Predicting Scientific Grid Data Transfer Characteristics

William Agnew
Georgia Institute of
Technology
wagnew3@gatech.edu

Michael Fischer
Univerity of
Wisconson-Milwaukee
fisch355@uwm.edu

Kyle Chard and Ian
Foster
University of Chicago
lastname@uchicago.edu

ABSTRACT

Big data scientists routinely transfer massive amounts of data. By understanding and modelling different aspects of these data transfers, we can make using big data more efficient and user-friendly. In this paper, we first develop a set of data storage location prediction heuristics. These heuristics help big data scientists manage and discover locations to transfer their data from and to. We show, via analysis of historical Globus operations, that our approaches can predict the storage locations accessed by users with 78.2% and 95.5% accuracy for top-1 and top-3 recommendations, respectively. Predicting transfer bandwidth allows for more optimal selections of data replicas to download from and for more optimal scheduling and routing of data transfers. We show that existing bandwidth prediction techniques perform poorly on real-world data and develop heuristics that (performance statistics).

1. INTRODUCTION

Big data is vital to the advancement of many areas of science, from high energy physics to materials science to molecular biology and will only continue to grow in importance. In this era of big data science, scientists are face with the challenge of managing and analysing huge amounts of data distributed over various storage repositories, compute resources, and personal computers. Therefore, we study how scientists use big data to make its use more efficient and user-friendly. By understanding and modelling aspects of big data use, we can target the development of new big data tools and allow existing systems to adapt to users' needs. [15], [14], [1] and others model and recommend scientific workflow components.

Transfers are a vital but often complex and time consuming aspect of big data. Scientists need to move vast quantities of data between collections sites, computing, visualization, and storage resources, and personal devices. This task is greatly complicated by the tens of thousands of available transfer locations. Commercial services, such as travel web-

sites, provide valuable user-specific recommendations derived from analysing huge amounts of usage data. These recommendations reduce the complexities associated with trawling through vast amounts of data and improve user experiences [2]. We explore here how recommendation approaches can be adapted and used to recommend storage locations to users, as shown in Figure 1.

Our approach builds upon a collection of specialized heuristics that consider unique features of scientific big data. We evaluate our approach using Globus [3], a hosted service that provides research data management capabilities across a vast network of distributed storage locations (called "endpoints").

Efficiently transferring data is also non-trivial. In scientific grids, it is common for copies of data to be stored in several locations [5], [6]. The location that data is transferred from may have a large impact on transfer time since thse grid's networks are generally complex, highly heterogeneous, and volatile. In addition, optimally scheduling and routing transfers poses a challenges. Static models of transfer throughput between endpoints fail to capture the highly dynamic nature of throughput; by developing a more accurate dynamic model of transfer throughput, one may route transfers more efficiently (cite predictive routing paper). In this paper, we develop such a dynamic model of transfer throughput between endpoints. While this problem has been studied before [11], [10], [9], [7], our work distinguishes itself in that we validate it on real transfer data from Globus, which, as we will later discuss, poses a much greater challenge to model than the synthetic datasets used in previous works.

We explore here how recommendation approaches can be adapted and used to recommend storage locations to users. Our approach builds upon a collection of specialized heuris-
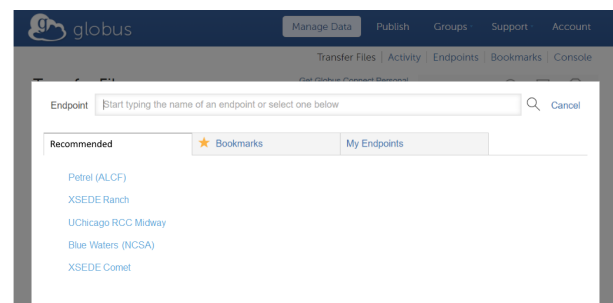
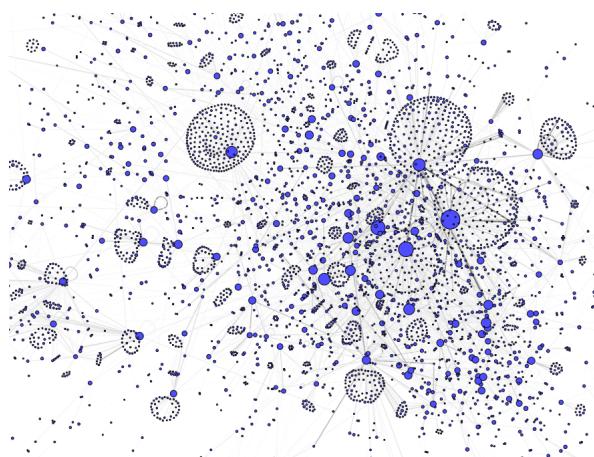Figure 1: Mockup of recommendation interface.

