IJCTA, 9(12), 2016, pp. 5577-5582

Host CPU Load Prediction Using Statistical Algorithms : A Comparative Study

***Shaik Naseera **G.K. Rajini ***P. Sunil Kumar Reddy**

Abstract : Resource sharing in heterogeneous and dynamic environments like Grid or Cloud requires mapping of tasks to resources for utilization of space capacity of the lightly loaded resources. To make the resource utilization more effective, we need to know their workload in advance. The load of these resources changes dynamically in Grid/Cloud environments. It is helpful for job scheduling algorithms, if one can predict the load of these resources in advance. The efficiency of job scheduling/resource sharing algorithms depends on how accurately these algorithms predict the resource's load. In this paper, we have considered the static and dynamic versions of prediction algorithms Homeo-static, Tendency-based, Step-Ahead based algorithms for predicting the host CPU load. These prediction algorithms predict the CPU load of the resource for future interval of the time based on the resource previous load history. The experimentation is done using these algorithms and results show that the dynamic Tendency based prediction algorithm gives better prediction accuracy compared to other algorithms.

Keywords : Load Prediction, Statistical Methods, CPU Load, Dinda Load Traces, Prediction Accuracy.

1. INTRODUCTION

Resource-sharing in a dynamic environment means that applications must be able to share the spare capacity of the resources spread over the network [6]. The resource sharing algorithms should adapt their behavior based on the changes in the system state. Prediction of future system load guide such adaptations and prediction of execution time of the application guides the scheduling algorithms to estimate whether the system is lightly loaded or heavily loaded. If one can predict the load during the execution of a task on a host, the execution time of the task on that host can be easily determined. Therefore host CPU load prediction is important for job scheduling and resource sharing algorithms.

Prediction Strategies

Several load prediction algorithms are proposed in the literature. The most popular prediction strategies are :

1. Homeostatic Prediction Strategy (HSPS)
2. Tendency-based prediction strategy (TBPS)
3. Step Ahead Prediction Strategy (SAPS)

Every prediction strategy predicts the value one step ahead based on the predefined number of past load values measured at a fixed time interval. The prediction strategies are further classified into static and dynamic. In static predictions, the prediction is by changing the current value with a predetermined amount for all prediction steps. Dynamic predictions are adaptive and self corrective in nature, the value predicted depends on the adaptation process and may be different for different prediction steps. In this paper, we consider the static and dynamic nature of all the prediction strategies listed in Sec.1.1.

* School of Computer Science and Engineering, VIT University, Vellore
** School of Electrical Engineering, VIT University, Vellore
*** Department of Computer Science and Engineering, SVCET, Chittoor

1. **Homeostatic Prediction Strategy :** Works on the strategy that if the current value is greater or less than the mean of history values then the next value is likely to decrease or increase respectively.
2. **Tendency Based Prediction Strategy :** This prediction strategies works on the strategy that, the next value is predicted based on the tendency of the change in the time series. *i.e*., if the current value increases then the next value also increases and accordingly when the current value decreases then the next value will also decrease.
3. **Step-Ahead Prediction Strategy :** This strategy predicts the next value using polynomial fitting method and similar patterns. In this method, the relationship between the independent variable x and the dependent variable y is represented by a function $y = f(x)$. Function $f(x)$ is then used to estimate the value of y over region x by applying the approximation.

   Section 2. gives the related work in the field. Section 3. gives the experimental evaluation Section 4 demonstrates the performance analysis of all the prediction strategies Section 5 present conclusions of the work.

Related Work

In a multi-user time sharing environment, applications share workload by sharing the resources in the network. In this connection, applications compete for the shared resources. This results varying availability and load for the resources.

Performance prediction is very useful for applications and schedulers for getting better performance in response to changes in the system state [9,14]. Schedulers can guide the scheduling strategy for efficient resource usage and application performance. Generally, CPU load generally affect the run time of the CPU-bound applications. Sometimes, the host CPU load and the application run time are directly proportional to each other [2,19]. Performance prediction is useful for both applications and schedulers.

