Proceedings of HPDC-2002, July 2002

Predicting Sporadic Grid Data Transfers

Sudharshan Vazhkudai [1,2] Jennifer M. Schopf [1,3]
{vazhkuda, jms}@mcs.anl.gov

[1] Mathematics and Computer Science Division, Argonne National Laboratory
[2] Department of Computer and Information Sciences, The University of Mississippi
[3] Computer Science Department, Northwestern University

Abstract

The increasingly common practice of replicating datasets and using resources as distributed data stores in Grid environments has led to the problem of determining which replica can be accessed most efficiently. Because of diverse performance characteristics and load variations of several components in the end-to-end path linking these various locations, selecting a replica from among many requires accurate prediction information of the data transfer times between the sources and sinks.

In this paper we present a prediction system that is based on combining end-to-end application throughput observations and network load variations, capturing whole-system performance and variations in load patterns, respectively. We develop a set of regression models to derive predictions that characterize the effect of network load variations on file transfer times. We apply these techniques to the GridFTP data movement tool, part of the Globus Toolkit™, and observe performance gains of up to 10% in prediction accuracy when compared with approaches based on past system behavior in isolation.

Keywords: *Grids, data transfer prediction, replica selection.*

1. Introduction

As the coordinated use of distributed resources, or Grid computing, becomes more commonplace, basic resource usage is changing. Many recent applications use Grid systems as distributed data stores [DataGrid02, GriPhyN02, HSS00, LIGO02, MMR+01, NM02, TPW+00], where pieces of large datasets are replicated over several sites. For example, several high-energy physics experiments have agreed on a tiered Data Grid architecture [Holtman00, HJS+00] in which all data (approximately 20 petabytes by 2006) is located at a single Tier 0 site; various (overlapping) subsets of this data are located at national Tier 1 sites, each with roughly one-tenth the capacity; smaller subsets are cached at smaller Tier 2 regional sites; and so on. Therefore, any particular dataset is likely to have replicas located at multiple sites.

More often, these datasets are replicated for performance or proximity reasons. Variations in performance characteristics among these replica locations are bound to exist because of different architectures, network connectivity, network traffic, and system load. Thus, users may want to be able to determine the site from which particular datasets can be retrieved most efficiently, especially as datasets of interest tend to be large (1–1000 MB). It is this *replica selection* problem that we address in this paper.

One way a more intelligent replica selection can be achieved is by having replica locations expose performance information about past data transfers, which can then be used to predict future behavior between the sites involved. In previous work [VSF02], we examined the use of logs of past data transfers of large files to predict future behavior. The results had errors on average of about 20%, because of the sporadic nature of the transfers and the lack of current information about the network conditions.

In this paper we examine the use of another data stream, lightweight regular sensors, and combine this information with past observations for better predictions. We develop a predictive framework that combines infrequent but very accurate end-to-end GridFTP [AFN+01] file transfer data with frequent Network Weather Service [Wolski98] small probe data. Since these values are correlated, we use regressive techniques to combine the two data. We analyze several approaches that use different data filling techniques, and we show a 5–10% improvement in prediction error over a small wide-area testbed.

2. Related and Previous Work

Our goal is to obtain an accurate prediction of file transfer times between a storage system and a client. Achieving this can be challenging because numerous devices are involved in the end-to-end path between the source and the client, and the performance of each (shared) device along the end-to-end path may vary in unpredictable ways.

One approach to predicting this information is to construct performance models for each system component (CPUs at the level of cache hits and disk access, networks at the level of the individual routers, etc.) and then use these models to determine a schedule for all data transfers [SC00], similar to classical scheduling [Adve93, Cole89, Crovella99, ML90, CQ93, Schopf97, TB86, ZLP96]. In practice, however, it is often unclear how to combine this data to achieve accurate end-to-end measurements. Also, since system components are shared, their behavior can vary in unpredictable ways [SB98]. Further, modeling individual components in a system will not capture the significant effects these components have on each other, thereby leading to inaccuracies [GT99].

Alternatively, observations from past application performance of the entire system can be used to predict end-to-end behavior, which is typically what is of interest to the user. This technique is used by Downey [Downey97] and Smith et. al., [SFT98] to predict queue wait times and by numerous tools (Network Weather Service [Wolski98], NetLogger [NetLogger97], Web100 [Web100Project02], iperf [TF01], and Netperf [Jones02]) to predict the network behavior of small file transfers.

A substantial difference in performance can arise between a small NWS probe (lightweight with 64 KB size) and an actual file transfer using GridFTP (with tuned TCP buffers and parallelism). We show this in Figure 1, which depicts 64 KB NWS measurements that indicate that the bandwidth is about 0.3 MB/sec, and end-to-end GridFTP measurements, that indicate a significantly higher transfer rate. In this case, the NWS by itself is not sufficient to predict end-to-end GridFTP throughput. In addition, we see a much larger variability in GridFTP measurements, ranging from 1.5 to 10.2 MB/sec (due to different transfer sizes and also load variations in the end-to-end components), so that it is unlikely that a simple data transformation will improve the resulting prediction.

