Adaptive Performance Prediction for Distributed Data-Intensive Applications*

Marcio Faerman† Alan Su† Richard Wolski‡ Francine Berman†

April 29, 1999

Abstract

The *computational grid* is becoming the platform of choice for large-scale distributed data-intensive applications. Accurately predicting the transfer times of remote data files, a fundamental component of such applications, is critical to achieving application performance. In this paper, we introduce a performance prediction method, ARM (**A**daptive **R**egression **M**odeling), to determine data transfer times for network-bound distributed data-intensive applications.

We demonstrate the effectiveness of the ARM method on two distributed data applications, SARA (Synthetic Aperture Radar Atlas) and SRB (Storage Resource Broker), and discuss how it can be used for application scheduling. Our experiments demonstrate that applying the ARM method to these applications predicted data transfer times in wide-area multi-user grid environments with accuracy of 88% or better.

1 Introduction

Ensembles of distributed computational, storage, and other resources, also known as *computational grids* [12, 14], are becoming an increasingly important platform for applications which perform calculations over large datasets. Such applications include image acquisition and processing calculations, digital library searches, high performance massive data assimilation, distributed data mining and others [11, 17, 15, 10, 1, 24, 3]. Aggregating distributed resources presents the opportunity to employ or acquire data from very large datasets which are too large to be stored at a single site.

For several distributed data-intensive applications, data movement across the network is a critical determinant of application performance. In particular, in addition to being data-intensive, such applications are *network-bound* as well, with application performance heavily determined by the bandwidth available on network links used during data transfers. Examples of network-bound distributed data-intensive applications are JPL's Synthetic Aperture Radar Atlas (SARA) application [24], which allows the user to select and view images generated from from a large, replicated, and distributed database of radar data, and SDSC's Storage Resource Broker (SRB) [3, 19], which provides a uniform interface for users to obtain data from a heterogeneous and distributed collection of data repositories.

Efficient execution of network-bound applications on computational grids can be challenging. Although these platforms offer considerable performance potential through aggregation of resources, performance may be difficult to achieve in practice. In particular, the load and availability of shared resources such as networks may be hard to predict, yet accurate predictions of file transfer times are critical to the development of performance-efficient application execution strategies.

In this paper, we present a method, Adaptive Regression Modeling (ARM), for predicting the duration of data transfer operations in network-bound distributed data-intensive applications. Our technique predicts performance in production, multi-user distributed environments by employing small network bandwidth probes, provided by the Network Weather Service (NWS) [25, 26], to make short-term predictions of transfer times for a range of file sizes. Our approach is based on the use of statistical regression methods to calibrate application execution performance to the dynamic state of the system. The result is an accurate performance model that can be parameterized by "live" NWS measurements to make time-sensitive predictions.

The development of performance methods such as ARM is critical to the application performance for network-bound distributed data-intensive applications in

*This research was supported in part by NASA Graduate Student Research Grant #NGT-2-52251, Department of Defense Modernization Contract 9720733-00, DARPA contract N66001-97-C-8531, National Science Foundation Grants ASC-9701333 and ASC-9619020, and CAPES/Brazil Grant BEX0488/95-2.

†Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92093-0114 ({mfaerman,alsu,berman}@cs.ucsd.edu)

‡Department of Computer Science, University of Tennessee, Knoxville, TN 37996-1301 (rich@cs.utk.edu)

multi-user computational grid environments. As part of the **App**lication **Le**vel **S**cheduling (AppLeS) project [2, 5], our experience shows that the development and parameterization of accurate performance models can be a difficult and error-prone process. ARM relies on observable performance measurements only, and thus can be continuously updated to adapt to current network conditions *automatically* and *in real-time*. We demonstrate the effectiveness of the ARM method for two network-bound data-intensive applications with dissimilar data requirements: the SARA image acquisition tool which targets relatively small files (1-3 MB), and the SRB query tool which is designed for much larger (16 MB or more) files.

