Adaptive Workload Prediction for Cloud-Based Server Infrastructures

Kritwara Rattanaopas and Pichaya Tandayya
Department of Computer Engineering, Faculty of Engineering,
Prince of Songkla University, Hat Yai, Songkhla, Thailand.
pichaya@coe.psu.ac.th

***Abstract*—Currently, data centers offer cloud computing platforms relying on virtualization technology and multi-tier architecture to handle an ever increasing scale and to provide elastic service. However, in order to achieve elasticity, efficient prediction is needed to control virtual machines. We present a new adaptive linear auto regressive technique for web server workload prediction with feedback loop control. We test the Adaptive-Feedback AR model with the Songkhla Rajabhat University's academic web which has a similar daily pattern of workloads and the learning management system (LMS) web which has unpredictable workloads. For the 1-minute interval, the suitable result for controlling the AR orders is in the range of 2-8 and previous historical value is in range of 10-25. Our new prediction approach predicts both web workloads with a root mean square error (RMSE) below 0.6, of which quality is better, in terms of the prediction accuracy resulting in a better performance.**

***Index Terms*—Adaptive Workload Prediction; Cloud Infrastructure; Web; LMS; AR Model; Workload Characteristics; Elastic Architecture; Multi-Tier Architecture.**

I. INTRODUCTION

Non-stop operations supporting data storage are necessary for data centers. At present, many companies started to design and build very large facilities with new technologies in order to handle the ever increasing scale and operational requirements of the large-scale operations. This new type of data centers relies on virtualization technology of CPUs, memory, disks, networks and software in order to provide their services. In general, the platforms can be classified into three services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Amazon Web Services (AWS) [1] has been highly successful as an IaaS. Google App Engine [2] is an example of SaaS in which users can use its software (e.g., Google docs). Microsoft Azure [3] is an example of PaaS that allows users to build and runs applications on the cloud computing platform.

The goal of cloud computing platforms is to provide elastic server resources, including the multi-tier architecture and workload prediction method. The multi-tier architecture [4] has been employed for Internet applications. In case of e-commerce, a multi-tier structure generally consists of three tiers including Web tier, Java enterprise tier and database tier.

In general, workload prediction for Internet applications [5,6] is widely known as a difficult problem in finding the best accuracy of mean-square errors. Workload prediction methods are widely discussed and used for data center resource management. Prediction approaches based on techniques including linear regression [7], neural network [8] and classification algorithms [9]. The linear auto regression is widely used for web server workload forecasting focusing on behavior patterns such as the request arrival rate.

In this paper, we focus on linear regression and employ the Auto Regressive (AR) model. We integrate the model with feedback loop control of the mean-square error value so that it is an adaptive AR model. Future workload parameters are predicted by this model. The prediction can be done by applying parameters including control order, past values and a time interval for sampling resources usage.

The target of this paper is to determine best parameters for the adaptive AR model [10,11,12,15]contributing to a less web server workload prediction error. In the workload forecast, we use the academic web workloads of the Songkhla Rajabhat University (SKRU) data center which include academic web and learning management system (LMS) servers. The academic web involves just content viewing and downloading but the LMS web has various activities involving viewing, quizzing, and uploading and downloading contents including assignments. In our case study, we have a one-year data of the two different web server workloads.

The remainder of this paper is organized as follows. Section 2 describes related works. Section 3 presents system overview, workload characteristics, the existing AR model and the proposed adaptive-feedback AR model. It presents the approaches for optimizing the parameters of our new adaptive AR model with feedback loop control. Section 4 describes results and discusses the proposed models on the academic web and LMS workloads. Finally, we conclude our work in Section 5.

II. RELATED WORKS

Web server workload prediction and characterization [5,6] have been presented for the ever increasing network traffic workload, starting from Mbytes to Terabytes or more per month. Resource management in a data center requires an optimized prediction model. The most popular linear prediction model is the Autoregressive (AR) model. There is a special-case model for web servers, which is the Autoregressive–Moving-Average model (ARMA) [10,12] that adds the moving-average terms to the AR model.