In [VSF02], we analyzed the use of GridFTP data in isolation by developing a series of predictors to predict transfer times. We observed a 15–24% error on average. While the log data used for the predictions reflected the end-to-end path accurately, the sporadic nature of large data transfers meant that often there was no data available about current conditions. A similar effect was addressed by Faerman et.al., [FSW+99] using the NWS and adaptive linear regression models for the Storage Resource Broker [BMR+98] and SARA [SARA02]. That work compared transfer times obtained from a raw bandwidth model (*Transfer-Time* = *ApplicationDataSize/NWS-Probe-Bandwidth*, with 64 KB NWS probes) with predictions from regression models and observed accuracy improvements ranging from 20% to almost 100% for the sites examined.

In this paper we consider similar techniques for GridFTP but extend the body of work by considering multiple data filling techniques (instead of throwing away the non-matching data, as described in Section 3.3.2) and regression models ranging from linear to quartic (polynomial) to improve prediction accuracy.

3. Predicting GridFTP Throughput by Using Regression

To obtain an accurate prediction for selecting replicas, we analyze the use of NWS bandwidth data in combination with GridFTP log data. In this section we describe the two monitoring approaches, an initial correlation test, and our regression techniques for predictions.

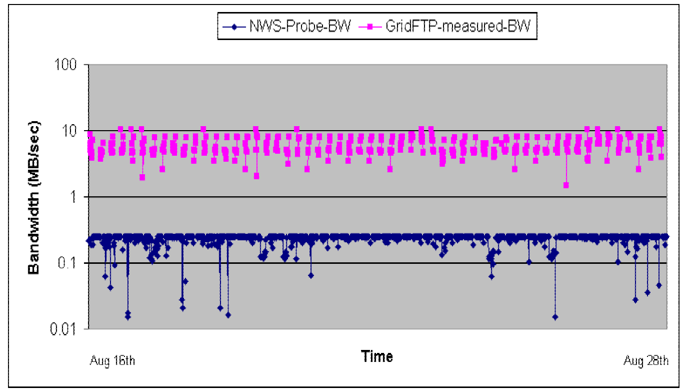


Figure 1: LBL-ANL GridFTP (approximately 400 transfers at irregular intervals) end-to-end bandwidth and NWS (approximately 1,500 probes every five minutes) probe bandwidth for the two-week August'01 dataset.

3.1. GridFTP and NWS Data Streams

GridFTP [AFN+01] is part of the Globus Toolkit™ [FK98] and is widely used as a secure, high-performance data transfer protocol [AFN+01, CFK+01, DataGrid02, GriPhyN02, SS01]. It extends standard FTP implementations with several features needed in Grid environments, such as security, parallel transfers, partial file transfers, and third party transfers.

We instrumented the GridFTP server to log the source address, file name, file size, number of parallel streams, stripes, TCP buffer size for the transfer, start and end timestamps, nature of the operation (read/write), and logical volume to and from which file was transferred [VSF02]. The GridFTP data measures the end-to-end application throughput, including component overheads, and is gathered only when a file is transferred between two sites.

The Network Weather Service monitors the behavior of various resource components by sending out probes at regular intervals [Wolski98]. NWS sensors exist for components such as CPU, disk, and network. Of interest to us is the network bandwidth sensor that uses small, lightweight, periodic probes (64 KB) to estimate the current network throughput.

3.2. Correlation

The first step in analyzing whether a combination of data streams will result in better predictions is to evaluate how highly correlated they are. The correlation coefficient is a measure of the linear relationship between two variables and can have a value between –1.0 and +1.0 depending on the strength of the relation. A coefficient near zero suggests that the variables may not be linearly related. However, they may exhibit nonlinear dependencies [Edwards84, OM88]. The correlation coefficient for GridFTP, (G), and NWS, (N), data is computed by using the formula

$$\text{corr} = \frac{\sum NG - (\sum N \sum G/\text{size})}{\sqrt{(\sum G^2 - (\sum G)^2/\text{size})}\ \sqrt{(\sum N^2 - (\sum N)^2/\text{size})}}$$

where "size" is the number of values in the data stream.

We compute the rank-order correlation for each of our two-week datasets. Rank correlation provides a distribution-free, nonparametric alternative to determine whether the observed correlation is significant. Rank correlation converts data to ranks by assigning a specific rank to each value in the data stream, as determined by the position of the value when the data stream is sorted.

Figure 2 shows a tabulated listing of the 95% confidence interval for the correlation coefficients. The confidence interval denotes that the correlation for 95% of the sample falls within a certain upper and lower limit and is determined by obtaining a Fisher transformation (a normal distribution) for the coefficient, finding the standard error for the distribution and then computing the interval [Edwards84]. From the figure, we can infer the presence of a moderate correlation between GridFTP and NWS data streams, encouraging the application of several regression models on these two datasets.

	Aug'01		Dec'01		Jan'02	
	Upper	*Lower*	*Upper*	*Lower*	*Upper*	*Lower*
LBL-ANL	0.8	0.5	0.5	0.3	0.6	0.2
LBL-UFL	0.7	0.5	0.7	0.4	0.6	0.1
ISI-ANL	0.8	0.5	0.6	0.4	0.7	0.3
ISI-UFL	0.9	0.4	0.6	0.2	0.5	0.1
ANL-UFL	0.5	0.2	0.6	0.2	0.6	0.1

Figure 2: 95% Confidence for the upper and lower limits of the rank-order correlation coefficient for the GridFTP and NWS datasets between four sites in our testbed. Denotes coefficients for our three datasets.