Figure 2: Globus Network. Endpoints are represented as vertices. Edges represent transfers between endpoints. Larger endpoints have transferred more frequently with more distinct endpoints. Shorter distances between endpoints represents more frequent transfers.

tics that consider unique features of scientific big data.

2. GLOBUS TRANSFERS

In this section we describe of the characteristics of Globus transfers to demonstrate the challenges of predicting transfer characteristics and motivate our solutions. Every Globus transfer is between a source and a destination endpoint, has a start and an end time, and transfers a certain number of bytes. Over 3.5 million Globus transfers have been conducted by over 16,000 users with over 23,000 endpoints. Below is a graph of considering endpoints as vertices and transfers as edges in Figure 2.

From this graph, we notice that there are a small number of endpoints that have been involved in many transfers with many endpoints, and a large number of endpoints that have only conducted a few transfers with a few endpoints. We examine this endpoint activity distribution in 3. These long-tailed distributions are a defining feature of Globus, and present a challenge for transfer characteristic prediction. The endpoints with low usage provide little historical information to base predictions off of, and the endpoints with high activity are often being used by different scientists with different usage patterns.

In addition, in 2 we see clusters of endpoints, often around a single high active endpoint. These clustered endpoints are likely related in some way, and we will try to characterize and use these clusters to make better predictions for endpoints with little historical activity.

Finally, although millions of transfer have been conducted, the data is very sparse. Only approximately 0.01% of potential user/endpoint pairs pairs are present, and only 0.006% of potential ordered endpoint pairs are present. This sparsity makes predicting transfer characteristic for pairs of endpoints, like throughput, challenging, as most potential pairs have never been seen before.

3. TRANSFER ENDPOINT RECOMMENDATION

In this part of the paper, we describe our methods for predicting the endpoints used in transfers. First we detail five endpoint recommendation heuristics. Then we combine these heuristics into a single superior heuristic using a neural network. Finally, we evaluate the accuracy of these heuristics and analyse some of their strengths and weaknesses. As we are interesting in improving user experience, we only consider endpoint recommendation for the approximately 1 million transfers initiated through the web GUI, and not through initiated programmatically through various APIs.

4. ENDPOINT PREDICTION HEURISTICS

Globus stores detailed records regarding users and their usage, including user institution and email. We use this information to develop a collection of endpoint recommendation heuristics. When queried with user ID, date, and n, each heuristic returns what it believes is the nth best endpoint recommendation for that user ID on that date.

History: The history heuristic does exactly what one would expect: it predicts that the nth best recommendation is the nth most recently used source (S) / destination (D) endpoint.

Markov Chain: The history heuristic matches users with endpoints, so we created a heuristic that correlates endpoints with endpoints, which we will call the Markov chain heuristic. The Markov chain heuristic recommends by maintaining a transition matrix of the probabilities of using each endpoint as a source/destination conditioned on a particular endpoint being previously used as a source/destination. These probabilities are estimated online by the Markov chain heuristic from the transitions is observes. According to this heuristic, the nth most likely endpoint for a user to use as a S/D endpoint is the nth most likely endpoint transition given that user's previous S/D endpoint choice.

Most Unique Users: The most unique users (MUU) heuristic, takes advantage of the long tailed distributions: a small number of endpoints are used by a larger number of different users. The nth best source/destination endpoint according to this heuristic the the endpoint with the nth most unique users who used that endpoint as a source/destination.

Thus far, all of our heuristics predict by using past transfers. This approach is known as collaborative filtering [cite]. Another major technique used in recommender system such as these are content based methods [cite], or heuristics that use information about the users and endpoints themselves, not just what they have transferred with. These methods have the advantage of not relying on the user or endpoint to have a transaction history, instead using other information like user email or institution to predict which endpoints the user will transfer with. On this section, we describe and benchmark two content based endpoint recommendation heuristics.

