# Using Disk Throughput Data in Predictions of End-to-End Grid Data Transfers

Sudharshan Vazhkudai [1,2] Jennifer M. Schopf [1]
[1] Mathematics and Computer Science Division, Argonne National Laboratory
[2] Department of Computer and Information Science, The University of Mississippi
{vazhkuda, jms}@mcs.anl.gov

**Abstract.** Data grids provide an environment for communities of researchers to share, replicate, and manage access to copies of large datasets. In such environments, fetching data from one of the several replica locations requires accurate predictions of end-to-end transfer times. Predicting transfer time is significantly complicated because of the involvement of several shared components, including networks and disks in the end-to-end data path, each of which experiences load variations that can significantly affect the throughput. Of these, disk accesses are rapidly growing in cost and have not been previously considered, although on some machines they can be up to 30% of the transfer time. In this paper, we present techniques to combine observations of end-to-end application behavior and disk I/O throughput load data. We develop a set of regression models to derive predictions that characterize the effect of disk load variations on file transfer times. We also include network component variations and apply these techniques to the logs of transfer data using the GridFTP server, part of the Globus Toolkit™. We observe up to 9% improvement in prediction accuracy when compared with approaches based on past system behavior in isolation.

## 1 Introduction

Increasingly, scientific discovery is driven by computationally intensive analyses of massive data collections. This recent trend has encouraged the research and development of sophisticated infrastructures for maintaining large data collections in a distributed, secure fashion and for improving the rapid access of large subsets of data files.

One example is in high-energy physics experiments, such as ATLAS [MMR+01] and CMS [HSS00], that have agreed on a tiered architecture [HJS+00, Holtman00] for managing and replicating the petascale data generated by the LHC experiment at CERN beginning 2006. The current architecture proposes to manage these petabytes of data, generated at CERN (Tier 0), by replicating subsets (approximately an order of magnitude reduction) across national (Tier 1) and regional (Tier 2) centers.

As data grid environments begin to be deployed and used, the amount of replication of data will likely grow rapidly as more users cache copies of datasets nearby for

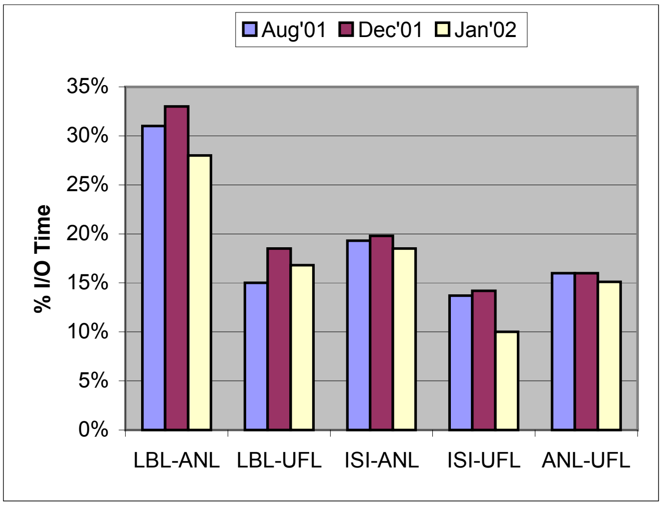



**Figure. 1.** Disk I/O time as a percentage of the total data transfer time for our experiments. Sites include Argonne National Laboratory (ANL), Lawrence Berkeley National Laboratory (LBL), University of Southern California's Information Sciences Institute (ISI), and University of Florida at Gainesville (UFL). Transfers include several file sizes ranging from 10 MB to 1 GB. Transfers were conducted over three distinct two-week periods.

better performance. Thus, a particular copy of a dataset will reside at multiple locations, and a choice of site to retrieve it from must be made.

In previous work [VS02, VSF02], we addressed this replica selection problem by having replica locations expose transfer performance estimates. Estimates were derived from past history of transfers (using the GridFTP server, part of the Globus Toolkit™) between sources and sinks, and by also factoring in the network link load to account for the sporadic nature of data grid transfers using regressive techniques. Our results showed prediction accuracy hovered around 15-24% error for predictors solely based on past transfer behavior, but improved 5-10% when network load variations were factored in. In this paper, we consider the effects of disk I/O as well.