Generally, the factor for making the accurate prediction is to suitably model the relationship between the past data and future data. Time series prediction model is a simple prediction model and is widely used for predicting the tendency in financial data [3,7,18], biomedical signal processing [4], networking [1,11], and earth and ocean sciences [5].

Network Weather Service [10, 15-17] is a adaptive prediction strategy and uses nine prediction models ranging from running average to autoregressive [10,17] for predicting the next value. NWS chooses one among these models dynamically by considering time series.

Dinda et al. evaluated multiple linear models like moving average (MA), autoregressive (AR), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and autoregressive fractionally integrated moving average (ARFIMA) models [12,13]. The authors have proved that, the simple Autoregressive model is the best model compared to others in terms of prediction power and overhead. This AR model is also used in NWS.

One-step prediction strategy predicts the value for the next time interval. Multiple step ahead prediction predicts the load for the longer time interval. It is found in [5] that one-step ahead prediction model is mostly a good estimate compared to multiple step prediction model for shorter intervals.

In [8] Yang et al. proposed various prediction strategies like homeostatic, tendency-based and one-step-ahead prediction strategies. Homeostatic prediction strategies are based on the assumption that the mean of a time series remains stable. In Tendency-based prediction strategy the “tendency” is assumed to be similar in the time series *i.e.*, if the current value increases then the next value also increases, or vice versa. The increment/decrement value used in adjusting the prediction error. It is shown that this technique performs better that the other methods in NWS for CPU load predictions.

Therefore, in this paper, we have chosen the prediction strategies like HSPS, TBPS and SAPS for performance analysis and we have presented a comparative study on the prediction accuracy of the algorithms present in the literature.

2. EXPERIMENTAL EVALUATIONS

Experimental Environment

We have run the prediction methods for the CPU load traces collected by Dinda [5] on variety of machines namely *Pitcairn.mcs.anl.gov, Mystere.ucsd.edu, Vatos.cs.uchicago.edu, Abyss.cs.uchicago.edu, Abyss.cs.uchicago.edu*, *Axp7.psc.edu Axp0.psc.edu, Sahara.cmcl.cs.cmu.edu, Themis.nectar.cs.cmu.edu*. The properties of these machines and the sampling interval, sampling period chosen are given in [5]. We also conducted a similar type of sampling on our server SITAMS (Sreenivasa Institute of Technology and Management Studies, Chittoor, India) and collected the CPU load traces as mentioned below.

Sitams-linux server is a moderately loaded desktop machine. The load on this machine has high standard deviation. The total number of data in this time series is 50,000 (for 2 days).

Sitams- netware-server is a moderately loaded desktop machine. This machine has higher standard deviation load. The total number of data in this time series is 20,000(for 2days).

Sitams-mail-server is a moderately loaded desktop machine. The machine load possess high standard deviation. The total number of data in this time series is 20,000(for 2 days).

Sitams-proxy-server is a moderately loaded desktop machine. This machine has higher standard deviation load. The total number of data in this time series is 20,000 (for 2 days).

Input Parameters

All of the prediction strategies mentioned here take input parameters to identify how much value to increment or decrement or how to change a prediction value over time. For every prediction step, the chosen increment or decrement value may be an independent value or a relative value proportional to the present measurement. The increment or decrement value can be static or dynamic. static is a fixed value fixed for all the prediction steps. Dynamic is the adapted value chosen based on the time series at each step. The different combinations of these strategies are applied to Homeostatic and Tendency based prediction strategies that results the following combinations.

1. Independent static homeostatic prediction strategy
2. Independent dynamic homeostatic prediction strategy
3. Relative static homeostatic prediction strategy
4. Relative Dynamic homeostatic prediction strategy
5. Independent static Tendency Based prediction strategy
6. Independent dynamic Tendency Based prediction strategy
7. Relative static Tendency Based prediction strategy
8. Relative Dynamic Tendency Based prediction strategy
9. Mixed Dynamic Tendency Based prediction strategy

In all the mentioned prediction strategies, the values are determined by running a set of experiments to search on the space of feasible selections. We did this using training data that is separate from the experiment data. We used 25 one-hour-long time series and evaluated increment and decrement values at intervals of 0.05 between 0 and 1 using the error formula.

