Effective Prediction of Job Times in a Large-Scale Grid Environment

Menno Dobber
Vrije Universiteit,
De Boelelaan 1081a,
1081HV Amsterdam,
The Netherlands
E-mail: amdobber@few.vu.nl

Rob van der Mei
CWI and Vrije Universiteit,
c/o Kruislaan 413,
1098SJ Amsterdam,
The Netherlands
E-mail: mei@few.vu.nl

Ger Koole
Vrije Universiteit,
De Boelelaan 1081a,
1081HV Amsterdam,
The Netherlands
E-mail: koole@few.vu.nl

Abstract

For years, grid environments expand globally in number and in size. Globally collaborating processors, that share their available capacities, construct a tremendous source of processing power. This source has proven its necessity for performing extensive computations, because these calculations require unacceptable amounts of time on a small number of processors. Key properties of grid environments are that the wallclock times of the jobs are highly bursty, mainly because of the changing load, and that the set of resources dynamicly changes over time. Those properties show the importancy of developing techniques that make grid applications robust against the dynamics of the grid environments. In particular, grid applications that use a considerable number of processors for their computations need effective predictions of the expected computation times on the different nodes. Currently, there are no effective prediction methods available that cope with those ever-changing dynamics of computation times in a grid environment. Motivated by this, in this paper we develop the Dynamic Exponential Smoothing (DES) method to predict job times in a grid environment. In order to compare predictions of DES to that of the existing prediction methods, extensive experiments in a real grid enviroment, Planetlab [1], have been performed. The results illustrate a strong and consistent improvement of DES in comparison with the existing prediction methods.

1. Introduction

For decades, grid environments have shown to be a very useful and inexpensive alternative to dedicated clusters of processors. The processors of grids are connected at a global scale via the Internet, and grid applications effectively share the processing resources, because reservation systems are not implemented in grids. Grid environments are highly unpredictable due to the following main properties: the job times of jobs on the shared processors change all the time, the processor capacities and link rates are often unknown, and the set of available processors changes continuously over time. Those properties may substantially increase the job times of grid applications. This raises the need for methods that make applications robust against those fluctuations of the grids. The quality of those techniques strongly depends on the effectiveness of the prediction methods: as demonstrated in an experimental setting by [2, 3, 6, 11], scheduling methods based on effective prediction methods can realize considerable speedups.
In this paper, we focus on the development of a prediction method for running times of jobs on shared processors. To this end, we analyze several existing methods that are potentially capable of predicting job times, and investigate their advantages and disadvantages. The results of previous work on the characteristics of job running times on grid nodes [8] provide a strong basis for those analyses. Next, we develop a new prediction method, called Dynamic Exponential Smoothing (DES), that takes into account all the observations of the analyses. Extensive experimental results demonstrate that predictions from the DES strongly outperforms the existing methods.

In [8] an extensive set of statistical properties of job times has been investigated. The results of the statistical data analysis demonstrate that: (1) the characteristics of the datasets very often differ completely, (2) the datasets show on average more long- as short-term fluctuations and the proportion differs per dataset, (3) the average job-time fluctuate continuously during the run, with differences in the amount of fluctuations between the different nodes, (4) the standard deviations fluctuate considerably during the run, with differences in the amount of fluctuations between the different nodes, and (5) the datasets contain a small amount of peaks that have a substantial influence on the standard deviation, the total amount of fluctuations, and the variance. This has raised the need for the development of effective

prediction methods of job times that are able to deal with those characteristics. To this end, these observations form the basis for both an investigation of the applicability of existing predictors for predicting job times, and, the development of a new prediction method.

In the literature, much research focusses on the development of prediction methods. Those methods can be broadly classified into two main types: linear and non-linear predictors. For the grids, various linear models have been proven effective in predicting grid properties: Autoregressive models (AR) for load [5] and network traffic [14], Exponential Smoothing (ES) for job times [7], a tendency-based predictor also for load [20], and Linear Regression (LR) for total running times of parallel applications [10]. Furthermore, the Network Weather Service (NWS) prediction algorithm that selects between different linear predictors (e.g. [19]) showed good performance for different grid predictions. Further, ES-based predictors (e.g. [15]) give very accurate predictions in other areas (e.g. economical, and weather-forecasting areas) and are potentially applicable in grids. Moreover, the applicability in grids of non-linear prediction models (e.g. Neural Networks) has not yet been investigated.

The main requirement of grid applications is the necessity of robustness against the dynamically changing circumstances. An example of creating robustness is developing dynamic load balancing schemes to control the fluctuations of the job times on the several nodes [6]. A compulsory requirement of efficient dynamic load balancing is that the predictor is fast and simple. Moreover, due to the restricted monitoring possibilities in a real grid, the required number of types of grid-property measurements (e.g. load, job times, latencies) for the prediction methods has to be small. In order to predict the job times on the basis of the measured load, Dinda et al. [4, 5] investigate the relation between processor load and job times. The results of Dinda show that the job times and the load show a significant negative correlation, but that, however, predicting job times by the load is infeasible due to the impact of many other factors (e.g. memory space) on the job times, and those are hard, or even technically impossible, to gather in practice. The prediction of job times is even further complicated by the fact that in a real grid environment even the load (i.e., CPU utilization) can often not be measured at all. In order to avoid this difficulty, we focus on methods to predict future job times *only* on the basis of the *past job times*, not requiring any additional - and possibly unavailable - measurement data.

In this paper, we develop a new prediction method for the running times of successive jobs on nodes in a grid environment. To this end, in Section 2 we combine the results of [8] (outlined above), and a theoretical analysis, to identify the strong and weak points for an extensive set of predictors (e.g., NWS, Autoregression, Adaptive Exponential Smoothing, STES predictors). The results show that the main reasons for inaccuracy of these existing methods are (1) delayed reaction to "level switches", and (2) overreaction to sudden peaks in the job times. In Section 3, these findings form the basis for the development of the prediction method Dynamic Exponential Smoothing (DES). The main idea behind DES is that it uses ES with a interpolating factor α that switches between multiple dynamically adapting values. Subsequently, in Section 4 the accuracy of the predictions based on DES are compared to that of the other prediction methods. For the comparison we use 45 datasets of 18 different nodes of Planetlab [1], a global-scale grid testbed environment. The results show that DES strongly outperforms the existing methods in the vast majority of the datasets. Finally, in Section 5 we address a number of topics for further research.

2. Analysis of Existing Prediction Methods

In this section, we analyze a variety of existing methods that can be used to predict the running times of jobs on shared processors. We analyze the methods in a theoretical and practical way, which is based on the 40 generated datasets and results from [8]. The data-collection details of those datasets are described in 4.2. In this section, the names of the datasets all start with 'pre_', followed by the abbreviation of the node location (e.g. 'au' indicates the node in Australia, and 'warsch' the node in Warsaw), and end with the number of the dataset.

The most commonly used predictors in grid studies are ES, NWS and AR (these predictors will be described below). Nevertheless, for predicting job times there are many other useful predictors applicable from other areas (e.g. the economics area). We selected the Adaptive Exponential Smoothing Predictors (AESP) and Smooth Transition Exponential Smoothing (STES) predictors as potential methods that can give accurate predictions in grids. From the AESP we discuss Trigg and Leach, Whybark, Mantzer, Pantazopoulos and Pappis, and from the STES predictors we discuss STES $|e|$ with $\gamma < 0$, STES $|e|$, STES e^2, and STES Whybark, which we selected from [15]. Below, we describe the different prediction methods and assess their strong and weak points for making forecasts based on the characteristics of our datasets discussed in the previous section.

