ISSN No. 0976-5697

Volume 8, No. 3, March – April 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Survey Paper on Cloud Demand Prediction and QoS Prediction

Rohan Garg
Computer Engineering Department
Institute of Technology, Nirma University
Ahmedabad, India

Prof. Vivek Prasad
Computer Engineering Department
Institute of Technology, Nirma University
Ahmedabad, India

Abstract: Cloud Computing is known for majorly two features which makes it attractive for consumers and providers: 1. Automatic Resource Provisioning 2. Strictly following the decided QoS . Resource Provisioning needs to be dynamic and quick. Even the state of the art technology might take time in minutes to provision a VM and makes Cloud unattractive than an on-premise system. So we present a study of ASAP - A Self Adaptive Prediction System for instant demand provisioning and Cyclic Window Learning Algorithm to predict the requests. QoS is closely related to resource allocation along with but includes strictly following the SLOs. QoS prediction can help the users to select the best service as per the need for which we study ALPINE which is a system based on Bayesian Networks and another model has been discussed here.

Keywords: Demand Prediction, QoS Prediction, ASAP,Cyclic Window Algorithm,ALPINE,Bayesian Networks

I. INTRODUCTION

Cloud Computing[5] provides a novel way to transfer the system that are on premise to on line. The features that makes the users get pulled towards it are illusion of infinite pool of resources , on demand resource provisioning, elasticity and trust on the service provider. However with increasing number of cloud providers and even more number of customers, the mismanagement of the factors that make Cloud a lucrative technology might cause people to abandon it.

Accurate prediction of demanded resources not only improves the response time of the Cloud but also helps in saving energy[6] which is otherwise wasted each time a resource is demanded and provisioned thereafter. Both the advantages help the Service Provider to effectively manage his resources for optimal utilization and result into more profit and a greener earth. ASAP, a Self Adaptive Prediction System provides for instant service of demand requests based on studying the past trends through temporal data mining. The Cyclic Window Learning Algorithm uses probability distribution for each time interval through Max Likelihood Estimation and Local Linear Regression.

Quality Of Service[7] is an important aspect provided by Cloud which deals with performance, reliability, scalability etc. There are a number of hardware and software factors which affect the QoS and yet majority of the existing prediction techniques fail to take them into account for QoS Prediction and Diagnosis. CloudHarmony.com lists 95 CSPs and with advancing years this is bound to grow, putting the consumer in a lot of confusion as to which Cloud to choose. QoS Prediction model which accounts for all factors such as CPU usage, the amount of memory, location of the data center, network and storage type will definitely fulfill the requirement of the user and prove to be useful. ALPINE, provides a front end for stakeholders to predict the cloud performance by tweaking the parameters uses BNs in the back end. The other BN model collects data from 3 layers of the Cloud and then trains and updates the network to perform prediction.

II. DEMAND PREDICTION

A. *ASAP*

The solution proposed by this framework is

a. From the request stream, extract high level information and generate a time series, for each image type.
b. For the time series set , a mechanism is required to select the models and train them to generate the prediction models.
c. This prediction model should be able to predict the demands at a later time based on historical request records

1.ASAP Framework

The ASAP system can be divided into three modules:

a)Raw Data Filtering and Aggregating Module

This modules brings out the required information from the raw data. In order to generate time series, the high level characteristics of the data need to be extracted.

b)Model Generating Module

This module would build up the models for different predictors. The models would be trained based on the latest inputs provided by the Raw Data Filtering module. Once the model is generated, the information of its parameters is stored in a file and the Demand Prediction module is sent a notification.

c)Demand Prediction Module

The models are reconstructed from the file generated in the previous module. Thereafter it makes use of the

regression ensemble and correlation ensemble to predict future demands.

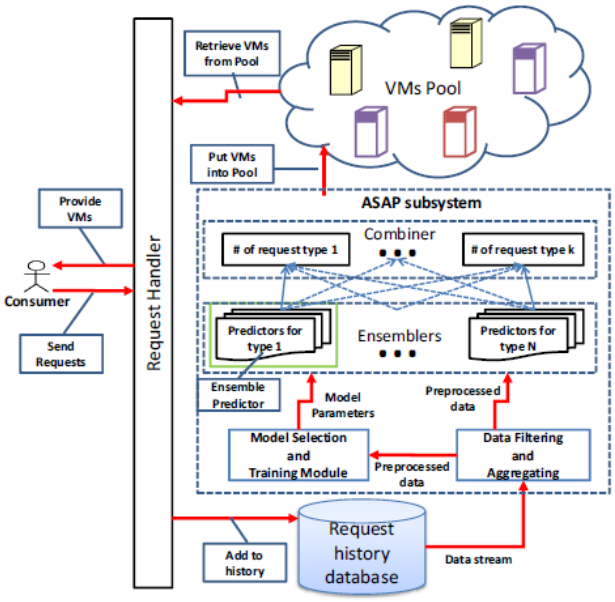

Figure 1. ASAP Framework[1].

