

# Multivariate Resource Performance Forecasting in the Network Weather Service<sup>‡</sup>

Martin Swany and Rich Wolski  
Department of Computer Science  
University of California  
Santa Barbara, CA 93106  
{swany,rich}@cs.ucsb.edu

## Abstract

*This paper describes a new technique in the Network Weather Service for producing multi-variate forecasts. The new technique uses the NWS's univariate forecasters and empirically gathered Cumulative Distribution Functions (CDFs) to make predictions from correlated measurement streams. Experimental results are shown in which throughput is predicted for long TCP/IP transfers from short NWS network probes.*

## 1 Introduction

Performance monitoring in Computational Grid settings is widely recognized to be an essential capability [7, 11, 16, 26, 29]. There are a variety of systems available for taking measurements to this end, particularly for network performance [3, 10, 12, 15, 18, 20, 24, 26, 30]. For the Grid, this information is used primarily to optimize application execution. Applications are directed (either by a human user or an automatic scheduler) to use the resource (network, CPU, memory system, etc.) that exhibits the best measured performance.

Frequently, monitor data gathered from these tools is used as a *prediction* of future performance. That is, an observation of past performance implies that future performance levels will be similar. For network bandwidth estimation, in particular, many users simply conduct lengthy data transfer between end-hosts, observe the throughput, and use that observation as a harbinger of future available bandwidth.

There are two significant problems with this methodology. First, it is not clear that the last observation, particularly of network throughput, is a good estimate of future levels since network performance can

fluctuate significantly [17, 32]. Secondly, it requires that the resource be loaded “enough” to yield a significant estimate. On high-throughput networks with lengthy round-trip times (e.g. the NSF Abilene [1] network) enough data must be transferred to match the bandwidth-delay product [14] for the end-to-end route. This intrusiveness can be costly, both in terms of wasted resource and lost time while the probe takes place.

Performance monitoring tools such as the Network Weather Service (NWS) [25, 30] address the first problem by making the predictions explicitly using statistical techniques. Application-level schedulers [2, 5, 19, 23] have been able to use these predictions (and measures of prediction error) to achieve good execution performance levels in a variety of Grid settings, despite fluctuating resource performance.

However, in making these forecasts, current NWS methodologies (described in [29, 30]) share the second problem with other performance monitoring tools for the Grid. Performance responses that are expensive to generate with probing, such as steady-state TCP/IP throughput, are unfortunately best forecast from a history of such responses. Moreover, the dynamics of the network are frequently changing, making old measurements increasingly less statistically valuable as time passes. As such, instrumenting Grid applications or tools (e.g. GridFTP [2]) and using the observations from the instrumentation may yield dramatic inaccuracies if the time between application runs is significant.

In this work, we describe a new multivariate forecasting technique that enables the NWS to automatically correlate monitor data from different sources, and to exploit that observed correlation to make better, longer-ranged forecasts. In particular, this new correlative method allows the NWS to combine infrequent and irregularly spaced expensive measurements (possibly obtained through instrumentation) with regularly

\*0-7695-1524-X/02 \$17.00 ©2002 IEEE

<sup>†</sup>This work was supported in part by NSF grant ANI-0123911.

spaced, but far less intrusive probes to make predictions that would be far too expensive to generate with probes alone.

We demonstrate the effectiveness of the technique by showing how it can be used to forecast long HTTP transfers using a combination of short NWS TCP/IP bandwidth probes and previously observed HTTP transfers. Using short TCP/IP messages to predict steady-state throughput is the subject of significant research [6, 13, 28]. Our particular experimental verification is similar to the work described in [27], but unlike that work, our new method does not rely on linear regression.

Our results indicate that the combination of short NWS bandwidth probes and previous HTTP history can be used to generate more accurate HTTP transfer bandwidth forecasts than has been previously possible, particularly when the time between observable HTTP transfers is long.

More generally, we observe that it is often advantageous to combine data from two or more measurement streams which may be correlated in some way. The HTTP experiment described herein is an example of how heavy-weight operations that would be too intrusive to duplicate as explicit probes can be monitored and combined with lightweight, inexpensive probes of a resource to produce a forecast of the heavy-weight operation. Similar correlation problems exist when predicting CPU availability [31] and real memory. As such, we believe that this new forecasting technique can be used to combine probe monitor data with application instrumentation data (e.g. internal code timings) gathered from a computation by a mechanism such as Autopilot [21] or Tau [22].

In summary, the contributions that this new work makes are:

- the description of a new correlation technique that greatly improves observed prediction error, particularly for longer-ranged predictions
- an empirical evaluation of the technique to the problem of long-message bandwidth prediction using short-message bandwidth probes

Our results show similar or better prediction accuracy than what has been previously reported in either the networking or Grid computing literature and indicate a general methodology for combining historical instrumentation data with periodic explicit measurements.