$$\text{Average Error Rate} = \frac{\sum_{i=1..N} abs(P_i - V_i)/V_i}{N} * 100\%$$

Where V_i is the measured value, P_i is the predicted value and N is the number of data in the time series to be tested. we found the best results with

$$\text{Increment Constant} = \text{Decrement Constant} = 0.1$$

$$\text{Increment Factor} = \text{Decrement Factor} = 0.05$$

$$\text{Adapt Degree} = 0.5$$

3. PERFORMANCE ANALYSIS

This section tabulates the mean and Standard Deviation(SD) of the prediction values predicted by static and dynamic versions of HSPS, TBPS and SAPS.

Table 1. Mean and SD of the prediction errors collected from AXP0.PSU.EDU

S.No	*Prediction strategy name*	*Mean (%)*	*Standard deviation*
1.	**ISHPS**	0.061103	0.03786
2.	**IDHPS**	0.058047	0.035967
3.	**RSHPS**	0.046582	0.037982
4.	**RDHPS**	0.044112	0.036139
5.	**IDTPS**	0.053148	0.045873
6.	**RDTPS**	0.037988	0.041737
7.	**MDTPS**	0.041774	0.047145
8.	**SHPS**	0.024198	0.043837

Table 2. Mean and SD of the prediction errors collected from AXP7.PSU.EDU

S.No	*Prediction strategy name*	*Mean (%)*	*Standard deviation*
1.	**ISHPS**	0.066632	0.023019
2.	**IDHPS**	0.063301	0.021868
3.	**RSHPS**	0.048037	0.024515
4.	**RDHPS**	0.045264	0.023129
5.	**IDTPS**	0.061357	0.046475
6.	**RDTPS**	0.03883	0.035561
7.	**MDTPS**	0.048391	0.041897
8.	**SHPS**	0.04171	0.030647

4. CONCLUSION

Prediction of resource availability generally benefit the applications and schedulers for making decisions related to utilization of time-shared resources. In this paper, we evaluated the performance of static and dynamic versions if different prediction strategies like HSPS, TBPS and SAPS for time series collected for 12 different machine's.

From the experiment results, we noticed that giving more weight to most recent values significantly affects the prediction accuracy.

Table 3. Mean and SD of the prediction errors collected from Shara CMCl.CS.CMU.EDU

S.No	*Prediction strategy name*	*Mean (%)*	*Standard deviation*
1.	**ISHPS**	0.998616	0.04437
2.	**IDHPS**	0.059538	0.01873
5.	**RSHPS**	0.038547	0.024863
6.	**RDHPS**	0.036785	0.023923
7.	**IDTPS**	0.059166	0.038566
8.	**RDTPS**	0.030419	0.032858
9.	**MDTPS**	0.038074	0.03933
10.	**SHPS**	0.032672	0.029873

Table 4. Mean and SD of the prediction errors collected from the MIS.NECTAR.CS.CMU.EDU

S.No	*Prediction strategy name*	*Mean (%)*	*Standard deviation*
1.	**ISHPS**	1.000212	0.033936
2.	**IDHPS**	0.0545504	0.018359
3.	**RSHPS**	0.04464	0.0182
4.	**RDHPS**	0.042042	0.017491
5.	**IDTPS**	0.52183	0.030088
6.	**RDTPS**	0.037287	0.029899
7.	**MDTPS**	0.040537	0.033598
8.	**SHPS**	0.21727	0.25978

Table 5. Mean and SD of the prediction errors collected from ABYSS.CS.UCHICAGO.EDU

S.No	*Prediction strategy name*	*Mean (%)*	*Standard deviation*
1.	**ISHPS**	0.21264	0.21264
2.	**IDHPS**	0.24713	0.24713
3.	**RSHPS**	0.22466	0.22466
4.	**RDHPS**	0.21264	0.21264
5.	**IDTPS**	0.24713	0.24713
6.	**RDTPS**	0.22466	0.22466
7.	**MDTPS**	0.21264	0.21264
8.	**SHPS**	0.22466	0.22466
9.	**SABPS**	0.21264	0.21264