Institution: The institution heuristic maps users to the institution they belong to based on that user's provided email suffix. For example, wagnew3@gatech.edu would be mapped to the institution "gatech.edu". The n best source/destination endpoint for a user is the endpoint owned by a user belonging to the same institution that has been used a source/destination by the most unique users.

Endpoint Ownership: The endpoint ownership recom-

(a) Transfer volume per user   (b) Transfers per user   (c) Distinct endpoints per user
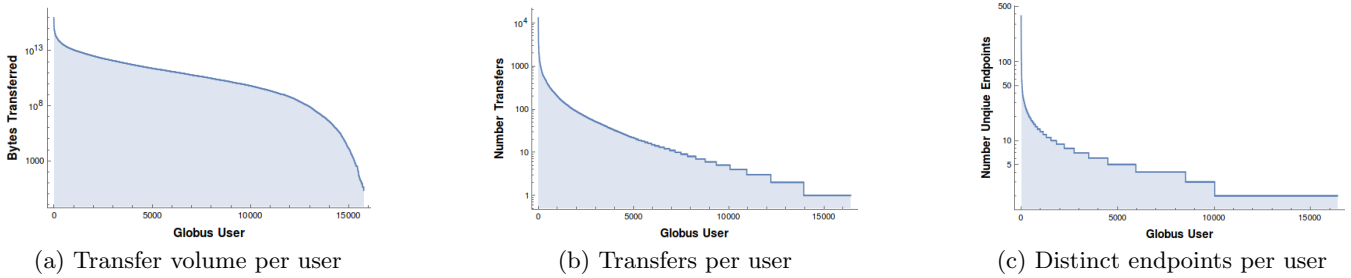
Figure 3: Long-tailed distributions of Globus usage

mends the nth most recent endpoint the user owns. The idea behind this heuristic is that if a user creates a new endpoint, that user is very likely to use it in a transfer soon.

5. COMBINING HEURISTICS

These heuristics model different aspects of the Globus ecosystem and therefore perform well for different classes of users. To combine the strengths of each heuristic we trained a deep recurrent neural network [4] on the series of endpoints chosen. Each heuristic, starting with the most accurate overall, outputs the top 3 recommendations that have not already been recommended and the weights it gives to those recommendations. The neural network is then given the heuristics' recommendation weightings and some additional user and endpoint information, and it re-weights heuristic recommendations and chooses the most highly re-weighted recommendation, as outlined in Figure 4. We give the neural network values representing the following information: date, user institution type (.edu, .gov, .com, or other), number of other Globus users in the same institution, relative user total number of transfers and total transfer volume, each heuristic opinion of how good its recommendations are overall (for example, the history heuristic reports how much transaction history that user has), each recommended endpoints's relative total number of transfers and total transfer volume, each heuristic's past accuracy for the user, and the which heuristic recommendations were correct and incorrect for the last recommendation made to that user. In short, we give the neural network a lot of potentially useful information about the recommendation problem and allow the network to determine which information is actually useful and how to integrate that information into its recommendations.

6. ENDPOINT PREDICTION RESULTS

We will measure our heuristic's accuracies with two metrics: transfer accuracy and user accuracy. Transfer accuracy is simply the fraction of all endpoint selections the heuristics correctly predicted, or $\dfrac{\text{number endpoints predicted correctly}}{\text{total number endpoints used in all transfers}}$. We give each heuristic's transfer accuracy in Figure 5. As we saw in the long-tailed distributions of the previous section, a few users have transferred a lot, and most users have transferred relatively little. While predicting the endpoints for these very active users is important, we cannot ignore these users with only few transfers, as they are likely least familiar with Globus and so would most benefit from good endpoint prediction. Therefore, to gauge how well our heuristics per-
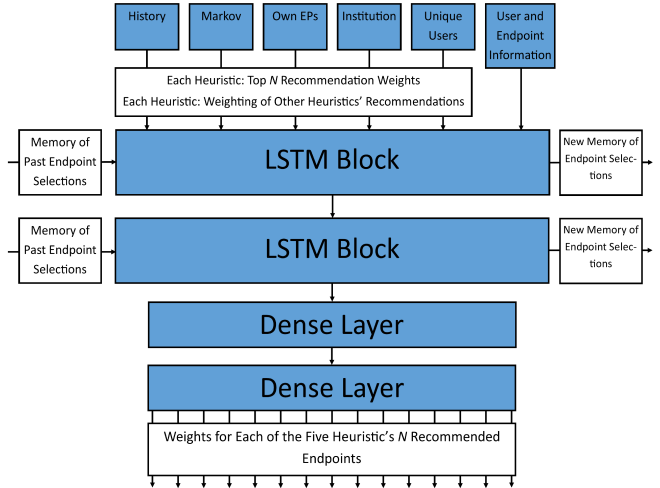

