Adaptive Hybrid Model for Long Term Load Prediction in Computational Grid

Yulai Yuan[1], Yongwei Wu[2], Guangwen Yang[3], Weimin Zheng[4]
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[1]*yuan-yl05@mails.tsinghua.edu.cn*
[2, 3, 4]*{wuyw, ygw, zwm-dcs}@tsinghua.edu.cn*

Abstract

Long term load prediction can assist task scheduling and load balancing greatly in distributed environment such as computational grid. Due to the dynamic property of grid environment, fixed-parameter prediction model can not exert its forecast capability completely. In this paper we first observe and analyze parameters' impact on prediction accuracy for our previous long term load prediction hybrid model (HModel) in detail. And then, a parameter-level adaptive method based on previous analysis is proposed in order to make HModel adapt to the time-varying characteristics of load in computational grid. The results of the experiments demonstrate that our adaptive hybrid model (AHModel) outperforms the widely used autoregressive (AR) model in long term load prediction significantly, and it also achieves obvious reduction in prediction mean square error comparing with HModel which uses fixed parameter value.

1. Introduction

Grid computing [1] is the high-performance and internet-based infrastructure which aggregates geographically distributed and diverse resources to deliver computational power to users in a transparent way, supporting large-scale, resource-intensive and distributed applications. Task scheduling and load balancing are two major mechanisms to improve resource usage efficiency and avoid resource consumption conflicts in such a highly dynamic distributed environment. Obviously, grid load prediction can be used to forecast task run time [18], guide task scheduling and load balancing strategies, thus to achieve high performance and more efficient resource usage [2] [3] [4].

Many previous load prediction works, like [5] [6] [10] [11] [14], use various load prediction models such as mean-based methods, median-based methods, autoregressive (AR) model, polynomial fitting, Markov model-based meta-predictor in addition to seasonal variation , etc., to implement 1-step-ahead or *n*-step-ahead load prediction. In highly distributed computational grid environment, tasks usually take long run time, thus task scheduler needs a comparatively long-time-ahead load prediction or an *n*-step-ahead load prediction with large *n*, which we call it long term load prediction. The further the load prediction of a computational grid can reach, the more appropriate the task scheduler can make scheduling strategies and balance the load.

Because load variation and resource consumption in distributed environment are time-varying and exhibit wide range of dynamics, such as *sudden local change*, *sudden level change* and *gradual level change* [15], adaptive techniques were brought into many previous load prediction works to capture these dynamic characteristics. Normally, two levels of adaptation are used to gain prediction accuracy improvement. One is model adaptation level, which is based on a hypothesis that best predictor varies from resource to resource [5] and the best predictor for any particular resource may change over time [7]. When target resource changes or resource consumption pattern switches, adaptive prediction works on this level select a best prediction model which has minimum prediction errors in last time frame to forecast the future. Works such as [5] [7] belong to this level. The other level is parameter adaptation level, and the works of this level, like [15], implement adaptive methods on parameters. When adaptation is triggered, these methods auto tune the parameters to adapt to the resource consumption variation, in order to achieve lower prediction errors than fixed parameter value models.

In this paper, we first observe and analyze parameters' (such as *HistoryWindowSize*) impact on prediction accuracy in our previous load prediction

978-0-7695-3156-4/08 $25.00 © 2008 IEEE
DOI 10.1109/CCGRID.2008.60

hybrid model (HModel) in detail. It shows that *HistoryWindowSize* has different optimal values in different types of load-varying patterns. Then based on our observation, we propose and evaluate a parameter adaptive hybrid model (AHModel), which can automatically adapt to the change of load-varying patterns and choose optimal parameter value to enhance prediction accuracy. Two kinds of adaptive methods are integrated in AHModel. One is *mean adaptation* which is embedded in HModel naturally; the other one is *HistoryWindowSize* parameter adaptation, which we present and evaluate in detail. The results of the experiments demonstrate that our AHModel outperform significantly the widely used AR model in long term load prediction, and it also achieves obvious reduction in prediction mean square error (MSE) compare with fixed parameter value HModel.

The rest of this paper is organized as follows. Section 2 lists and summarizes the related work. Then HModel is briefly introduced in Section 3. Section 4 analyses the impact of different HModel parameter values on load prediction accuracy, and mainly focus on one key parameter *HistoryWindowSize*. Section 5 presents AHModel in detail. Section 6 describes the experiment results of our AHModel on several real load traces of computational grids. Finally, we conclude our work and describe our future work in section 7.