Table 6. Mean and SD of the prediction errors collected from VATOS.CS.UCHICAGO.EDU

S.No	*Prediction strategy name*	*Mean (%)*	*Standard deviation*
1.	**ISHPS**	1.000212	0.033936
2.	**IDHPS**	0.0545504	0.018359
3.	**RSHPS**	0.04464	0.0182
4.	**RDHPS**	0.042042	0.017491
5.	**IDTPS**	0.52183	0.030088
6.	**RDTPS**	0.037287	0.029899
7.	**MDTPS**	0.040537	0.033598
8.	**SHPS**	0.21727	0.25978

5. REFERENCES

1. H.-W. Braun, B. Chinoy, K. C. Claffy, and G. C. Polyzos, "Analysis and Modeling of High Speed Networks: 1993 Annual Status Report," San Diego Supercomputer Center and University of Californina, San Diego CS92-237, 1993.
2. D. Lu, H. Sheng, and P. Dinda, "Size-based scheduling policies with inaccurate scheduling information," *12th IEEE Int'l Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems(MASCOTS'04)*, pp. 31-38, 2004.
3. G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis Forecasting and Control*, Prentice Hall, 1994.

4. H. Kato and H. Kawahara, “An Application of the Bayesian Time Series Model and Statistical System Analysis for F0 Control,” *Speech Communication*, vol. 24, pp. 325-339, 1998.
5. http://www.cs.cmu.edu/~pdinda/LoadTraces/
6. I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, San Fransisco, CA, 1999.
7. J. Liang, K. Nahrstedt, and Y. Zhou, “Adaptive Multi- Resource Prediction in Distributed Resource Sharing Environment,” *2004 IEEE Int’l Symp. on Cluster Computing and the Grid*, pp. 1-8, 2004.
8. L. Yang, I. Foster, and J.M. Schopf, “Homeostatic and tendency-based CPU load predictions,” *Int’l Parallel and Distributed Processing Symp. (IPDPS’03)*, pp. 42-50, 2003.
9. L. Yang, J.M. Schopf, and I. Foster, “Conservative scheduling: Using predicted variance to improve decisions in dynamic environment,” *Supercomputing’03*, pp. 1-16, 2003.
10. M. Swany and R. Wolski, “Multivariate resource performance forecasting in the network weather service,” *Supercomputing’02*, pp. 1-10, 2002.
11. N. K. Groschwitz and G. C. Polyzos, “A Time Series Model of Long-Term NSFNET Backbone Traffic,” presented at Proceedings of the IEEE International Conference on Communications(ICC’94), 1994.
12. P.A. Dinda, “A prediction-based real-time scheduling advisor,” *Proc. 16th Int’l Parallel and Distributed Processing Symp.(IPDPS 2002)*, pp. 35-42, 2002.
13. P.A. Dinda, “The statistical properties of host load,” Technical report, CMU, pp. 1-23, 1998.
14. P.A. Dinda, D.R. O’Hallaron, “Host load prediction using linear models,” *Cluster Computing 3(4)*, pp. 265-280, 2000.
15. R. Wolski, “Dynamically forecasting network performance using the network weather service,” *Journal of Cluster Computing*, Vol.1, pp.119-132, 1998.
16. R. Wolski, “Experiences with predicting resource performance on-line in computational grid settings,” *ACM SIGMETRICS Performance Evaluation Review*, Vol.30, No.4, pp. 41-49, 2003.
17. R. Wolski, N. Spring, and J. Hayes, “Predicting the CPU Availability of Time-shared Unix Systems on the Computational Grid,” *Proc. 8th IEEE Symp. on High Performance Distributed Computing*, pp. 1-8, 1999.
18. S. Akioka and Y. Muraoka, “Extended forecast of CPU and network load on computational grid,” *2004 IEEE Int’l Symp. on Cluster Computing and the Grid*, pp. 765-772, 2004.
19. S. Jang, X. Wu, and V. Taylor, “Using performance prediction to allocate grid resources,” Technical report, GriPhyN 2004-25, pp. 1-11, 2004.