Figure 4: Neural Network Block. Takes as input heuristic recommendation weights and memory from past recommendations to the user, and outputs reweighed endpoint recommendations and updated recommendation memory.

form for each user, we introduce user accuracy, which is the average accuracy per user. The accuracy for a single user is simply $\dfrac{\text{number endpoints predicted correctly for user's transfers}}{\text{total number endpoints used in user's transfers}}$. User accuracy weights frequent Globus users equally with infrequent or new Globus users, giving us a better idea of how our heuristics will perform with new users. We give each heuristics user accuracy in 6.
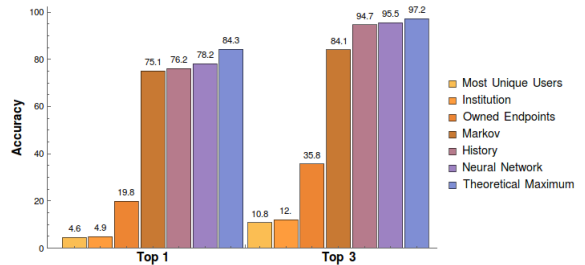

Figure 5: Transfer recommendation accuracy

The *most unique users*, *institution*, and *owned endpoints*

heuristics perform significantly worse than the other heuristics. However, when we compare heuristic accuracy against user history size (Figure 7), we see the *history* and *Markov* heuristics perform poorly when users have few previous transfers. By combining heuristics, the neural network is able to outperform all individual heuristics. For example, when there is little user history, the neural network increases the weighting of the *unique users*, *institution*, and *owned endpoints* heuristics.
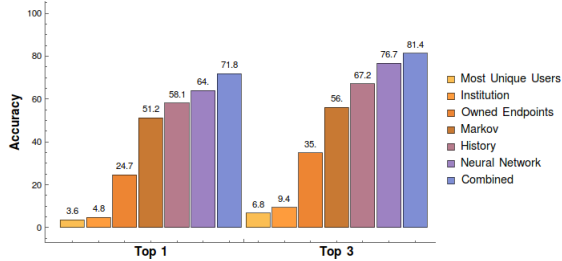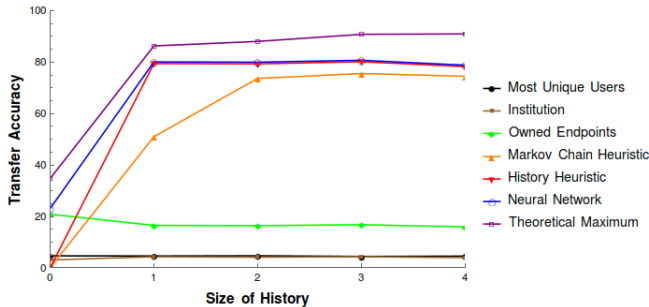

Figure 6: User recommendation accuracy

Figure 7: Top-1 transfer accuracy vs. history size

7. PREDICTING THROUGHPUT

In this part of the paper, we study predicting the throughput (transfer rate) between a source and a destination endpoint. Predicting this value accurately would allow for improved replica selection and transfer routing algorithms, increasing cloud transfer speed and reducing load on overutilized network links. Small transfers to probe throughput may seem like a simple solution, but as shown in [11] and [13], transfer probes must actually be quite large to be very accurate. This renders the approach impractical for our applications of interest, where we want to quickly estimate the throughput between many pairs of endpoints. To give an intuition for the structure of Globus throughput rates between different endpoints, we include a heatmap of top endpoints in Figure 8.
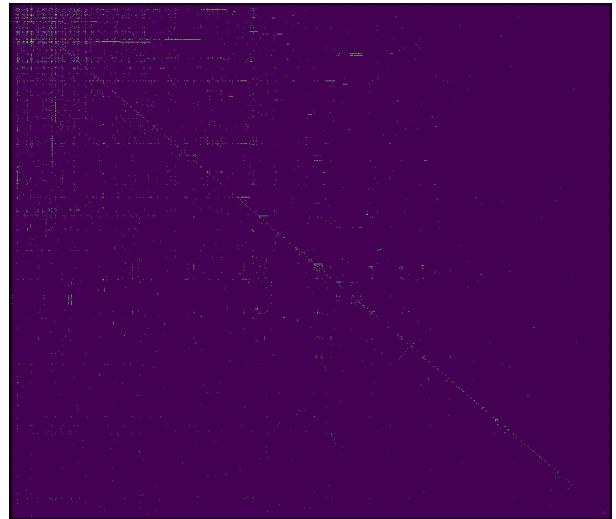