The addition of disk I/O behavior in our predictions is motivated by three main factors: (1) disk I/O currently plays a large role, up to 30% on our testbed, in large data transfer times (as detailed below); (2) this role will become more important as a result of trends in disk size and network behavior; (3) and having access to additional data streams become more important as Grid environments grow, and not all resources will have the same information available about them.

In fact, we observe that disk I/O can account for up to 30% of the transfer time. In Figure 1, we show the percentage of I/O time spent on an average data transfer. We compare the cost of performing a local GridFTP read/write (source disk to device abstraction at source, essentially eliminating the network) with the wide-area transfer cost (source disk to device abstraction at sink). For these experiments, the disks on the source ends were all high-end RAID servers. On lower-end disk systems the effect would be even more significant.

In addition to current behavior, trends in disk storage and networking suggest that disk I/O will play an even larger role in the future. Disk capacity is increasing at the rate of about 100x per decade [GS00]. The ratio between disk capacity and disk throughput, however, is increasing at only 10x per decade, indicating that storage capacity is far outpacing disk speeds. Further, Gilder's law predicts that network bandwidth will triple every year for the next 25 years [GS00], so both network throughput and storage capacity are outpacing advances in disk speeds. Therefore, as link speeds increase, the network latency significantly drops and disk accesses are likely to become the bottleneck in large file transfers across the Grid.

In addition to the proportionality of the disk I/O time to the full transfer time, we must consider that data grids are potentially highly dynamic, with resources joining and leaving communities. The availability of data sources (required for obtaining forecasts) can also vary unpredictably as a result of failures in the various components, monitoring sensors and so forth. Thus, we need to be able to derive forecasts from several combinations of "currently available" data sources. For example, we can build predictions by using (1) just past GridFTP transfer logs, (2) transfer logs combined with current network load observations, (3) transfer logs with disk I/O load data, or (4) a combination of past transfer logs, network, and disk load traces. In our previous work, we investigated (1) and (2), this paper explores techniques to derive predictions for the (3) and (4).

In this paper, we extend our previous work to combine transfer log data with disk throughput data. Specifically, we develop multiple regression models, deriving predictions from past transfer logs, disk I/O, and network load data combined. Our results denote an improvement in prediction accuracy of up to 4% when using regression techniques between GridFTP transfers and disk I/O throughput data when compared with predictions based on past GridFTP behavior in isolation; we achieved a 9% improvement when combining all three data sources.

In the remainder of the paper we present related and previous work (Section 2), our prediction model (in Section 3), an evaluation of our techniques (in Section 4), and suggestions for future work (Section 5).

## 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, CQ93, Crovella99, ML90, 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 [NetLogger02], Web100 [Web100Project02], iperf [TF01], and Netperf [Jones02]) to predict the network behavior of small file transfers. We used this technique in [VSF02] but found that it had large errors because of the sporadic nature of GridFTP transfers and that we needed to be able to include additional data about current system conditions in order to improve the predictions.

In our previous work [VS02], we combined end-to-end throughput observations from past GridFTP data transfers and current network load variations, using regression models to obtain better predictions. Faerman et al. [FSW+99] addressed the issue 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. Swany and Wolski have also approached the problem by constructing cumulative distribution functions (CDF) of past history and deriving predictions from them as an alternative to regressive models. This has been demonstrated for 16 MB HTTP transfers with improved prediction accuracy when compared with their univariate prediction approach [SW02].

## 3 Prediction Model

In this section, we examine the various data sources we used, their relations, regressive models and our prediction algorithm.

### 3.1 Data Transfer Logs and Component Data

Our three data sources are GridFTP, NWS and iostat. We use a GridFTP server [AFN+01] to perform the data transfers and to log the behavior every time a transfer is made, thereby recording the end-to-end transfer behavior. Since these events are very sporadic, however, we also need to capture data about the current environment to have accurate predictions. We use the iostat disk throughput data to measure disk behavior and the Network Weather Service network probe data as an estimate of bandwidth for small data transfers.

