PREDICTING THE PERFORMANCE OF DATA TRANSFER IN A GRID ENVIRONMENT

A predictive framework for efficient data transfer in a Grid environment

A.B.M. Russel, Savitri Bevinakoppa
School of Computer Science & Information Technology,
Royal Melbourne Institute of Technology (RMIT) University,
GPO BOX 2476V, Melbourne, Victoria-3001, Australia

Keywords: Performance prediction heuristics, Data transfer, Node selection, Grid environment, Globus, GridFTP, NWS

Abstract: In a Grid environment, implementing a parallel algorithm for data transfer or multiple parallel jobs allocation doesn't give reliable data transfer. There is a need to predict the data transfer performance before allocating the parallel processes on grid nodes. In this paper we propose a predictive framework for performing efficient data transfer. Our framework considers different phases for providing information about efficient and reliable participating nodes in a computational Grid environment. Experimental results reveal that multivariable predictors provide better accuracy compared to univariable predictors. We observe that the Neural Network prediction technique provides better prediction accuracy compared to the Multiple Linear Regression and Decision Regression. Proposed ranking factor overcomes the problem of considering fresh participating nodes in data transfer.

1 INTRODUCTION

A Grid environment is a network of geographically distributed resources including computers, scientific instruments and data. Schedulers in the Grid ensure that jobs are completed in a particular order like priority, deadline, urgency and load balancers distribute tasks across systems to decrease the chance of overload.

To determine the source of grid performance problems (Lee et al., 2002) require detailed end-to-end instrumentation of all components, including applications, operating systems, hosts, networks etc. But in this environment, available communication and computational resources are changed constantly. For this purpose, raw historical data or forecasts of future end-to-end path characteristics between the source and each possible sink can be used.

The computational grid environment is dynamic in nature. Here, participating nodes can be connected and disconnected at the user's wish or detached by power failure or any kind of failure as shown figure 1. Thus, predicting the performance of nodes, whether previously used or not, can increase the overall performance of data transfer in a grid environment.

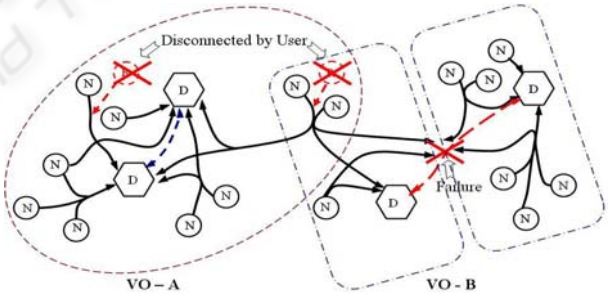

Figure 1: Node detachment in a computational Grid environment

The figure 1 shows two different types of failures: i) resource disconnected by user and ii) resource disconnected by other failures. Here, D denotes Directory Services and N denotes Nodes.

In any scenario mentioned above, grid management middleware is used which automatically resubmits jobs to other nodes on the grid when a failure is detected. In real-time, multiple copies of the important jobs run on different nodes throughout the grid according to their previous characteristics and performances. Their results are

B. M. Russel A. and Bevinakoppa S. (2005).
PREDICTING THE PERFORMANCE OF DATA TRANSFER IN A GRID ENVIRONMENT - A predictive framework for efficient data transfer in a Grid environment.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 176-181
DOI: 10.5220/0002537401760181

checked for any kind of inconsistency in data transfer. There still remain the problems of selecting the suitable nodes that are more reliable and give better performance. In this paper we are addressing the above-mentioned problem for selecting more reliable and efficient nodes by predicting their performance before allocating parallel tasks in a computational Grid environment.

2 RELATED WORK

Vazhkudai and Schopf (Vazhkudai, 2002, Vazhkudai, 2003) raised the necessity of multiple data-streams as input as these data streams can be periodic in nature. Similar to this work, Swany and Wolski (Swany & Wolski, 2002) also used multivariate predictors by constructing cumulative distribution functions for 16 MB HTTP transfers. Vazhkudai et al. (Vazhkudai et al., 2002) proposed a context-insensitive and context sensitive factor concept by considering the data size "on the fly" by categorizing into 4 different ranges: 50MB, 50MB-250MB, 250MB-750MB and more than 750MB for wide area data transfer. We propose an idea of considering the data size in ranking phase for calculating a ranking factor. Liu et al. (Liu et al., 2002) used a similar ranking policy for selecting resources by considering the longest subtask execution time with the following formula:

$$Rank = \frac{1}{Max(SubTaskExecutionTime)}$$

Moreover, Vazhkudai et al. (Vazhkudai et al., 2001) used a ranking method among the resources based on their available space for achieving the best match. Similarly, Wolski (Wolski, 2003) proposed a method for estimating the resource performance characteristics of the resources with accuracy ranking using the most recently gathered history. All these still do not fulfil the requirement of categorization data for source and sink nodes. After estimating the GridFTP throughput we need to rank the available nodes based on their performance. The higher the ranking factor will indicate more reliable nodes. Thus among all participating nodes, we can allocate our parallel processes for transferring data to efficient nodes based on their ranking factor.