2.1. Common grid predictors

In this subsection, we consider the most commonly used predictors in grid environments: Exponential Smoothing (ES), Network Weather Service (NWS) and Autoregression (AR).

2.1.1 Exponential Smoothing

ES is a simple prediction method that suprisingly often works very good in practice. We define y_t as the measured value at time t, $\hat{y}_t$ as the prediction for y_t, α as a chosen parameter between 0 and 1. Next, the prediction for y_t is defined as:

$$\hat{y}_t = \alpha y_{t-1} + (1-\alpha)\hat{y}_{t-1}, \text{ where } 0 \leq \alpha \leq 1. \quad (1)$$

A strong point of ES is that on the one hand, it does not react completely on peaks and on the other hand it reacts fast enough to level switches, which are observed frequently in the datasets. The most suitable value of the interpolation parameter α depending on the characteristics of the dataset. A weak point is that once a parameter is chosen, it always reacts in the same way to peaks and level switches, even when the structure is are changed completely, which were found to occur frequently in [8]. For example, Figure 1 shows that ES with parameter 0.5 does not react properly to a peak, even when there were many peaks in the history data. In this paper we use the ES parameter 0.5, because in [6] is stated that 0.5 is the value that leads to the most accurate predictions for job times.

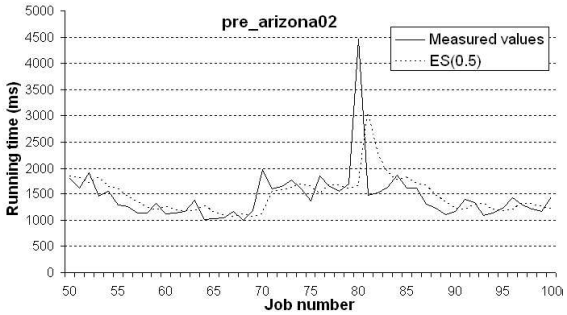
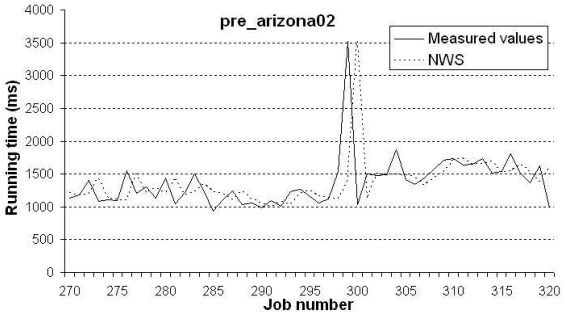


Figure 1. ES reacting on a peak on pre_arizona02

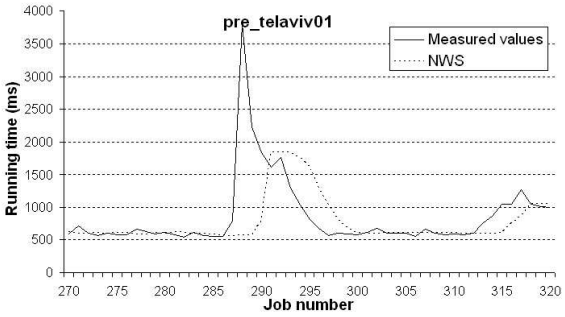
2.1.2 The Network Wheather Service

The Network Wheather Service (NWS) prediction method conducts postcasts using different windows of previous data (always starting with the most recent data and working backwards in time) and records the "winning" forecaster for each window size. Each window size of previous history is subsequently treated as a separate forecaster and a final accuracy tournament determines which forecaster will be used. The predictor selects from a set of the following types of predictors: median-based, mean-based, and exponential-smoothing based predictors. For brevity, further details can be viewed in [19].

A strong point of this method is that it contains a wide set of predictors. The broad set of predictors represents many



(a) **Peak on pre_arizona02**



(b) **Level switch on pre_telaviv01**

Figure 2. NWS-predictor reacting on a peak and a level swith in the job times

different charateristics of the datasets, and, therefore, the NWS prediction method is able to deliver accurate predictions in many situations. Further investigations with the 40 analysis-phase datasets show that some parts of the datasets contain more than 20% peaks, or show an alternating characteristic. For those cases a predictor based on the average is the most accurate. Some parts of the datasets show a homeostatic character, for which a median-based predictor gives the best predictions. Both types of predictors are represented in the NWS set. Another good point of this selection method is the case of changing charateristics, this predictor rapidly chooses another predictor that predicts the new situation more accurately. A weak point is that it does not always react properly to peaks. Figure 2(a) shows that when there were small level switches in history, the NWS method often chooses a predictor that overreacts to peaks. That is even the case when peaks in history never introduced big changes. Moreover, Figure 2(b) shows that the NWS prediction method reacts too slow on the new level, despite many peaks introduced a new level for previous values. Clearly, peaks in datasets have to be interpreted differently than regular values. However, the NWS does not make distinction between those types of values.

2.1.3 Autoregression

Autoregressive (AR) predictors multiply previous datapoints with some parameter between 0 and 1 to compute the next prediction. AR models have a parameter p which indicates the number of datapoints it uses from the history. Generally, AR predictions (denoted as $AR(p)$) are calculated in the following way:

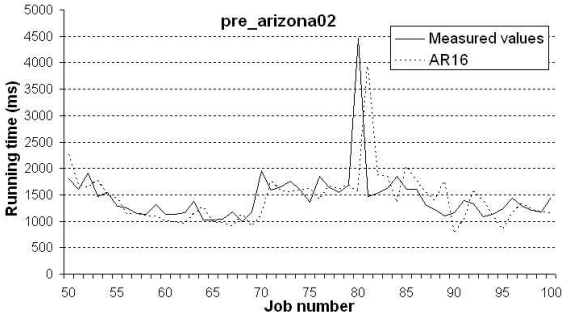
$$\hat{y}_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + .. + \alpha_p y_{t-p}, \quad (2)$$

with

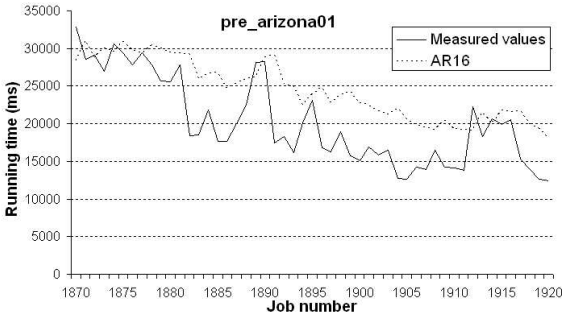
$$\sum_{i=1}^{p} \alpha_i = 1 \text{ and } 0 \leq \alpha_i \leq 1. \quad (3)$$

When the expectation of the y_t's does not equal 0, the mean of the fitted values is substracted from the y_t's and finally added to the $\hat{y}_t$. Dinda [5] has also analyzed a set of other prediction models, related to AR models, and showed that $AR(16)$ performs the best in forecasting processor loads and job times. However, as stated in [5], they only work well in case of periodicity. As can be seen in [8], the datasets do not show periodicity. Dinda used k-steps-ahead prediction methods and implemented the Yule-Walker technique. However, we only use the software part of Dinda that implements the one-step ahead prediction methods to be able to compare the predictions with the other one-step-ahead predictions and because analyses showed that those predictions are significantly more accurate.