GridFTP [AFN+01] is part of the Globus Toolkit™ [FK98, Globus02] and is widely used as a secure, high-performance data transfer protocol [ACF+02, AFN+01, DataGrid02, GriPhyN02]. 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 GT 2.0 wuftp-based 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/from which file was transferred. [VSF02].

The iostat tool is part of the sysstat [SYSSTAT02] system-monitoring suite. It collects disk I/O throughput data. Iostat can be configured to periodically monitor disk transfer rates, block read/write rates and so forth of all physically connected disks. We are particularly interested in the disk transfer rate that represents the throughput of a disk.

The Network Weather Service [Wolski98] monitors the behavior of various resource components by sending out lightweight probes or querying system files at regular intervals. NWS sensors exist for components such as CPU, disk, and network. We used the network bandwidth sensor with 64 KB probes to estimate the current network throughput.

In subsequent sections, we see how forecasts can be derived from these correlated data streams by using regressive techniques.

### 3.2 Correlation

Correlation gives a measure of the linear strength of the relationship between two variables and is often used as a test of significance before linear regression analysis is performed [Edwards84]. For our data sources, namely, GridFTP logs, iostat load, and NWS traces we computed rank-order correlation (a distribution free test) and observed moderate correlation between the variables from 0.2 to 0.7 (95% confidence interval for the correlation).

### 3.3 Algorithm

Our three data sources (GridFTP, disk I/O, and NWS network data) are collected exclusive of each other and rarely had same timestamps. To use common statistical techniques on the data streams, however, we need to line up the values to be considered. Hence, we must match values from these three sets such that, for each GridFTP value, we find disk I/O and network observations that were made around the same time.

For each GridFTP data point $(T_G, G)$, we match a corresponding disk load $(T_D, D)$ and NWS data point $(T_N, N)$ such that $T_N$ and $T_D$ are the closest to $T_G$. By doing this, the triplet $(N_i, D_j, G_k)$ represents an observed end-to-end GridFTP throughput ($G_k$) resulting from a data transfer that occurred with the disk load ($D_j$) and network probe value ($N_i$). At the end of the matching process the sequence looks like the following:

$$(N_i, D_j, G_k)(N_{i+1}, D_{j+1}, \_)\ldots(N_{i+m}, D_{j+m}, G_{k+1}),$$

where $G_k$, and $G_{k+1}$ are two successive GridFTP file transfers, $N_i$ and $N_{i+m}$ are NWS measurements, and $D_j$ and $D_{j+m}$ are disk load values that occurred in the same time-frame as the two GridFTP transfers. The sequence also consists of a number of disk load and NWS measurements between the two transfers for which there are no equivalent GridFTP values, such as $(N_{i+1}, D_{j+1}, \_)$. Note that these interspersed network and disk load values also need to be time aligned, as they seldom have same timestamps.

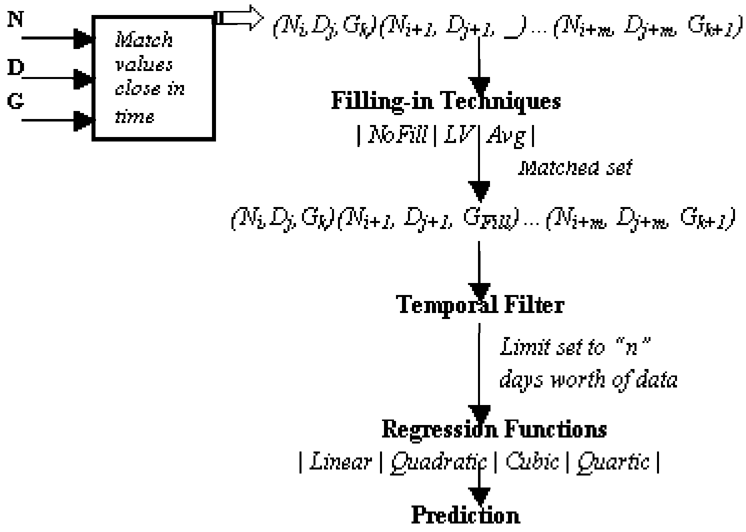



**Figure. 2.** Algorithm for deriving predictions from GridFTP (G), disk load (D), and NWS (N) data streams by using regression techniques.