3.3. Regression Techniques and Algorithm

Regression uses various models to support relationships between datasets and is a powerful tool that can be used to derive predictions. Regression provides techniques to study the impact of the independent variable NWS (N), on the dependent variable GridFTP (G). Additional processing on the data must be done, however, to result in the one-to-one mapping expected by these techniques.

In Figure 3 we show the process we use to derive predictions. The key components are the two data sources (G and N), filling-in techniques, temporal filter, and the set of possible regression functions. Each dataset entry consists of a timestamp, the observed throughput value pair (T_G, G) for GridFTP and (T_N, N) for NWS.

3.3.1. Matching

In our datasets, two points from the two data sources rarely have the same timestamp. Therefore, before this data can be analyzed, the closest related pairs between the two data streams must be matched. For each GridFTP data point (T_G, G), we match a corresponding NWS data point (T_N, N), such that T_N is the closest to T_G, is established. By doing this, the pair (N_i, G_j) represents an observed end-to-end GridFTP bandwidth (G_j) resulting from a data transfer that occurred with the network probe value (N_i). At the end of the matching process the sequence looks like the following:

$$(N_i, G_j)(N_{i+1}, \_)\ldots(N_{i+(k-1)}, \_)(N_{i+k}, G_{j+1}),$$

where G_j, and G_{j+1} are two successive GridFTP file transfers and N_i, and N_{i+k} are NWS measurements that occurred in the same timeframe as the two GridFTP transfers. The sequence also consists of a number of NWS measurements between the two transfers for which there are no equivalent GridFTP values, such as $(N_{i+1}, \_)$.

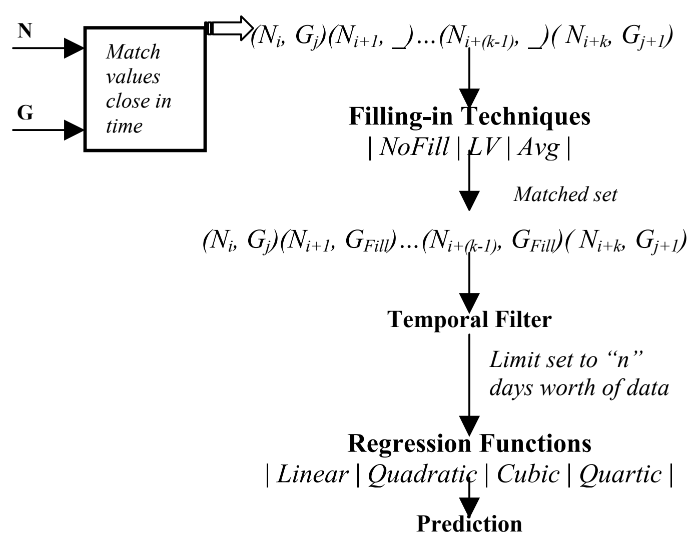
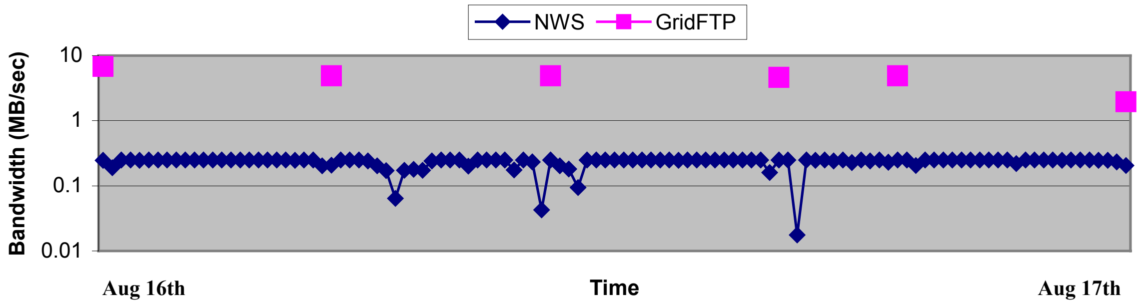


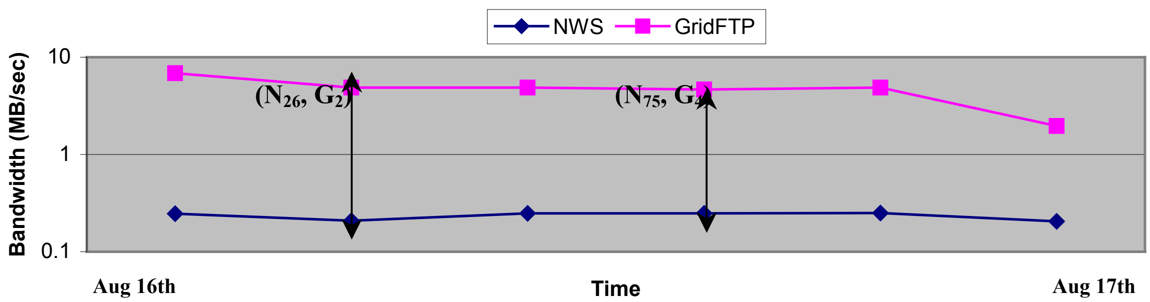
Figure 3: Algorithm for deriving predictions from GridFTP (G) and NWS (N) data streams by using regression techniques.