## 2 Forecasting using Correlation

The generalized NWS forecasting system can be thought of as having access to a collection of measurement series. These series are characterized by units and frequency. In order to make multivariate predictions, a system needs to operate on some subset of these measurement series with a combination of *correlation* and *forecasting*. The correlator serves to map  $X$  values onto  $Y$  values where the  $X$  values are plentiful and “cheap” and the  $Y$  values are “rare” and expensive. The univariate forecaster produces forecasts, which represent the current and future performance of a single dataset. At present, the NWS provides us with a rich set of tools for the latter, and our goal is to augment those tools by addressing the former.

Given two correlated variables, knowledge of one at a given point in time provides us some information as to the value of the other. Stated more formally, being able to fix the value of one of the variables allows us to make reasonable assumptions about the probability of various values of the other variable, based on their history of correlation. The question is how to exploit that empirically to make a prediction.

### 2.1 Correlator Methodology

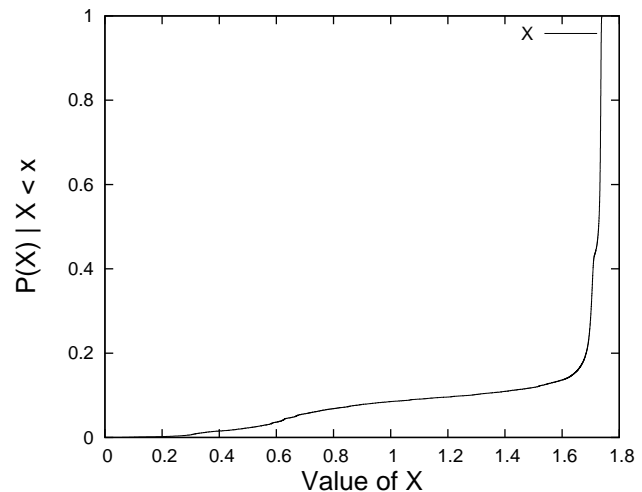
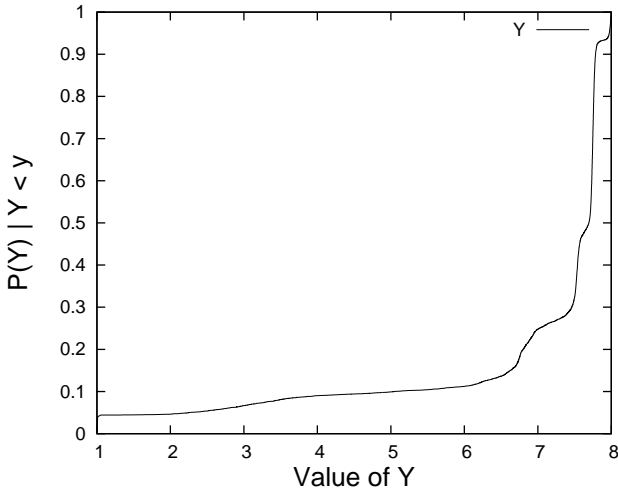


Figure 1. CDF of NWS data ( $X$ )

The correlator can determine the mapping between the  $X$  and  $Y$  values in a number of ways. Methods such as linear regression and traditional correlation operate on datasets of equal sizes and assume that the variables in question are related linearly. However, in the case of measurement data gathered from a variety of sources:



**Figure 2.** *CDF of HTTP data (Y)*

- data sets may be of different sizes
- data items within each set may be gathered with different frequencies and/or with different regularities.
- the units of measure may be different between data sets.

The mechanism that our correlator uses is similar to rank correlation techniques. A rank correlation measure sorts datasets and does linear correlation based on their position in the sorted list, (their “rank”). For our purposes, rank correlation is attractive because it is non-parametric. A non-parametric technique is appropriate since we cannot be certain of the distribution of the data. However, if there are different amounts of data in the respective datasets, something must be done to make the set sizes equal.

The Cumulative Distribution Function (*CDF*) is defined as the probability for some set of sample points that a value is less than or equal to some real valued  $x$ . If the probability density function (*PDF*) is known, then it is defined as follows:

$$CDF_X(x) = \int_{-\infty}^x PDF_X \quad (1)$$

Empirically, we can approximate the *CDF* by:

$$CDF_X(x) = \sum_{i=1}^{position_x} \frac{1}{count_x} \quad (2)$$

Where  $count_x$  is the total elements in the sample set. Which is equivalent to:

$$CDF_X(x_{target}) = \frac{count_{x'} \mid x' \leq x_{target}}{count_x} \quad (3)$$

So, we are able to compute the empirical *CDF* for both the  $X$  and  $Y$  variables ( $CDF_X$  and  $CDF_Y$ , respectively.) Now we have a way to rank measurements in datasets of different sizes. By computing the *CDF* for multiple datasets, we are able to compare them in a way similar to the rank correlation technique.

However, since our goal is forecasting, we are not interested in using this to quantify the correlation between the  $X$  and  $Y$  datasets, but rather in using them as a mapping function from  $X$  to  $Y$ <sup>1</sup>. We can use the position of the value in  $X$  ( $x_{target}$ ) to produce a value of  $Y$  ( $y_{forecast}$ ). This allows us to inexpensively compute a relationship between the two sets.