After matching the data streams we need to address the tuples that do not have G values. Because of the sporadic nature of data grid transfers we will have more disk I/O and network data than GridFTP data. Regression models expect a one-to-one mapping between the data values, so we can either discard unaccounted network and I/O data (for which there are no equivalent GridFTP data) or fill in synthetic transfer values for the unaccounted data. We use three strategies to fill in missing values. These filling-in strategies are as follows: discard unaccounted disk I/O and network data (NoFill), use last GridFTP transfer values as a filling (LV) for unaccounted data, and use average of previous transfers as a filling (Avg) for unaccounted data. After the G values are filled in, these datasets are fed to the regression models (Figure 2).

### 3.4 Regressive Techniques

To predict the end-to-end GridFTP throughput and study the effect of the disk I/O component, we use standard regressive techniques. Regression provides the necessary mechanisms to analyze the impact of several independent variables (in our case, I/O traces or NWS bandwidth data) on the dependent variable (GridFTP throughput).

### 3.4.1 Simple Regression

In our previous work, we developed simple regression techniques between GridFTP (G) and NWS network data (N). We built a set of linear and nonlinear regression models between the two variables and derived forecasts from it. In this paper, we employ similar techniques to analyze the effect of disk I/O variations (D) on end-to-end GridFTP bandwidth. We construct a linear model between two variables D and G as follows: $G^{|}=a+bD$, where $G^{|}$ is the prediction of the observed value of G for the corresponding value of D. The coefficients, a and b, are calculated based on a regression function that accounts for previous Ds and Gs, using the method of least squares,

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

while the coefficient b is calculated by using the formula

$$b = \frac{\sum DG - (\sum D \sum G/\text{size})}{\sum G^2 - (\sum G)^2/\text{size}},$$

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

### 3.4.2 Multiple Regression

In addition to simple regression, we study the effect of deriving predictions from all three data sources. For this purpose, we construct multiple regression strategies that allow us to study the effect of several independent variables on a dependent variable.

We construct multiple regression models by adding terms corresponding to various components to the simple regression equation. Similar to the disk component discussed earlier, to include network variations into the equation, we add a network load term. Thus, the multiple regression model is as follows: $G^{|}=a+b_1D+b_2N$, where $G^{|}$ is the prediction of the observed value of G for the corresponding values of N and D. The regression coefficients are calculated [Edwards84] as follows

$$a = \text{Mean}(G) - (b_1 * \text{Mean}(D)) - (b_2 * \text{Mean}(N))$$

$$b_1 = \frac{(\sum DG \sum N^2) - (\sum NG \sum DN)}{(\sum D^2 \sum N^2) - (\sum DN)^2}$$

$$b_2 = \frac{(\sum NG \sum D^2) - (\sum DG \sum DN)}{(\sum D^2 \sum N^2) - (\sum DN)^2}$$

Including further components (which contribute to the end-to-end data path) would mean adding terms to the multiple regression equation, whose coefficients can then be computed by using the method of least squares [Edwards84]. To summarize, we are interested in predicting the performance of the dependent variable, GridFTP, by studying the impact of adding independent components such as disk and network link loads to the regression model.

# 4 Evaluation

To analyze the performance of our predictors, we conducted several wide-area experiments between our testbed sites comprising resources from Argonne National Laboratory (ANL), Lawrence Berkeley National Laboratory (LBL), University of Southern California's Information Sciences Institute (ISI) and University of Florida at Gainesville.

First, we set up GridFTP experiments between these sites, transferring files ranging from 10 MB to 1 GB at random intervals in twelve-hour durations for a two-week period (during August 2001, December 2001 and January 2002). All transfers were made with tuned TCP buffers size of 1 MB and eight parallel streams. Disk I/O throughput data was collected by using the iostat tool logging transfer rates every five minutes. NWS was set up to monitor network bandwidth between these sites at five-minute intervals using 64 KB probes. All logs were maintained at the respective sites.

We analyze the performance of our regressive techniques in the following cases: (1) regression between GridFTP transfer data and disk I/O trace data, and (2) regression between GridFTP, disk I/O, and NWS network data. We compare the results from these approaches with predictions based on GridFTP data in isolation [VSF02] and predictions based on regressing GridFTP and NWS data [VS02]. In all of the above, we compare several of our filling strategies.