2.Demand Prediction:

This is a two level ensemble algorithm. The first level considers the same VM types and aggregates the results of the prediction models. The second level considers different VM types and makes use of correlation to make the prediction accurate and robust.
The algorithms used for prediction are as follows:

Table I. Time Series Prediction Algorithm[1]

Sr. No.	*Prediction Method*	*Description*
1	Moving Average	Naïve
2	Auto Regression	Linear Regression
3	Artificial Neural Network	Non Linear Regression
4	Gene Expression Programming	Heuristic Algorithm
5	Support Vector Machine	Linear Learner with Non linear kernel

3. Prediction Ensemble:

The Prediction algorithm presented for use in ASAP is inspired from a classification based online ensemble algorithm. Say for a predictor p ε P ,the predicted value is v_p and the weight for that instance is w^t_p then the predicted demand represented by v^t would be

$$v^t = \Sigma_p w_p{}^t v_p \text{ subject to } \Sigma_p w_p{}^t = 1$$

.At t = 0, we would have $w_p = 1/|P|$ for every predictor and hence each predictor would give the same contributions to the final result. To recalculate the weights we need to consider the relative error $e_p{}^{t-1}$caused by a predictor p at time t-1 as:

$$e^t = \Sigma_p c_p{}^{t-1} w^{t-1}/c^{t-1}$$

$c_p{}^{t-1}$ is the cost predicted by the methods shown in the table. Also these are the relative errors, so they cannot be used to update the weights without normalization. The final predicted value will be a linear combination of the results given by each of the individual predictors and the weight given by $w^t = e^t/\Sigma_p e_p{}^t$ will be used in accordance with $\Sigma_p w_p{}^t = 1$.

4.Correlation Ensemble:

In order to improve the prediction a time series correlation has also been considered . These will be used post process the prediction. Say the covariance between the resource i and its j^{th} resource is given by $cov_{ij}{}^t$, then the post processed predicted demand would be as follows

$$u_i{}^t = \Sigma_{j=1}{}^k cov_{ij}{}^{t-1} s_{ij} v_k{}^t / \Sigma_{j=1}{}^k s_{ij} cov_{ij}$$

where $s_{ij} = v_i/v_j$ shows the difference of scale between two time series and k represents the number of strongly related time series. Negatively related time series have not been taken into consideration.

5.Reservation Controller:

It reserves the unused VMs and asks the VM Manager to prepare new VMs once all the existing ones have been used up. It helps in reduction of wastage by over provisioning.

B. Cyclic Window Learning Algorithm

The algorithm proposed would make prediction of the probability distribution of the incoming request. It is not supposed to predict the actual value of requests. Prediction of requests of high probability would help the CSP for handling the demand bursts. A model of probability distribution is fitted on the histogram generated from the historical data of requests. The parameters which vary with time are evaluated using Maximum Likelihood Estimation. Once enough data is collected, prediction of these parameters is made using Local Linear Regression following a cyclic window approach. Old data of the parameters is also replaced by new one, in order to account for the changing trends.
Also an assumption is made that the trends i.e. the patterns will be repeated after a certain period of time.

1.The Algorithm

Consider 3 time scales:

- Pattern Period or PP : The period over which pattern gets repeated.
- Target Period or TP : The period for which the prediction is to be made.

- Utilization Period or UP : The cyclic window which the algorithm will use for prediction in TP .

Here in order to improve upon the efficiency of the system, data of several PPs is collected. In each ,even if the trend is same, the amount of traffic will be varying giving us accurate predictions.

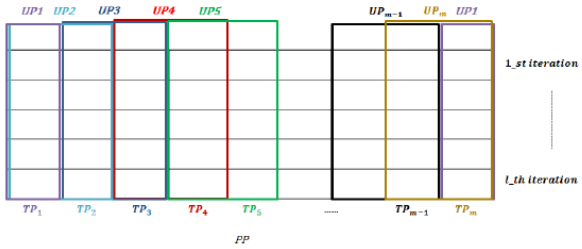

Figure 2. The Matrix Formed [2].