Our research is to investigate and extend the Middleware Service of the Globus project (Globus) by constructing a predictive framework for data transfer incorporating the following aspects:

a) Applying GridFTP log data, dynamic Network Weather Service (NWS) data and Disk throughput data for predicting the network performance and calculating a ranking factor when nodes are involved in previous data transfer.

b) Predicting performance when there was no previous transfer of data between source and sink nodes.

3 PREDICTIVE FRAMEWORK

We construct a predictive framework for performance prediction of already participated nodes and potential participating nodes by:

(1) Recording the performance of end-to-end file transfers employing integrated instruments of high performance data transfers.

(2) Estimating future transfer performance by predictors.

(3) Calculating ranking factor for identifying efficient nodes and

(4) Constructing a data delivery infrastructure for providing users with the raw data and predictions by Monitoring & Discovery Service (MDS) provided by Globus (Globus).

Our predictive framework performs activities in four different phases to identify reliable nodes for performing data transfer. The four phases are Monitor, Prediction, Ranking and Delivery. The framework is shown below:

3.1 Monitor Phase

In the Computational Grid environment, simply allocating parallel processes is not enough for data transfer as many applications require gigabyte or even petabyte transfer, GridFTP log Data, Network Weather Service Data, Disk I/O Data, Bandwidth or effective throughput, Bottleneck Bandwidth, Round-trip delay, and Data Size.

3.2 Prediction Phase

This is the second phase of our framework. In this phase several predictors can be used to calculate future performance from the historical value. The aim of this phase is to achieve accurate prediction calculation. To achieve accurate prediction we can use univariable predictors (Vazhkudai et al. 2002, Faerman, 1999, Wolski, 1997, Wolski, 1998, Wolski et al., 1999) or multivariable predictors (Vazhkudai & Schopf, 2002 Vazhkudai & Schopf, 2003).

Univariable predictors use only one source of data. These can be only file transfer performance data, network performance data, disk throughput

data etc. Researchers found that multivariable predictors provide better performance accuracy compared to single variable predictors (Vazhkudai & Schopf, 2002, Vazhkudai & Schopf, 2003, Swany & Wolski, 2002).

Multivariable predictors are in the case of nodes that participated in previous data transfers we propose a prediction calculation model for multivariable predictors modified from the predictive model of Vazhkudai and Schopf (Vazhkudai & Schopf, 2003) which considers three different data sources for efficient prediction calculation: GridFTP Data, NWS Data, Disk I/O Data. Our prediction calculation model works in several stages, which begins with making datasets from historical performance datasets. Then apply some filling techniques (Vazhkudai & Schopf, 2002, Vazhkudai & Schopf, 2003) and then apply this dataset to different prediction techniques and achieve future performance value. Our prediction model is shown in figure 2.

a) Make Datasets: In the first step we make the dataset by closely comparing timestamp values of three different data sources to make it single common dataset.

b) Filling: Although these three source datasets are correlated these are not available in all nodes with the same timestamp. To overcome this problem, we use some proposed filling techniques (Vazhkudai & Schopf, 2003) in the second step:

- **NoFill:** We discard the unmatched NWS data and Disk I/O data. By discarding unmatched data we may lose useful data.
- **LastValue:** We fill in the last GridFTP value for each unmatched NWS and Disk I/O value.
- **AverageValue:** We consider an average over the historical data transfers is calculated for each timestamp.

c) Prediction Techniques: To calculate a more accurate predictive value from three different data sources we compare three different prediction techniques. These are Multiple Linear Regression, Decision Regression Tree and Neural Network. We calculate prediction accuracy for these three different prediction techniques using the following formula:

$$\%Error = \frac{\sum \left| measuredBW - predictedBW \right|}{(size * meanBW)} * 100$$