2. Related Work

Two levels of adaptive methods are widely used in distributed environment's resource consumption prediction areas. Some previous works such as [5][7] did adaptation on the model adaptation level, while other works like[15] implemented adaptive methods in their models on parameter adaptation level.

The Network Weather Service (NWS) [5] provided a dynamical monitoring and forecasting method to implement 1-step-ahead prediction on network and computational resources. NWS uses various prediction methods, such as mean-based methods, median-based methods, and AR methods to forecast the future system state at the same time. NWS tracks the accuracy of all predictors, and selects the one exhibiting the lowest cumulative error measure at any given moment to generate a forecast. In this way, NWS automatically identifies the best forecasting technique for any given resource.

Adaptive predictor integration proposed in [7] adopts an approach for predictor integration based on the learning of historical predictions. Classification algorithms such as *k-Nearest Neighbour* (k-NN) *based supervised learning* are used in its work to predict the optimal predictor for the workload, then only the forecasted best predictor will be selected to do prediction. Evaluation presented in [7] shows that it achieved higher best predictor forecasting accuracy than the cumulative MSE based predictor selection approach used in the popular Network Weather Service system. In addition, it achieves prediction accuracy improvement compare with the observed most accurate single predictor in its predictor pool.

Paper [15] proposed a multi-resource prediction model (MModel) that use auto-correlation of a single resource plus cross correlation between two resources to achieve higher prediction accuracy. It also presented two parameter adaptive techniques that enable MModel to adapt to the time-varying characteristics of the resource consumption. Firstly *mean adaptation* is considered to address the changing mean of the resource, once a new measurement is available, MModel subtracts a new *mean* from the new data and use this new *mean* to feed back to the final prediction; Secondly in order to adapt to the changing cross correlation between two resources in an efficient way, a heuristic *cross correlation adaptation* algorithm was used in [15].

3. Hybrid Model

In previous work [21], we proposed a hybrid model (HModel) which use interval value to predict load variation in computational grid. This model can forecast far more *n*-step-ahead load state than previous models and methods, with comparatively less prediction errors and acceptable prediction interval length for task scheduler and other utilities. In this section, we introduce our HModel briefly and use it to generate point value prediction, in order to make it easier to be compared with other models and methods.

To catch the dynamic nature of the load of computational grid and meet the requirements of task schedulers, HModel integrates AR model with confidence interval estimate, where AR model is used as a basic *n*-step-ahead point value prediction method, and confidence interval which future load state might fall into is estimated based on both historical point values and predicted point values generated by AR model. In order to enhance prediction accuracy, Kalman Filter [22] is used to minimize the measurement errors of historical load values, and Savitzky-Golay filter function [17] is used as a tendency revealing tool for data noise eliminating.

When a new load measurement data is available at time *t*, the main steps of HModel are as follows:

1. Using Kalman filter to minimize the measurement errors of the historical data $X_{measure}(t)$, generate measurement error minimized data $X_{filter}(t)$.

2. Using Savitzky-Golay filter to smooth the data $X_{history}(t)$ with the former data $X_{filter}(t-1), X_{filter}(t-2), \ldots,$ generate the data $X_{smooth}(t)$.

3. Compute the AR(p) coefficients $\varphi_1, \varphi_2, \ldots \varphi_p$ based on the $X_{smooth}(t), X_{smooth}(t-1)\ldots$

4. For the given n-step-ahead parameter N, using the $X_{smooth}(t), X_{smooth}(t-1)\ldots$ and AR(p) coefficients $\varphi_1, \varphi_2, \ldots \varphi_p$ to compute the predict future value of $X_{predict}(t+1), X_{predict}(t+2)\ldots X_{predict}(t+n)$.

5. Compute the confidence interval ($X_{lower}(t+n), X_{upper}(t+n)$) with sample *HistoryWindow+ PredictionWindow*, at confidence level 95%, using the data $X_{history}(t-HistoryWindowSize+1)\ldots, X_{history}(t-2), X_{history}(t-1), X_{history}(t), X_{predict}(t+n-PredictionWindowSize+1)\ldots X_{predict}(t+n-1), X_{predict}(t+n)$.

6. Using Savitzky-Golay filter to smooth the prediction interval value ($X_{lower}(t+n), X_{upper}(t+n)$) with the former data ($X_{lower}(t+n-1), X_{upper}(t+n-1)$), ($X_{lower}(t+n-2), X_{upper}(t+n-2)$),…, generate the smoothed prediction interval value($X_{smoothed\_lower}(t+n), X_{smoothed\_upper}(t+n)$).