Figure 8: Heatmap of average throughput speed between top 1000 endpoints by total bytes transferred. Purple represents 0 bytes/sec (or no transfers between endpoint pair).

Our methodology is the same as endpoint prediction: we develop a set of heuristics, some utilizing content specific to scientific grids to produce accurate predictions for new endpoints pairs, and then combine these heuristics into a single superior heuristic using a neural network. Unlike endpoint prediction, we will predict throughput for all transfers. Much work has been done on throughput prediction already, and we include the best previous throughput prediction heuristics for comparison.

8. THROUGHPUT PREDICTION HEURISTICS

When given the source endpoint, destination endpoint, transferring user, date, and amount to be transferred, each throughput heuristic outputs its prediction of the throughput for that transfer and a rating of how confident it is in that transfer.

The first three heuristics, mean, median, and ARIMA, are taken from [11].

Mean: The mean heuristic predicts that the transfer rate between two endpoints is the mean of their past transfer rates. In addition to considering all past transfers, we use a version of the mean heuristic that only considers transfer that occurred in the last day, which may better adapt to changing network conditions.

Median: The median heuristic predicts that the throughput between two endpoints is the median rate of their past transfers. Like with the mean heuristic, we also use a version of the median heuristic that only considers transfers that occurred in the past day.

ARIMA: We use the version of ARIMA, or Auto-Regressive Integrated Moving Average, that is presented in [11]. This heuristic is in the same spirit as the endpoint prediction Markov heuristic: given the previous throughput between a pair of endpoints, it predicts the next throughput. More specifically, the ARIMA heuristic fits a line to the set of (previous throughput, next throughput) points for each pair of endpoints, and predicts throughput using this line model.

Polynomial Regression: As originally described in [8], the polynomial regression heuristic correlates the throughput of one endpoint pair with that of another, and fits a polynomial model to relate the two throughput variables. However, [8] only considers a grid with two endpoint pairs; Globus has nearly 40,000, so we must modify this algorithm to ensure that correlated endpoint pairs are used to model each other and that we do not have to re-fit 40,000 polynomials every time we update this heuristic. We do this by calculating the Pearson correlation between every pair of endpoints and only including the 10 most correlated endpoint pairs in each model. The throughput estimation of each model is scaled by the corresponding endpoint pair's relative correlation, and all throughput estimations are summed to obtain the heuristic's throughput estimation. In addition, after the first 5 transfers between a pair of endpoints, we only re-fit the models for that pair of endpoints every 20 transactions.

Recursive Least Squares and Kalman Filtering: As presented in [10], the recursive least squares and Kalman filtering heuristics build models that predict future throughput between a pair of endpoints from the past throughputs. Both models have a similar structure; we refer the reader to section 3.2 of [10] for more detail.

CDF Matching: [9] describes a throughput prediction heuristic that works by correlating one endpoint pair's throughputs with another's. More specifically, CDFs of throughput are created for both endpoint pairs by observing transfers. To predict the throughput of a new transfer for one endpoint pair, the position in the CDF of the last transfer throughput for the other endpoint is calculated. The heuristic then estimates the throughput as the throughput value in that position of the CDF of the first endpoint pair. This heuristic is also specialized to a two endpoint pair cloud, which is unrealistic. We extend it to an arbitrary number of endpoint pairs in a similar way as we did the polynomial regression heuristic: we only use the 10 most correlated endpoint pairs in each model, and we weight each endpoint pair's contribution towards the final throughput estimation by that endpoint pair's relative correlation.

NWS Ensemble [12] describes a set of heuristics-exponential smoothing and median with a variety of parameters- that are then combined into a single heuristic by choosing the historically most accurate heuristic to use.