A strong point is that AR methods adapt the parameters in a way that it efficiently reacts to periodicity. In case of many peaks, this method will adapt the parameters in a more 'averaging' way, such that the peaks do not influence the predictions too much. When more level switches occur, this method chooses for higher values for the first α parameters, such that it reacts fast on those switches. However, from [8] we know that the datasets do not clearly show periodicity. Therefore, the aspect of adapting to periodicity of AR methods is not an advantage for our datasets. Another weak point is that the choice of parameters is not optimal. For example, when the AR method takes a high first parameter because the set shows some small level switches, it will give a highly inaccurate prediction when a higher peak occurs (see Figure 3(a)). Another drawback of this method is that when the structure of datapoints is changed it takes a long time before the method is adapted due to the fact that this method uses more than 100 datapoints to fit the parameters. Also in case of a trend upwards or downwards (see fro example Figure 3(b)) this method adapts the predictions too slowly, because a significant part of the prediction is based on the average of the whole dataset.



(a) **Peak on pre_arizona02**



(b) **Trend down on pre_arizona01**

Figure 3. AR16-predictor reacting on a peak and a trend down in the job times

2.2. Adaptive Exponential Smoothing Predictors

Adaptive Exponential Smoothing Predictors (AESP) [15] use the following formula:

$$\hat{y}_t = \alpha_{t-1} y_{t-1} + (1 - \alpha_{t-1}) \hat{y}_{t-1}. \quad (4)$$

and focus on adapting α_t such that the α_t will always get a good value that is independent of the start value, and adapts when the structure of the values changes. We consider the following variants of AESPs: Trigg and Leach [16], Whybark [17], Mentzer [12], Pantazopoulos and Pappis [13], and the STES predictors [15].

2.2.1 Trigg and Leach

The method of Trigg and Leach [16] defines the smoothing parameter, α_t of (4), as the absolute value of the ratio of the smoothed forecast error to the smoothed absolute error:

$$\alpha_t := |\frac{A_t}{M_t}|, \quad (5)$$

$$A_t := \phi e_t + (1 - \phi) A_{t-1}, \quad (6)$$

$$M_t := \phi |e_t| + (1 - \phi) M_{t-1}, \quad (7)$$

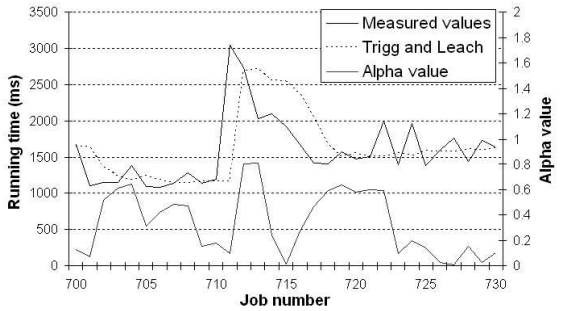


Figure 4. A trend up and down on pre_arizona02

where e_t is the forecast error at period t ($e_t = y_t - \hat{y}_t$) and ϕ is set arbitrarily, with 0.2 being a common choice [16]. Trigg and Leach explain that this formulation enables α_t to vary according to the degree to which biased forecasts are obtained.

A main advantage of this prediction method is that it chooses a low α parameter when there is a lot of noise in the dataset and a high α when there is less noise and there are some level-switches. These properties improve the accuracy of the predictor. A disadvantage is that the proposed method is that it is difficult to find a suitable choice of the parameter ϕ and that the approach sometimes delivers unstable forecasts [9]. Another weak point is that when a few successive datapoints show a trend up (in Figure 4 from value 710 to 711) and then a trend down (see Figure 4 from value 711 to 717) in the α value will first increase to close to 1, which increases the accuracy, and then decreases to 0 and finally increases back to 1. Consequently, as is shown by the Trigg and Leach prediction values 713 to 718 in Figure 4, the predictor does not give very accurate predictions. In this case, it would be better when the α-value was kept fixed to 1. Similar problems occur when there are many level switches up- and downwards.

2.2.2 Whybark

AESP Whybark [17] defines the control limits in terms of multiples of the forecast error standard deviation, σ. An indicator variable, δ_t, is defined as:

$$\delta_t := \begin{cases} 1 & \text{when } |e_t| > 4\sigma, \\ 1 & \text{when } |e_t| > 1.2\sigma, |e_{t-1}| > 1.2\sigma, \\ & \text{and } e_t e_{t-1} > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The value of δ_t determines whether α_t takes a base value, B, a medium value, M, or a high value, H. Whybark suggests $B = 0.2, M = 0.4$ and $H = 0.8$. Altogether, the α_t in (4) of Whybark is computed in the following way:

$$\alpha_t := \begin{cases} H & \text{when } \delta_t = 1, \\ M & \text{when } \delta_t = 0 \text{ and } \delta_{t-1} = 1, \\ B & \text{otherwise.} \end{cases} \quad (9)$$

A strong point of Whybark is that it distinguishes between normal situations and those where there was a huge difference between the prediction and the measured value; characteristics of predictions are always different when big differences occur. However, further analysis with the 40 datasets shows that the Whybark predictor gets a higher accuracy when it would distinguish between (1) 'normal' fluctuations, (2) fluctuations that are higher than 2 times the measured standard deviation, and (3) fluctuations that are higher than 10 times the measured standard deviation. A second strong point is that Whybarks' predictor contains a *correction part*; that is, when two successive measurements both deviate in the same direction with more than 1.2 times the standard deviation from the predictions, another ES parameter has to be applied. Further analysis with our analysis-phase datasets shows that this correction part increases the accuracy of the predictor, but that a value of 1.0 makes better distinctions. A weak point is that does not distinguish between up- and downward fluctuations; it is not likely that those situations can be interpreted the same. Moreover, Whybark reacts in the same way to huge peaks as on two successive small differences between the prediction and the measured value. Subsequently, Whybark always uses the same parameters for the same kind of peaks; 0.8 for high peaks, 0.4 for the value after a high peak and also for two successive differences, and 0.2 as a base value. In practice, however, characteristics for each dataset are different. Figure 5 shows that the Whybark predictor does not react properly to peaks. Whybark will only work well when peaks introduce a new 'level' of values. It would be smarter when Whybarks predictor learns from historical data whether a huge difference introduces a new level or is a peak. Finally, we conclude that the Whybark predictor contains a useful fluctuations classification, but that following improvements can be made: (1) distinction between the above described 3 classes of fluctuations, (2) a correction part parameter of 1.0, (3) different treatment of down- and up-wards fluctuations, (4) different treatment of a datapoint after a huge peak and two successive significant, but not enormous, differences, and (5) adaptive parameters.