## 4.1 Metrics

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 is the average measured GridFTP throughput. We show our results based on the August 2001 dataset. Results for all our datasets can be found at [Traces02].

In addition to evaluating the error of our predictions, we evaluate information about the variance. Depending on the use case, a user may be more interested in selecting a site that has reasonable performance bandwidth estimates with a relatively low prediction error instead of a resource with higher performance estimates and a possibly much higher error in prediction. In such cases, it can be useful if the forecasting error can be stated with some confidence and with a maximum/minimum variation range. These limits can also, in theory, be used as catalysts for corrective measures in case of performance degradation.

In our case, we can also use these limits to verify the inherent cost of accuracy of the predictors. Comparing the confidence intervals of these prediction error rates, we can determine whether the accuracy achieved is at the cost of greater variability, in which case there is little gain in increasing the component complexity of our prediction approach.

Thus, for any predictor (for any site pair), the information denoted by the following triplet can be used as a metric to gauge its accuracy:

*Accuracy-Metric = [Throughput, % Error-Rate, Confidence],*

where *Throughput* is the predicted GridFTP value (higher the better), with a certain percentage error (lower the better) and a percentage confidence interval (smaller the better). Interested parties can use a function of this accuracy metric to choose one site from the other.

### 4.2 Results

Table 1 presents the average normalized percent error based on all transfers for the site pairs we examined. They are classified as follows: MovingAvg corresponds to prediction based on GridFTP in isolation [VSF02]; G+N corresponds to regression between GridFTP and NWS network data [VS02]; G+D corresponds to regression between GridFTP and disk I/O; and G+N+D corresponds to regressing all three datasets. We have shown all results in the interest of continuity.

**Table 1.** Normalized percent prediction error rates for the various site pairs for the August 2001 dataset. The figure denotes four categories: (1) prediction based on GridFTP data in isolation (Moving Avg), (2) regression between GridFTP and NWS network data with the three filling in techniques (G+N), (3) regression between GridFTP and disk I/O data with the three filling in techniques (G+D), and (4) regression based on all three data sources (G+N+D). Shaded portions indicate a comparison between our approaches.

| | GidFTP Logs [VSF02] | Linear Regression between GridFTP Logs and Network Load [VS02] | | | Linear Regression between GridFTP Logs and Disk Load | | | Linear Regression Using All Three Data Sources | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Moving Avg | G+N NoFill | G+N LV | G+N Avg | G+D NoFill | G+D LV | G+D Avg | G+N+D NoFill | G+N+D LV | G+N+D Avg |
| LBL-ANL | 24.4 | 22.4 | 20.6 | 20 | 25.2 | 21.7 | 21.4 | 22.3 | 17.7 | 17.5 |
| LBL-UFL | 15 | 18.8 | 11.1 | 11 | 20.1 | 11.6 | 11.9 | 11.1 | 8.7 | 8 |
| ISI-ANL | 15 | 12 | 9.5 | 9 | 13.1 | 13 | 11.4 | 11 | 8.9 | 8.3 |
| ISI-UFL | 21 | 21.9 | 16 | 14.5 | 22.7 | 19.7 | 18.8 | 14.7 | 13 | 12 |
| ANL-UFL | 20 | 21 | 20 | 16 | 21.8 | 19.9 | 19.3 | 15.3 | 16.7 | 15.5 |

From Table 1, we can observe that including disk I/O component load variations in the regression model provides us with gains of up to 4% (G+D Avg) when compared with MovingAvg (first and third shaded columns in Table 1). Different filling techniques (G+D Avg and G+D LV) perform similarly.

Further, from Table 1, we see that all variations of G+N perform better than G+D in general, that is, regression using network data performs better than regression using disk I/O data. This observation agrees with our initial measurements that only 15-30% of the total transfer time is spent in I/O, while the majority of the transfer time (in our experiments) is spent in network transport.

When we include both disk I/O and NWS network data in the regression model (G+N+D) along with GridFTP transfer data, we see that the prediction error drops up to 3% when compared with G+N (second and fourth shaded columns in Table 1). Overall, we see up to 9% improvement when we compare G+N+D with our original prediction based on Moving Avg. As disk sizes grow and speeds stay the same, we believe this will be even more significant.