This extended abstract is organized as follows: In Section 2, we briefly describe the characteristics of network-bound distributed data-intensive applications, and in particular, SARA and SRB. Section 3 presents several performance models for predicting data transfer times for this application class. We propose the ARM prediction method and present experiments which demonstrate its effectiveness for both SARA and SRB in Section 4. In Section 5, we summarize and briefly touch on related and future work.

2 Network-Bound Distributed Data-Intensive Applications

We use the term **data-intensive applications** to denote computations which access and perform operations on numerous or massive datasets. Within this application class, we identify a subclass of **network-bound distributed** data-intensive applications for which a prime determinant of application performance is movement of data across the network.

To illustrate the characteristics of network-bound distributed data-intensive applications we provide a brief description of JPL's Synthetic Aperture Radar Atlas application and SDSC's Storage Resource Broker.

The Synthetic Aperture Radar Atlas (SARA) [24, 20], developed at JPL and SDSC, is a web-based distributed data-intensive application which allows users with access to the World-Wide Web to view images of the Earth's surface taken by a synthetic aperture radar. The SARA datasets are replicated across several high-capacity storage sites. Via a Java applet, users of the SARA system can request an image of an arbitrary sub-region with certain features of the data highlighted. The size of SARA files transmitted across the network typically ranges from 1 to 3 MB.

SDSC's Storage Resource Broker (SRB) [3, 19] is middleware that provides data-intensive applications with a uniform API to access heterogeneous distributed storage resources systems including file systems, databases, and hierarchical and archival storage systems. SRB provides users the capability to access and aggregate massive quantities of data scattered across wide area networks.

Note that there are several important differences between SARA and SRB. Most obviously, SARA files are typically small whereas SRB files can be quite large. In order to achieve performance, SARA must select a data server among the data servers which house the target replicated file. To do that, SARA needs to compare the predicted file transfer times for the replicated file to all the data servers. For SRB, the data transfer is often part of a larger application framework, and thus, file transfer time may be a component of a larger performance model. Applications which use SRB for data access require reasonably **accurate** transfer time estimates in order to perform scheduling decisions with them.

3 Performance Models

It would be reasonable to expect that a simple performance model of remote file transfer time for a network-bound distributed application would suffice for scheduling and application execution. In particular, the straightforward model **RBW** (**R**aw **B**and**W**idth model) shown below:

$$FileTransferTime = \frac{DataSize}{AvailableBandwidth}$$

could be used to estimate the time for transferring files between various servers based on the value of $AvailableBandwidth$ between the client and each of the servers. This value could be supplied by a network monitor such as the Network Weather Service (NWS) [25, 26], which measures and forecasts the load and availability of system resources (including network bandwidth).

The RBW model can be used to predict the file transfer performance of both SARA and SRB in a wide-area grid environment. However, this simplified model does not account for differences in network conditioning (message sizes, buffer sizes, etc.) between NWS probes and those used by the two applications. In particular, NWS probes are typically 64 kB packets, whereas SARA has data transfer sizes of 1-3 MB and SRB may have data transfer sizes of 16 MB and above. While NWS probes must be small in order to minimize intrusiveness on the network, they are insufficient for predicting the available bandwidth for much larger transfers directly.

In addition, RBW fails to consider application computational and internal message buffering overheads, which may have non-trivial effects on performance for even the most network-bound applications. It is not surprising then that RBW model predicts performance relatively inaccurately (with errors up to 100%), as illustrated in Table 1. What is interesting is that the peaks and valleys