Specifically:

$$y_{forecast} = CDF_Y^{-1}(CDF_X(x_{target})) \quad (4)$$

This gives us the value in  $Y$  (in  $CDF_Y$ ) associated with the position  $x_{target}$ . If the *CDF* position of  $x_{target}$  ( $CDF_X(x_{target})$ ) is between measured  $Y$  values, linear interpolation is used to determine the value of  $y_{forecast}$ . This mapping is similar in spirit to quantile-quantile comparisons (e.g. qq-plot) that are familiar. Again, however, the goal is forecasting and not the quantification of relationship.

The  $x_{target}$  can be chosen in a variety of ways. Tests have shown that the most effective way to choose  $x_{target}$  is by using the NWS forecasters to produce the value. This is because the univariate forecast is less sensitive to transient outliers (i.e. for all the same reasons that univariate forecasting is effective at all.) This is simply:

$$x_{target} = Forecast(X) \quad (5)$$

As an illustration of the *CDF* forecaster, consider the plots of the *CDF*s computed from the data described in Section 4. The NWS ( $X$ ) values are shown

<sup>1</sup>Indeed, no correlation coefficient is defined for this correlation technique, although we are investigating this for selection among datasets.

in Figure 1 and the HTTP ( $Y$ ) values are shown in Figure 2. To compute the forecast for  $Y$  we use the position in the  $CDF$  of  $x_{target}$ , which yields some real number between 0 and 1. The forecast is the value of  $y$  at that point in the  $CDF$  of  $Y$ .

This predictive method then forms both the  $X$  and  $Y$   $CDF$ s. At each prediction cycle, a univariate forecast of  $X$  is taken and that  $X$  value’s position is determined. That value is used to determine the value in  $Y$  at the corresponding position, which is the prediction.

We have implemented an adaptive version of this correlator that uses varying amounts of history used to form the  $CDF$  of each dataset. These different versions are selected against based on their accuracy in the same way that the univariate forecasters do [30]. We have also implemented a prototype that alters the computation of the  $CDF$  and weights current values more heavily when creating the mapping function. However, the data presented in this paper uses only the most simple case in that the computed  $CDF$ s contain all measurement data.

The  $CDF$  is a useful measure for a number of reasons. One reason deals with the practicality of delivering this information to clients of this system – Grid programs and schedulers. A  $CDF$  dataset may be compressed with varying degrees of loss. In simple cases, a handful of points can describe the  $CDF$  with linear interpolation between specified points.

## 2.2 Error

The NWS uses prediction error internally to choose optimal forecasts and provides that information to clients as a measure of the forecast accuracy.

Some notion of normalized error is necessary so that over-prediction and under-prediction (and excessively high or low measurement values) are represented fairly. For instance, 1Mbit prediction error is quite different for transfers that observe 100Mbits/second than for those that see 10Mbits/second. So, to determine the accuracy of various datasets which might not have the same units and to evaluate the overall performance of the system, we compute the Mean Normalized Error Percentage (MNEP) as the last prediction’s error over the current average value.

$$MNEP = \frac{\sum |value_t - prediction_t| / mean_t}{observations} \quad (6)$$

Again, this is required due to the forecast-oriented application of the system. The mean used ( $mean_t$ ) is the current mean and is reevaluated at each timestep.

## 3 Experimental Methodology

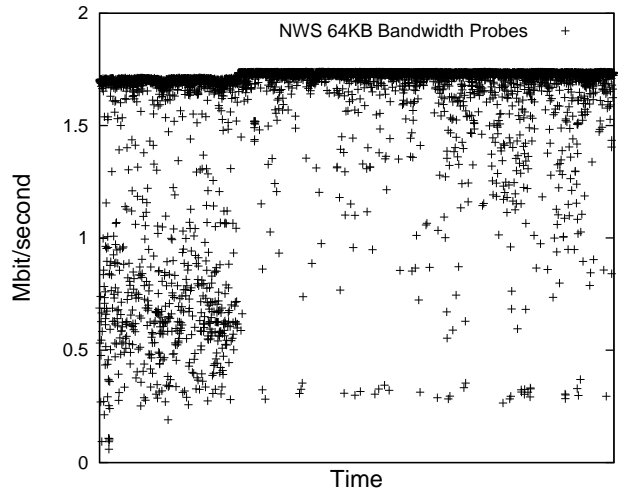


Figure 3. 64KB NWS Probes.