Because we have no information about the measurement error of the load traces that we use in our experiments in this paper, we let parameter R in Kalman filter to be 0 in all the experiments of this paper. In this case $X_{filter}(t)$ equals to $X_{measure}(t)$ in step 1.

In order to compare with other prediction models and methods easily and fairly, in this paper we adopt formula (3.1) to generate point value prediction be the final prediction result.

$$X_{prediction-median}(t) = \frac{X_{lower}(t) + X_{upper}(t)}{2} \qquad (3.1)$$

4. Parameter Impact Analysis

The impact of parameter value in HModel has briefly been evaluated in [21]. We noticed that the optimal *HistoryWindowSize* value maybe different along with the varying load, while other parameters, such as p in AR(p) model, cause comparatively much smaller impacts on prediction MSE when use different values than that of *HistoryWindowSize*. Unfortunately, in highly distributed environment like computational grid, load is always fluctuant and its varying pattern switches frequently. If we still use fixed parameter value HModel to predict dynamic load, it is significantly unadvisable and might not achieve good prediction accuracy in the long run, because fixed parameter value can not react to the dynamic load variation.

To observe and analyze the impact that different *HistoryWindowSize* value would bring to HModel on load prediction accuracy, we evaluate HModel on 6 load traces with different *HistoryWindowSize* values. All these load traces are collected from AuverGrid[9]. We classify these traces into 3 groups, inspired by the idea in paper [15], which distinguished the traces by three kinds of load-varying patterns: *sudden local change*, *sudden level change* and *gradual level change*. *Sudden local change* means the load suddenly varies up or down in a wide range and then quickly returns to the previous level; *sudden level change* means the load suddenly changes to another level and stay there for a comparatively long period; *gradual level change* means level of the load gradually changes. But we notice that when load is varying acutely in computational grid, it exhibits *sudden local change* phenomenon singularly, and more generally, when load suddenly rises or drops, it always keeps stable on another level for a period, or quickly changes on the other direction. We call the latter one *sudden fluctuation*. So in this paper we use notion *sudden fluctuation* to replace *sudden local change*. We choose 6 load traces those typically exhibit the characteristics of these three load-varying patterns from AuverGrid. All the traces' load variation types are summarized in Table 1. Figure 1 shows the impact of different *HistoryWindowSize* values on these load traces.

From Figure 1 we can see that optimal *HistoryWindowSize* value changes in different load traces, especially in large n-step-ahead predictions. In small n-step-ahead predictions such as 10-step-ahead, HModel seems to prefer small *HistoryWindowSize* on all kinds of load-varying patterns, though the optimal *HistoryWindowSize* values are different in these traces. In large n-step-ahead predictions where n is bigger than 10, the trend of optimal *HistoryWindowSize* values' varying on different load-varying patterns changes. Moderate values of *HistoryWindowSize* are preferred by HModel when it processes these traces, and what is more noticeable and important is that optimal values of *HistoryWindowSize* on these 6 traces are significantly different. These phenomenons indicate that when load is varying in a computational grid, optimal *HistoryWindowSize* value of HModel would also dynamically change along with grid load variation.

5. Adaptive Hybrid Model

Based on the observation and analysis in the last section, we propose a parameter adaptive hybrid model (AHModel) in this section by implementing adaptive method on HModel. There are two adaptations integrated in AHModel, one is *mean adaptation* which is embedded in AHModel naturally, and the other one

Table 1. Trace types (Sample Interval is 5 minutes, and load is denoted 100%)

Trace Index	Size	Mean	Standard deviation	Load change type	Source
Trace1	1000	68.8262	22.6156	*sudden level change*	AuverGrid
Trace2	1000	69.0613	34.6666	*sudden level change*	AuverGrid
Trace3	1000	46.5152	23.5334	*sudden fluctuation*	AuverGrid
Trace4	1000	46.6994	25.8609	*sudden fluctuation*	AuverGrid
Trace5	1000	70.0289	22.3820	*gradual level change*	AuverGrid
Trace6	1000	70.6977	13.5794	*gradual level change*	AuverGrid