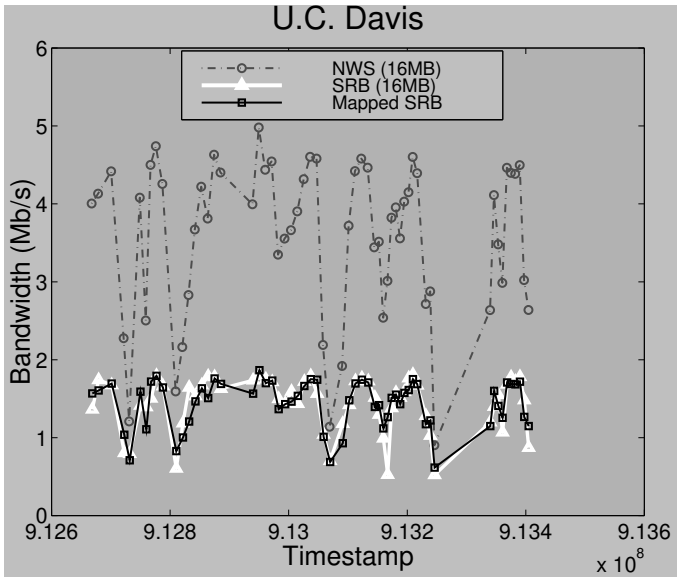


Figure 1: Linear Regression mapping NWS (16 MB messages) to SRB (16 MB file transfers)

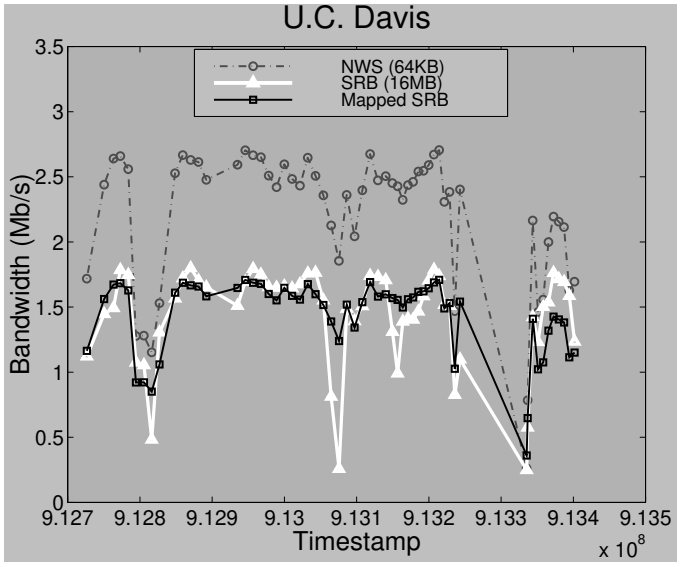


Figure 2: Linear Regression mapping NWS (64 kB messages) to SRB (16 MB file transfers)

as predicted by RBW seem to correlate with observed file transfer behavior.

3.1 Regression Modeling

Regression modeling is in general a simple method for establishing a functional relationship among variables [8, 9, 6]. In order to achieve more accurate predictions, we considered the use of a linear regression model to address the discrepancy, exhibited in the RBW model, between the performance behavior of small NWS probes and larger file transfers. We developed two linear models that map NWS bandwidth measurements to the observed file transfer behavior of the network-bound distributed data-intensive application within a specified time-frame.

The first model demonstrates an upper bound on the expected accuracy of this technique. We started by regressing large-file transfer times with 16MB NWS bandwidth probes. This probe size is too large to be practical for the NWS in general, but it allowed the NWS to more accurately mimic the actual network load during file transfer. Figure 1 shows the results of using this technique to model file transfer times from the University of California at Davis to the University of California at San Diego. The experiments show this regression model parameterized by data from 16MB data transfers and 16MB NWS message size probes. The results were very impressive. We observe in the graph how the modeled file transfer performance tracks very closely the actual throughput measurements. Our next step was to investigate a less intrusive approach to determine if the technique could be implemented practically.

3.2 A Practical Approach — NWS with 64 kB Probes Only

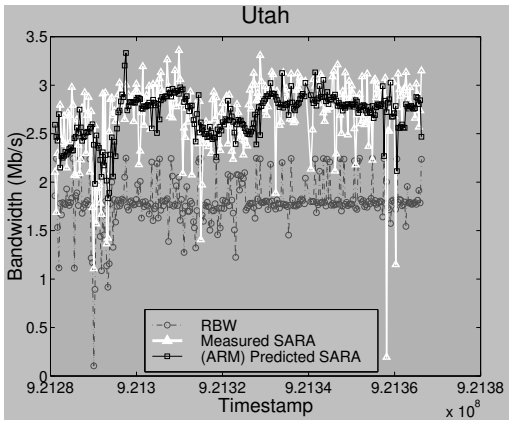
In order to decrease overhead, we considered a new regression model, which uses NWS inputs with 64 kB probes. This is the probe size that is commonly used by the Network Weather Service to perform bandwidth measurements and forecasts. Using small probes, the NWS is able to maintain a low level of intrusiveness over the network [25, 26].