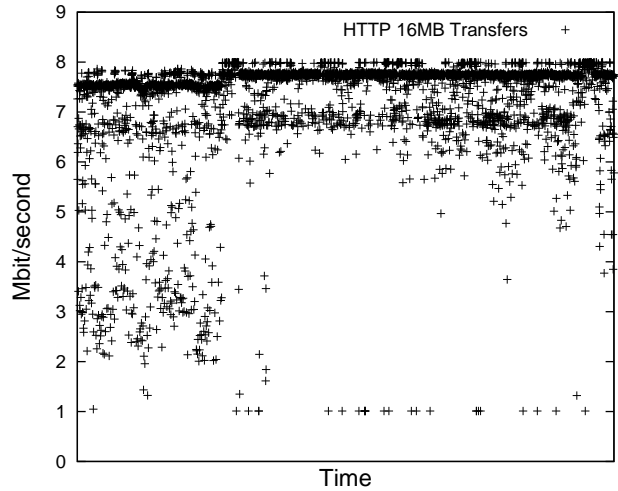
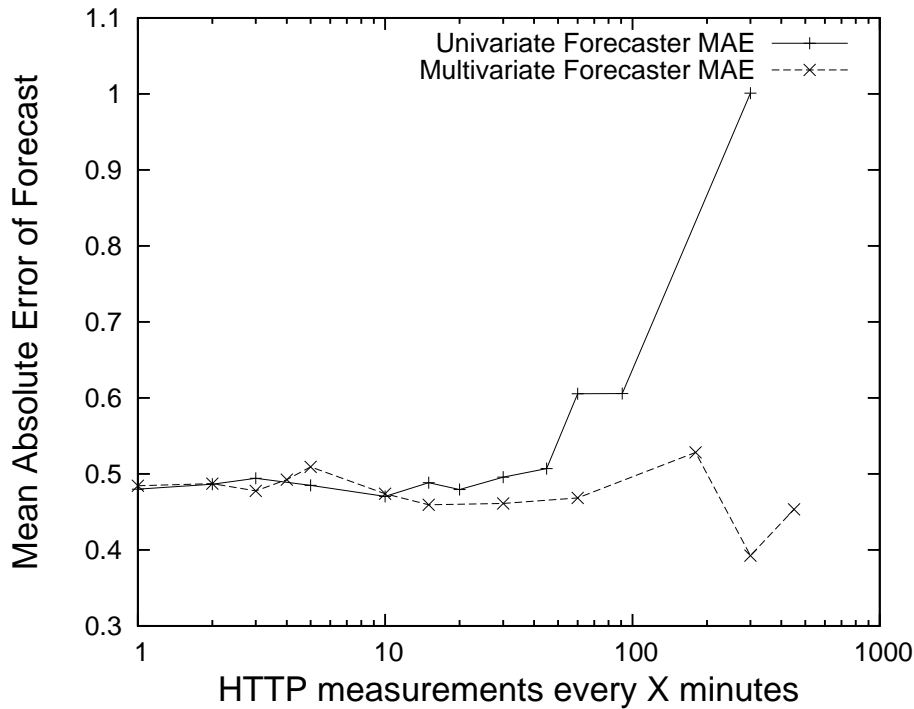


Figure 4. 16MB HTTP Transfers.

To investigate the effectiveness of our new technique, we used the following experimental procedure. We generated default-sized NWS bandwidth measurements (64K bytes), using the TCP/IP end-to-end sensor between a pair of machines every 10 seconds. Every minute, we initiated a 16MB HTTP transfer and the recorded the observed transfer bandwidth. The NWS data is depicted in Figure 3 and the HTTP data is shown in Figure 4. While the exact correlative relationship is not clear visually, the shapes of each data set appear to share similar characteristics.



**Figure 5. Comparison of Mean Absolute Error (MAE) between univariate and multivariate forecasts for different frequencies of HTTP measurements.**

Our goal was to use this data to investigate the accuracy with which the short 64K NWS transfers could be used to predict the long 16MB HTTP transfers given successively longer periods between 16MB transfers. This experimental configuration emulates the case where instrumentation data is gathered from an application (e.g. HTTP/web client). We examine the effects as this data ranges from plentiful (frequent 16MB transfers) to scarce. In particular, we wish to study how well NWS 64K messages can be used to predict large messages that are separated by several hours as the network conditions often change over this time scale.

We chose to use HTTP transfers as a prediction target for several reasons. HTTP is, of course, widely used in the Internet. Further, HTTP is a key component of SOAP and Web services and we are likely to see this used more and more in high-performance computing. This data represents two of the cases mentioned previously. That is, it is representative both of the relationship between short and long network experiments, and also of instrumentation data (which it indeed is) being fed into the forecasting service.