2.2.3 Mentzer

AESP Mentzer [12] uses the absolute forecast error fraction from the most recent period as α_t. In order to restrict α_t to the interval [0, 1], if the absolute error fraction exceeds 1.0, then α_t is set to 1:

$$\alpha_t := \min\left(\left|\frac{y_t - \hat{y}_t}{y_t}\right|, 1\right). \quad (10)$$

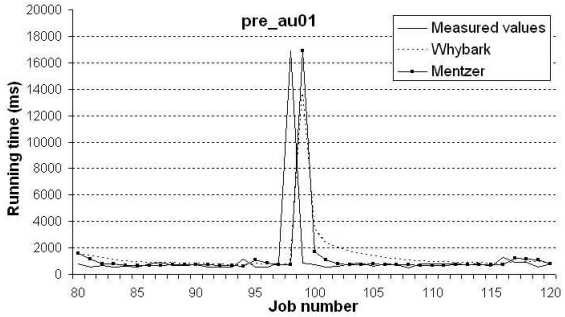


Figure 5. Whybark and Mentzer predictors reacting on a peak in pre_au01

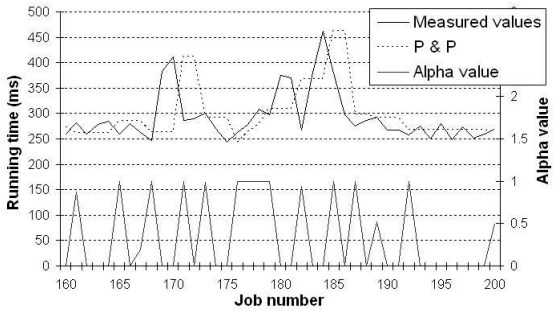


Figure 6. P&P-predictor in pre_warsch01

A strong point of this method is that when the forecasting error increases (decreases), the α value also increases (decreases) to improve the forecast. A weak point of this method is the reaction on peaks: when a peak occurs, the predictor will have a large forecasting error, which leads to a high next prediction, which in turn causes another large prediction error. This is illustrated in Figure 5. Another weak point is that it is not clear why there should be a 1- 1 relation between the α-value and the last forecasting error. For example, a forecasting error of 50% is a worse prediction and probably needs an α value that is higher than 0.5. The height of the α value also depends on the height of the standard deviation of the value.

2.2.4 Pantazopoulos and Pappis

Pantazopoulos and Pappis [13] argue that, since the ideal value for α_t would lead to $\hat{y}_{t+1} = y_{t+1}$, this ideal value can be derived by substituting y_{t+1} for $\hat{y}_{t+1}$ in (4) and solving for α_t to give

$$\alpha_t := \left(\frac{y_{t+1} - \hat{y}_t}{y_t - \hat{y}_t} \right). \tag{11}$$

Since y_{t+1} is unknown at time t, the ideal value of at for period $t-1$ is used for period t to give

$$\alpha_t := \left(\frac{y_t - \hat{y}_{t-1}}{y_{t-1} - \hat{y}_{t-1}} \right), \tag{12}$$

which can be substituted into the standard ES-Formula (4). In order to restrict α_t to the interval [0, 1], Pantazopoulos and Pappis propose that if $\alpha_t \notin (0, 1)$ then α_t is set to either 0 or 1, whichever one is closer.

A strong point of this predictor is that it calculates the optimal value of the α_t's for the last predictor-value combination. With high probability, that optimal choice of α_t is also a very good choice for the next value. Unfortunately, brief consideration of (12) suggests that the approach is unlikely to be of use. Since the one-step-ahead forecast is also the multi-step-ahead forecast for simple exponential smoothing, the numerator is a two-step-ahead forecast error, while the denominator is a one-step-ahead forecast error. This suggests that the expression in (12) will very often lead to values larger than 1. Following the rule of Pantazopoulos and Pappis (P&P), α_t would then take a value of 1. The result is that α_t very often takes a value of 1, which is shown by Figure 6. Moreover, this figure shows that the interpolation parameter α of the P&P predictor is quite unstable, and in many cases results in inaccurate predictions, see for example datapoints 171, 172, 185 and 186.

2.3. STES predictors

A smooth transition adaptive exponential smoothing (STES) method [15] has a smoothing parameter α_t defined as a logistic function of a user-specified transition variable, V_t, and is written as:

$$\hat{y}_t = \alpha_{t-1} y_{t-1} + (1 - \alpha_{t-1}) \hat{y}_{t-1}, \tag{13}$$

where

$$\alpha_t := \frac{1}{1 + e^{\beta + \gamma V_t}}. \tag{14}$$

If $\gamma < 0$, then α_t is a monotonically increasing function of V_t. Hence, as V_t increases, the weight on y_t increases, and consequently, the weight on $\hat{y}_t$ decreases. The logistic function restricts α_t to lie between in the interval (0,1). Historical data is used to calibrate the adaptive smoothing parameter, α_t, through the estimation of β and γ in (14). The derived values for β and γ in (14), govern the degree to which the variation in the transition variable influences the STES smoothing parameter. The choice of the transition variable, V_t, is of crucial importance to the success of the method. Consideration of the adaptive methods described in the previous section leads to a number of different possible transition variables. The value of the smoothing parameter in all of the existing adaptive methods depends to varying degrees on the magnitude of the most recent periods forecast error. At the end of this subsection we describe the different methods we used: STES $|e|$, STES e^2, and STES Whybark.

In [15], 80 data points are used to fit the parameters, and those parameters are used to make one-step-ahead predictions of the next 20 values. To make a fair comparison between the STES- and the other predictors, we fitted the β and the γ after every new measured value. Moreover, for the STES predictors, we compared 10, 80, and "all values" as the number of data points used for the fit.

Next, we describe the three specific STES prediction methods we used in the predictor analysis.

STES |e|

An obvious choice for the transition variable is the absolute value of the forecast error from the most recent period:

$$V_t := |y_t - \hat{y}_t|. \tag{15}$$

In our analyses in Section 4 we make distinction between the situation in which we let the predictor fit β and γ freely, and in which we restrict the γ to be lower than 0. The purpose of restricting the γ to negative values is to get a high α value when the absolute error is high.

STES e^2

Another obvious choice for the transition variable is the square of the forecast error from the most recent period:

$$V_t := (y_t - \hat{y}_t)^2. \tag{16}$$

STES Whybark

It is possible to use the α_t outcome values of all the AESPs. We choose to use the Whybark parameter as the transition variable, because short analysis showed that the Whybark parameter was the best transition variable to choose:

$$V_t := \begin{cases} H & \text{when } \delta_t = 1, \\ M & \text{when } \delta_t = 0 \text{ and } \delta_{t-1} = 1, \\ B & \text{otherwise,} \end{cases} \tag{17}$$

where

$$\delta_t := \begin{cases} 1 & \text{when } |e_t| > 4\sigma, \\ 1 & \text{when } |e_t| > 1.2\sigma, |e_{t-1}| > 1.2\sigma, \\ & \text{and } e_t e_{t-1} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

A strong point of a STES predictors discussed above is that it does not assume a linear relation between the transition variable and the value α_t. With the logistic function and the freedom of the parameters β and γ many different relations can be created. Another strong point is that the relation between the transition variable and the α_t's is fitted with two parameters, based on the history. Therefore, the α_t is adapted in a way that it is optimal according to the history. Moreover, when the transition variable has no correlation with the right α_t, the method automatically adapts the γ to 0 and the smoothing parameter will be constant. The STES method, therefore, enables recalibration of the existing adaptive methods.

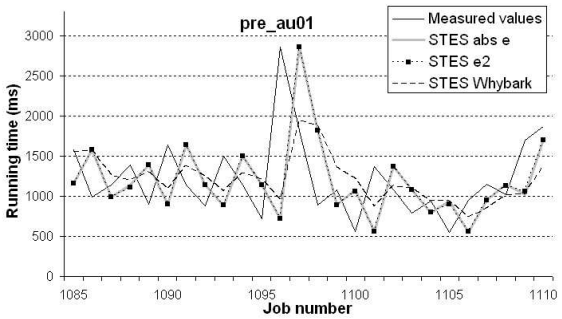


Figure 7. STES predictors reacting on a peak in pre_au01

A weak point, however, is that it is not clear why a logistic function would work better than other functions. Although previous experimental results have shown that logistic functions work quite well, it is the question whether it is an optimal function for the values of job times. Further on, this method uses the same formula for peaks as for stable situations. Figure 7 shows the predictions around a peak of three different STES predictors, which are described below. The figure illustrates that using the same parameters for peaks as for stable situations leads to bad predictions. Besides, the peaks have a high impact on the optimal parameters β and γ. Consequently, the α will behave in a stable situation highly influenced by previous peaks.