We next present three heuristics which, unlike all other we have discussed, do not rely on an endpoint's history to make predictions, and instead exploit other correlations.

Institution Median The institutional median heuristic recommends the median throughput of all transactions from endpoints at the same institution as the source endpoint to endpoints at the same institution as the destination endpoint. As with the median heuristic, we also use a version of the institution median heuristic that only considers transactions from the past day.

User Median The user median heuristic recommends the median throughput of all transactions from endpoints owned by the same user as the source endpoint to endpoints owned by the same user as the destination endpoint. We also use a version of the user median heuristic that only considers transactions from the past day.

Global Median The global median heuristic recommends the median throughput of all transactions. This heuristic is targeted at endpoints from new users with no discernible in-stitution. Again, we also use a version of the global median heuristic that only considers transactions from the past day.

9. COMBINING HEURISTICS

As we did with endpoint prediction, we combine our collection of heuristics using a recurrent neural network on the series of throughputs of each endpoint pair.

10. RESULTS

Throughput prediction is significantly more difficult than endpoint prediction. The described heuristics are computationally intensive, and even the best do only about 10% better than just guessing the global median throughput every time.

11. CONCLUSION

In our work on endpoint prediction, developed and evaluated a collection of heuristics for recommending data locations. By combining these heuristics using a neural network, we correctly predict endpoints with 78.2% and 95.5% accuracy for top-1 and top-3, respectively. In future work we will integrate our recommendation system into Globus. We are also interested in characterizing, modelling, and improving the usage of scientific big data by analysing the performance of heuristics for different user profiles.

In our work on throughput prediction, we implemented and tested leading throughput prediction heuristics on real world data. We found that these heuristics' performances were much worse on this real data than on the contrived data they were originally tested on. In addition, we described weaknesses in the metric previously used to evaluate these heuristics, relative error, and, based off of network flow algorithms, proposed a new evaluation metric, absolute error, that more accurately describes how well a particular heuristic will perform in an application.

12. REFERENCES

[1] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 423–424. IEEE, 2004.

[2] A. Ansari, S. Essegaier, and R. Kohli. Internet recommendation systems. *Journal of Marketing research*, 37(3):363–375, 2000.

[3] I. Foster. Globus online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing*, 15(3):70, 2011.

[4] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[5] K. Holtman. Object level replication for physics. *Proceedings of 4th Annual Globus Retreat, Pittsburgh*, 2000.

[6] W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, and K. Stockinger. Data management in an international data grid project. In *International Workshop on Grid Computing*, pages 77–90. Springer, 2000.

[7] C. Lee, H. Abe, T. Hirotsu, and K. Umemura. Predicting network throughput for grid applications

on network virtualization areas. In *Proceedings of the first international workshop on Network-aware data management*, pages 11–20. ACM, 2011.

[8] J. Schopf and S. Vazhkudai. Predicting sporadic grid data transfers. In *11th IEEE International Symposium on High-Performance Distributed Computing (HPDC-11), IEEE Press. Edinburg, Scotland*, 2002.

[9] M. Swany and R. Wolski. Multivariate resource performance forecasting in the network weather service. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–10. IEEE Computer Society Press, 2002.

[10] A. T. Vakhitov and M. A. Panshenskov. Methods of linear transfer speed estimation in the data grid. In *Proceedings of the 1st ACM workshop on Data grids for eScience*, pages 29–34. ACM, 2009.

[11] S. Vazhkudai, J. M. Schopf, and I. Foster. Predicting the performance of wide area data transfers. In *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM*, pages 10–pp. IEEE, 2001.

[12] R. Wolski. Experiences with predicting resource performance on-line in computational grid settings. *ACM SIGMETRICS Performance Evaluation Review*, 30(4):41–49, 2003.

[13] E. Yildirim, J. Kim, and T. Kosar. Modeling throughput sampling size for a cloud-hosted data scheduling and optimization service. *Future Generation Computer Systems*, 29(7):1795–1807, 2013.

[14] J. Yu and R. Buyya. A taxonomy of scientific workflow systems for grid computing. *ACM Sigmod Record*, 34(3):44–49, 2005.

[15] J. Zhang, W. Tan, J. Alexander, I. Foster, and R. Madduri. Recommend-as-you-go: A novel approach supporting services-oriented scientific workflow reuse. In *Proceedings of the IEEE International Conference on Services Computing (SCC)*, pages 48–55, July 2011.