3.3.2 Filling-in Techniques

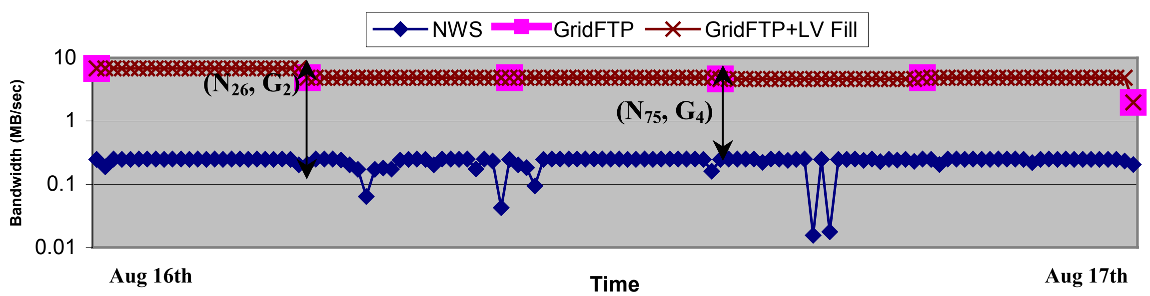
Two successive GridFTP transfers are almost always interspersed with many NWS values, given the nature of these datasets. In the matched sequence above, these values are represented as $(N_{i+1}, \_)\ldots(N_{i+(k-1)}, \_)$. Regression techniques expect a one-to-one mapping between NWS and GridFTP datasets, so we need mechanisms to compensate for the lack of sufficient GridFTP data. We use three techniques: NoFill, LV, and Avg.



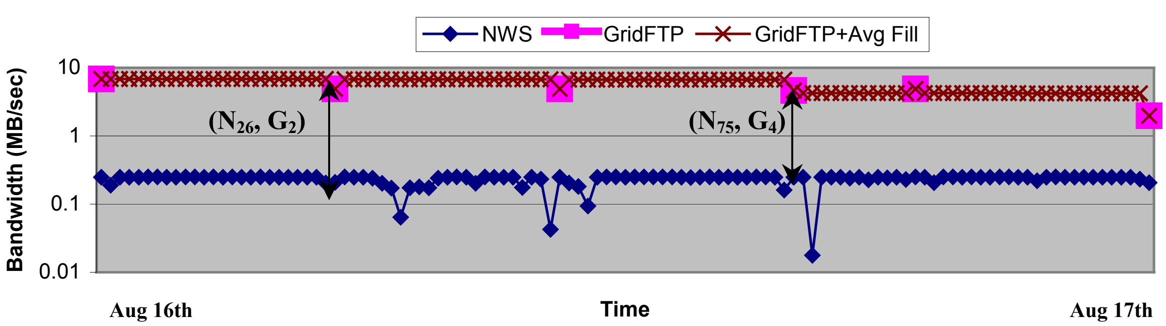
(a) Measured GridFTP and NWS



(b) NoFill



(c) Last Value Filling (LV)



(d) Average Filling (Avg)

Figure 4: (a) Six measured successive GridFTP transfers and NWS observations during those transfers between LBL and ANL (August 2001). (b) Discarding NWS values to match GridFTP transfers. Here (N_{26}, G_2) denotes that the 26th NWS measurement and the 2nd GridFTP transfer occur in the same timeframe. (C) Filling-in the last GridFTP value for NWS values between six successive file transfers. The graph follows a step function. Actual transfers are shown along with the filling. (d) Average of previous GridFTP transfers as a filling for NWS values.

I. **Discard unaccounted NWS data (NoFill):** With no-fill we simply omit the unmatched NWS data. The drawback of this approach, shown in Figure 4b, is that we throw away useful data. This was the approach used in [FSW+99].

II. **Last Value filling (LV):** In this approach we fill in the last GridFTP value for each unmatched NWS value, as shown in Figure 4c.

III. **Average filling (Avg):** This is similar to the LV approach except that, instead of the last GridFTP value, an average over the past day's worth of data transfers is computed and is used as a filling, as shown in Figure 4d.

3.3.3. Temporal Filters

Regression techniques can function over a variety of data sizes with differing results. We use a temporal filter to truncate the dataset, much in the same way as a sliding window is used in averages.

3.3.4 Regression Models

After this preprocessing, a set of pairs is fed to the regression function to calculate the coefficients necessary to obtain predictions. We use regression models ranging from linear to quartic to account for diverse relations between variables.

Linear regression attempts to build linear models between NWS (N) and GridFTP (G) data. We constructed a linear model between the two variables N and G as follows: $G^{|}=a+bN$, where $G^{|}$ is the prediction of the observed value of G for the corresponding value of N. The coefficients a and b are calculated based on a regression function that accounts for previous Ns and Gs, using the method of least squares. The regression coefficient a is calculated by using the formula

$$a = \text{Mean}(G) - b * \text{Mean}(N)$$

while the coefficient b is calculated by using the formula

$$b = \frac{\sum NG - (\sum N \sum G/\text{size})}{\sum G^2 - (\sum G)^2/\text{size}}$$

where "size" is the total number of values in the dataset [Edwards84].