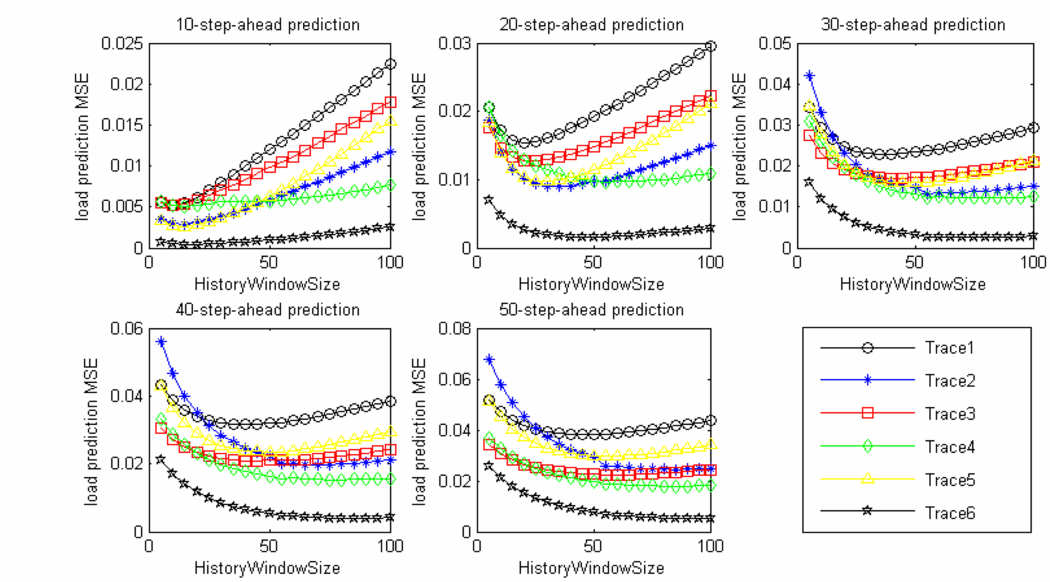

Figure 1. Impact of *HistoryWindowSize* in HModel on load prediction. Default parameter values are $pd = 4$, $span_{\text{Savitzky-Golay}} = 51$, $p = 32$

is parameter adaptation on parameter *HistoryWindowSize*, which auto detects the load-varying patterns change and adjusts the value of *HistoryWindowSize* to enable HModel adapt to the dynamic behavior of computational grid's load.

In HModel presented in Section 3, one implicit *mean adaptation* is considered and embedded in HModel, which would adapt to the changing mean of the load in AR(p) model. Inputs of AR(p) model should be white noise signals, so mean value of historical load is kept in AR(p) model as an internal state. When a new load measurement value X_t is available, we first compute the mean of input historical load value array $\{X_{t-p}, X_{t-(p-1)}, \ldots X_{t-1}, X_t\}$, then subtracts the mean from all the input values and generate a white noise signal sequence as the inputs of AR(p) model. Finally the future load value is noise part ε_t pluses future state predicted based on the input white noise signal. For load with a constant mean, this conversion does not affect the prediction accuracy. But in computational grid environment, where load is varying up and down all over the time and the mean of load changes frequently and significantly, such *mean adaptation* would enable AR(p) model to adapt to the load variation and address the problem caused by the changing mean of load values.

HModel assumes that future load is correlated with historical load, and *HistoryWindowSize* reflects this correlation: large *HistoryWindowSize* means that load is stable for the past long time and therefore future load value would have a much greater possibility of maintaining this stable state; while small *HistoryWindowSize* indicates that load might be varying so quickly and HModel would do prediction on the assumption that future load would keep varying and has little correlation with previous historical load.

As we can see from the load traces of several computational grids such as AvuerGrid[9] and Grid'5000[12], load is always varying and could not keep stable more than a day in most cases. HModel with fixed-value *HistoryWindowSize* can not properly capture this dynamic property of grid's load. Therefore we design and develop a parameter adaptive method to make HModel auto adjusts *HistoryWindowSize*, in order to react to load-varying patterns switching phenomenon and reduce prediction error.

Based on the hypothesis that load is periodically stable and the optimal parameter value does not change during this period, we use the notion of MSE adaptation to form AHModel. By periodically computing the prediction MSE under different *HistoryWindowSize* values, AHModel get the optimal *HistoryWindowSize* value in a past time frame of certain length and use this optimal *HistoryWindowSize* value to forecast future period's load values. The algorithm of AHModel is as follows:

First, in *n*-step-ahead load prediction, we define a notion of adaptive interval *Ia* and set it to be *n*. It means that when HModel receives *Ia* input load measurements $X_{t-(Ia-1)}$,$X_{t-(Ia-2)}$, …, X_{t-1} ,X_t, and predicts *Ia* *n*-step-ahead load values $X_{t-(Ia-1)+n}'$,$X_{t-(Ia-2)+n}'$, …, X_{t-1+n}' ,X_{t+n}' , an adaptation process will be triggered.

