can be for real money or it can refer to a more abstract notion of exchange, depending on your individual cloud computing general policies.

We describe a Proactive metering model which provides for dynamic pricing of the cloud service based on the load condition of the cloud infrastructure. Proactive Metering to the need of a dynamic pay-per-use model in the cloud environment based on real-time pricing of the cloud service. The aim of this dynamic pay-per-use pricing model is regulation and improvement of overall utilization of the cloud infrastructure.

The premise of this model is to record the resource usage per cloud instance and to generate a bill based on two factors, *viz.*, the consumer's utilization and the load on the cloud during the period of utilization. The Proactive metering model facilitates improving the overall utilization of the cloud infrastructure by implementing a varying tariff model based on the prevailing load.

Commercial cloud vendors provide cloud instances which function based on a bid amount which is the maximum price a consumer is willing to pay for the instance. Resource provisioning is based on this bid amount. Once the service price amount increases beyond the bid amount, the cloud instance is liable to be terminated without warning. This restrictive model is not advantageous to the consumer. An example for this is the Amazon spot instances [4].

Iyer *et al.* [5] describe a "virtual platform architecture" (VPA) for resource metering in data-centers which provides for a *pay-as-you-go* billing mechanism. Resource usage of each datacenter application is metered and customers are charged accordingly. The metering and chargeback model is based on VPA which abstracts physical resources in the infrastructure and treats them as sets of virtual entities. This implementation does not provide for dynamic metering. The billing mechanism is based on predefined static pricing information.

The rest of the paper is organized as follows. Section II details our system Design which includes the architecture, pricing, billing model and proactive load prediction mechanism used. Section III details the implementation of various elements described in the design. Conclusions for further development are presented in Sections IV and V respectively.

II. SYSTEM DESIGN

A. Architecture

A cloud setup typically comprises of the following components [9] as seen in Fig 1

- Cloud controller and cluster controller.
- Node controller.
- Storage controller.

The *cloud controller* is the front end of the cloud infrastructure. The main functions of the cloud controller are: monitoring the resource availability; running instances; and resource arbitration. The *cluster controller* manages one or more nodes in the infrastructure and is responsible for deploying cloud instances on the nodes. The *node controller* runs on each node in the infrastructure. A virtualization technology [9] enabled server capable of running a hypervisor is called a node in the cloud. Virtual machines or cloud instances are deployed on these nodes. The *Proactive metering* module is either co-located with the cloud controller on the same server, or is deployed on a machine that is directly connected to the cloud controller. This deployment helps reduce the effect of network delays in recording the utilization data.

The *Application Nodes* are application replicas, such as Web Server, or Web Application. Each Node is assigned to run in a domain on the computing platforms of the Cloud Providers. We can envision having one Node per Cloud, and the number of Nodes depends on the level of intrusion tolerance required by the user's application. The clean image of the Application Node must be safely stored locally in the Cloud Storage [9].

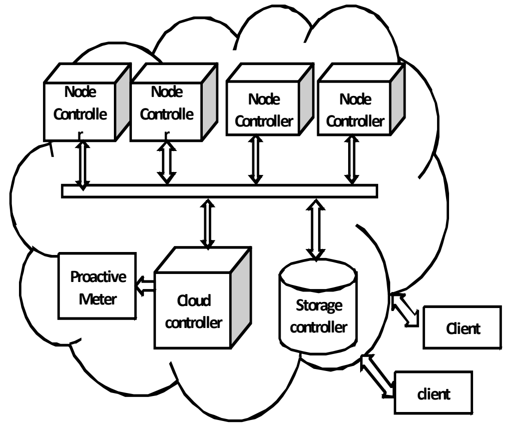

Figure 1: Architecture of cloud setup

Each Application Node is managed by a *Local Controller*, which is a new component for cloud controller. A Local Controller is responsible for maintaining the clean image for the Node, and directing its associated Node through the four states of Life Spare, Active, Grace Period, and Inactive. The link between an Application Node and a Local Controller is unidirectional from the latter to the former so that any compromise to the Application cannot affect the Local Controller. The existence of Local Controllers aims at limiting the amount of messages and data to be exchanged between the Nodes and the Central Controller over the WAN Operation.

The *Central Controller* coordinates the cleansing procedure by initiating an operational message to the Local Controllers running in a virtualized platform at each Cloud Service Provider. This control includes the assurance that the images of the. The Controller's domain can be on the customer's premise or a third-party Cloud.

In this section we present the architecture of the metering module, introduce the billing model used and the resulting pricing mechanism. And i introduce the proactive load predictor model for predicting the load when it changes dynamically based on customer increases. Then the bill for predicted load will be displayed.

B. Monitoring

Monitoring functions in a cloud are responsible for the health and availability of the components of the cloud. The monitoring tools are used to collect data that are fed to various other components in the cloud.