To improve prediction accuracy, we also developed a set of nonlinear models adding polynomial terms to the linear equation. For instance, a quadratic model is as follows: $G^{|}=a+b_1N+b_2N^2$; cubic and quartic models have additional terms b_3N^3 and b_4N^4, respectively. Similar to the linear model the coefficients in quadratic, cubic, and quartic models b_2, b_3, and b_4 are computed by using the method of least squares. Adding polynomial terms to the regression model can reach a saturation point (no significant improvement in prediction accuracy observed), suggesting that a particular model sufficiently captures the relationship between the two variables [OM88, Pankratz91].

4. Results and Analysis

We evaluated the performance of our regression techniques on datasets collected over three distinct two-week durations: August 2001, December 2001, and January 2002. In the following sections we describe the experimental setup, prediction error calculations, and our results obtained from these datasets.

4.1. Experimental Setup

Our experiments comprised controlled GridFTP transfers and NWS network sensor measurements between four sites in our testbed: Argonne National Laboratory (ANL), the University of Southern California Information Sciences Institute (ISI), Lawrence Berkeley National Laboratory (LBL) and the University of Florida at Gainesville (UFL).

GridFTP experiments included transfers comprising several file sizes ranging from 10 MB to 1 GB, performed at random time intervals within 12-hour periods. These transfers were performed with tuned TCP buffer settings (1 MB) and eight parallel streams to achieve enhanced throughput. Logs of these transfers were maintained at the respective sites and can be found at [Vazhkudai02]. In our previous work [VSF02] we observed that GridFTP throughput varied with transfer file sizes, and thus we grouped several file sizes into categories: 0–50 MB as 10M, 50–250 MB as 100M, 250–750 MB as 500M, and more than 750 MB as 1G, based on the achievable bandwidth, for the sites we examined. Our results in the next section are based on these settings.

Configuring NWS among a set of resources involves setting up a nameserver and memory to which sensors at various sites can register and log measurements [Wolski98]. In our experiments, we used ANL as a registration and memory resource. NWS network monitoring sensors between these sites were setup to measure bandwidth every five minutes with 64 KB probes.

The accuracy of a regression function depends on the size of the dataset, which can be minimal initially. For this reason we use a training set of 15 GridFTP and NWS data points so the regression function can adjust.

4.2. Performance

In this section we discuss the performance of our regressive techniques, compare the various approaches used to account for network data, compare linear and nonlinear models, and analyze the effect of window sizes on prediction error. We use our August 2001 dataset to illustrate these points. Complete results for all our datasets can be found at [Vazhkudai02].

We calculate the prediction accuracy using the normalized percentage error calculation

$$\% \text{ Error} = \frac{\sum | \text{Measured}_{BW} - \text{Predicted}_{BW} |}{(\text{size} * \text{Mean}_{BW})} * 100$$

where "size" is the total number of predictions and the Mean_{BW}, is the average measured GridFTP throughput.

In Figure 5, we show the average performance (based on all transfer sizes) for our predictors. We compare the normalized percent errors for predictors based on past GridFTP behavior and predictors based on linear regression between our various site pairs. For our datasets, we consistently observed a 5 to 10% improvement in prediction accuracy when regression techniques with LV or AVG filling were used. Regression with NoFill provides us with no significant improvement when compared with past predictors.

	Moving Avg	Regression NoFill	Regression LV	Regression Avg
LBL-ANL	24.4%	22.4%	20.6%	20%
LBL-UFL	15%	18.8%	11.1%	11%
ISI-ANL	15%	12%	9.5%	9%
ISI-UFL	21%	21.9%	16%	14.5%
ANL-UFL	20%	21%	20%	16%

Figure 5: Average normalized percentage errors for GridFTP predictions based on (1) past GridFTP behavior using moving average predictor (previous work), (2) regression with no filling, (3) regression with last value fills, and (4) regression with average fills. All error rates are for linear regression models

Figure 6, depicts the regression results between LBL and ANL for the 100M category. The figure shows how predictions using Avg fill regression and past moving average closely follow the measured values. Corresponding error rates for the Avg and PastMavg are shown in Figure 7, where we can see at least a 5% improvement in accuracy.

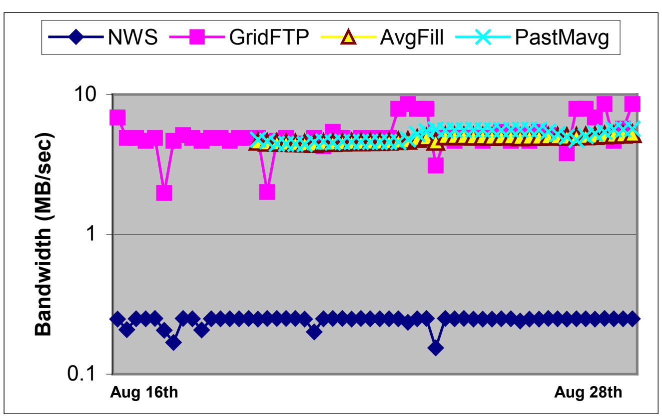


Figure 6: NWS, measured GridFTP, and GridFTP predictions using linear regression with average filling and predictions using the moving average past predictor for 100 MB file transfers between LBL and ANL. Predictions include an initial training set of 15 values.