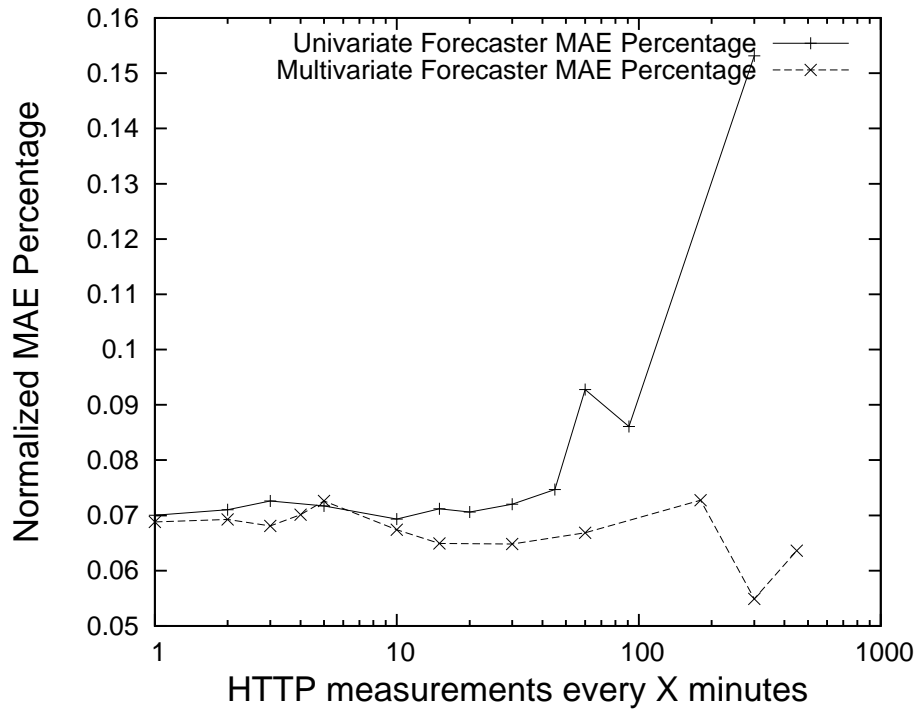
Our basis of comparison is prediction error measures as either the mean absolute difference between prediction and corresponding measurement, or the square root

of the mean square prediction error. We compare three prediction methods: the *last value* method which uses the last observed value as a prediction, the univariate NWS forecast applied solely to the 16MB probe history, and the new correlative method.

We chose 16MB as our test case to limit intrusiveness. Also, there is less variability, in general, in the observed bandwidth as the amount of data transferred gets larger (evidenced one of many places by the better predictive accuracy of longer transfers seen in [27]). This can be intuitively explained by the notion that any transient periods of high congestion are amortized over the longer transfer time. So, 16MB transfers are not so long as to make things easy, but long enough to represent real applications and real data.

## 4 Results

The following data represents 10 days of measurement data between Argonne National Laboratory (ANL) and UC Santa Barbara (UCSB). These results are based on a version of the forecaster we have implemented that automatically computes the empirical *CDF* for each forecast incrementally, based on all previous measurement data. That is, the forecaster “learns”



**Figure 6. Comparison of Moving Normalized Error Percent (MNEP) of the Mean Absolute Error (MAE) between univariate and multivariate forecasts for different frequencies of HTTP measurements.**

as it goes and acquires more data. The  $X$ -axis of these graphs represents the time between 16MB transfers. That is, we gradually decimate the  $Y$  dataset until there are measurements approximately every 500 minutes.

Figure 5 shows the Mean Absolute Error (MAE) of the NWS univariate forecasters compared to the MAE of the multivariate forecasters. The NWS uses the MAE as it is a unit-preserving metric of success. We can see that the multivariate forecaster enjoys slightly better, but comparable, performance as the 16MB  $Y$  values are being measured frequently. However, while the multivariate forecaster maintains effectiveness as the  $Y$  values become less frequent, the univariate forecasters begin to lose accuracy. Notice that the mean absolute prediction error is only 0.47 megabits/second at 500 minutes. That is, using the new forecasting technique, the NWS can predict 16MB transfer bandwidth using 64K measurements with a mean error of 0.47 megabits/second.

To give some notion of magnitude, we normalize the MAE by the average transfer bandwidth resulting in the MNEP. Figure 6 shows the MNEP of the MAE. The MNEP shows us the percentage of the current average value that an error represents. This is not unit-preserving, but depicts the error values in relation to the

moving average of the data. Again, at 500 minutes, the 64K transfers, when used in our forecaster, can predict 16MB transfer bandwidth to within 7% on the average.

Figure 7 shows the square root of the Mean Square Error (MSE) of both types of forecasts. This value bears resemblance to the traditional notion of standard deviation and tends to indicate the variance of the error. Using a combination of the univariate NWS forecasting technique and incrementally constructed empirical  $CDF$ s, our new technique can predict actual throughput with relatively high levels of accuracy.

For comparison purposes, we also include a “forecaster” based on the last value. This is the “accuracy” of a prediction that assumes the the performance of a given transfer will be the same as it was the last time it was performed – actually, a common way that monitor data is used in the “real” world. Figure 8 show the MNEP of the MAE for the  $CDF$  forecaster and the “Last Value” predictor. This shows that using the last value as a predictor for this dataset yields “forecasts” with over 100 percent error. Clearly, forecasting yields predictive accuracy that significantly better than the the base case “last-value” approach.