Here, "size" is the total number of predictions and the meanBW is the average measured GridFTP throughput and GridFTP throughput by using the following formula

$$throughput = \frac{datasize}{datatransfertime}$$

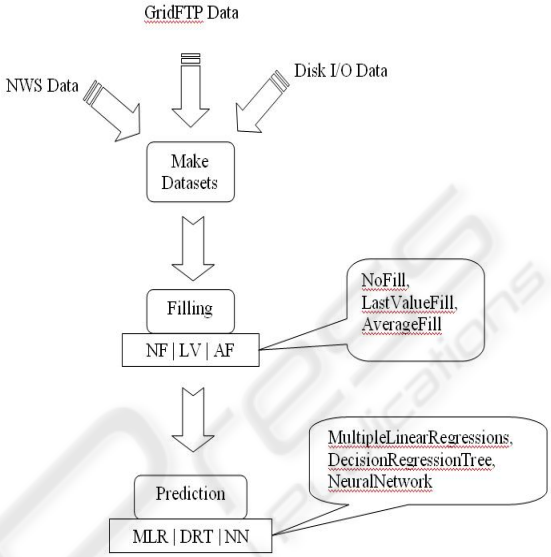

Figure 2: Prediction

Three different prediction techniques can be used for calculating future performance:

I. Multiple Linear Regression
II. Decision Regression Tree
III. Neural Network

In the next section we discuss each prediction technique in detail

I. Multiple Linear Regressions:

Linear regression can be used to make predictions. Simple linear regression involves discovering the equation for a line that nearly fits the given data. That linear equation is then used to predict values for the data. If there are more than one predictor variable we then use multiple linear regression technique to calculate the predicted value.

Vazhkudai and Schopf (Vazhkudai & Schopf, 2003) used this technique for calculating a predictive value from correlated datasets. Correlation describes the strength, or degree, of linear relationship and specifies to what extent the two variables behave alike or vary together.

II. Decision Regression Tree

A decision tree can be used as predictive model (Syed & Yona, 2003).

III. Neural Network

A Neural network is also a predictive model of highly interconnected neurons, that accept inputs as neurons by applying weighting coefficients and feed their output to other neurons. The weights applied to each input at each neuron are adjusted to optimize the desired output. Neural networks are trained to deliver the desired result by an iterative process. This process continues through the network. Some neurons may send feedback to earlier neurons in the network.

3.3 Ranking Phase

After estimating the GridFTP throughput we need to rank the available nodes based on their performance. The higher-ranking factor will provide us with the more reliable nodes. Thus among all possible nodes, we can allocate our parallel processes for transferring data to efficient nodes based on their ranking factor which could be calculated as:

$$RankingFactor = \frac{DataSize}{(BottleneckBandWidth - predictedBandWidth) * RoundTripDelay}$$

We consider two different scenarios which are:

1) Nodes that have already participated in data transfer
2) Novice nodes that did not participate in any data transfer

3.4 Delivery Phase

In this phase we can use Globus Resource Allocation Manager (GRAM) and Monitoring and Discovery Service (MDS) of the Globus project (Globus). GRAM provides resource allocation and process creation, monitoring and management services. It maps requests via RSL (resources specification language) into commands to local schedulers.

MDS of Globus (Globus) provides better performance among different monitoring services (Zhang et al., 2003). In the computational Grid environment, resources do dynamic registration via Grid Resource Registration Protocol (GRRP) and a client can query the resource via Grid Resource Information Protocol (GRIP). Using these two protocols, Grid Index Information Servers (GIIS) of MDS knit together arbitrary GRIS services to identifying resources and Grid Resource Information servers (GRIS) of MDS to provide uniform facilities of resource discovery. Using MDS, our forecasted information can be published in directory services along with other information to any authorized client node for data transfer. Moreover, based on the predicted information, GRAM can allocate jobs to suitable nodes for scheduling or load balancing purpose.

4 EXPERIMENTAL TEST-BED SITES AND DATA SOURCES.

In this paper, we consider the data sources (GridFTP log data, NWS log data, Disk IO data) (Trace-Data) achieved among four different testbed sites (see Figure 3): Argonne National Laboratory (ANL), Information Sciences Institute (ISI) of University of Southern California, Lawrence Berkeley National Laboratory (LBL) and the University of Florida at Gainesville (UFL). All sites comprised of 100 Mb/sec Ethernets with high-end storage.