Several prediction models had claimed to be able to predict the future workload per-server required. The Adaptive Hybrid method (AHModel) combined with the Kalman and Savitzky-Golay filters is presented by Yongwei et al. [14] in order to predict the range of future workload,

called the confidence window, which is used for the Grid performance guided. Daniel et al. in [16] developed load balancer integration with the AR model to detect overloading in a data center. Vilalta et al. [15] presented short-term and long-term prediction algorithms to estimate various performance variables in a computer system including abnormal events such as QoS violations and system failures. Kandasamy et al. [17] proposed a forecasting model to predict request arrival rates using key characteristics of some representative e-commerce applications. D. Shen et al. [15] provided an AR model with a Kalman filter to predict trends in network traffic. Most researches about web server workload prediction were based on an AR model, which achieves a high precision in forecasting trends and seasonal patterns. Jiang et al. [20] used an AR model to implement a workload forecasting framework for monitoring workload in real time. The AR model integrated with neural network and other machine learning in [8] helped increase the prediction accuracy.

Recently, Calheiros et al. [19] proposed the Autoregressive Integrated Moving Average (ARIMA) model for predicting the workload of Cloud-based Software as a Service (SaaS) applications. The simulation results showed the average accuracy at 91%. In [21], Balaji et al. compared the prediction accuracy between Holt-Winter and ARIMA models applying the NASA WWW server workload. The result showed that the Holt-Winter model has a better performance.

We now propose an adaptive forecasting model to predict request arrival rates with feedback loop control technique. Our model is able to dynamically predict the results in situations with load variations.

III. METHODOLOGY

This section starts with system overview, compares workload characteristics and then presents our autoregressive model.

A. System Overview

Our experiments investigate on the SKRU academic web and LMS web workloads. The academic web infrastructure exploits a big virtual machine which has two vCPUs and 4 GB RAM. The LMS web is a multi-tier architecture, including load balance tier, web server tier, database server tier, and storage server tier as seen in Figure 1. The LMS exploits two hosts, each host server is Xeon E5520, 2.27 GHz, Quad cores with 16 GB RAM and 2-gigabit network interface.

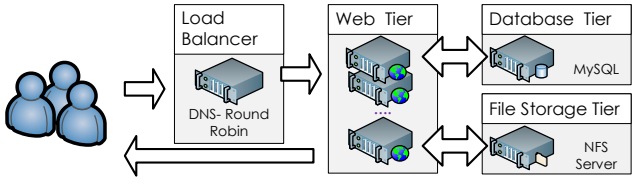

Figure 1: Elastic LMS web multi-tier architecture

In Figure 1, each computer runs the CentOS 6.2, 64 bits, and the KVM hypervisor on top of it. A virtual machine runs the CentOS 6.2, 64 bits, the same as that of the host computer. We use Moodle version 1.9.5+ which has been widely used in LMS and many standard learning service modules. We split the services of Moodle into web, database and file storage. The Web Server Tier handles requests and sends responses directly to the user, bypassing DNS load balancing. The Database Tier runs MySQL. The File Storage Tier server employs NFS 4.0 to store media files.

B. Workload Characteristics

The testing has been conducted using the access logs of the SKRU academic web and LMS web servers. The Apache access log is an ASCII text file of which format is one line per request. For example, the log format depicted *"%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\" %U "* can be described below:

- %h : Remote host
- %l : Remote logname (from daemon, if supplied)
- %t : Time the request was received
- %s : Status for requests
- %b : Size of response in bytes
- %U : The URL path requested, not including any query string.

In this section, we compare the access logs information, including SKRU academic web and LMS web, with the general world web server data (World Cup 98) in Table 1.

Table 1
Comparison of workloads of access logs (SKRU LMS, SKRU cademic web and World Cup 98 [22])

Workload/Web Type	SKRU academic Web	SKRU LMS	World Cup 98
Access Log Duration	22/11/2010 – 22/11/2011 (12 months)	22/11/2010 – 22/11/2011 (12 months)	1/5/1998-23/6/1998 (2 months)
Avg. Requests/Day	310,676	22,970	15,546,240
Avg. Bytes/Day (MB)	7,646.1	722.80	58,752.0
Avg. Byte(KB)/ Requests	24.6	32.2	3.8

Table 1 shows the number of average requests and data transfer per day on each type of web servers. The LMS has the number of average requests per day lower than the other. On the other hand, its average data transfer per request is more than the other. The LMS workloads vary and are different by semester due to different courses and activities. The average SKRU LMS workload per hour on each weekday can be demonstrated in Figure 2.