These results show that by combining instrumentation data with NWS forecasting, our technique is able

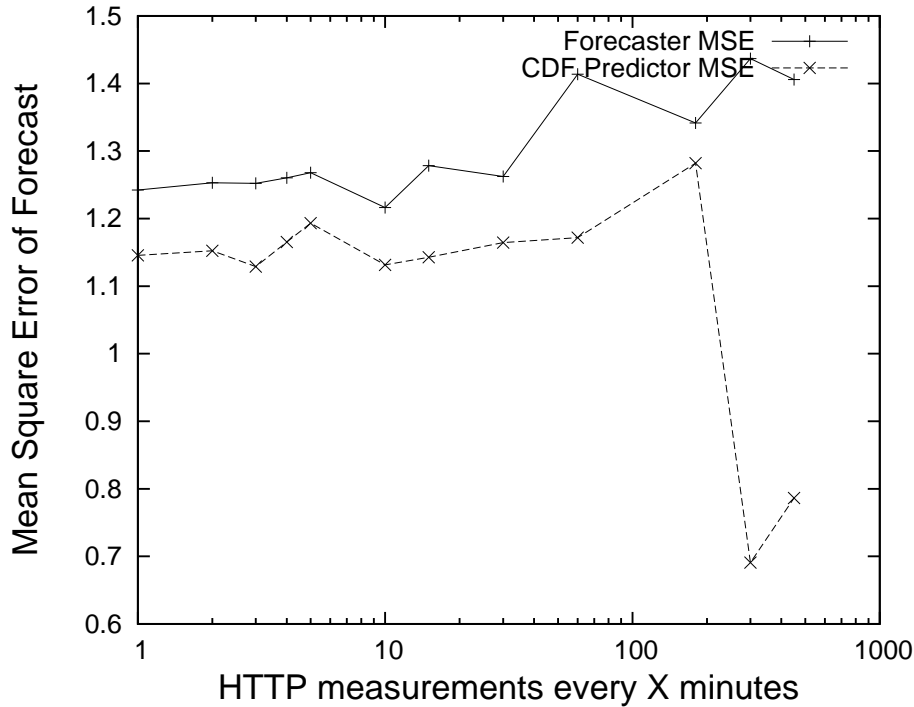


Figure 7. Comparison of the square root of the Mean Square Error (MSE) between univariate and multivariate forecasts for different frequencies of HTTP measurements.

Measurement Target	Moving Normalized Error Pct.	MNEP (square error)	Last Value MAE MNEP
10MB-Transfers	0.184883	1.454097	1.008486
100MB-Transfers	0.102253	1.216345	1.033892
500MB-Transfers	0.074396	0.737839	1.030498
1GB-Transfers	0.096729	1.367443	1.040794

Table 1. ISI to ANL – Normalized Error Percentages when using *NWS* data to forecast *GridFTP* [8] performance compared to the “last value” prediction. (data from ANL [27]).

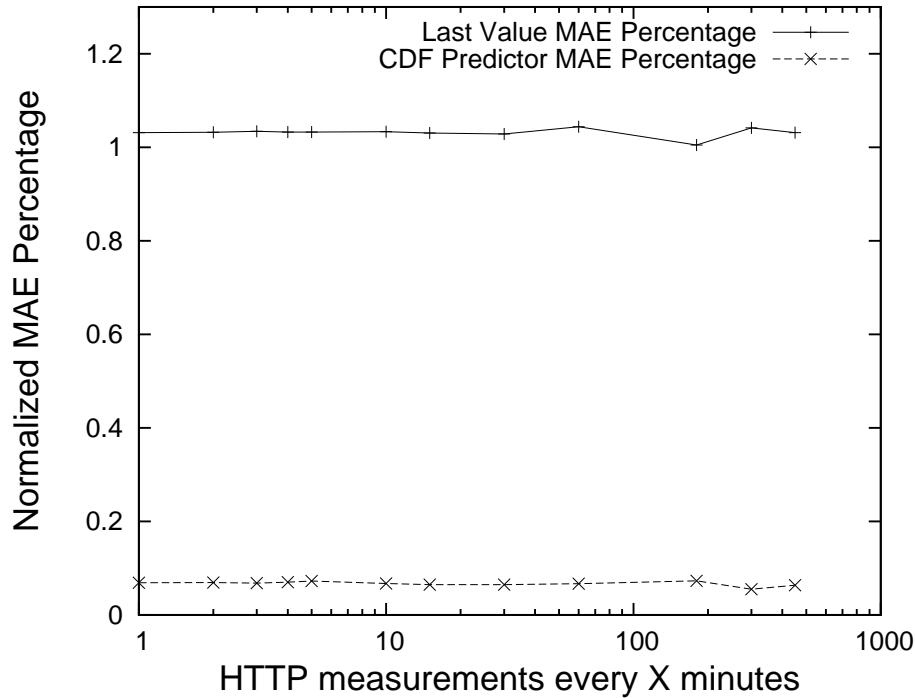
to use relatively short transfers to predict long-message throughput. Moreover, the predictive accuracy does not degrade substantially when forecasts of events occurring hours apart are used as inputs. Our work indicates that the statistical techniques used by the NWS can extract and exploit the inherent performance relationship that must exist between message transfers of different sizes, and does so automatically.