Next, since the hypothesis that load-varying pattern is periodically stable and the optimal parameter value does not change during a stable period, we adjust the value of *HistoryWindowSize* and use HModel to process a past period of historical load values $X_{t-(Ia+Sa)}$, $X_{t-(Ia+Sa-1)}$,…X_{t-Ia-1},X_{t-Ia} to do *n*-step-ahead prediction, where we define *Sa* as *adaptation training size*. Then we can verify the prediction accuracy by using load values X_{t-Sa}, $X_{t-(Sa-1)}$,…X_{t-1},X_t and find an optimal *HistoryWindowSize* value *OptimalHWS* of this period. We call this process a parameter "*retraining*" of HModel.

Thirdly, we let *HistoryWindowSize* equals to *OptimalHWS* generated in the last step and continue to forecast future *n*-step-ahead load on next *Ia* input load measurement values X_{t+1} ,X_{t+2} , …,X_{t+Ia}.

Table 2 shows the pseudo-code algorithm of AHModel.

6. Evaluation

As presented in Section 5, there are two adaptations in AHModel: one is naturally embedded in AR(*p*) model which is used in HModel, and the other one implements adaptation by periodically adjusting parameter *HistoryWindowSize* in HModel. Because HModel also contains first adaptation, in this section we mainly evaluate the second adaptive method on load traces in the Grid Workloads Archive[8], which are collected from AuverGrid[9] and Grid'5000[12].

Table 2. Adaptive Hybrid Model Algorithm

Algorithm parameter adaptation
Count = 0, *HistoryWindowSize* = initial value;
While(new measurement data X_t is available)
do
Predict *n*-step-ahead load X_{t+n}' using HModel;
Count++;
if Count%*Ia* == 0 **then**
retrainingLoads = $X_{t-(Ia+Sa)}$, $X_{t-(Ia+Sa-1)}$,…X_{t-Ia-1},X_{t-Ia} ;
verificationLoads = X_{t-Sa}, $X_{t-(Sa-1)}$,…X_{t-1},X_t ;
Using HModel to do prediction on *retrainingLoads* and verify the prediction accuracy using *verificationLoads* under different *HistoryWindowSize* values, and get *OptimalHWS* who can minimize prediction MSE among these *HistoryWindowSize* values;
HistoryWindowSize = *OptimalHWS* ;
end if
end while

6.1. Load Traces Introduction

All load traces used to evaluate our AHModel in this paper come from two famous grid projects, AuverGrid and Grid'5000. The Grid Workloads Archive collected workload traces from these 2 grids and did a primary processing. Based on the works of the Grid Workloads Archive, we compute the total load as the number of CPUs multiplied by the duration of a fixed time interval, 5 minutes, and then we divide the values by the number of total processors of each grid, finally we get the load of these 2 grids as shown in Figure 2, denoted from 0% to 100%.

AuverGrid is a production grid platform consisting of 5 clusters located geographically in the Auvergne region, France and it has 475 CPUs. By trace analysis report [16], from January, 2006 to December, 2006, average CPU load of this Grid is 58.481%, and between April 2006 and November 2006, its system load is more than 80% in most of the time. From Figure 2 we can see that AuverGrid is a heavy loaded grid and in some months like March, April and November, it fluctuates dramatically.

Grid'5000 is an experimental grid platform gathering 9 sites geographically distributed in France. Because we fail to get the accurate number of processors of Grid'5000, we define the total processors of Grid'5000 to be 5000. It load variation from January 2006 to October 2006 is showed in Figure 2, with an average load of 10.683%. During these 10 months, Grid'5000 is light loaded and comparatively fluctuant compare with its average load.