2.4. Other predictors

As stated in [20] a homeostatic or a tendency-based predictor can be very effective in predicting CPU load. A homeostatic predictor uses the assumption that if the current value is greater (less) than the mean of the history values, then the next value is likely to decrease (increase). A tendency-based predictor uses the assumption that when a current value is greater (less) than the mean of the history values, then the next value is likely to increase (decrease), because of a trend up- or downwards. A disadvantage is that before the prediction process is started, a choice between homeostatic or tendency has to be made. We know from the data analysis that it is possible that the characteristics in the dataset change, and that during the run the other type of predictor seemed to be more accurate. During our data analysis we noticed that trends do not appear very often in the datasets, there are many level switches, and sometimes homeostatic situations appear. For that reason we think that tendency-based predictors are not very accurate in predict-

ing the job times. A homeostatic predictor will probably work fine for some datasets, but for the nodes with many level switches this method is very unlikely to be effective in the grid context. For these reasons, we have not implemented this prediction method.

Another well-known method is Linear Regression (LR). This method is quite similar to AR, when only historical data is used: the predictor is computed by multiplying historical data with parameters, which are fitted by using a least-squares method and the history. When not only data from the previous history is used, but also other information, like load, this method differs from AR. The parameter used to multiply with other data will also be fitted by its historical data. Since we focus on prediction methods based on the use historical data of job times only (see Section 1), LR is beyond the scope of this study.

Besides the linear prediction methods, experiments with a Neural Network, which is an example of a non-linear predictor, have been performed. The results show that the predictor is not accurate enough in predicting grid properties, and that is it computationally infeasible.

2.5. Conclusions of analyses

We conclude that although each prediction method has its strong points, none of the predictors properly reacts to peaks and level switches, which are two of the most important characteristics of the evolution of job times in a grid environment. This raises the need for the development of a new prediction method particularly suited for the specifics of a grid environment. For this reason, in the next section we use the insights in the pros and cons of the existing prediction methods to develop a new prediction method that effectively reacts to the peaks and level switches observed in a real grid environment.

3. New prediction method

3.1. DES prediction method

In this section we propose the method that effectively reacts to the peaks and level switches. Taking into account the benefits and drawbacks of the existing prediction methods, we propose the following predictor for $t = 2, 3, \ldots$:

$$\hat{y}_t = \begin{cases} \hat{y}_{DES,t} & \text{when } \kappa_{DES} = \kappa_{min}, \\ \hat{y}_{\mu,t} & \text{when } \kappa_{\mu} = \kappa_{min} \neq \kappa_{DES}, \\ \hat{y}_{median,t} & \text{else,} \end{cases} \quad (19)$$

with

$$\hat{y}_{DES,t} := \alpha^*_{t-1} y_{DES,t-1} + (1 - \alpha^*_{t-1}) \hat{y}_{DES,t-1}, \quad (20)$$

$$\hat{y}_{\mu,t} := \mu((y_1, ..., y_{t-1})), \quad (21)$$

$$\hat{y}_{median,t} := median((y_{t-l}, ..., y_{t-1})), \quad (22)$$

$$\kappa_i := \sum_{s=1}^{t-1} (y_s - \hat{y}_{i,s})^2, \;\; i \,\epsilon\, \{DES, \mu, median\}, \quad (23)$$

$$\kappa_{min} := min_{j \epsilon \{DES,\mu,median\}}(\kappa_j), \quad (24)$$

where

$$\alpha^*_t := \frac{\sum_{s=2}^{t} \alpha_{s-1,opt} (y_{s-1} - \hat{y}_{s-1})^2 1_{\{\delta_s = \delta_t\}}}{\sum_{s=2}^{t} (y_{s-1} - \hat{y}_{s-1})^2 1_{\{\delta_s = \delta_t\}}}, \quad (25)$$

with

$$\alpha_{s-1,opt} := \frac{y_s - \hat{y}_{s-1}}{y_{s-1} - \hat{y}_{s-1}}, \quad (26)$$

$$\delta_s := \begin{cases} H1 & \text{when } |y_s - \hat{y}_s| > 10\sigma((y_{s-k}, .., y_{s-1})), \\ H2 & \text{when } (y_s - \hat{y}_s) > 2\sigma((y_{s-k}, .., y_{s-1})), \\ H3 & \text{when } (y_s - \hat{y}_s) < -2\sigma((y_{s-k}, .., y_{s-1})), \\ M & \text{when } |y_s - \hat{y}_s| > \sigma((y_{s-k}, .., y_{s-1})), \\ & \quad |y_{s-1} - \hat{y}_{s-1}| > \sigma((y_{s-k}, .., y_{s-1})), \\ & \text{and } (y_s - \hat{y}_s)(y_{s-1} - \hat{y}_{s-1}) > 0, \\ B & \text{otherwise.} \end{cases} \quad (27)$$

Note that $\mu((y_1, ..., y_{t-1}))$ is the average of values $y_1, \ldots, y_{t-1}$, the $median((y_{t-l}, ..., y_{t-1}))$ is the median of values $y_{t-l}, \ldots, y_{t-1}$, and that $\sigma((y_{t-k}, .., y_{t-1}))$ is the standard deviation of values $y_{t-k}, \ldots, y_{t-1}$.

The basic idea of the DES predictor is that it (1) treats different classes of fluctuations differently, (2) always computes the optimal parameter for those classes, and (3) has the possibility to select another predictor from a set, because of performance reasons. As can be seen in (19), the DES prediction method selects the best prediction method from a set of 3 predictors, by comparing the sum of the squared errors of the previous measurements. The set contains the running average, the l-sliding window median, and the DES predictor. The l-sliding window median takes the median of the last l measurements. The DES-predictor part classifies different types of fluctuations: 3 huge fluctuations classes ($H1$, $H2$, and $H3$), 1 medium fluctuations class (M), and 1 base class (B). When the deviation between the measured value and its prediction is more than 10 times the standard deviation of the last k values, the measurement is of class $H1$. Furthermore, when the deviation between the measurement and the prediction is higher than two times (smaller than minus two times) the standard deviation, the measurement is of class $H2$ ($H3$). Two consecutive measurements that both deviate in the same direction with more than standard deviation from the predictions are of class M. B is the base class and is defined for the rest of the cases. This classification is defined in l (27). Subsequently, the DES

predictor computes the prediction by using the exponential-smoothing equation with an α parameter that equals the weigthed average of the optimal α's (see (25)) of the previous measurements that are from the same class as the most recent measurement. The derivation of (25) is a straightforward computation of the α's with the lowest MSE of the predictions for historical data. Further details are omitted for brevity.

We implemented the predictor with $k = 20$, and $l = 31$, and fixed the maximal number of data points to estimate each of the optimal α values for classes $H1$, $H2$, $H3$, M and B at 500. On the one hand, when k is higher than 20, the predictor reacts slower on changes in standard deviations. On the other hand, when less values are taken, the estimated standard deviations get less accurate. While 20 is a good value for k, a higher or lower value does not affect the results significantly. The value 31 as the value for l corresponds to one of the set of medians in [18]. Accuracy comparisons showed that this sliding-window median performs better than the other medians. The number 500 ensures that on the one hand the α_t^* is reliable and stable, because it is computed by enough measurements, and that on the other hand the α_t^* is still able to adapt to new charateristics in the dataset.