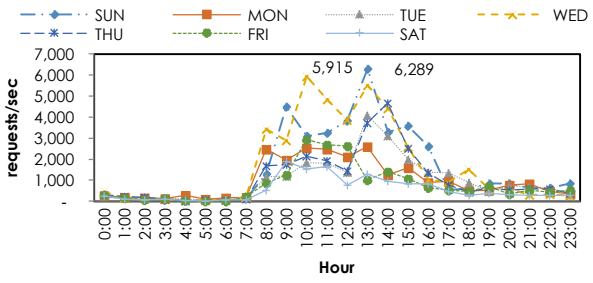

Figure 2: Comparison of the average SKRU LMS web workload per hour on each weekday

In Figure 2, each weekday has different courses scenario and number of students. The maximum average workloads are 6,289 requests per second on Wednesday from 9.00 to 10.00 and 5,915 requests per second on Sunday from 13.00 to 14.00. As depicted in Table 2, we can presume that

course schedules and activities in a semester affect with the LMS web workload.

Table 2
Comparison of SKRU LMS web workloads from Semester 2/2010 to Semester 1/2011

Workload/Semester	2/2010	3/2010	1/2011
Avg. Requests/Day	22,994.87	3,921.57	39,458.90
Avg. Bytes/Day (MB)	672.99	106.33	1,285.66
Avg. Byte(KB)/Request	30.0	27.8	33.4

In Table 2, the number of average requests per day in Semester 1/2011 is more than that of Semester 2/2010, reflecting increasing demands and interactions. Semester 3/2010 has the least number as it is not a main semester. The activities of workload requests include presentation files, course e-books, self-examination and unpredictable course activities. The effects of unpredictable activities workload are presented in Figure 2.

In Figure 3, we start with the workload of LMS in each semester and then present the workload of the SKRU academic web in order to compare the workload characteristics. Figure 3 shows a similar trend in the SKRU academic web workload characteristics of each weekday. The workload peeks from 9.00 to 16.00 every day.

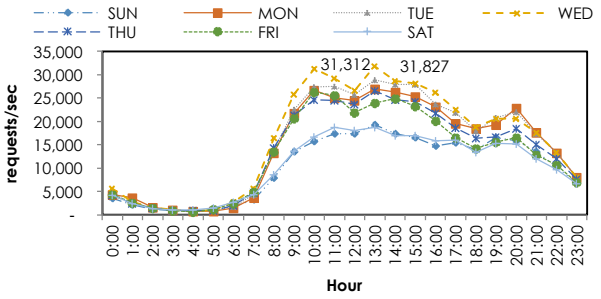

Figure 3: Comparison of the average SKRU academic web workload per hour on each weekday

The SKRU academic web and LMS web servers obtain different characteristics, shown by the maximum, mean and standard deviation of workload per hour of each weekday. The LMS web provides an unpredictable workload from 8.00 to 16.00, which is the classroom time. The highest numbers of LMS average requests per second in different times on different days vary unpredictably and seem not related with each other. On the other hand, the SKRU academic web shows a similar workload trend at the same time on each day. In this paper, we experiment by setting up predictable and unpredictable workload scenarios for testing our prediction approach later described in the section as the Adaptive-Feedback AR Model.

C. Autoregressive Model

Our prediction manager module employs the AR model to compute the future workload of the system. The main idea in the AR model [10],[11],[14] is described as follows:

$$X(n) = \sum_{m=1}^{M} a_m X(n\text{-}m) + a_0 + \epsilon(n) \quad (1)$$

where $X(n)$ is the future workload at time n, $\epsilon(n)$ is a white noise, and M denotes the order of the AR model. The higher the order, the more accurate it is. The AR coefficients are a_m and a_0 determined by the historical previous request values of the servers. The workload of which value, in our case, is decided based on the client request utilization. For this prediction, the future workload $X(n+1)$ at time n is based on the M number of previous historical values $\{X(n), X(n\text{-}1), \dots, X(n\text{-}M+1)\}$ using the coefficients $\{a_1, a_2, \dots, a_M\}$ and a_0.

The model has been applied for a new load balancing scheme on the existing cloud services [8],[16]. In this work, we use the model for helping DNS round robin load balancing in a virtual data center system.