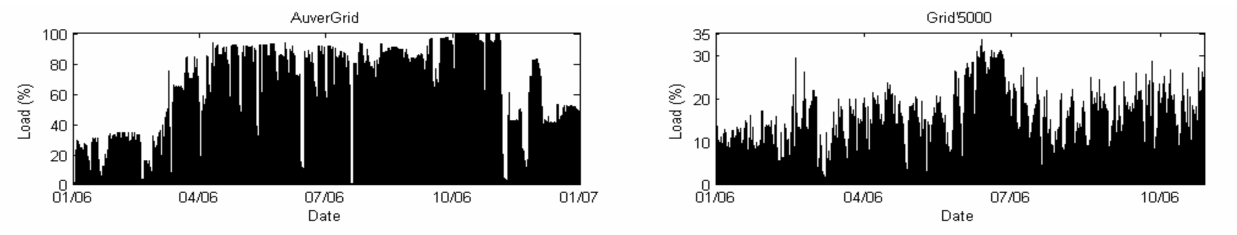

Figure 2. Load variance of AuverGrid and Grid'5000

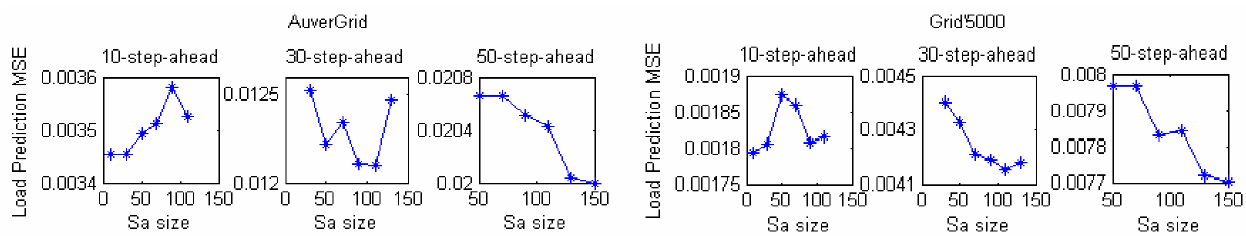

Figure 3. Examples of Optimal *Sa* in AHModel on Load Traces, *pd* = 4, $span_{\text{Savitzky-Golay}}$ = 51, *p* = 32.

6.2. The Optimal *Sa* in AHModel

When AHModel predicts *Ia* future load values, it will use historical load of size *Sa* to "retraining" AHModel and get a new optimal *HistoryWindowSize*. In our experiment, we found not all values of *Sa* can properly adapt to the load changes in computational grid and improve load prediction accuracy of HModel. We analyze the impacts of different *Sa* values on load traces from AuverGrid and Grid'5000, and from observation we find that when *Sa* to be *Nstep+80*, AHModel exhibits best adaptive ability and achieve significant prediction accuracy improvement than other values on most of these two grids' traces. So we use *Sa* to be *Nstep+80* in all the following experiments. Figure 3 gives examples of *Sa*'s impacts on prediction accuracy. Both of these two load traces are random selected from these two grids and sample sizes are both 8000.

6.3. Optimal *HistoryWindowSize* Selection

This set of experiments demonstrates the optimal *HistoryWindowSize* selection of AHModel. Through the experiments we find that *HistoryWindowSize* generally reflects the dependence degree of future load on historical load. If load is unstable and fluctuate widely, optimal *HistoryWindowSize* is always small. This is reasonable because when load is varying acutely, future load is not likely to rely on past historical load values; in other words, there is little possibility for future load to fall into the scope of recent historical load values in this situation. Contrarily, if load is varying slightly or keeping stable, optimal *HistoryWindowSize* value is always larger than that in fluctuant states. Figure 4 shows the optimal *HistoryWindowSize* selection. It is clear that when load is fluctuating widely, optimal *HistoryWindowSize* value would become small; On the contrary, optimal *HistoryWindowSize* value would increase or keep in large value if load is varying slightly.

6.4. Performance Comparison

We evaluate the performance of AHModel on a set of load traces of AuverGrid and Grid'5000 introduced before, and compare it with AR model and HModel. We divide load traces of these 2 grids by months, which are 12 load traces from AuverGrid, from January 2006 to December 2006; and 10 load traces from Grid'5000, from January 2006 to October 2006.

The average prediction MSE of our AHModel and AR model on this set of load traces are shown in Figure 5(a). Obviously, AHModel outperforms the "best" AR model (we use p = 4,8,12…,64 to do prediction on each load trace and choose the AR(p) model with lowest prediction MSE to be the "best" AR model of this load trace)on both AuverGrid and Grid'5000 load traces. The prediction MSE of AHModel only slowly increase with n, and for 50-step-ahead prediction (which is 250 minutes into the future), the prediction MSE of AHModel on load traces of AuverGrid and Grid'5000 are both below 0.04. On average, prediction MSE of AHModel is 93.10% and 91.35% less than that of "best" AR model in n-step-ahead load prediction on the traces of