3.2. Discussion

Formula (19) shows the selection algorithm of the DES prediction method. As we have seen for the NWS method in subsection 2.1.2, the datasets sometimes show characteristics for which the average or the sliding median with window size 31 gives the most accurate predictions. Around 10 % of the measurements show these kinds of characteristics. Analysis show that measuring the κ_i's (see (24)) is an effective way of comparing the different predictors: in all the 10 % of the cases the average or the sliding-window median is chosen. We note that comparing the κ_i's is the same as comparing the Root of the Mean Squared Errors (as will be described in Section 4).

We chose to adapt the α_t's to the situation of the characteristics (see (25)). First, in [8] we concluded that the datasets may have completely different and continuously changing characteristics. We concluded from the analysis of the AESPs, the STES predictors and the AR method in Section 2 that adapting the α_t to the characteristics is an effective way to make the predictor robust against all the completely different and ever-changing characteristics of the datasets. However, we showed that improvements were possible on the following aspects: the choice of the α_t was not always clear and the α_t's are unstable in many predictors. The choice of taking the optimal α of the measured data is very clear and leads to stable α_t's within the classes of fluctuations.

Next, as shown in (27) we classified different types of fluctuations. We concluded in subsection 2.2.2 that Whybark consists of useful classification elements, but that improvements are necessary to be made to develop a useful classification of job times. We incorporated those improvements in our prediction method, as can be seen in (27).

4. Experimental Results

In this section we compare the performance of the DES prediction method with those of the predictors described in Section 2.

4.1. Quality metrics for prediction methods

To make a fair comparison we first have to select a metric for the prediction error. There are two commonly used error evaluators: the Root of the Mean Squared Error (RMSE) and the Mean Absolute Errors (MAE). Sometimes the MSE is taken instead of the RMSE. We selected the RMSE because in the kind of applications we used it is very important to minimize the number of high forecasting errors. Since the RMSE is more sensitive to high forecasting errors, we prefer to use this metric. The RMSE of a given predictor X is defined as

$$RMSE_X = \sqrt{\frac{1}{N-1}\sum_{t=2}^{N}(y_t - \hat{y}_t)^2}. \quad (28)$$

We also need to define a metric that quantifies the improvement of a predictor in comparison to another predictor. As can be seen in [8], peaks have a strong impact on the standard deviation or the RMSE of the dataset, which is highly correlated with the standard deviation. When the peaks do not show up periodically, it hard to predict when peaks occur. Consequently, peaks always have a strong effect on the RMSE of a predictor. For that reason, we are interested in a measure that is able to compare the performance of different predictors and is independent of the influence of the peaks on the standard deviation. Therefore, to compare different predictors we define $\Delta\%(A, B)$ to be the percentage improvement of predictor A versus B:

$$\Delta\%(A, B) := \frac{RMSE_A - RMSE_B}{RMSE_B - RMSE^*} * 100\%, \quad (29)$$

where the value $RMSE^*$ is the optimal *postcast* selected from the set of the NWS predictors. It indicates the theoretically maximal forecasting performance (minimum error) that the method could have achieved if the best predictor at each step were known (see for more details about the $RMSE^*$ the definition of the *Optimum* in [18]). When a peak occurs, even the optimal forecasting method has the property that it was not able to predict that peak. But for the

successive measurement, the optimal forecast mostly has a low RMSE. Consequently, the $RMSE^*$ is the RMSE that a prediction method would have when it would predict the behavior after peaks perfectly.

4.2. Experimental setup

The next necessary step of the comparison analysis is to collect a significant amount of data from a real grid environment. In [8] 40 experiments have been performed on 18 shared processors in Planetlab [1], to generate datasets for the statistical properties analyses. Planetlab [1] is a commonly used grid test bed environment shared by many users. For the comparison in this paper between the existing prediction methods and the DES, 45 additional datasets from 18 different Planetlab nodes were generated for the comparison phase. Version 2.0 of Planetlab was installed on the nodes, which is the same version as is used for the experiments in [8]. Each dataset is generated by a single run and consists of the running times (i.e. wallclock times) of 2000 consecutive and identical jobs. The average duration of a run is about two hours. We observed significant differences in the durations; some run durations even exceeded the 10 hours, especially the runs on the Arizona node. In order to correlate the statistical properties of the job times at the different nodes, at each day all runs were kicked off *simultaneously* at 9:00 CET. The experiments were done during six additional consecutive days. We used nodes in Amsterdam, The Netherlands (ams); Arizona, USA (ar); Australia (au); California, USA (cal); British Columbia, Canada (ca); China (china); Denmark (dk); Inria, France (inria); Madrid, Spain (mad); Moscow, Russia (mos); San Diego, USA (sandiego); Santa Barbara, USA (santab); Singapore (sing); Taiwan (tw); Telaviv, Israel (telaviv); Utah, USA (utah); Warsaw, Poland (warsch); and Washington, USA (wash). The selection of which nodes were used at which days is based on both the availability, and the global distribution of the nodes. Moreover, we never started a new run on a node on which a previous run was interupted. Table 1 shows which nodes were used during these six days.

	Day					
Run nr	1	2	3	4	5	6
Run 1	ams01	ams02	ams03	ams04	ams05	inria01
Run 2	ar01	ar02	ar03	ar04	ar05	ar06
Run 3	warsch01	cal01	cal02	cal03	cal04	cal05
Run 4	china01	china02	wash01	wash02	wash03	wash04
Run 5	au01	au02	au03	sandiego01	sandiego02	sandiego03
Run 6	utah01	utah02	utah03	ca01	ca02	mad01
Run 7	mos01	mos02	telaviv01	telaviv02	tw01	tw02
Run 8	santab01	dk01	sing01			

Table 1. Run-schedule

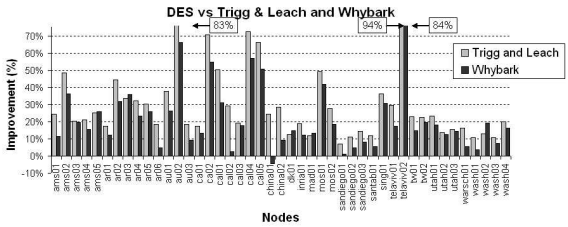


Figure 8. DES-predictor improvements compared to the Trigg & Leach and the Whybark predictor

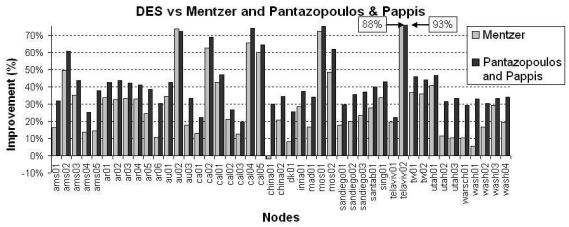


Figure 9. DES-predictor improvements compared to the Mentzer and Pantazopoulos & Pappis predictor

4.3. Comparison results

In this subsection, we compare the predictions of the DES (see Section 3) with those of the existing predictors (see Section 2). To this end, we compute the percentage improvements (see 4.1) of DES compared to the existing predictors for the 45 datasets, described in 4.2. The results are outlined below.

Figure 8 shows the performance improvements of the DES prediction method compared to the Trigg and Leach predictor and the Whybark predictor. Similarly, Figure 9 shows the results for DES compared to the Mentzer and Pantazopoulos & Pappis predictor.