Representative results using the low overhead model, are shown in Figure 2. Note that the model still tracks very closely the actual file transfer measurements.

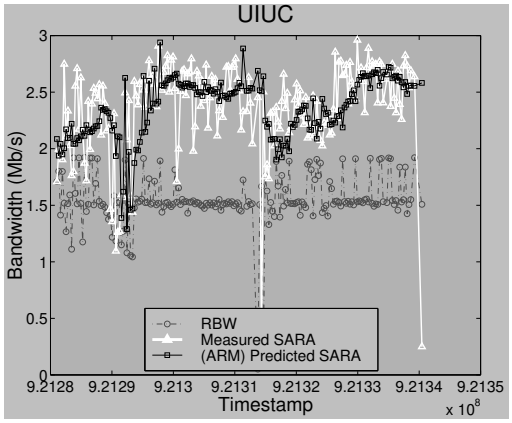
In the next section, we focus on the development of a performance model using the less intrusive regression approach which we call **A**daptive **R**egression **M**odeling **(ARM)**.

4 Using ARM to Dynamically Predict Execution Performance

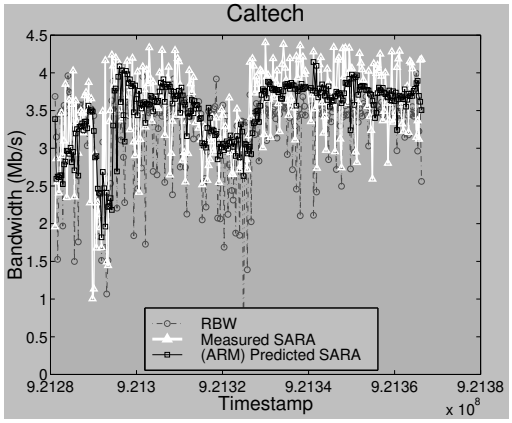
The previous section discusses the goodness of the regression "fit" between traces of NWS probe data and actual file transfers. In this section we describe how to use the regression technique to actually **predict** the execution performance of the SRB and SARA grid applications at run-time — the **ARM** forecasting method.



(a) perigee.chpc.utah.edu



(b) sitar.cs.uiuc.edu



(c) spin.cacr.caltech.edu

Figure 3: Forecasting SARA file transfer behavior with the **ARM** method.

SRB – NMAE		
site	RBW	ARM
U.C. Davis	54.10%	11.09%
NCSA	105.20%	9.20%
W.U. St. Louis	96.31%	11.95%
Rutgers	101.56%	1.15%

SARA – NMAE		
site	RBW	ARM
Utah	34.43%	9.97%
UIUC	36.06%	11.04%
Caltech	15.80%	11.67%

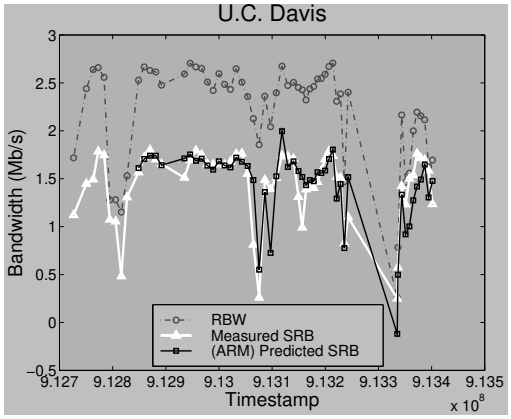
Table 1: **Normalized Mean Absolute Errors (NMAE)** for file transfer throughput forecasts obtained directly from bandwidth measurements **RBW** and from the **A**daptive **R**egression **M**odeling **(ARM)** forecaster.