The computation of the future workload for each time n requires a few hundreds of previous requests to the server. The value of future workload at n is calculated within a few milliseconds. This computational cost is rather low, and thus, applicable in real practices.

D. Adaptive-Feedback AR Model

In this paper, we propose a new approach for linear prediction by using integrated feedback loop control. An Adaptive-Feedback AR model is integrated with feedback for controlling the minimum value of the Root Mean Square Error (RMSE). We use a feedback control to detect and monitor the RMSE of real and predicted workloads. The autoregressive parameters of this model can be changed when the RMSE is close to the set-point in order to provide adaptive workload prediction.

The basic idea of our workload prediction model can be summarized in Figure 4. The key parameters for adaptive prediction model include RMSE from the RMSE module, i_{best} which is the AR order, j_{best} which is the number of previous historical values and $A[0..i_{best}]$ which is an AR coefficient matrix. The new model aims for less prediction error.

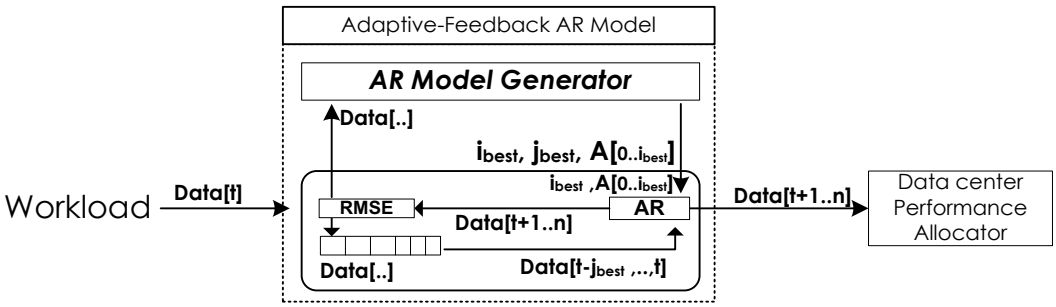

Figure 4: The process diagram of our Adaptive-Feedback AR model

In Figure 4, the general structure of our Adaptive-Feedback AR model includes AR Model Generator, Autoregressive (AR) and RMSE modules. The AR Model Generator describes assumptions and contains the AR algorithm modified for finding new optimized parameters of AR as shown in Algorithm 1 that includes the order of AR, number of previous historical values and coefficient matrix. A workload value is the input of the RMSE module for calculating an RMSE with prediction values from AR when the RMSE is close to the set-point. RMSE data is then sent to the AR Model Generator. The AR used for estimating *n* steps ahead of the workload values to be sent to the Data Center Performance Allocator.

In Algorithm 1, the Autoregressive (AR) Model Generator has two loops to find out a better root mean square error at the time *t*. In the first loop, we determine RMSE for the AR order from 2 to 16. In the second loop, we must start with the number of *i* on each round of the first loop and repeat the loop up to *i* = 40. The process inside the loop is the Revert AR method and RMSE test in j steps. If the value is lower than the previous RMSE, it replaces i_{best}, j_{best} and A[0..i_{best}] with *i*, *j* and A[0..i] of the round.

Algorithm 1: Autoregressive (AR) Model Generator
Input : array workload data (1-minute-interval)
Output: AR order i_{best}, Previous values j_{best} and array regression coefficient $A[0..i_{best}]$

/*Calculate $a_0..a_{Order}$ for finding the minimized the Root Mean Square */
1 Data[t] = Number of Requests at time *t*
2 for *Order* i 2 to 16 do
3 for *Previous Historical Value* j Order(i) to 40 do
4 A[0..i] = *Revert Autoregressive Method(*i, j, Data[t]*)*
5 *Root Mean Square Error(* j *)*
6 If RMSE is minimum then
7 i_{best}, j_{best}, A[1..i_{best}] ← i, j, A[1..i]
8 end for
9 end for
/* A better RMSE at time *t* is i_{best}, j_{best}, A[1..i_{best}]*/
10 Return i_{best}, j_{best}, A[0..i_{best}]

IV. Results and Discussion

We have applied the Adaptive-Feedback AR model and tested the system with the SKRU academic and LMS web workloads. We exploit the data during the period starting from November 22, 2010 to November 22, 2011 which is exactly one year and record only the control order and previous values which produce the least RMSE value in each minute interval as shown Figures 5 - 10 and then plot a 24-hour graph of each day. Applying the proposed model on both types of SKRU webs, the results have achieved the RMSE values around 0 to 1. The objective of this result is to replace the range of order on the 2nd line and historical values on the 3rd line of the prototype AR model generator in Algorithm 1.