Monitoring of the cloud components is vital to maintain the performance of the cloud. The status information of the components in cloud will be collected and processed in monitoring solutions of the cloud.

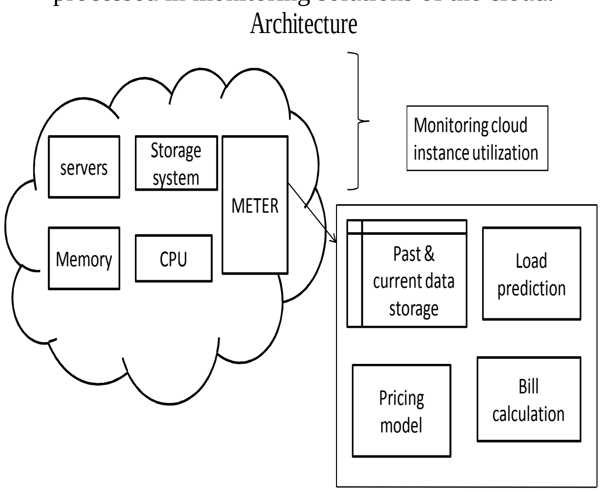

Figure 2 Proactive load Prediction model

Resource utilization data from the hosting server farm and the deployed cloud instances are monitored. Utilized data are recorded in database. Monitoring involves two activities:

- Infrastructure resource utilization monitoring
- Cloud instance resource utilization monitoring

Database maintains the record about customer id, transaction id, request, response, utilized resources etc. In response, the system quickly created additional instances of resources to balance workload demands dynamically.

C. Billing

Dynamic billing on cloud is a function of the instantaneous load on the cloud and the pricing information obtained as per the service provider. Billing calculations involve determining the overall load on the cloud over a recent interval of history and obtaining a weighted sum of the load on the entities and the corresponding pricing information.

The overall load Lt on the cloud infrastructure at time t is the sum of the load on every node as given by:

D. Pricing model

Pricing for a cloud service can be applied based on multiple considerations. Current service providers like Amazon [8] and Rackspace [9] price their cloud instances mostly based on configuration and duration of use. Another prevalent practice is to charge the consumers a fixed price for a lease period like the Amazon *reserved instance.*

Under elastic pricing for cloud pricing, customers are charged based on their usage and consumption of a service. And with awareness of the costs comes more efficient and selective usage, thus resulting in less waste and lower costs.

E. Load prediction

Based on the historical data collected through the monitoring, the load for the next time interval of operation is predicted. The auto regressive model assumes that the current value of a process xt, can be described by a finite linear aggregate of the previous values of the process along with the current value. By the monitoring tool current load can be obtained. By that based on the pricing model we can predict the load for the future using ARIMA Model.

III. IMPLEMENTATION

Proactive metering proposes pricing based on the dynamic operational cost of running the service. In this pricing model, the base cost of running the service is specified by the service provider, C_{base}. Pricing rules which define the pricing overhead for running the service under various load conditions are specified by the service provider. This pricing overhead is given by α(l, t), where l is the load at time instance of operation t. The current price, Pt for the interval at a given load condition is given as,

$$Pt = C_{base} \times \alpha (l, t) \quad (1)$$

Pt = operational price at time t

C_{base} = base operational cost

α(l, t) = pricing overhead for running the service under load l at time t.

A.Billing

Dynamic billing on cloud is a function of the instantaneous load on the cloud and the pricing information obtained as per the configuration specified by the service provider. Billing calculations involve determining the overall load on the cloud over a recent interval of history and current usage of the cloud. It is obtaining by a weighted sum of the load on the entities and the corresponding pricing information as described above. The overall load Lt on the cloud infrastructure at time t is the sum of the load on every node as given by:

Rajeshwari et al. / IJAIR Vol. 2 Issue 5 ISSN: 2278-7844

$$L_t = \text{SUM (i=I to n) } l_i, \quad (2)$$

where,

L_t - total load on the cloud at time t

l_i - load index of the individual cloud component.

The load obtained in (2) is mapped to a corresponding pricing value at a given interval of time, t. The bill amount is computed as a summation of the product of instantaneous pricing obtained in (1) and the utilization of the consumer, ut. The total bill amount is obtained as,

$$\text{Bill} = \text{SUM (t=1 to n) } P_t \times ut \times Ct, \quad (3)$$

Where,

Pt - operational price at time t

ut - resource utilization of the consumer at time t.

C_t - Current resource using by the cloud customers.

B. Load Prediction

It is known that price plays an important factor in deciding the cloud service utilization. We obtain the indicative price by predicting the load on the infrastructure and obtaining the corresponding pricing information as mentioned earlier and current usage of the cloud. Based on the historical data collected through the monitoring framework, the load for the next time interval of operation is predicted using the auto regressive integrated moving average (ARIMA) statistical model.