Since linear regression is cheap to compute, we start by deriving an initial model from historical application and NWS performance data. As the application executes, we monitor its performance and add the monitored values to the performance history of the program. When a prediction is required, we recalculate the regression coefficients "on-the-fly" using the original performance history, the most recent performance measurements, and the corresponding NWS data for the most recent time frame. In this way, the model evolves and adapts in response to changing performance conditions.

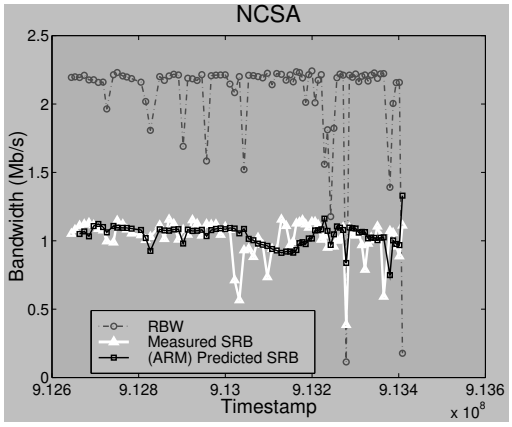
We term the initial set of samples required to "heat-up" the regression model the *Start-up Window*. Having an initial regression model derived from the *Start-Up Window*, we obtain a new bandwidth sample from NWS, and use this value in the regression model to generate a prediction of the first file transfer throughput value.

While the application executes, the regression function is updated at each new application file transfer. The new file transfer sample and the network bandwidth value (from the NWS) are incorporated as a new pair within history of network and file transfer samples that will be used in the calculation of the next regression model. Then, the updated regression model is used to forecast the next file transfer, and so on.

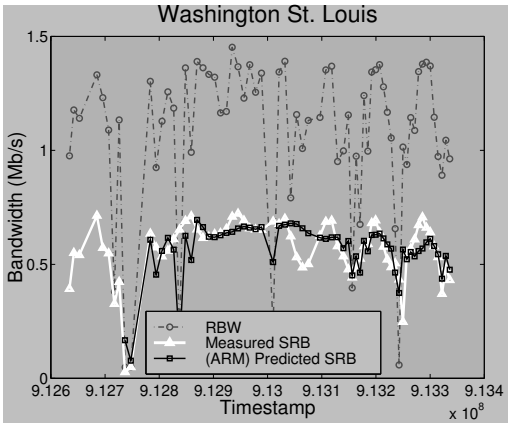
We refer to the set of samples used to generate each new regression model at execution time as the *Running Time Window*. The *Running Time Window* slides over the past history of network and application measurements at each new application transfer. The regression model is updated at each new sample assuring that the forecaster will adapt to the most recent application resource requirements and observed network behavior. At each update, the oldest sample pair of the *Running Time Window* is



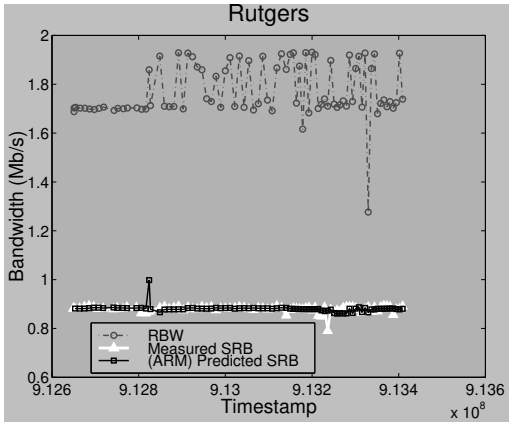
(a) mahler.cipic.ucdavis.edu



(b) vor.ncsa.uiuc.edu



(c) brainmap.arl.wustl.edu



(d) bionic.rutgers.edu

Figure 4: Forecasting SRB file transfer behavior with the **ARM** method.

discarded at each new transfer. In this way, we filter out of the regression model past history that is no longer relevant for new trends of system behavior.