A. Learning Management System (LMS) web

In this session, we present the results of our model on the LMS web workload in Figures 5-7. The model has been tested applying the data of an LMS web with various requirements on each day and each time of the day. The optimized AR order is shown in Figure 5.

In Figure 5, referring to the server usage in the LMS workload, it shows a continuous system usage only from around 8 a.m. to about noon. After 1 a.m., the usage has gradually decreased so that the AR order value is inconsistent. After 8 a.m., the best AR order values are in the range of 2-6 for the optimized previous historical value (requests) that can be calculated at the AR model generator as shown in Figure 6.

In Figure 6, the result of the previous historical values after 1 a.m. also is inconsistent because it has lower usage or no request to the web server, similar to that of the AR order values. After 8 a.m., the optimized previous historical values are the best for AR model calculation, in the range of 10-25, during the period with low server usage that affects the prediction. On the other hand, in the day time, there are different usages on different weekdays. The prediction has a consistent result in the range of 15-25. The RMSE calculated from the AR order value and the previous historical values can be shown in Figure 7.

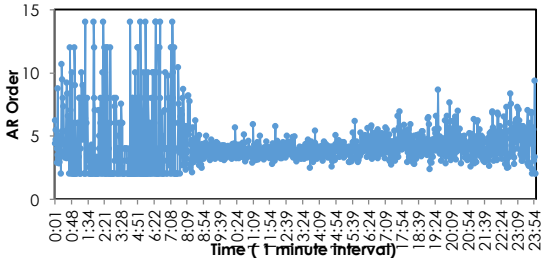

Figure 5: Optimized AR order values of LMS web (1-mintue-interval)

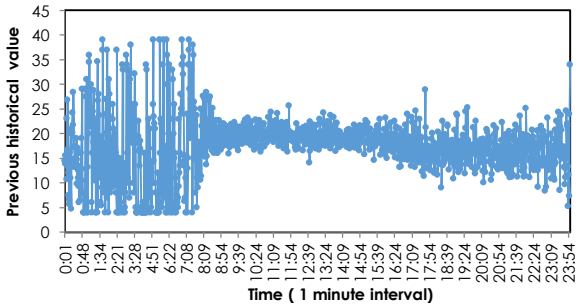

Figure 6: Optimized previous historical values of LMS web (1-mintue-interval)

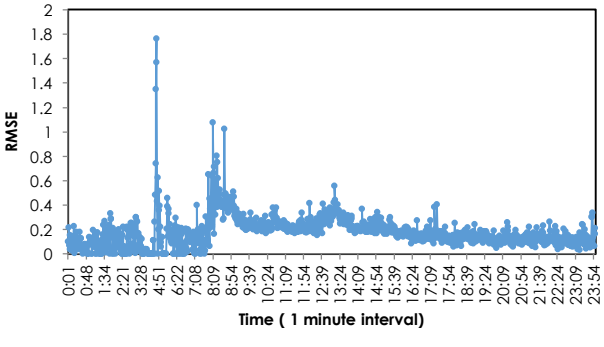

Figure 7: RMSE values of LMS web (1-mintue-interval)

In Figure 7, due to the period of lower usage or no usage, after 1 a.m., the results of RMSE are inconsistent. However, after 8 a.m., the RMSE is not more than 0.4, of which quality is a quite good; referring to the prediction accuracy that the better performance value for general web servers is in the range of 0.2-0.82 in [14]. Then, for the period with lower usage, after 4 p.m., the results of RMSE are not more than 0.2, of which quality is good in terms of the prediction accuracy, similar to the SKRU academic web workload during 1 a.m. to 8 a.m.

B. SKRU academic web

In case of the academic web, the pattern is consistent for all weekdays. According to the Adaptive-Feedback AR model predictor, the AR order and the previous historical values can be shown in Figures 8-10.