The results presented in Figures 8 and 9 show that the DES prediction method strongly and consistently outperforms the other predictors. The improvements are remarkably high: for many datasets the DES prediction method shows improvements of more than 30%. From the four AESP predictors considered here, the Whybark predictor gives the most accurate predictions, but is still strongly outperformed by our DES predictor.

Figure 10 shows the prediction performance of in total 12 STES predictor-parameter combinations for the three randomly-chosen datasets ar07, tw01, and wash01. We tested the STES |e| predictor with parameter $\gamma < 0$, the STES |e| predictor with no restrictions on γ, the STES e^2 predictor and the STES Whybark predictor. For all the four predictors we used 10, 80 and 2000 data points used for the

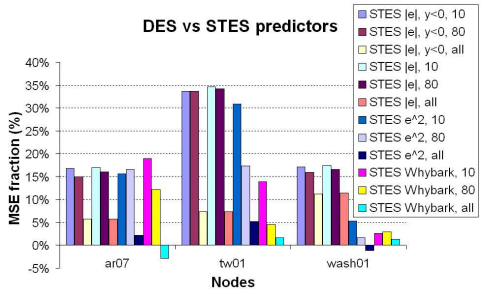


Figure 10. DES-predictor improvements compared to the STES predictors

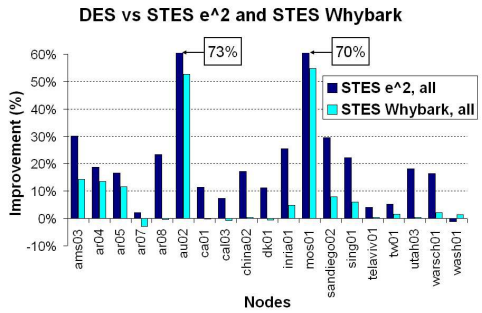


Figure 11. DES-predictor improvements compared to the STES e^2 and STES Whybark predictors

fit. We tested those predictors only with three datasets due to the fact that the STES predictors take too much time (a whole day per dataset). The results in Figure 10 show that it is always better to use all the data (at least 2000 values) for the parameter fit, and that the STES e^2 and the STES Whybark are the best STES predictors and need further comparison analysis with more datasets to conclude more about the quality of their predictions. We used the following randomly-chosen datasets for the further analysis: ams03, ar04, ar05, ar07, ar08, au02, ca01, cal03, china02, dk01, inria01, mos01, sandiego02, sing01, telaviv01, tw01, utah03, warsch01, and wash01.

In Figure 11 we compare the improvements of the DES prediction method in comparison with the STES e^2 and the STES Whybark predictor for all those datasets. We clearly observe that the DES prediction method outperforms the other two predictors. For two of the datasets the DES shows even more than 50% improvements for both of the other predictors.

Figure 12 below shows the improvements of the DES prediction method compared to the ES predictor with $\alpha = 0.5$, denoted as the ES(0.5) predictor. Despite the fact that DES is based on ES, the method shows a completely different accuracy: the difference between the RMSE's of the ES(0.5) predictions and the DES predictions ranges from

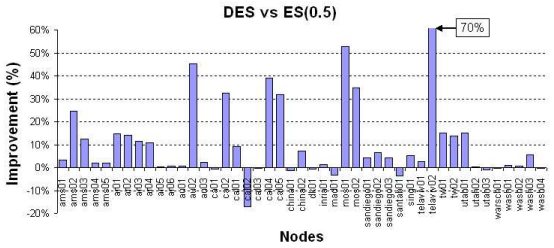


Figure 12. DES-predictor improvements compared to the ES(0.5) predictor

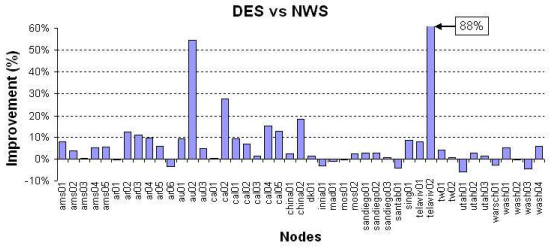


Figure 13. DES-predictor improvements compared to the NWS prediction method

-17% to +70% and differs for each dataset. Only for single dataset the ES(0.5) predicts significantly better than the DES. On average the DES shows 11% improvement.

Figure 13 illustrates the improvement of the DES prediction method in comparison with the NWS prediction method. The figure shows that DES mostly outperforms the NWS prediction method. For 6 of the 45 datasets the NWS prediction method is only 1 to 5 percentages more accurate as the DES prediction method. However, on average 8% improvements can be performed by implementing the DES prediction method instead of the NWS prediction method.

Figure 14 illustrates the performance improvement of the DES prediction method in comparison with the AR(16) predictor. Figure 14 shows a more bursty characteristic than Figure 13, because the DES and the NWS prediction methods have more similarities. For five datasets the AR(16) predictor shows more than 5% accuracy improvements compared to the DES prediction method. In fourteen cases the DES prediction method shows more than 10% improvement. For three datasets the DES prediction method is even more than 40% more accurate than the AR(16) predictor. On average, the DES prediction method is 9% more accurate as the AR(16).

As we see in Figures 12 to 14 DES strongly outperforms all other predictors for datasets au02 and telaviv02. Further analysis show that the peaks in those datasets have a huge influence on the standard deviation. We investigate that the main reason for the enormous outperformance is that the

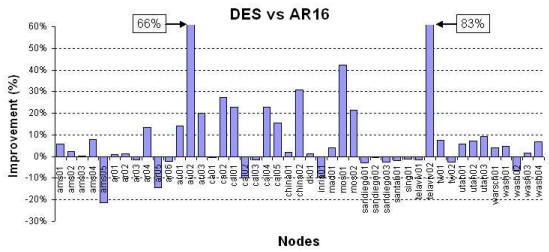


Figure 14. DES-predictor improvements compared to the AR16 predictor

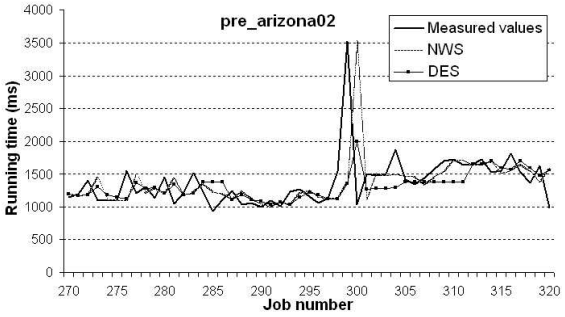
DES prediction method is the only method that reacts properly on the high peaks. Nevertheless, when we filter out the peaks with a deviation above 10 times the standard deviation we still clearly outperform the others predictors.

Next, we investigate the reason for the performance improvement of the DES prediction method in comparison with the best of the other predictors, the NWS prediction method. To this end, we analyze the situations in the datasets where DES clearly outperforms NWS. We observe two main reasons why DES outperforms the NWS prediction method: (1) DES reacts more properly on peaks, and (2) DES reacts more properly on level switches. To investigate this more carefully, we add the predictions of the DES prediction method in Figure 2. The results are graphically represented in Figure 15.