As seen in the figure we have an m X l matrix. Here m denotes the number of TPs and l is the PP which we collect. We start in the 1st iteration by filling the first block i.e. of TP_1 and move up to TPm. Then we move to the next PP. In this way the entire matrix is filled and after this we again begin from first block. Now onwards, we will take into account the updated matrix to consider the changing trends. In order to make a prediction we will use the data of the UPs. Say for TP_m prediction we will use UP_m.The number of UPs is decided based on how many of them will affect our current TP prediction.

Based on MLE, the nearest probability distribution model is formed which fits the histogram of historical data. The author proposes to use Poisson Distribution in the MLE. LLR is then used to estimate the parameters of probability distribution of the future TPs.

In the algorithm, consider the following:

- PData is the dataset for prediction
- PT_t are the predicted parameters of a particular TP
- AT_t are the actual parameters of that TP
- m is the number of Target Periods included in Utilization Period
- n is the number of Target Periods when the Pattern Period is functioning
- l is the number of cycles stacked on the PData
- TPData is the incoming requests at time t

Algorithm 1 Cyclic Window Learning Algorithm

Require: $PData_{m\times l}$,$TPdata_t$, m, n, and l
Ensure: PT_t and AT_t
1: $t = 1$,$p = 1$, and $w = 1$
2: **while** System operate **do**
3: **if** $p < n$ **then**
4: $UT_t = PData_{(m-n-p)\times l} : PData_{m\times l}$, $PData_{1\times l} : PData_{p\times l}, \forall l$
5: **else**
6: $UT_t = PData_{(p-n)\times l} : PData_{p\times l}, \forall l$
7: **end if**
8: Implement LLR in terms of UT_t
9: $PT_t = \hat{Y}(UT_p)$ and select kth value
10: Update databased with actual parameter
11: $AT_p = MLE(TPdata_t)$
12: Update databased with AT_p
13: $PData(p, w) = AT_p$
14: $t = t + 1$
15: **if** $p < m$ **then**
16: $p = p + 1$
17: **else**
18: $p = 1$
19: **if** $w < l$ **then**
20: $w = w + 1$
21: **else**
22: $w = 1$
23: **end if**
24: **end if**
25: **end while**

Figure 3. The Cyclic Window Algorithm [2].

- Line 1 initializes t which is the particular instance for which prediction is being made, p whose value ranges from 1 to m and w which indicates the row number in our PData
- Line 2 to Line 7 extracts the required data in the form of UT. Here if p < n then we select the data from m-n-p column to m and from 1 to p column because p is cyclic in nature. If not, then we can directly select the data from p-n to p column which in both cases gives us a total of n columns.
- In Line 8 and Line 9 the prediction is being made using Local linear Regression
- In Line 10 to Line 13 the actual probability distribution parameters are obtained by applying Maximum Likelihood Estimation on the Target Period Data requests. Then the PData data block is updated.
- Next from Line 15 to Line 22 we move to the next p for for next prediction. Here p<m means a row is not yet completed and we move to the next column. The else condition moves p back to 1 showing that a row is completed. Also then we have to move the row down and check for conditions on w. If w<l means we still have un updated rows and hence we move down. The else condition here moves us right back to the first row.

III. QoS Prediction

A. *ALPINE*

The ALPINE system is different from CloudHarmony, Amazon CloudWatch and CloudWorkbench where it claims that the others only provide raw measures of Cloud

performance and also where it can provide predict the performance even with sparse data. The system uses 5 Bayesian Networks whose development and modelling is not described in the paper. The paper talks about a Web Client which hides the complexity of these BNs and caters to them by using a Web Service.

1. The ALPINE Web Service

The Web Service has been written in java 8 and makes use of the SMILE library and is based on the JAX-RS Jersey framework. It runs on GlassFish and uses Docker to create runtime environment and Maven to build and manage the profiles. The RESTful HTTP API in JSON format is used for interpretation of the data.