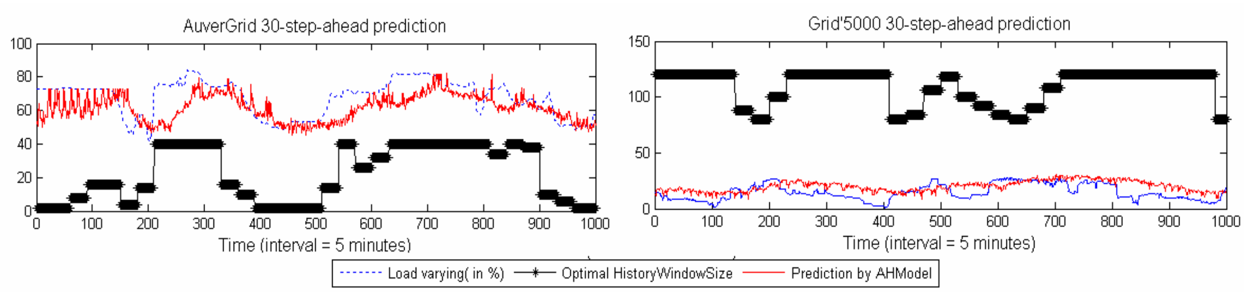

Figure 4. **Example of Optimal *HistoryWindowSize* selection on AuverGrid and Grid'5000 load traces, pd = 4, $span_{\text{Savitzky-Golay}}$ = 51, p = 32, Sa = 110**

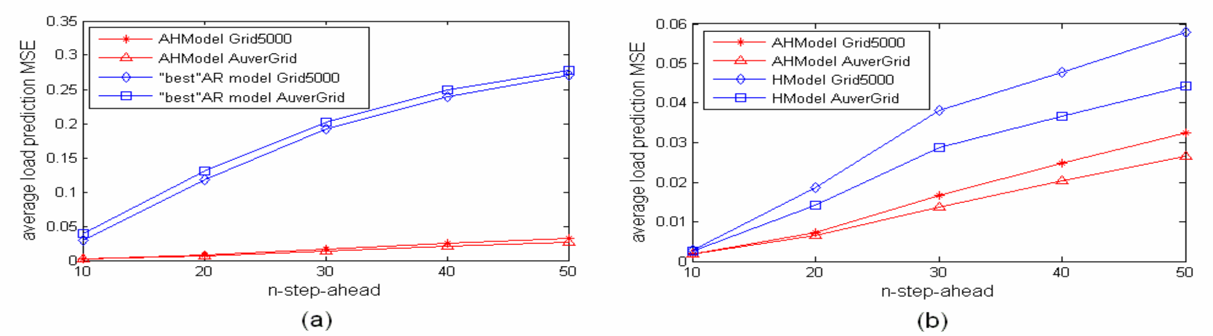

Figure 5. **Average Prediction MSE of AHModel (pd = 4, $span_{\text{Savitzky-Golay}}$ = 51, p = 32, Sa = n+80), HModel (pd = 4, $span_{\text{Savitzky-Golay}}$ = 51, p = 32) and the "best" AR model**

AuverGrid and Grid'5000 respectively.

Figure 5(b) demonstrates the power of our adaptive method mentioned in section 5 which is integrated with HModel to form AHModel. We compare load prediction MSE of AHModel with that of HModel. For every load trace, we use first 1000 sample load values to training the *HistoryWindowSize* for HModel, and use this *HistoryWindowSize* value as initial input of our AHModel. Then we do prediction for the rest of load trace's data. It is clear that AHModel can reduce long term load prediction MSE effectively, compare with HModel who use fixed parameter value. The prediction MSE of AHModel is on average 43.44% less than that of HModel on AuverGrid's load traces and 49.76% less on load traces of Grid'5000.

7. Conclusion and Future Work

In this paper, the impact of parameters in HModle on long term load prediction accuracy is analyzed in detail first. Furthermore, a parameter adaptive prediction model is proposed in order to enable HModel to adapt to the time-varying characteristics of load in computational grid. From the observation of experiments, we notice that when load is varying from time to time, optimal *HistoryWindowSize* value in HModel is also changing. This phenomenon indicates that implementing adaptation on parameter *HistoryWindowSize* might make HModel adapt to the varying load in computational grid more properly. So we design and develop a parameter adaptation AHModel. The results of the experiments on two real grid load traces demonstrate that the prediction MSE of AHModel is significantly less than that of widely used AR model, and AHModel also achieves markedly less prediction MSE than that of HModel for n-step-ahead prediction where n can be as large as 50.