Figures 7 through 10 study the effect of filling in techniques for various transfer sizes. We show our predictions for various site pairs and for several file sizes, highlighting the fact that our predictors work well for several transfer sizes. For almost all transfer sizes, filling-in techniques performed better than discarding network data. We observe error rate improvements of up to 10% when we use last value (LV) or average (Avg) filling as against simply discarding (NoFill) NWS data or using past predictors.

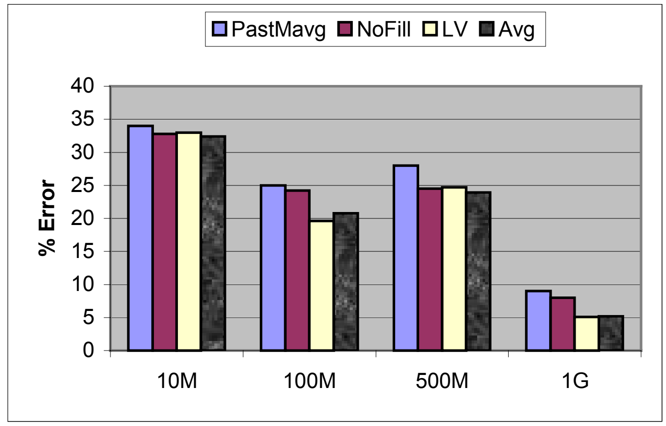


Figure 7: Prediction accuracy using filling-in techniques for all file transfers between LBL and ANL.

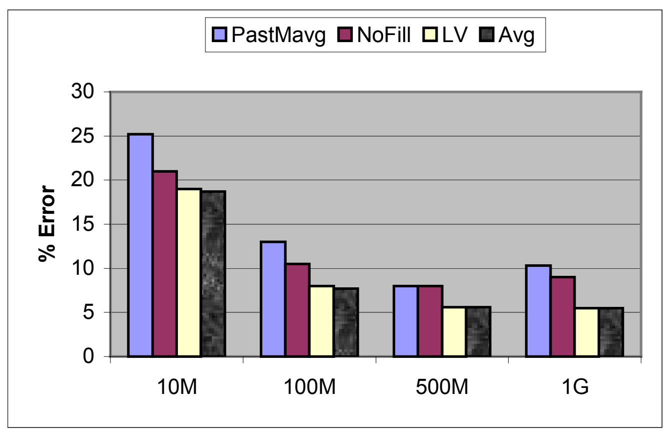


Figure 8: Prediction accuracy using filling-in techniques for all file transfers between ISI and ANL.

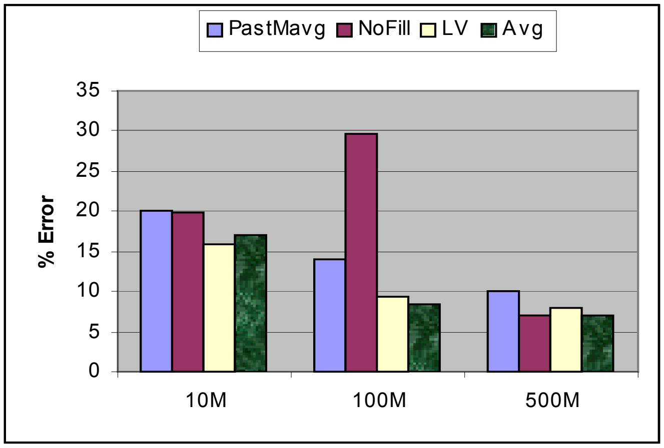


Figure 9: Prediction accuracy using filling-in techniques for all file transfers between LBL and UFL.

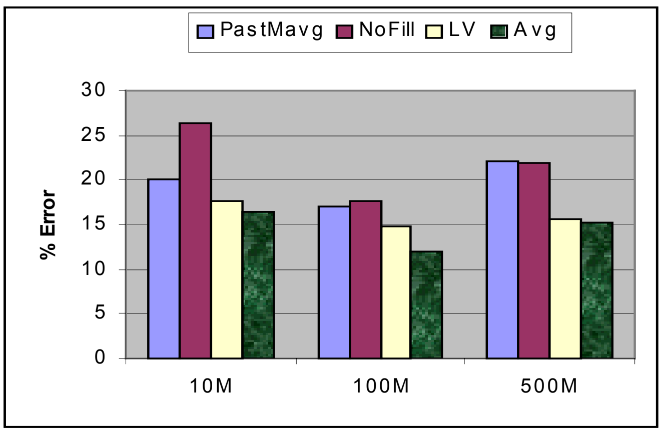


Figure 10: Prediction accuracy using filling-in techniques for all file transfers between ISI and UFL.