We investigated the effect of varying the size of the *Start-up Window* and *Running Time Window* respectively. After simulating the forecaster for several *Start-Up Window* and *Running Time Window* lengths, we found that the the modeling technique remains accurate for a wide range of parameterizations. Usually a *Running Time Window* length between 8 and 20 samples will yield good predictions for the environments we have investigated. The algorithm is even less sensitive to the length of the *Start-Up Window*, since its effect is felt only initially. **A more detailed discussion of the effects of *Start-Up Window* and *Running Time Window* parameters on performance, and the ARM pseudo-code will be provided in the final paper.**

Figures 3 and 4 show a representative and comprehensive set of experiments to predict file transfer times using the ARM method for SARA and SRB respectively. In the experiments, SARA transferred files of 3MB, at every 5 minutes from data-servers running at the University of Utah, University of Illinois Urbana-Champaign and Caltech. The SRB client transferred 16MB files at intervals ranging from 2 to 25 hours from the University of California at Davis, University of Washington at St. Louis, NCSA and Rutgers. Both clients ran at the University of the California, San Diego. In the figures, the differences (the error) between predicted (*in black*) and actual execution times (*in white*) are represented by the vertical distance between each pair of points. In Table 1, we summarize the error results for this data.

4.1 Analysis

To determine prediction accuracy, we use the *Normalized Mean Absolute Error* **(NMAE)**, given by

$$\sum_{i=1}^{N} \frac{|EstimatedPerf_i - MeasuredPerf_i|}{N \times \overline{MeasuredPerf}} \times 100\%$$

where N is the total number of predicted measurements and $\overline{MeasuredPerf}$ is the mean measured file transfer performance for a particular site. The concept behind the *Normalized Mean Absolute Error* is to calculate the mean prediction error, then normalize it by the mean file transfer throughput which is given by $\overline{MeasuredPerf}$. In this way we provide an effectiveness metric comparable among different applications and sites, since each different environment might have distinct performance characteristics. In other words, the error is given as a proportion or percentage of the average performance perceived by an application.

From observing the graphs it is striking how the ARM predictions very closely track the actual measured performance of both SARA and SRB applications. Moreover, Table 1 numerically confirms the efficacy of this method. As it can be observed, the highest relative errors (NMAE) for ARM predictions are on the order of only 10%, whereas the errors for predictions using the RBW model reach up to 100%. The largest difference occurs for the Rutgers data server, where the RBW model resulted in an error of 101.56%, while the ARM forecasting method predicted with an error of just 1.15%.

Furthermore, it is important to emphasize that we were able to successfully obtain this high level of prediction accuracy with a low level of intrusiveness on the system — using only small 64 kB messages to probe the network behavior.

Although our analysis in this extended abstract is confined to NMAE values, we also include Table 2, showing mean square errors (**MSE**) for the predictions for all sites in the experiments. Table 3 shows the mean file transfer throughput for each data-server. **In the final version of the paper, we will discuss the MSE values in greater detail.**

4.2 Using ARM for Scheduling

The **App**lication-**Le**vel **S**cheduling (AppLeS) approach incorporates both application-specific system requirements and dynamic resource performance information to schedule distributed applications in multi-user distributed environments [5, 20, 27]. AppLeS application-level schedulers use a performance model (based on the application's communication and computational needs), which is parameterized by values representing system characteristics (such as available network bandwidth). An AppLeS scheduler delays evaluation of the model until run-time, at which point the model parameters are supplied by a forecaster such as the Network Weather Service (NWS) [25, 26], which reports current conditions and generates forecasts of various system performance measures, such as network bandwidth and CPU load.

The initial AppLeS schedulers we developed [2, 5, 27, 20, 18] focused on the development of adaptive custom schedules for individual grid applications. We are currently developing AppLeS templates for scheduling structurally similar classes of grid applications. We are focusing on several application classes, including network-bound distributed data-intensive applications. For certain applications in this class, like a basic SARA application, using the RBW model to *rank* potential data servers suffices [20]. However, applications for which predictions using RBW are insufficient require methods like ARM, which provide reasonably accurate run-time estimates of file transfer performance. Such estimates are used to determine an execution schedule for the target application which is implemented by the AppLeS template.