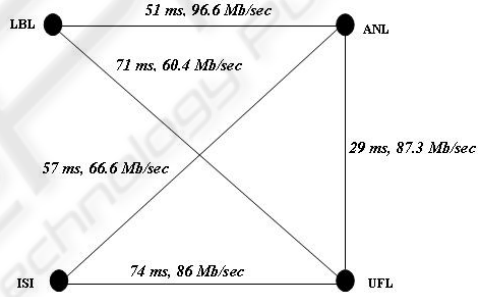

Figure 3: Network settings for testbed sites. All sites are connected through OC-12 or OC-48 network links. For each site pair round trip times and network bottleneck bandwidths for the link between them is shown

4.1 Experimental Results

We design our experiments to compare different prediction techniques by investigating several univariable predictors and multivariable predictors. We calculate the prediction error using the following formula:

$$\%Error = \frac{\sum |measuredBW - predictedBW|}{(size * meanBW)} * 100$$

and the meanBW is the average measured GridFTP throughput and GridFTP throughput is achieved by using the following formula:

$$throughput = \frac{datasize}{datatransfertime}$$

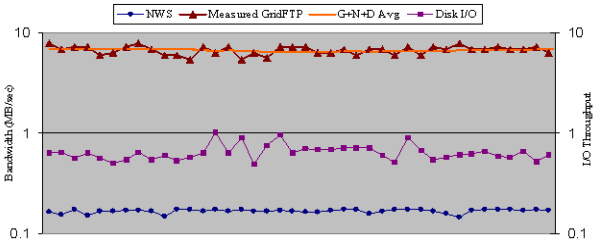

Figure 4: Relationship between NWS, Disk I/O data, GridFTP data and Multiple Linear Regression with GridFTP + Disk I/O + NWS data

We investigate three different prediction techniques ie. Decision Regression Tree (DRT), Multiple Linear Regression (MLR) and Neural Network (NN) with a common dataset as shown in Table 1 below:

Table 1: Sample dataset used in prediction techniques for comparison purpose.

NWS	DiskIO	GridFTP
0.164971	0.64	7.876
0.154224	0.65	6.826
0.173058	0.56	7.314
0.151951	0.64	7.314
……	……	……

To compare these three prediction techniques we use seven different sized training sets for each prediction technique.

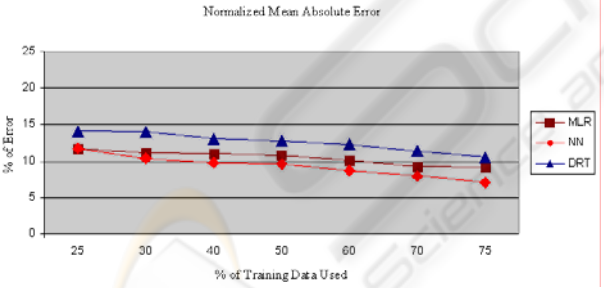

Figure 5: Comparison among three different prediction calculation models with Normalized Mean Absolute Error at different training datasets inputs with 50% window shift

We calculate the Ranking factor to identify the most efficient and reliable nodes for performing 10MB data transfer. According to our formula we achieve the rank order among all participating nodes for allocating parallel processes for data transfer:

$$R_{UFL,\,ANL} = \frac{10MB}{(87.3Mbps - 0.0MBps) * 29ms} = 31.59$$

$$R_{UFL,\,ISI} = \frac{10MB}{(86Mbps - 5.12MBps) * 74ms} = 24.00$$

$$R_{UFL,\,LBL} = \frac{10MB}{(60.4Mbps - 6.40MBps) * 71ms} = 122.47$$

In this case, ANL is a fresh node to UFL and ISI performed previous data transfer with UFL. Although ISI performed data transfer with UFL, our ranking factor provides a higher rank to ANL than ISI for UFL. Thus the ranking factor calculation shows: $R_{UFL,\,LBL} > R_{UFL,\,ANL} > R_{UFL,ISI}$. This calculation solves the problem with novice participating node consideration in data transfer.

4.2 Summary of Results

After performing several experiments we achieved the following conclusion about choosing reliable and efficient nodes based on their performance for data transfer in the Grid environment:

- Univariable predictors provide accuracy ranges between 15% and 25%.
- Average 25-value predictor provides better prediction accuracy among all univariable predictors examined.
- Multivariable predictor's prediction accuracy is better than univariable predictor's prediction accuracy.
- Average filling technique gives better prediction accuracy by considering a one-to-one mapping among multivariable datasets.
- Multiple Linear Regression (MLR) prediction technique gives better prediction accuracy using more number of training input data compared with Decision Regression Tree (DRT) and Neural Network (NN) prediction techniques.
- DRT prediction technique provides less prediction accuracy and the number of nodes increases with the increase of training input data thus consumes more resources.
- NN prediction technique provides better prediction accuracy with less number of training input data but consumes more resources and the accuracy increases with more training input data.