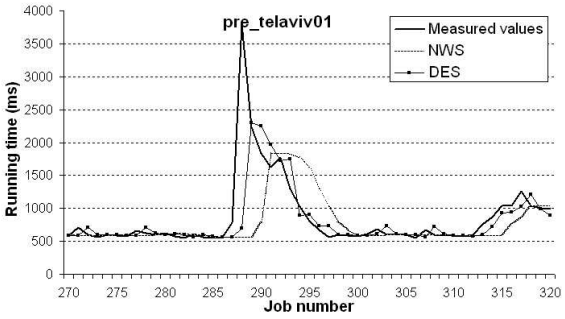
We notice the following from Figure 15(a). Due to the values before the peak, the NWS chooses a predictor that gives a high weight to the last value. When the peak appears, the NWS prediction for the next job time also gets very high, unless that there is a high probability that the peak appears once, because almost all the peaks in history appeared only once. The DES prediction method also gives a prediction that is slightly too high due to the fact that one peak in history did introduce a new level. But, nevertheless, the DES prediction method is far more accurate as the NWS prediction method.

In Figure 15(b), we have a different situation: peaks introduce a new level of job times. In this example we see that the NWS also shows a bad performance. Due to the fact that the values before the peaks need a predictor that has a more averaging property, the NWS also uses the average to predict the value after the peak despite the fact that the peaks in history very often introduced a new level. It takes three values (i.e., jobs durations) before the NWS prediction method realizes that the predictions are not accurate. The DES prediction method clearly shows a better pattern of predictions in the level switch.

Despite the two previous mentioned situations where the NWS prediction method does not react very properly on peaks, there are a lot of situations where the NWS prediction method does react appropriately. There are two situations where the NWS does react very accurately: (1) when a more averaging predictor is chosen because of the values before a peak, and the peak does not introduce a new level, (2) when a last value predictor is chosen before a peak and the peak introduces a new level, and (3) when there are less number of values between the peaks, the NWS still *remembers* what the best predictor was during the last peak, because peaks have a high impact on the RMSE, and therefore on the choice of the predictor.



(a) **Peak on pre_arizona02**



(b) **Level switch on pre_telaviv01**

Figure 15. DES- and NWS prediction method reacting on a peak and a level swith in the job times

To summarize, the comparison results show that in general the DES prediction method by far outperforms the existing predictions methods. Another well performing predictor for predicting job times is the NWS prediction method.

5. Conclusions and challenges

Extensive theoretical and practical analysis, based on the results of [8] (outlined in the Introduction), of a variety of prediction methods have been performed. The results show that none of these methods is well-suited for dealing with

the specifics of job times in a grid environment, including the presence of level switches and sudden peaks. Due to the absence of an effective prediction method for running times of jobs, we have developed a new prediction method, called DES, that overcomes the shortcomings of the existing methods by properly reacting to level switches and peaks. The power of DES lays in the fact that it is able to deal with on the one hand with peaks and on the other hand with level switches.

Extensive comparisons with a large number of datasets show that DES is a highly effective method for predicting running times of jobs on shared processors. DES consistently outperforms the common grid prediction methods, such as the NWS predictor, AR(16), and ES, and moreover, gives much more accurate predictions than the Adaptive Exponential Smoothing Predictors Trigg and Leach, Whybark, Mentzer, Pantazopoulos and Pappis, and four kinds of promising Smooth Transition Exponential Smoothing predictors.

The results presented in this paper lead to a number of challenges for further research. First, there is room for refinement of the DES predictor. For the job times, the current choice of parameters for the heights of peaks (i.e. a difference between two consecutive values of more than 2 times the standard deviation is a peak, and more than 10 times the standard deviation is a high peak) leads to significant improvements in the predictions. We expect that those parameters partly depend on the properties of the dataset. We plan to investigate whether it is possible to adapt these parameters to the statistical properties of the datasets. Second, we aim to apply the DES prediction method for predicting other types of grid property measurements (e.g., latency, CPU utilizations). Finally, the ultimate goal of the development of effective prediction methods in the context of the computational grid is to decrease the effective running times of distributed applications by triggering effective load rebalancing actions. Therefore, the next step is to quantify the actual improvements that can be obtained in the running times of distributed applications, which addresses a challenging area for further study.

References

[1] http://www.planet-lab.org.

[2] H. Dail, G. Obertelli, F. Berman, R. Wolski, and A. Grimshaw. Application-aware scheduling of a magnetohydrodynamics application in the legion metasystem. In *HCW '00: Proceedings of the 9th Heterogeneous Computing Workshop*, page 216, Washington, DC, USA, 2000. IEEE Computer Society.

[3] P. A. Dinda. A prediction-based real-time scheduling advisor. In *IPDPS '02: Proceedings of the 16th International Parallel and Distributed Processing Symposium*, page 35, Washington, DC, USA, 2002. IEEE Computer Society.

[4] P. A. Dinda. Online prediction of the running time of tasks. *Cluster Computing*, 5(3):225 – 236, July 2002.

[5] P. A. Dinda and D. R. O'Hallaron. Host load prediction using linear models. *Cluster Computing*, 3(4):265–280, 2000.

[6] A. M. Dobber, G. M. Koole, and R. D. van der Mei. Dynamic load balancing for a grid application. In *Proceedings of HiPC 2004*, pages 342–352. Springer-Verslag, December 2004.

[7] A. M. Dobber, G. M. Koole, and R. D. van der Mei. Dynamic load balancing experiments in a grid. In *Proceedings of the 5th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005)*, pages 123–130. IEEE Press, May 2005.

[8] A. M. Dobber, R. D. van der Mei, and G. M. Koole. Statistical properties of task running times in a global-scale grid environment. In *Submitted for publication*, 2005.

[9] R. Fildes. Quantitative forecasting-the state of the art: extrapolative models. *Journal of the Operational Research Society*, 30:691–710, 1979.

[10] B. Lee and J. Schopf. Run-time prediction of parallel applications on shared environments, poster-paper. In *Proceedings of Cluster2003*, December 2003.

[11] C. Liu, L. Yang, I. Foster, and D. Angulo. Design and evaluation of a resource selection framework for grid applications. In *HPDC '02: Proceedings of the 11 th IEEE International Symposium on High Performance Distributed Computing*, page 63, Washington, DC, USA, 2002. IEEE Computer Society.

[12] J. T. Mentzer. Forecasting with adaptive extended exponential smoothing. *journal of the Academy of Marketing Science*, 16:62–70, 1988.

[13] S. N. Pantazopoulos and C. P. Pappis. A new adaptive method for extrapolative forecasting algorithms. *European Journal of Operational Research*, 94:106–111, 1996.

[14] Y. Qiao and P. A. Dinda. Network traffic analysis, classification, and prediction. *Tech. Rep. NWU-CS-02-11*, January 2003.

[15] J. W. Taylor. Smooth transition exponential smoothing. *Journal of Forecasting*, 23:385–404, 2004.

[16] D. W. Trigg and A. G. Leach. Exponential smoothing with an adaptive response rate. *Operations Research Quarterly*, 18:53–59, 1967.

[17] D. C. Whybark. Comparison of adaptive forecasting techniques. *Logistics Transportation Review*, 8:13–26, 1973.

[18] R. Wolski. Forecasting network performance to support dynamic scheduling using the Network Weather Service. In *HPDC*, pages 316–325, 1997.

[19] R. Wolski. Experiences with predicting resource performance on-line in computational grid settings. *SIGMETRICS Performance Evaluation Review*, 30(4):41–49, 2003.

[20] L. Yang, I. Foster, and J. M. Schopf. Homeostatic and tendency-based cpu load predictions. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 42.2, Washington, DC, USA, 2003. IEEE Computer Society.