SRB – MSE		
site	RBW	ARM
U.C. Davis	0.7536	0.0534
NCSA	0.3320	0.0195
W.U. St. Louis	1.2147	0.0112
Rutgers	0.8129	0.0004

SARA – MSE		
site	RBW	ARM
Utah	0.9674	0.1446
UIUC	0.8669	0.1515
Caltech	0.5128	0.2975

Table 2: **Mean Square Errors (MSE)** for file transfer throughput forecasts obtained directly from bandwidth measurements **RBW** and from the **A**daptive **R**egression **M**odeling **(ARM)** forecaster. These values are not normalized.

SRB – Mean Throughput	
site	Mbits/s
U.C. Davis	1.4399
NCSA	1.0218
W.U. St. Louis	0.5659
Rutgers	0.8792

SARA – Mean Throughput	
site	Mbits/s
Utah	2.6838
UIUC	2.3569
Caltech	3.4825

Table 3: Mean Measured File Transfer Throughput values for SRB and SARA

5 Conclusions and Future Work

This paper presents ARM, a dynamic forecasting method to predict the performance of data transfer operations for network-bound distributed data-intensive applications. Our method achieves a high level of accuracy for exemplar applications SARA and SRB. In summary, this paper makes the following contributions:

- An ARM forecaster is derived automatically and in real-time, by using a regression model to map measurements of system behavior to observed application performance.
- ARM forecasters dynamically adapt in time to changes in the environment. In particular, changes in the workload and system reconfigurations are used to parameterize frequent updates of the model.
- The overhead of computing predictions using ARM is low and relatively straightforward – the linear regression is applied to a sliding window of a small number of sample pairs, minimizing computational overhead. Moreover, generating predictions of application performance using the model is inexpensive, making it possible reevaluate the model often to adapt to rapidly changing network conditions.
- The ARM method hides the internal details of the underlying system. No knowledge about communication protocols, network topology, or local file systems was necessary to achieve good predictions. For example, even though the SARA application exhibits higher file throughput than the NWS measured bandwidth and exactly the opposite happens for the SRB transfers, the forecaster is able in both instances to make accurate predictions.

Regarding related work, performance analysis and scheduling of data-intensive applications are described by the ADR group from University of Maryland in [23] and by Thakur in [21]. However, they focus on parallel data servers running over local area networks. Performance monitoring and forecasting of wide-area networks is discussed in works such as the NWS [25, 26], GloPerf [13] and [7, 4]. The Netlogger system [22] presents a profiling framework for distributed storage systems. **A more detailed *related work section* will be included in the final version of this paper.**

In the future, we intend to investigate adaptive Running Window length choices, adapting the past history length according to the current statistical trends regarding the relation between system and application behavior. In addition, we intend to extend this work to account for data transfers performed on hierarchical storage systems such as HPSS [16], including system with tertiary storage (tapes). We should also be able to use the ARM method to generate network forecasts for varied communication protocols and configurations and accurate relative ranking among several data servers. In extending the scope of this work to new scenarios, we also intend to look at the possibility of on-demand refinement of the regression model to include additional factors, such as disk or tape behavior and server load, in order to do a better job of assessing end-to-end file transfer, and hence application, performance.

Acknowledgements

The authors would like to thank NPACI researchers Reagan Moore, Chaitanya Baru, Arcot Rajasekar and Michael Wan for their invaluable help for providing us

access to the SRB infrastructure, their constant support, and insightful comments. The SARA application was developed by Roy Williams at CalTech and George Kremenek at SDSC, both of whom helped immensely in our experiments with SARA. We are also very grateful for the important ideas resulting from discussions with our colleagues from the AppLeS group. Essential help was also provided by Jim Hayes in adding new capabilities to the NWS, necessary to run our experiments. Finally, we would like to thank the NPACI sites that provided the data servers for this work.