C. ARIMA

The auto regressive model assumes that the current value of a process xt, can be described by a finite linear aggregate of the previous values of the process along with the current value of a white noise signal at. Generally, autoregressive (AR) models (denoted as AR (p)) are described as,

$$xt = \varphi_1 x_{t-1} + \varphi_2 x_{t-1} + L + \varphi_c x_{t-c} + \varphi_p x_{t-p} + a_t \quad (4)$$

where

a_t - the unpredictable white noise sequence with zero mean and variance of σ^2_a , and each φ_i and L is a coefficient of AR

Model with p as the number of coefficients. If the p coefficients of AR models are estimated from past data, the one-step-ahead prediction of this AR model can be estimated using these p coefficients along with the p historical data, x_{t-1}, x_{t-2},L, $x_{t-p,}$ $x_{t-c.}$

Mathematically this is described as:

$$xt = \varphi_1 x_{t-1} + \varphi_2 x_{t-1} + L + \varphi pxc + t\text{-}p_t \quad (5)$$

where,

x_t is a one-step-ahead prediction value for time t.

The parameter p of AR models is the number of historical data points it uses. Runtime cost increases if a higher order of differencing is used in the AR model to predict the load. If the differencing order chosen is too low, it results in an erroneous prediction. However, we cannot use this directly in our work as the number of data points available to predict the next interval utilization is restricted to about two to four hours of data.

IV. ALGORITHM FOR LOAD PREDICTION AND BILL CALCULATION

A. Price prediction

```
Data: historicalLoadValues
   begin PublishServicePrice
      while True do  predictedPrice
                     PredictNextIntervalServicePrice
                     (historicalLoadValues)
                     Publish on the Internet:
                     predictedPrice
WAIT (interval)
      end
  end

Data: historicalLoadValues
Constant: LoadToPriceMapping
begin PredictNextIntervalServicePrice
         predictedLoad   PredictLoadValue
         (historicalLoadValues)
         predictedPrice   LoadToPriceMapping
         [predictedLoad ]
return : predictedPrice
end

Data: historicalLoadValues
Begin PredictLoadValue

         ARIMA(historicalLoadValues)

/*ARIMA implementation is available
```

```
in statistical packages like R. */
return : predictedLoad
end
```

B. Bill Calculation

```
Data: cloudInstance, BillAmount
begin CalculateBill
    foreach instance in cloudInstance do
            BillAmount   0
        foreach timeInterval in interval do
            BillAmount   BillAmount +
            RetrieveStoredIntervalBill
            (instance,timeInterval+ current load)
            Publish the BillAmount on
            consumer portal
        end
    end
end

Data: utilizationData, cloudInstance, interval
begin CalculateBillForInterval
  foreach timeInterval in interval do
      foreach instance in cloudInstance do
            utilizationData
            RetrieveStoredUtilizationData
            (instance)
            intervalBillAmount   utilizationData ×
            priceAtInterval
            Store (intervalBillAmount)
        end
    end
```

Load and cost Prediction

1. Number of Users request for the service (application) in the cloud.
2. For each user's utilization, find the load for Virtual Machine (VM) and load for application.
3. For each application calculate the load. And for individual user calculate the load in the cloud.
4. Record the periodic load information which is used to predict load for next time.
5. Fine the threshold value (peak) for load (each application). If it exceed additional VM is needed to run the application. Calculate the resource utilization (RAM, processor, memory etc) of additional VM by using prediction algorithm.
6. After the threshold value is reduced to average level, additional VM is get terminated.
7. Finally estimate the cost for additional VM utilization and current VM utilization. Based on pricing scheme bill amount will be predicted and displayed to the customer.
8. Customer can make the decision to continue with the new predicted price.

V. EXPERIMENTAL RESULT

After giving the username and password. Client portal will open in that web page we can specify our needed resource. Load prediction option will provide information about future use of load and bill amount for the predicted load. Here we can find the load and bill amount for next 4 hrs, 8 hrs, 12 hrs, 16 hrs.(figure 3).

Figure 3: Load Prediction

For finding the future load by using prediction Method. The user can make the decision to continue with the new predicted value. Load can be dynamically will get change for every time. In the peak time after reaching the threshold level availability of resource increases. So more resource is needed. That time cost wills increase. The amount will be displayed to customer (figure 4).