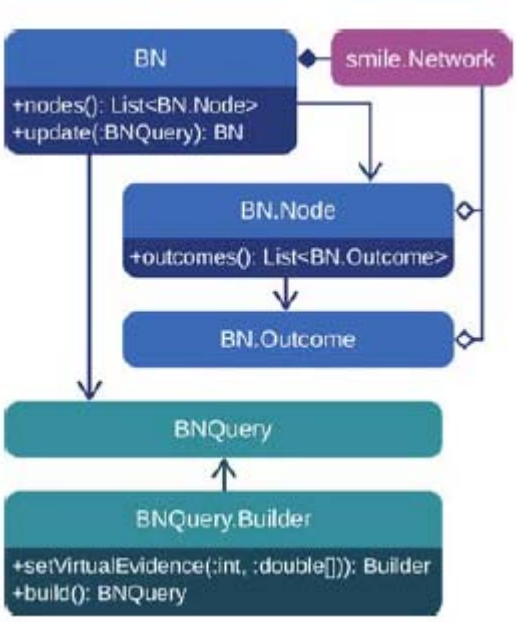

Figure 4. BN API UML [3].

As seen in figure 4, when a request for a BN is received, object of BayesianNetworkQuery class is created which is provided to the update() method of BayesianNetwork class(whichever is relevant) which is a wrapper for smile.Network class. The update() method returns the modified object relating to the query.

The following are the service endpoints provided by the ALPINE Web Service:

HTTP Path	Method	Description
/networks	GET	Lists known BNs
/networks/{tag}	GET	Serves BN with {tag}
/users	GET	List registered users
/users	POST	Creates new user
/users/{id}	GET	Serves user with {id}
/users/{id}/permissions	POST	Grants permission to user

Figure 5. Service Endpoints [3].

2. The ALPINE Web Client

Using the client, the users can use the BN nodes as per their will.
A single page web based application serves as the web client. Parts of it have been constructed using HTML5,CSS, JQuery, ECMAScript 5 and the SVG. Object Oriented Programming based constructs have also been used. D3.js library is used to develop the more complicated widgets and Google+ JavaScript API library was used as the service requires Google+ authentication. The client also uses Docker as the Container and is run in GlassFish environment.

The presentation is made as follows:

- Reader: Presents the distribution of probability in form of charts
- Writer: Presents users to select random distributions

Figure 6. CSP Writer Block [3].

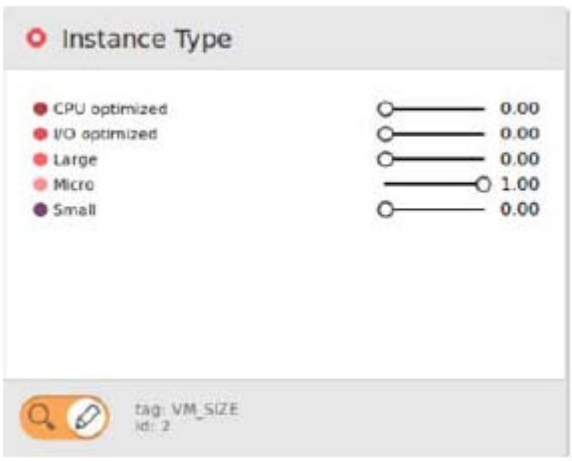

Figure 7. Instance Type Writer Block [3].

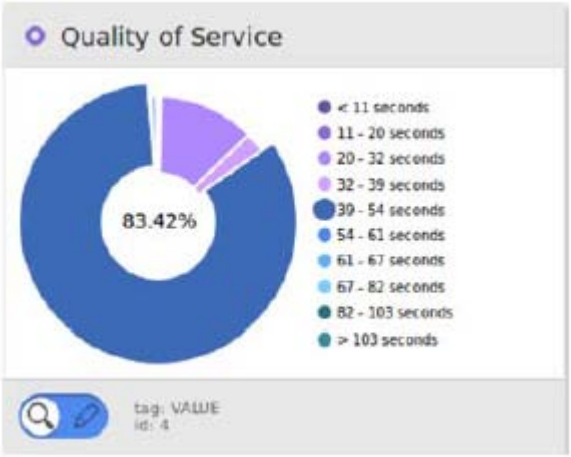

Figure 8. Output Block [3].

Once a user has been authenticated and has selected the required Bayesian Network , the system provides only the reader blocks which can be toggled to be Writer blocks. After setting the required parameters, the output such as shown in Figure 8 will be generated(Using AWS as Cloud Provider and micro as the Instance type, what is the chance that a task will be completed in given number of seconds.)