Figure 3a compares the forecasting error in Moving Avg, G+D Avg, G+N Avg, and G+N+D Avg for all of our site pairs (the shaded columns in Table 1) and also presents 95% confidence limits for our prediction error rates. The forecasting accuracy trend is as follows:

*Moving Avg < (G+D Avg) < (G+N Avg) < (G+N+D Avg).*

From Figure 3b we can observe that the interval does in fact reduce with more accurate predictors, but the reduction is not significant for our datasets.

Figure 4 depicts the performance of predictors G+D Avg and G+N+D Avg. Graphs show the relevant data sources and the associated predictions. We can see how predictors closely track the measured GridFTP values. Predictions were obtained by using regression equations that were computed for each observed network or disk throughput value. Sample regression equations with computed coefficients (based on discussion from Section 3.4) for the last observed N and D values in Figures 4a and 4b are as follows:

$$G' = 6.9 - 0.18*D$$

for the simple regression case and

$$G' = 7 - 0.38*N - 0.18*D$$

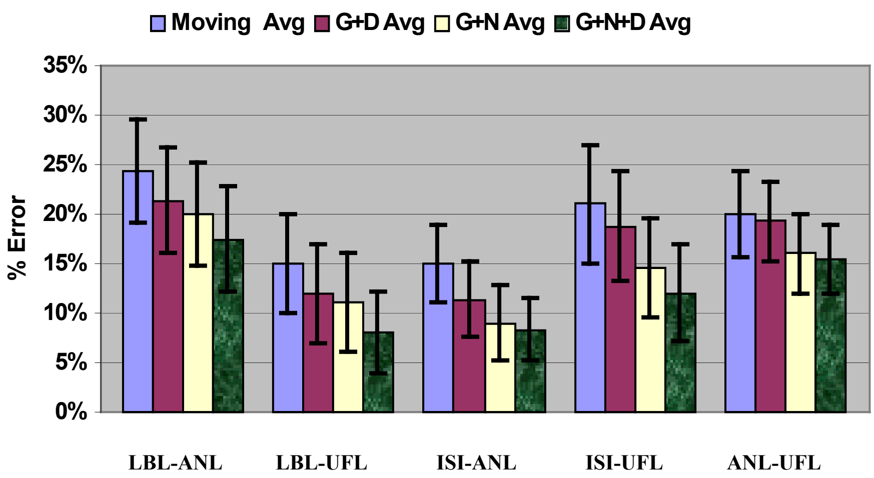
for the multiple regression.

## 5 Conclusion

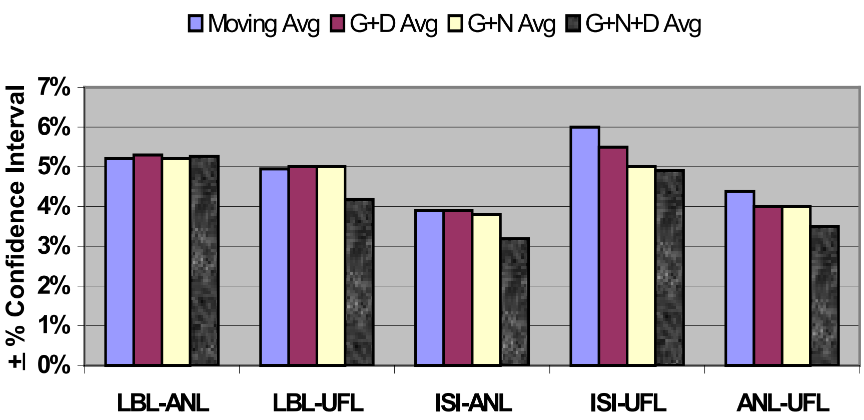
In this paper, we present techniques to combine observations of end-to-end application behavior and disk I/O throughput load data. We develop a set of regression models to derive predictions that characterize the effect of disk load variations on file transfer times. For deriving predictions we use simple statistical tools that are reasonably straightforward and easy to implement and therefore easy to apply to other datasets.