References

[1] A. Amoroso, K. Marzullo, and A. Ricciardi. Wide-Area Nile: A Case Study of a Wide-Area Data-Parallel Application. In *ICDCS'98 - International Conference on Distributed Computing Systems*, 1998.

[2] AppLeS webpage at `http://www-cse.ucsd.edu/groups/hpcl/apples`.

[3] C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC Storage Resource Broker. In *IBM CASCON '98*, 1998.

[4] S. Basu, A. Mukherjee, and S. Klivansky. Time Series Models for Internet Traffic. *IEEE Comput. Soc. Press*, 1996.

[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] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis — Forecasting and Control*. Prentice Hall, 1994.

[7] R. L. Carter and M. E. Crovella. Dynamic Server Selection Using Bandwidth Probing in Wide-Area Networks. Technical Report TR-96-007, Computer Science Department, Boston University, 1996.

[8] S. Chatterjee and B. Price. *Regression Analysis by Example*. John Wiley & Sons, Inc., 1991.

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

[10] U. Fayyad and R. Uthurusamy. Data Mining and Knowledge Discovery in Databases. *Communications of the ACM*, 1996.

[11] R. Ferreira, B. Moon, J. Humphries, A. Sussman, J. Saltz, R. Miller, and A. Demarzo. The Virtual Microscope. In *Proc. of the 1997 AMIA Annual Fall Symposium*, 1997.

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

[13] GloPerf webpage at `http://www-fp.globus.org/details/gloperf.html`.

[14] A. S. Grimshaw, W. A. Wulf, and the Legion team. The Legion Vision of a Worldwide Virtual Computer. *Communications of the ACM*, 1997.

[15] R. Grossman, S. Kasif, R. Moore, D. Rocke, and J. Ullman. Data Mining Research: Opportunities and Challenges - A Report of three NSF Workshops on Mining Large, Massive, and Distributed Data, 1998. Available at `http://www.ncdm.uic.edu/M3D-final-report.htm`.

[16] High Performance Storage System webpage at `http://www.sdsc.edu/hpss/hpss1.html`.

[17] B. R. Schatz. High-Performance Distributed Digital Libraries: Building the Interspace on the Grid. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*, 1998.

[18] N. Spring and R. Wolski. Application level scheduling of gene sequence comparison on metacomputers. In *Proc. 12th ACM International Conference on Supercomputing*, Jul 1998.

[19] SDSC's **Storage Resource Broker** project webpage at `http://www.npaci.edu/DICE/SRB/index.html`.

[20] A. Su, F. Berman, R. Wolski, and M. M. Strout. Using AppLeS to Schedule a Distributed Visualization Tool on the Computational Grid. *International Journal of Supercomputer and High-Performance Applications*, 1999.

[21] R. Thakur, W. Gropp, and E. Lusk. A Case for Using MPI's Derived Datatypes to Improve I/O Performance. In *Proc. of SC98: High Performance Networking and Computing*, 1998.

[22] B. Tierney, W. Johnston, B. Crowley, G. Hoo, C. Brooks, J. Lee, D. Gunter, and S. Kim. The NetLogger Methodology for High Performance Distributed Systems Performance Analysis. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*, 1998.

[23] M. Uysal, T. Kurc, A. Sussman, and J. Saltz. A Performance Prediction Framework for Data Intensive Applications on Large Scale Parallel Machines. In *Proc. of 4th Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers*. 1998.

[24] R. Williams. Caltech's **Synthetic Aperture Radar Atlas** project webpage at `http://www.cacr.caltech.edu/roy/sara/index.html`.

[25] R. Wolski. Dynamically Forecasting Network Performance to Support Dynamic Scheduling Using the Network Weather Service. In *Proc. 6th IEEE Symp. on High Performance Distributed Computing*, August 1997.

[26] R. Wolski, N. Spring, and J. Hayes. The network weather service: A distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems (to appear)*, 1999. available from `http://www.cs.utk.edu/~rich/publications/nws-arch.ps.gz`.

[27] D. Zagorodnov, F. Berman, and R. Wolski. Application Scheduling on the Information Power Grid. *International Journal of High-Performance Computing*, 1998.