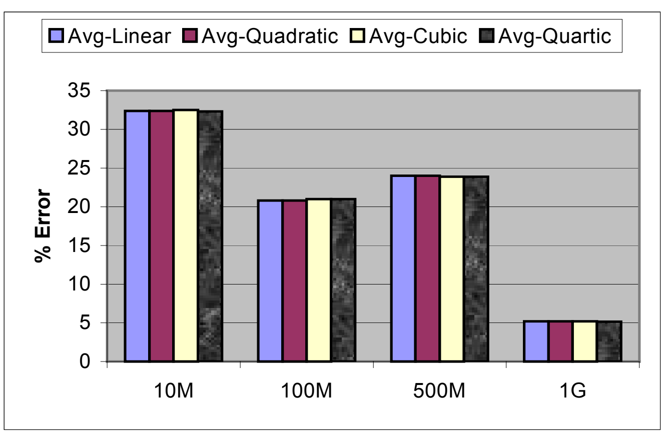


Figure 11: Average filling (LBL-ANL) for the linear, quadratic, cubic, and quartic regression models.

For our datasets, we observed no noticeable improvement in prediction accuracy by using polynomial models for our site pairs. Figure 11 shows the performance of linear, quadratic, cubic, and quartic regression models for various transfer sizes between LBL and ANL with Avg filling. All our models performed similarly.

We also studied the impact of different window sizes on regression error rates. In general, we observed that regression functions perform better with more data. Figure 12 depicts regression with Avg filling over five days, over ten days, and over all the data. We noticed no substantial improvements, this is likely due to the fact that our datasets were collected over short durations.

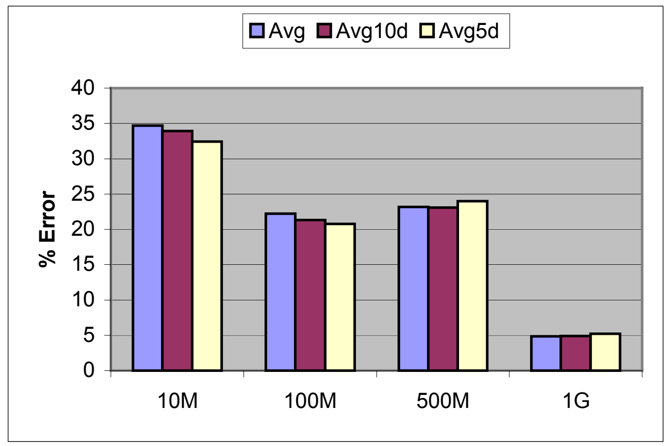


Figure 12: Average filling (LBL-ANL) with temporal windows of 5 days (Avg5d), 10 days (Avg10d) and over all data (Avg).

5. Conclusion

In this paper we have described the need for predicting the performance of GridFTP data transfers in the context of replica selection in Data Grids. This extends our previous work predicting transfer times based on past GridFTP behavior alone by extending it to compensate for the spondaic nature of large data transfers

We used regressive techniques between GridFTP and NWS network data streams to generate better predictions. By including information about current conditions, we saw a significant improvement in the resulting predictions. We used several data-filling techniques (NoFill, LV and Avg) and developed a set of regression models (linear, quadratic, cubic and quartic) to account for the relations between GridFTP and NWS data. With this approach, we obtained a 5-10% increase in prediction accuracy when compared with predictions based on only past GridFTP behavior. For our datasets, nonlinear regression offered no significant benefits.

As a next step, we plan to examine the effects of disk I/O load on application throughput using similar techniques explained in this paper.

Acknowledgments

We thank all the system administrators of our testbed sites for their valuable assistance. We also thank Matthew Knepley of Argonne National Laboratory for several useful discussions. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under contract W-31-109-Eng-38, and by NSF Awards ACI-00933000 and PACI#763-04/ACI-9619019.

References

[Adve93] V. S. Adve, Analyzing the Behavior and Performance of Parallel Programs, Ph.D. Thesis, Technical Report TR1201, Department of Computer Science, University of Wisconsin, December 1993.

[AFN+01] W. Allcock, I. Foster, V. Nefedova, A. Chevrenak, E. Deelman, C. Kesselman, A. Sim, A. Shoshani, B. Drach, and D. Williams, High-Performance Remote Access to Climate Simulation Data: A Challenge Problem for Data Grid Technologies, *Proceedings of Supercomputing (SC'01)*, November 2001.

[BMR+98] C. Baru, R. Moore, A. Rajasekar, and M. Wan, The SDSC Storage Resource Broker, *Proceedings of CASCON'98,* 1998.

[CFK+01] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets, *Journal of Network and Computer Applications*, 23:187–200, 2001.

[Cole89] M. Cole, Algorithmic Skeletons: Structured Management of Parallel Computation, *Pitman/MIT Press*, 1989.

[CQ93] M. J. Clement and M. J. Quinn, Analytical Performance Prediction on Multicomputers, *Proceedings of Supercomputing '93*, 1993.

[Crovella99] M.E. Crovella, Performance Prediction and Tuning of Parallel Programs, Ph.D. Thesis, University of Rochester, 1999.

[DataGrid02] The European Data Grid Project, http://www.eu-datagrid.org.

[Downey97] A. Downey, Queue Times on Space-Sharing Parallel Computers, *Proceedings of the 11th International Parallel Processing Symposium*, 1997.

[Edwards84] A. L. Edwards, *An Introduction to Linear Regression and Correlation,* W.H. Freeman and Company, 1984.