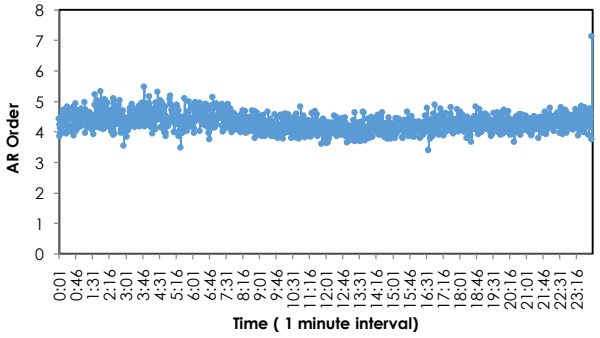

Figure 8: Optimized AR order values of SKRU academic web (1-mintue-interval)

In Figure 8, referring to the server usage in the SKRU academic web workload, it shows a similar system usage around 4-5 every hour of the day so that the optimized previous historical values (requests) can be calculated on the AR model generator as shown in Figure 9.

In Figure 9, the results of the previous historical values are similar to that of the AR order values. After 8 a.m., the previous historical values are the best for AR model calculation, in the range of 15-20, during the period with a similar pattern of user requests. However, in the day time, the usages are similar on all weekdays. The RMSE calculated from the AR order values and the previous historical values can be shown in Figure 10.

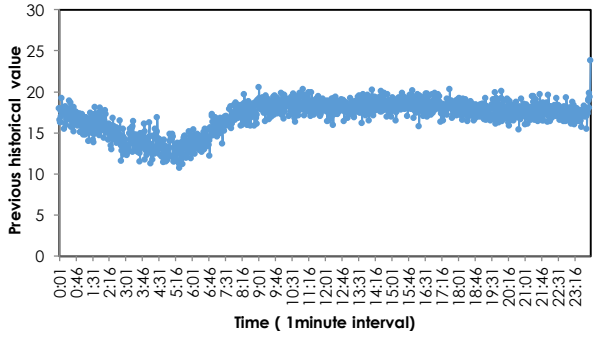

Figure 9: Optimized previous historical values of SKRU academic web (1-mintue-interval)

In Figure 10, due to the similar usage after 1 a.m., the results of RMSE values are better at 0.1. After 8 a.m., the RMSE is in the range of 0.3-0.6, of which quality is good in terms of the prediction accuracy resulting in a better performance with SKRU academic web workload. The RMSE values of the SKRU academic web are more than that of the LMS web workload. The RMSE result of the Adaptive-Feedback AR model is compatible with the similar pattern workload of the SKRU academic web as it is lower than the unpredictable workload of the LMS web.

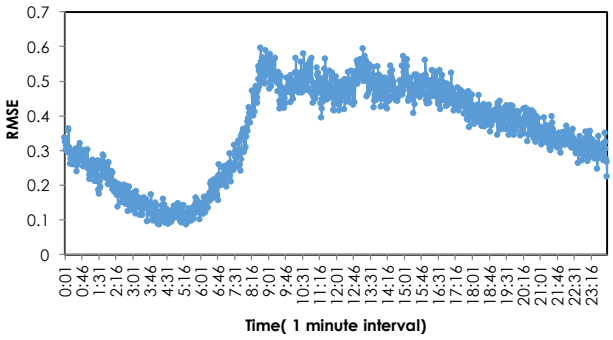

Figure 10: RMSE values of SKRU academic web (1-mintue-interval)

V. CONCLUSION

In our experiments, we have applied the new prediction approach based on autoregressive techniques set for predicting web server workload with feedback control for calculating the AR model parameters for the SKRU academic and LMS webs. Both webs differ in workloads and service requirements as analyzed within the 1-minute-interval. The results show that the workload prediction applying RMSE is below 0.6. Referring to the RMSE values for workload prediction of a web server is in the range of 0.2-0.82 in [14], our method is considered effective in case of continuous workload like the academic web. In case of inconsistent workload like the LMS web that varies unpredictably due to courses management and periodically in relation with the time of day, the suitable AR order value is in the range of 2-6 and previous historical value is in the range of 15-25. Owing to the determined values, we can narrow down the calculation set values of the AR model generator. As a consequence, the prediction speed is faster for controlling virtual machines in real time. The AR model parameters affect the accuracy of future workload prediction in the reality. Finding suitable parameters for each generic type is highly recommended. The ARIMA is an example of that using the simulated environment [19].

ACKNOWLEDGEMENT

The authors would like to thank Prince of Songkla University for the Graduate Study Scholarship and Songkhla Rajabhat University for workload information.