The prediction accuracy improvement by AHModel on long term load prediction in computational grid indicates that our adaptive method captures grid load-varying patterns switching in certain degree. In this paper we use MSE adaptation; meanwhile we noticed that best *HistoryWindowSize* follows some rules along with load variation. So in future we would like to study the rules of best *HistoryWindowSize* value variation along with load variation. AHModel works well in minimizing prediction MSE, but for load peak prediction, which is the hardest in load prediction area especially long term load prediction, we still have to do more work to enhance load peak prediction accuracy.

Acknowledgment

This Work is supported by ChinaGrid project of Ministry of Education of China, Natural Science Foundation of China (90412006, 90412011, 60573110, 90612016, 60673152), National Key Basic Research Project of China (2004CB318000, 2003CB317007), National High Technology Development Program of China (2006AA01A108 ， 2006AA01A111 ， 2006AA01A101), and IST programme of the European Commission (DG Information Society and Media, project n° 034442). We acknowledge Dr. Alexandru Iosup of TU Delft for his help in the Grid Workloads Archive and grid load generation. We are grateful to Dr. Emmanuel Medernach and the AuverGrid team for their support by supplying the AuverGrid traces, and also thank Dr. Franck Cappello, Dr. Olivier Richard and their Grid'5000 team for the Grid'5000 traces.

References

[1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[2] C. Liu, L. Yang, I. Foster, and D. Angulo, "Design and Evaluation of a Resource Selection Framework for Grid Applications", Proceedings of the 11th IEEE International Symposium on High-Performance Distributed Computing (HPDC 2002), Edinburgh, Scotland, 2002.

[3] D. Lu, H. Sheng, and P. Dinda, "Size-based scheduling policies with inaccurate scheduling information", 12th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS 2004), pp. 31-38, 2004.

[4] S. Jang, X. Wu, and V. Taylor, "Using performance prediction to allocate grid resources", Technical report, GriPhyN 2004-25, pp. 1-11, 2004.

[5] R. Wolski, "Dynamically Forecasting Network Performance Using the Network Weather Service", *Journal of Cluster Computing 1*, pp.119–132, January 1998.

[6] P. A. Dinda and D. R. O'Hallaron, "Host load prediction using linear models", *Journal of Cluster Computing 3(4)*, 2000.

[7] J. Zhang, R.J. Figueiredo, "Adaptive Predictor Integration for System Performance Prediction", 2007 IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007), March. 2007

[8] The Grid Workloads Archive. [Online]. Available: http://gwa.ewi.tudelft.nl/

[9] AuverGrid. [Online]. Available: http://www.auvergrid.fr/

[10] R. Wolski, N. Spring, and J. Hayes, "Predicting the CPU availability of Time-shared Unix Systems", Proceedings of 8th IEEE High Performance Distributed Computing Conference (HPDC 1999), 1999.

[11] J. Schopf and F. Berman, "Performance prediction in production environments", 12th International Parallel Processing Symposium, Orlando, FL (April 1998) pp. 647-653.

[12] Grid'5000. [Online].Available:https://www.grid5000.fr/

[13] Marple, S. Lawrence and Jr, "Digital Spectral Analysis with Applications. Englewood Cliffs", *Journal of the Acoustical Society of America 86*, 5, November 1989

[14] 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 (CCGrid 2004), pp. 765-772, 2004.

[15] J. Liang, K. Nahrstedt, and Y. Zhou, "Adaptive Multi-Resource Prediction in Distributed Resource Sharing Environment," 2004 IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2004), pp. 1-8, 2004.

[16] AuverGrid Workload Analysis Report. [Online]. Available: http://gwa.ewi.tudelft.nl/pmwiki/reports/gwa-t-4/trace_analysis_report.html

[17] Orfanidis, S.J., *Introduction to Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[18] Y. Zhang, W. Sun, and Y. Inoguchi, "Predicting Running Time of Grid Tasks based on CPU Load Predictions", 7th IEEE/ACM International Conference on Grid Computing(Grid 2006) ,Sept. 2006

[19] (2006) The ChinaGrid Super Vision website. [Online]. Available: http://www.chinagrid.edu.cn/CGSV/.

[20] (2006) The ChinaGrid website. [Online]. Available: http://chinagrid.hust.edu.cn or http://www.chinagrid.edu.cn.

[21] Y. Wu, Y. Yuan, G. Yang and W. Zheng, "Load prediction using hybrid model for computational grid", 2007 8th IEEE/ACM International Conference on Grid Computing (Grid 2007), pp. 235 – 242.

[22] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Transaction of the ASME—Journal of Basic Engineering*, pp. 35-45, March 1960.