[FK98] I. Foster and C. Kesselman, The Globus Project: A Status Report, *Proceedings of IPPS/SPDP '98 Heterogeneous Computing Workshop*, pp. 4–18, 1998.

[FSW+99] M. Faerman A. Su, R. Wolski, and F. Berman, Adaptive Performance Prediction for Distributed Data-Intensive Applications, *Proceedings of the ACM/IEEE SC99 Conference on High Performance Networking and Computing*, Portland, Oregon, November 1999.

[GriPhyN02] The GriPhyN Project, http://www.griphyn.org.

[GT99] J. Geisler and V. Taylor, Performance Coupling: Case Studies for Measuring the Interactions of Kernels in Modern Applications. *Proc. SPEC Workshop on Performance Evaluation with Realistic Applications,* April 1999.

[Holtman00] K. Holtman, Object Level Replication for Physics, *Proceedings of 4th Annual Globus Retreat*, Pittsburgh, July 2000.

[HJS+00] W. Hoschek, J. Jaen-Martinez, A. Samar, and H. Stockinger, Data Management in an International Grid Project, *2000 International Workshop on Grid Computing (GRID 2000)*, Bangalore, India, December 2000.

[HSS00] M. Hafeez, A. Samar, and H. Stockinger, A Data Grid Prototype for Distributed Data Production in CMS, *7th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT2000),* October 2000.

[Jones02] R. Jones, The Public Netperf Homepage, http://www.netperf.org/netperf/NetperfPage.html

[LIGO02] The LIGO Experiment, http://www.ligo.caltech.edu/.

[ML90] V. W. Mak and S. F. Lundstrom, Predicting the Performance of Parallel Computations, IEEE Transactions on Parallel and Distributed Systems, pp. 106–113, IEEE Computer Society Press, July 1990.

[MMR+01] D. Malon, E. May, S. Resconi, J. Shank, A. Vaniachine, T. Wenaus, and S. Youssef, Grid-enabled Data Access in the ATLAS Athena Framework, *Proceedings of Computing and High Energy Physics 2001 (CHEP'01) Conference*, 2001.

[NetLogger97] NetLogger: A Methodology for Monitoring and Analysis of Distributed Systems, http://www-didc.lbl.gov/NetLogger.

[NM02] H. Newman and R. Mount, The Particle Physics Data Grid, www.cacr.caltech.edu/ppdg.

[OM88] B. Ostle and L. C. Malone, *Statistics in Research,* Iowa State University Press, 1988.

[Pankratz91] A. Pankratz, *Forecasting with Dynamic Regression Models,* John Wiley & Sons Inc., 1991.

[SARA02] SARA: The Synthetic Aperture Radar Atlas, http://sara.unile.it/sara/

[SB98] J. M. Schopf and F. Berman, Performance Prediction in Production Environments, *Proceedings of IPPS/SPDP '98*, 1998.

[SC00] X. Shen and A. Choudhary, A Multi-Storage Resource Architecture and I/O, Performance Prediction for Scientific Computing, *Proceedings of the 9th IEEE Symposium on High Performance Distributed Computing,* pp. 21–30. IEEE-Press, 2000.

[Schopf97] J.M. Schopf, Structural Prediction Models for High Performance Distributed Applications, *Proceedings of the Cluster Computing Conference (CCC '97)*, March 1997.

[SFT98] W. Smith, I. Foster, and V. Taylor, Predicting Application Run Times Using Historical Information, *Proceedings of the IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, 1998.

[SS01] A. Samar and H. Stockinger, Grid Data Management Pilot (GDMP): A Tool for Wide Area Replication, *IASTED International Conference on Applied Informatics (AI2001)*, Innsbruck, Austria, February 2001.

[TB86] A. Thomasian and P. F. Bay, Analysis Queuing Network Models for Parallel Processing of Task Systems, *IEEE Transactions on Computers c-35* 12, December 1986.

[TF01] A. Tirumala, J. Ferguson, Iperf 1.2 – The TCP/UDP Bandwidth Measurement Tool, http://dast.nlanr.net/Projects/Iperf/, May 2001.

[TPW+00] I. Terekhov, R. Pordes, V. White, L. Lueking, L. Carpenter, H. Schellman, J. Trumbo, S. Veseli, and M. Vranicar, Distributed Data Access and Resource Management in the D0 SAM System, *Proceedings of HPDC 2000,* San Francisco, August 2000.

[Vazhkudai02] S. Vazhkudai, GridFTP Predictor Trace Data, http://www.mcs.anl.gov/~vazhkuda/Traces.

[VSF02] S. Vazhkudai, J.M. Schopf, and I. Foster, Predicting the Performance of Wide Area Data-Transfers, *Proceedings of the 16th Int'l Parallel and Distributed Processing Symposium (IPDPS 2002)*, to appear.

[Web100Project02] The Web100 Project, http://www.web100.org/

[Wolski98] R. Wolski, Dynamically Forecasting Network Performance Using the Network Weather Service, *Journal of Cluster Computing*, 1: 119-132, January, 1998.

[ZLP96] M. J. Zaki, W. Li, and S. Parthasarathy, Customized Dynamic Load Balancing for Network of Workstations, *Proceedings of HPDC'96*, 1996.