REFERENCES

[1] Amazon. Amazon Web Services (AWS). http://aws.amazon .com.
[2] Google. Google App Engine. http://code.google.com/ appengine.
[3] Microsoft. Windows Azure. http://www.microsoft.com/ windowazure.
[4] Urgaonk ar B., Pacifici G., Shenoy P., Spreitzer M. and Tantawi A., 2005. An analytical model for multi-tier internet services and its applications. in In *Proc. of the ACM SIGMETRICS'*2005. 291–302.
[5] Joseph F. Z. and Hellerstein L., 2000. Characterizing Normal Operation of a Web Server: Application to Workload Forecasting and Problem Detection.
[6] Arlitt M. F. and Williamson C. L., 1996. Web server workload characterization: the search for invariant. in Proceedings of the *1996 ACM SIGMETRICS International Conference on Measurement and modeling of computer systems*, New York, NY, USA. 126-137.
[7] Akaike H., 1969. Fitting Autoregressive Models for Prediction, Annals of the Institute of Statistical Mathematics. 21 (1):243-247, Dec.
[8] Prevost J. J., Nagothu K., Kelley B., and Jamshidi M., 2011 Prediction of cloud data center networks loads using stochastic and neural models. in *2011 6th International Conference on System of Systems Engineering (SoSE)*. 276–281.

[9] Gmach D., Rolia J., Cherkasova L., and Kemper A., 2007 Workload Analysis and Demand Prediction of Enterprise Data Center Applications. in Proceeding of *IEEE's 10th International Symposium on Workload Characterization*, 2007. IISWC 2007. 171-180.

[10] Brockwell P. J. and Davis R. A., 1991. Time Series: Theory and Methods. New York, NY: Springer New York.

[11] Chandra M. and Ray M., 2015. Comparative Study of PCM, LPC, and CELP Speech Coders Used for VoIP Applications. in Intelligent Computing, Communication and Devices, L. C. Jain, S. Patnaik, and N. Ichalkaranje, Eds. Springer India. 579–587.

[12] J. M. Tirado, D. Higuero, F. Isaila, and J. Carretero, Predictive Data Grouping and Placement for Cloud-Based Elastic Server Infrastructures, in 2011 *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid),* 2011, pp. 285–294.

[13] Vercauteren T., Aggarwal P., Wang X., and Li T.-hsin, 2007. Hierarchical Forecasting of Web Server Workload Using Sequential Monte Carlo Training. *Ieee Transactions On Signal Process*. 55:1286-1297,

[14] Wu Y., Hwang K., Yuan Y.i, and Zheng W., 2010. Adaptive Workload Prediction of Grid Performance in Confidence Windows. *IEEE Transactions on Parallel and Distributed Systems.* 21(7):925-938, Jul.

[15] Shen D. and Hellerstein J. L., 2000. Predictive Models for Proactive Network Management: Application to a Production Web Server, in Server. Proc. Network Operations & Management Symp. 833–846.

[16] Daniel S. and Kwon M., 2015. Prediction-based virtual instance migration for balanced workload in the cloud datacenters, 2011.[Online].Available:https://ritdml.rit.edu/handle/1850 /14203. [Accessed: 27-Sep-2015].

[17] Vilalta R. et al., 2002. Predictive Algorithms in the Management of Computer Systems, *IBM Systems Journal*. 41(3):461-474.

[18] Kandasamy N., K N., Abdelwahed S., and Hayes J. P., 2004. Self-Optimization in Computer Systems via Online Control: Application to Power Management.

[19] Calheiros R., Masoumi E., Ranjan R., and Buyya R., 2014. Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS. *IEEE Transactions on Cloud Computing.*99:1–1,

[20] Jiang S., Chen H., and Hu F., 2014. Workload forecasting framework for applications in cloud. in *2014 International Conference on Cloud Computing and Internet of Things (CCIOT).* 31–38.

[21] Balaji M., Rao G. S. V., and Kumar C. A., 2014. A Comparitive Study of Predictive Models for Cloud Infrastructure Management. in 2014 *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid),* 923–926.

[22] Arlitt M. and Jin T. 1999. Workload Characterization of the 1998 World Cup Web Site. Tech. Report, HPL-9935R1, Hewlett-Packard Labs. September.