Using disk I/O data improves prediction accuracy by up to 4% when compared with predictions based on past GridFTP behavior. Similarly, predicting based on I/O, NWS, and GridFTP data improved accuracy further, by up to 9%. By adding additional data streams, each of which describing a piece of the end-to-end GridFTP transfer path, we see improvements in the accuracy of the predictions generated. For our datasets, we observe no improvements in using polynomial regression.

Future work includes exploring rank functions to evaluate the accuracy of predictors and using the variance information of predictors to perform scheduling decisions.

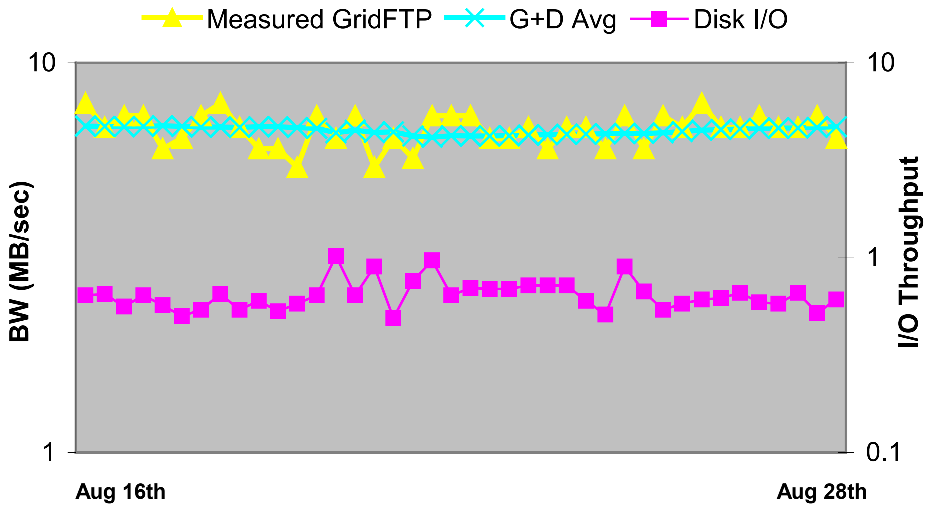



**(a) Comparison of normalized percent errors for the predictors with 95% confidence limits**

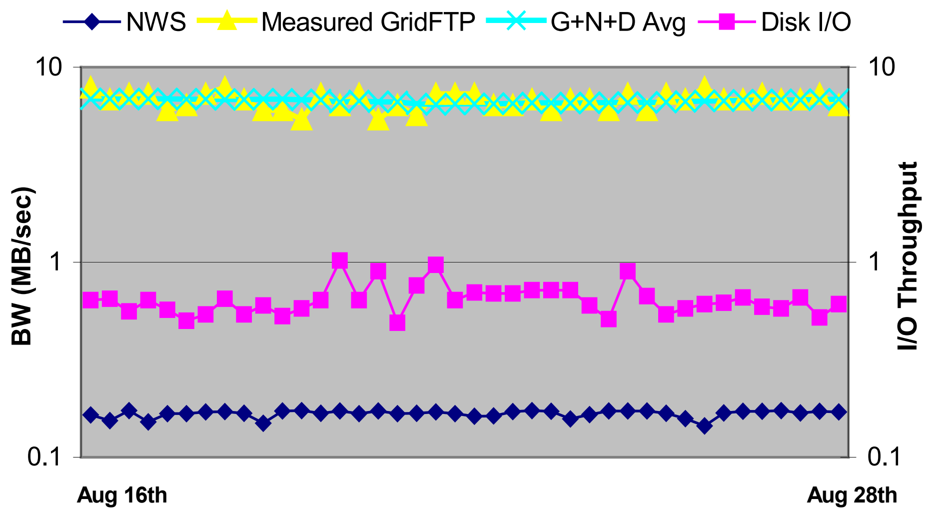



**(b) Comparison of intervals for the predictors**

**Figure. 3.** (a) Normalized percent prediction error and 95% confidence limits for August 2001 dataset from (1) prediction based on GridFTP in isolation (MovingAvg), (2) regression between GridFTP and disk I/O with Avg filling strategy (G+D Avg), (3) regression between GridFTP and NWS network data with Avg filling strategy (G+N Avg), and (4) regressing all three datasets (G+N+D Avg). Confidence Limits denote the upper and lower bounds of prediction error. For instance, the LBL-ANL pair had a prediction range of 17.3% ± 5.2%. (b) Comparison of the percentage of variability among the predictors.