- Ranking factor calculation solves the problem with fresh node consideration for data transfer.

Thus if we use a predictive framework for data transfer by allocation jobs on reliable and efficient participating nodes, it will improve the overall performance of data transfer in Grid environment.

5 CONCLUSIONS AND FUTURE WORK

In this paper we investigated the problem for selecting reliable and efficient nodes by predicting their performance before allocating parallel jobs in a computational Grid environment. Allocating jobs to predicted reliable nodes gives better performance and is also more cost effective than using hot pluggable expensive hardware. To solve this we proposed a predictive framework with different phases for providing information about efficient and reliable participating nodes in the computational grid environment. We discussed, examined and compared various prediction techniques. We also solved the problem of considering new nodes with ranking factor calculations.

Although our proposed framework solves the performance prediction problem, there may arise a critical situation that many ranking factors will become same for a large set of nodes in the vast internet. In that case priority based factors might be used. These can be any one of the measured factors based on the network topology. Moreover, based on the topology and the availability of the network, we can choose either centralised or decentralised data repository mechanism.

The extension of this work will be improving the ranking policy by considering more factors in the Grid environment. Improvement will be possible in our prediction data delivery phase with XML-based framework for better integration with web services.

Acknowledgement: We would like to acknowledge Mr Panu Phinjareonphan and Prof Bill Appelbe for their valuable comments on the experimentation.

REFERENCES

Faerman, M., Su, A., Wolski, R., and Berman, F., 1999. Adaptive Performance Prediction for Distributed Data-Intensive Applications. In SC'99 *ACM/IEEE conference on Supercomputing.*

Globus. Globus Project. http://www.globus.org

Lee, J., Gunter, D., Stoufer, M., Tierney, B., 2002. Monitoring Data Archives for Grid Environments. In *SC'02, **ACM/IEEE conference on Supercomputing.***

Liu, C., Yang, L., Foster, I., and Angulo, D., 2002. Design and Evaluation of a Resource Selection Framework for Grid Applications. In *HPDC'02, 11th IEEE Symposium on High-Performance Distributed Computing.*

Swany, M., Wolski, R., 2002. Multivariate Resource Performance Forecasting in the Network Weather Service. In *SC'02, ACM/IEEE conference on Supercomputing.*

Syed, U., Yona, G., 2003. Using a Mixture of Probabilistic Decision Trees for Direct Prediction of Protein Function. In *RECOMB'03, 7th Annual International Conference on Computational Biology.*

Vazhkudai, S. and Schopf, J., 2002. Predicting sporadic grid data transfers. In *HPDC'02, 11th IEEE Symposium on High Performance Distributed Computing.*

Vazhkudai, S., Schopf, J.M., 2003. Using Regression Techniques to Predict Large Data Transfers. In *IJHPCA*'03, *The International Journal of High Performance Computing Application.*

Vazhkudai, S., Schopf, J.M. and Foster, I, 2002. Predicting the Performance of Wide Area Data Transfers. In *IPDPS'02, 16th International Parallel and Distributed Processing Symposium.*

Vazhkudai, S., Tuecke, S. and Foster, I., 2001. Replica Selection in the Globus Data Grid. In *CCGRID'01 1st IEEE/ACM International Conference on Cluster Computing and the Grid).* IEEE Press.

Wolski, R., 1997. Forecasting Network Performance to Support Dynamic Scheduling Using the Network Weather Service. In *HPDC'97, 6th IEEE Symposium on High Performance Distributed Computing.*

Wolski, R., 1998. Dynamically Forecasting Network Performance Using the Network Weather Service. In *JCC'98, Journal of Cluster Computing.*

Wolski, R., 2003. Experiences with Predicting Resource Performance On-line in Computational Grid Settings. In *ACM SIGMETRICS PER'03, ACM SIGMETRICS Performance Evaluation Review.*

Wolski, R., Spring, N., Hayes, J., 1999. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. In *JFGCS'99, Journal of Future Generation Computing Systems.*

Trace-Data. http://www.csm.ornl.gov/~vazhkuda/GridFTP-Predictions.zip

Zhang, X., Freschl, J., Schopf, J., 2003. , A Performance Study of Monitoring and Information Services for Distributed Systems. In *HPDC'03, **12th IEEE** International Symposium on High Performance Distributed Computing.*