B. The Bayesian Network Model

This model collects information from 3 layers of Cloud i.e. Infrastructure, Platform and Application layer. A Bayesian network consists of a directed cyclic graph that helps to derive the probability information of some variables based on information of other variables. For simplicity, the prediction of response time and availability has been discussed.
The entire process leading to prediction has been divided into 4 steps as follows:

1. Data Collection
In this stage, not only the data related to Response time and Availability but also corresponding to the underlying hardware from the platform and infrastructure layer is collected.
2.Data Pre Treatment
Here the collected data is transformed as per a pre-specified format so that it can be fed into the model.

- P_n: Denotes the number of processes. Whether they lie within a specific value.
- C_r: Denotes the usage of CPU. If the usage lies in the range of stable changes?
- P_r: Denote usage of physical memory. If the usage lies in the range of steady changes?
- R_t: Denotes Response time. Whether it lies in the acceptable variation range?
- A_y: Denotes the availability. Is the service being accessed and responds within the given time interval?

3.Training the model

This consists of three steps:

a. Defining the variables: This step is covered in the data pre treatment where the variables have been defined.

b. Constructing the BN structure: This step uses the data without pre treatment. Network training method is used to identify the correlated variables on the data collected previously.
A fitting degree is used to calculate the correlation:

$$R = 1 - (Q/q)^{1/2}$$

here Q is the residual sum of squares and q is the sum of squares of the actual values.

c. Learning the parameters: This step consists of creating a directed acyclic graph and determination of distribution parameters of each node, where each node would correspond to a Conditional Probability Table. The edges drawn show the dependency of the variables and the table shows the strength of these dependencies.

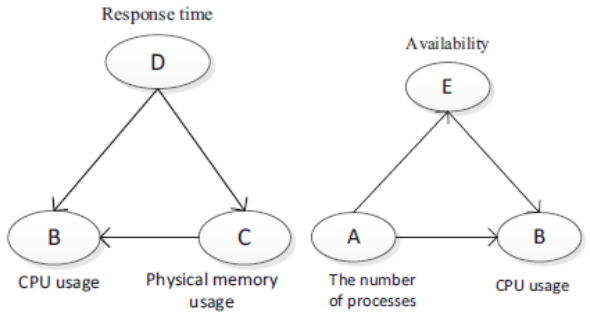

Figure 9. BN Structure[4].

4. QoS Prediction

In this phase, an inference algorithm, which goes by the name of Clique Tree propagation is used. The formula for conditional probability defined by Bayes is used to calculate P(b|a), where b is a node and a is its parent. The author proposes to generate a probability distribution table for each node and then feed it to MATLAB to construct the BN. Using the pre defined functions, Bayesian network prediction can be made.

IV. CONCLUSION

The review of ASAP and Cyclic Window Algorithm shows that they have their individual merits over the systems proposed before them for demand prediction. With time and growing technology, newer algorithms based on AI can be developed to remedy the shortcomings of the above two techniques. In case of QoS Prediction too, ALPINE claims to be better than the popularly used services in the market but the results are based on an experimental data set. Also it skips the details of implementation which makes it difficult to replicate the results. An overview of the implementation of the second model has also been discussed.

V. REFERENCES

[1] Y. Jiang, C. s. Perng, T. Li and R. Chang, "ASAP: A Self-Adaptive Prediction System for Instant Cloud Resource Demand

Provisioning," 2011 IEEE 11th International Conference on Data Mining, Vancouver,BC, 2011, pp. 1104-1109.

[2] M. S. Yoon, A. E. Kamal and Z. Zhu, "Requests Prediction in Cloud with a Cyclic Window Learning Algorithm," 2016 IEEE Globecom Workshops (GC Wkshps), Washington, DC, 2016, pp. 1-6.

[3] E. Palm, K. Mitra, S. Saguna and C. Åhlund, "A Bayesian System for Cloud Performance Diagnosis and Prediction," 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg City, 2016, pp. 371-374.

[4] P. Zhang, Q. Han, W. Li, H. Leung and W. Song, "A Novel QoS Prediction Approach for Cloud Service Based on Bayesian Networks Model,"2016 IEEE International Conference on Mobile Services (MS), San Francisco, CA, 2016, pp. 111-118.

[5] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing" NIST special Publication 800-145 ,2011

[6] Chen, Hanxiong, et al. "Resource Monitoring and Prediction in Cloud Computing Environments." Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on. IEEE, 2015.

[7] Buyya, Rajkumar, James Broberg, and Andrzej M. Goscinski, eds. Cloud computing: Principles and paradigms. Vol. 87. John Wiley & Sons, 2010