#### 4.1 Additional Results

This section further validates our experimental results by using additional datasets that were not gathered by us. The first of these is from similar work from Vazhkudai and Schopf [27]. This work is closely related to ours in that NWS measurements are used to

predict *GridFTP* [8] data transfers. Table 1 shows the predictive performance of our approach over transfers of various sizes from the Information Sciences Institute (ISI) to ANL. We present the Moving Normalize Error Percentage (MNEP) and the MNEP of the Mean Square Error (MSE). Finally, we include the “last value” prediction as before. Table 2 shows the results from ISI to the University of Florida (UFL). Table 3 depicts the results from Lawrence Berkeley Laboratory (LBL) to ANL, and Table 4 presents results from LBL to UFL. In all cases we see error that is comparable to, or slightly lower than, the results presented in [27].

Finally, we validated our technique on data gathered as part of the Internet End-to-End Performance Monitoring (IEPM) [9] project at the Stanford Linear Accel-



**Figure 8. Comparison of Moving Normalized Error Percent (MNEP) of the Mean Absolute Error (MAE) between “Last Value” and multivariate forecasts.**

Measurement Target	Moving Normalized Error Pct.	MNEP (square error)	Last Value MAE MNEP
10MB-Transfers	0.148436	1.819804	1.002759
100MB-Transfers	0.077140	1.305875	1.024719
500MB-Transfers	0.107906	1.731765	1.045038

**Table 2. ISI to UFL – Normalized Error Percentages when using *NWS* data to forecast *GridFTP* [8] performance compared to the “last value” prediction (data from ANL [27]).**

erator Center (SLAC). This data consists of a variety of tests run from SLAC to a set of other sites over a period of roughly 6 months. The tests are performed roughly every 90 minutes and the timestamp is taken at the beginning of a testing regime (avoiding the problems of data matching mentioned in [27].) We compared the measurements taken using *iperf* with the performance of a *bbftp* [4] transfer. Table 5 shows the results for various datasets. Again, the “last value” prediction for the *bbftp* transfers is also presented. We note that in many cases, the *iperf* measurements are quite predictive of *bbftp* transfers using our approach.

## 5 Conclusion

We have described a general Grid performance prediction architecture as well as our implementation of

this predictive technique. We have shown a novel method for prediction based on the Cumulative Distribution Functions (*CDFs*) of two time-series and observed that its performance is promising – both on data that we have gathered and two independently collected data sets.

## 6 Acknowledgements

We would like to thank our colleagues at Argonne National Laboratory – Jennifer Schopf and Sudharshan Vazhkudai – for access to their data. Thanks also go to Les Cottrell and the IEPM team at SLAC for providing such a valuable resource.



Measurement Target	Moving Normalized Error Pct.	MNEP (square error)	Last Value MAE MNEP
10MB-Transfers	0.246149	2.002583	0.997348
100MB-Transfers	0.177217	1.945978	1.069952
500MB-Transfers	0.248009	2.191534	1.062821
1GB-Transfers	0.050055	1.163226	0.994377

**Table 3. LBL - ANL – Normalized Error Percentages when using NWS data to forecast GridFTP [8] performance compared to the “last value” prediction (data from ANL [27]).**

Measurement Target	Moving Normalized Error Pct.	MNEP (square error)	Last Value MAE MNEP
10MB-Transfers	0.171448	3.227998	1.024326
100MB-Transfers	0.251636	3.913424	1.044074
500MB-Transfers	0.123307	1.770330	0.981202

**Table 4. LBL - UFL – Normalized Error Percentages when using NWS data to forecast GridFTP [8] performance compared to the “last value” prediction (data from ANL [27]).**