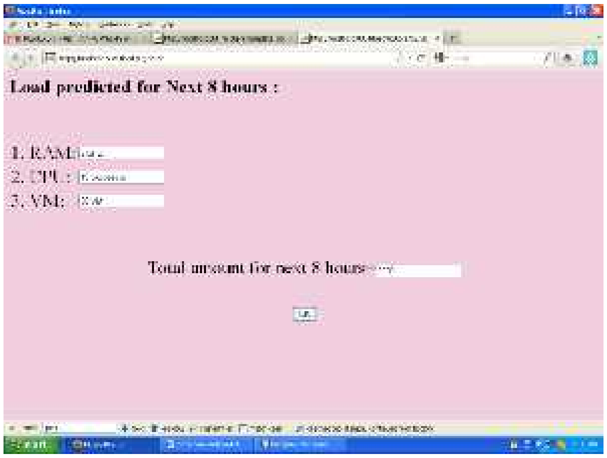

Figure 4: Bill Calculation for next 8 Hours

VI. CONCLUSION

An effective mapping between price and load was obtained using. By this mapping, dynamic pricing of cloud services based on the proactive load condition of the cloud infrastructure was achieved. Load needed for the next time future can be predicted using ARIMA model. And the bill based on the load predicted is calculated and displayed to the customer.

REFERENCES

[1] Akshay Narayan, Shrisha Rao, Gaurav Ranjan and Kumar Dheenadayalan," *Smart Metering of Cloud Services*" in IEEE international Conference 2012.

[2] Jorge Ejarque, Andras Micsikz, Ra¨ul Sirvent, Peter Pallingerz, Laszlo Kovacsz and Rosa M. Badia," *Semantic Resource Allocation with Historical Data Based Predictions*" The First International Conference on Cloud Computing, GRIDs, and Virtualization 2010.

[3] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, "*Time Series Analysis:Forecasting and Control"*. Prentice Hall PTR, 2008.

[4] Eyal Zohar, Israel Cidon, and Osnat Mokryn "PACK: Prediction-Based Cloud Bandwidth and Cost Reduction System" IEEE/ACM TRANSACTIONS ON NETWORKING

[5] Padmavathi, S. Rajeshwari, P. Pradheeba, P. Mythili, R. "*Achieving cost efficiency using CaaS model in the cloud*" (2012) IEEE Conference Publication.

[6] P.Rajeshwari, P.Sundaramoorthi, S.Padmavathi "*Cloud-based Self Cleansing Intrusion Tolerance with Hardware Enforced Security System*"5th International conference on Science Engineering and Technology November 2012.

[7] G. Neiger, A. Santoni, F. Leung, D. Rodgers, and R. Uhlig, "*Intel virtualization technology: Hardware support for efficient processor virtualization*,"
Intel Technology Journal, vol. 10, pp. 167–178, August 2006.

[8] Huazheng Qin, Xing Wu,, Ji Hou,Hanyu Wang, Wu Zhang "*Self-Adaptive Cloud Pricing Strategies with Markov Prediction and Data Mining Method***"** 2012 IEEE

[9] H. Akaike, "Fitting autoregressive models for prediction," *Annals of the Institute of Statistical Mathematics"*, vol. 21, pp. 243–247, 1969.

[10] Prasad Saripalli, GVR Kiran, Ravi Shankar R, Harish Narware and Nitin Bindal "*Load Prediction and Hot Spot Detection Models for Autonomic Cloud Computing*" 2011 IEEE.

[12] "*Amazon Pricing*," [Online], Available at http://bit.ly/f9K88S.T.Rajendran, Dr. P. Balasubramanie "*An Efficient Architecture for Agent-Based Dynamic Web Service Discovery with Qos*" 2005-2010 JATIT.

[13] T. Rajendran, P. Balasubramanie "*An Efficient Multi-Agent-Based Architecture for Web Service Registration and Discovery with QoS*", European Journal of Scientific Research ISSN 1450-216X Vol.60 No.3 (2011), pp.439-450.

AUTHORS BIOGRAPHY

P.Rajeshwari received her Bachelor of Engineering in Computer Science and Engineering from Anna University, Trichy. She is currently pursuing Master of Engineering in Computer Science and Engineering at Angel College of Engineering and Technology, affiliated to Anna University, Chennai. Her areas of interests are Cloud Computing and Networking She is an active member of IAENG. She has presented 4 international and 2 national conferences. She has published 2 papers in international journals.

P.Sundaramoorthy received his Bachelor of Engineering degree in Computer Science and Engineering stream and Master of Engineering degree in Computer Science and Engineering. He is currently doing his research in cloud computing. He has published 2 paper in international conference and 3 papers in national conferences. His area of interest includes cloud computing and network security.

Dr. T.Rajendran completed his PhD degree in 2012 at Anna University, Chennai in the Department of Information and Communication Engineering. Now he is working as a Dean for Department of CSE & IT at Angel College of Engineering and Technology, Tirupur, Tamilnadu, India. His research interest includes Distributed Systems, Web Services, Network Security, SOA and Web Technology. He is a life member of ISTE & CSI. He has published more than 51 articles in International/ National Journals/Conferences. He has visited Dhurakij Pundit University in Thailand for presenting his research paper in International conference. He was honored with Best Professor Award 2012 by ASDF Global Awards 2012, Pondicherry.