**(a) Regression between GridFTP and Disk I/O**




**(b) Regression between GridFTP, NWS and Disk I/O**

**Figure. 4.** Predictors for 100 MB transfers between ISI and ANL for August 2001 dataset. In both graphs, GridFTP, G+D Avg, G+N+D Avg, and NWS are plotted on the primary y-axis; while Disk I/O is plotted on the secondary y-axis. I/O throughput denotes transfers per second.

## Acknowledgments

We thank all the system administrators of our testbed sites for their valuable assistance. 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.

## References

[ACF+02] Allcock, W., 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.* Network and Computer Applications, 2002.

[Adve93] Adve, V.S., *Analyzing the Behavior and Performance of Parallel Programs*, in *Department of Computer Science*. 1993, University of Wisconsin.

[AFN+01] Allcock, W., 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*. in *Supercomputing'01*. 2001.

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

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

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

[Crovella99] Crovella, M.E., *Performance Prediction and Tuning of Parallel Programs*, in *Department of Computer Science*. 1999, University of Rochester.

[DataGrid02] *The Data Grid Project, http://www.eu-datagrid.org, 2002*.

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

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

[FK98] Foster, I. and C. Kesselman. *The Globus Project: A Status Report*. in *IPPS/SPDP '98 Heterogeneous Computing Workshop*. 1998.

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

[Globus02] *The Globus Project, http://www.globus.org, 2002*.

[GriPhyN02] *The GriPhyN Project, http://www.griphyn.org, 2002*.

[GS00] Gray, J. and P. Shenoy. *Rules of Thumb in Data Engineering*. in *International Conference on Data Engineering ICDE2000*. 2000. San Diego: IEEE Press.

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

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

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

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

[Jones02] Jones, R. *The Public Netperf Homepage*, http://www.netperf.org/netperf/NetperfPage.html. 2002.

[ML90] Mak, V.W. and S.F. Lundstrom, *Predicting the Performance of Parallel Computations*. IEEE Transactions on Parallel and Distributed Systems, 1990: pp. 106-113.

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

[NetLogger02] *NetLogger: A Methodology for Monitoring and Analysis of Distributed Systems*. 2002.

[SARA02] *SARA: The Synthetic Aperture Radar Atlas, http://sara.unile.it/sara/, 2002*.

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

[SC00] Shen, X. and A. Choudhary. *A Multi-Storage Resource Architecture and I/O, Performance Prediction for Scientific Computing*. in *9th IEEE Symposium on High Performance Distributed Computing*. 2000: IEEE Press.

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

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

[SW02] Swany, M. and R. Wolski. *Multivariate Resource Performance Forecasting in the Network Weather Service*. in *Supercomputing'02*. 2002.

[SYSSTAT02] *SYSSTAT Utilities Homepage, http://perso.wanadoo.fr/sebastien.godard/, 2002*.

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

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

[Traces02] *GridFTP predictor Trace Data, http://www.mcs.anl.gov/~vazhkuda/Traces, 2002*.

[VS02] Vazhkudai, S. and J. Schopf. *Predicting Sporadic Grid Data Transfers*. in *11th IEEE High Performance Distributed Computing (HPDC-11)*. 2002. Edinburgh, Scotland: IEEE Press.

[VSF02] Vazhkudai, S., J. Schopf, and I. Foster. *Predicting the Performance Wide-Area Data Transfers*. in *16th International Parallel and Distributed Processing Symposium (IPDPS)*. 2002. Fort Lauderdale, Florida: IEEE Press.

[Web100Project02] *The Web100 Project, http://www.web100.org, 2002*.

[Wolski98] Wolski, R., *Dynamically Forecasting Network Performance Using the Network Weather Service*. Cluster Computing, 1998.

[ZLP96] Zaki, M.J., W. Li, and S. Parthasarathy. *Customized Dynaimic Lad Balancing for Network of Workstations*. in *High Performance Distributed Computing (HPDC'96)*. 1996.