## References

- [1] Abilene. <http://www.ucaid.edu/abilene/>.
- [2] B. Allcock, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Lee, A. Sim, A. Shoshani, B. Drach, and D. Williams. High-performance remote access to climate simulation data. In Proc. SC2001, Denver, Colorado, November 2001.
- [3] Active measurement project (AMP). <http://amp.nlanr.net>.
- [4] BBFTP. <http://doc.in2p3.fr/bbftp/>.
- [5] F. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao. Application level scheduling on distributed heterogeneous networks. In *Proceedings of Supercomputing 1996*, 1996.
- [6] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? IEEE Infocom, April 2001.
- [7] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., 1998.
- [8] GridFTP. <http://www.globus.org/datagrid/gridftp.html>.
- [9] Internet End-to-End Performance Measurement. <http://www-iepm.slac.stanford.edu/>.
- [10] IPerf. [dast.nlanr.net/Projects/Iperf/](http://dast.nlanr.net/Projects/Iperf/).
- [11] The nasa information power grid. <http://science.nas.nasa.gov/Pubs/NASnews/97/09/ipg.html>.
- [12] V. Jacobson. A tool to infer characteristics of internet paths. available from <ftp://ftp.ee.lbl.gov/pathchar>.
- [13] M. Mathis and J. Madhavi. Diagnosing internet congestion with a transport layer performance tool. In *Proceedings of INET '96*, 1996.
- [14] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communications Review*, 27(3), July 1997., 1997.
- [15] Net100. [www.net100.org](http://www.net100.org).
- [16] Netlogger. <http://www-didc.lbl.gov/NetLogger>.
- [17] V. Paxson. End-to-end internet packet dynamics. *IEEE/ACM Transactions on Networking*, 7(3):277–292, 1999.
- [18] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis. An architecture for large-scale internet measurement. *IEEE Communications*, 1988.
- [19] A. Petitet, S. Blackford, J. Dongarra, B. Ellis, G. Fagg, K. Roche, and S. Vadhiyar. Numerical libraries and the grid. In *(to appear) Proc. of SC01*, November 2001.
- [20] PingER. <http://www-iepm.slac.stanford.edu/pinger/>.
- [21] R. L. Ribler, J. S. Vetter, H. Simitci, and D. A. Reed. Autopilot: Adaptive control of distributed applications. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*, Aug 1998.
- [22] S. Shende, A. Malony, J. Cuny, K. Lindlan, P. Beckman, and S. Karmesin. Portable profiling and tracing for parallel scientific applications using c++, 1998.
- [23] N. Spring and R. Wolski. Application level scheduling: Gene sequence library comparison. In *Proceedings of ACM International Conference on Supercomputing 1998*, July 1998.
- [24] Surveyor. <http://www.advanced.org/surveyor/>.
- [25] M. Swamy and R. Wolski. Representing dynamic performance information in grid environments with the network weather service. In *2nd IEEE International Symposium on Cluster Computing and the Grid*, May 2002.

Measurement Target	Moving Normalized Error Pct.	MNEP (square error)	Last Value MAE MNEP
node1.cacr.caltech.edu	0.028043	0.228811	0.998498
node1.ccs.ornl.gov	0.038328	0.288154	1.000417
node1.cern.ch	0.208558	0.509544	0.986665
node1.clrc.ac.uk	0.192432	0.620723	1.144837
node1.dl.ac.uk	0.205493	0.511433	0.953166
node1.ece.rice.edu	0.217942	0.836363	1.135521
node1.jlab.org	0.077436	0.260612	0.987048
node1.kek.jp	0.108600	0.167450	0.996402
node1.lanl.gov	0.009649	0.160156	0.998672
node1.mcs.anl.gov	0.024630	0.134512	1.002156
node1.mib.infn.it	0.057261	0.107567	0.997080
node1.nersc.gov	0.087924	0.812037	0.981166
node1.nikhef.nl	0.424183	1.443438	0.882736
node1.nslabs.ufl.edu	0.072351	0.312970	1.002630
node1.rcf.bnl.gov	0.040560	0.570013	0.988761
node1.riken.go.jp	0.215419	0.432769	1.041622
node1.roma1.infn.it	0.387890	0.560886	0.931576
node1.sdsc.edu	0.042680	0.294245	0.995550
node1.stanford.edu	0.149007	0.489753	0.996682
node1.triumf.ca	0.147486	0.247430	1.023609
node1.utdallas.edu	0.295524	0.670600	1.067989

**Table 5. Normalized Error Percentages when using *iperf* [10] data to forecast *bbftp* [4] performance compared to the “last value” prediction (data from IEPM [9] Project).**

- [26] B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolski, and M. Swamy. A Grid Monitoring Architecture. Grid forum working group document, Grid Forum, February 2001. <http://www.gridforum.org>.
- [27] S. Vazhkudai and J. Schopf. Predicting sporadic grid data transfers. In *Proceedings 11th IEEE Symposium on High Performance Distributed Computing*, July 2002.
- [28] S. Vazhkudai, J. Schopf, and I. Foster. Predicting the performance of wide area data transfers. In *Proceedings of the 16th Int’l Parallel and Distributed Processing Symposium (IPDPS 2002)*, April 2002.
- [29] R. Wolski. Dynamically forecasting network performance using the network weather service. *Cluster Computing*, 1998. also available from <http://www.cs.utk.edu/rich/publications/nws-tr.ps.gz>.
- [30] R. Wolski, N. Spring, and J. Hayes. The network weather service: A distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems*, 1999.
- [31] R. Wolski, N. Spring, and J. Hayes. Predicting the cpu availability of time-shared unix systems on the computational grid. In *Proc. 8th IEEE Symp. on High Performance Distributed Computing*, 1999.
- [32] Y. Zhang, V. Paxson, and S. Shenker. The stationarity of internet path properties: Routing, loss, and throughput. *ACIRI Technical Report*, 2000.