Touring Dataland?
Automated Recommendations for the Big Data Traveler

William Agnew
Georgia Institute of Technology
wagnew3@gatech.edu

Kyle Chard (advisor)
University of Chicago
chard@uchicago.edu

Ian Foster (advisor)
University of Chicago
foster@anl.gov

ABSTRACT

In this paper, we develop features and models to predict three important aspects of big data transfers: location, throughput, and errors. We develop and evaluate our models using more than 4 million historical transfers conducted by Globus. Our work is the first to study user-managed transfers in wide area networks comprised of user-owned endpoints, as opposed to previous work which focuses on experimental workloads, dedicated networks, and powerful computers. This real-world data presents several new challenges, including sparsity and lack of historical data, which we overcome by applying powerful ensemble machine learning algorithms and recurrent neural networks to summarize previous transfer information. Our approaches can predict the storage locations used with 78.2% and 95.5% accuracy for top-1 and top-3 recommendations, respectively. We model the throughput of transfers to within a factor of nearly two, and file transfer failures within approximately 1%.

ACM Reference format:
William Agnew, Kyle Chard (advisor), and Ian Foster (advisor). 2016. Touring Dataland?
Automated Recommendations for the Big Data Traveler. In *Proceedings of Supercomputing '16, Salt Lake City, UT, USA, November 2016 (SC'16),* 5 pages.
DOI: 10.475/123_4

1 PROBLEM AND MOTIVATION

Big data scientists, like travelers to a new land, are faced with the daunting tasks of discovering which (storage) locations contain interesting attractions (i.e., data), how to efficiently move between locations (i.e., optimizing transfer settings), and avoiding problems throughout their travels (i.e., transfer failures). These tasks are complicated by the rapid growth of research data, the increasing number of accessible storage systems, and the complexity of efficiently moving data between locations that span heterogeneous networks, hardware, and storage software. The Globus [7] research data management aims to simplify some theses activities for its more than 50,000 registered users. However, users are faced with more than 10,000 active data locations distributed around the world, each operating different storage and transfer infrastructure, and each connected by heterogeneous networks (e.g., low and high performance local and wide-area networks). A new user connecting to the Globus network has few guideposts as to which of these locations may be useful for finding or placing data, nor do they know how to optimize settings to maximize data transfer performance and minimize errors.

Commercial services, such as travel and video streaming services, provide valuable user-specific recommendations derived from analyzing huge amounts of usage data. These recommendations reduce the complexities associated with trawling through vast amounts of data and improve user experiences [2]. Recommendations have also been applied to support scientific users, for example by predicting workflow components [25], datasets [17], and coauthors [18]. However, to the best of our knowledge they have not been applied to the challenges of big data discovery and transfer.

We explore here how recommendation approaches can be applied to support users of big data. We specifically focus on three challenging recommendation problems: (1) recommending data storage locations to users; (2) predicting when errors are likely to occur when transferring data; and (3) predicting transfer performance based on implicit and explicit features (e.g., settings). In each case, we develop a collection of specialized heuristics that consider unique features of scientific big data. We then combine these heuristics into an online recommendation engine by developing an ensemble model using either recurrent neural networks or XGBoost [4].

1.1 Globus

Globus [7] provides a suite of data management capabilities primarily aimed at the research community. Amongst these capabilities it provides highly efficient and performant methods for accessing and transferring very large amounts of data. Globus comprises a globally distributed network of "endpoints"—storage systems that implement Globus interfaces to enable high performance data access and transfer. Over the past seven years Globus has been used to move more than 260 PB of data, in more than 4 million transfers that include more than 41 billion files. More than 65,000 Globus endpoints have been used to transfer data.

The complexity of our recommendation task is illustrated by Figures 1 and 2. Figure 1 shows a vast network of storage systems connected by user-specified data transfers. This figure highlights the sparsity of Globus usage: many endpoints have no transfers between them. (Similar sparsity is observed between users and endpoints). However, the figure also shows obvious clusters, often centered around a single popular endpoint. This information can, as we show, be used to develop accurate prediction heuristics. Figure 2 shows data transfer performance as a function of data transfer size and average file size. Transfers with the same data and average file size vary by as much as a factor of 10,000. In general, Globus

SC'16, Salt Lake City, UT, USA

DOI: 10.475/123_4

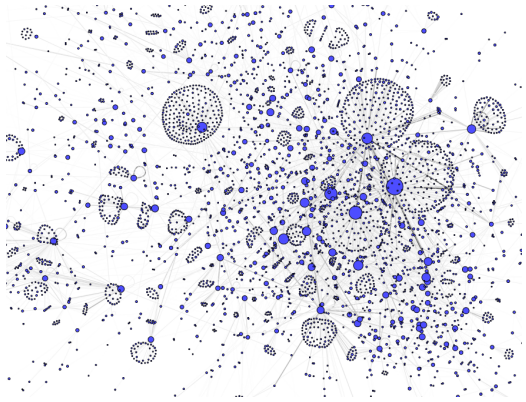

Figure 1: Portion of Globus Network. Endpoints are represented as vertices. Edges represent transfers between endpoints.

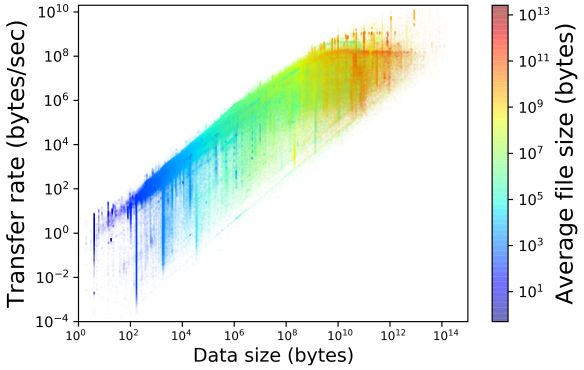

Figure 2: Transfer performance as a function of transfer size; color encodes average bytes per file.

usage follows long tail models: for example the number of transfers, amount of transfer, and transfer performance for users and endpoints [1]. These characteristics of the Globus network (especially sparsity) mean that traditional recommendation techniques, such as collaborative filtering, do not perform well on our target problems—as we showed in previous work [1].

2 BACKGROUND AND RELATED WORK

Our approach builds upon existing work to model data transfer performance and develop recommendation engines.

Researchers have long strived to accurately predict network transfer throughout and error conditions to optimize transfer parameters and identify and resolve software and hardware bottlenecks [3, 11, 13, 15, 19–23]. In addition to creating throughput optimization schemes, recent studies of throughput prediction have studied the factors that affect transfer rate, including the effects of disk and raid controller bandwidth, data compression, and parallelism [10]; the effects of transfer protocol characteristics, such as buffer and window size [24]; and the detrimental effects of concurrent transfers and prioritization of transfers [12]. Our work is differentiated both by its focus and also its approach, as none of these efforts use recurrent neural networks to create powerful yet concise summaries of past transfers for use in prediction.

To the best of our knowledge, no prior research has focused on predicting data transfer locations. Perhaps the most similar efforts are those that relate to predicting groups of files that are frequently transferred together. By identifying these groups, transfer performance can be improved by using better caching and job scheduling algorithms [6]. The families of heuristics we have developed may also provide value in this domain.

General approaches for recommendations are typically based on either collaborative or content-based filtering. Collaborative filtering [16] builds models of a user's past behavior to recommend items based on similar decisions made by others. Content-based filtering relies on a set of item characteristics to recommend items with similar characteristics. While others have used hybrid models to combine these recommendation approaches, our use of neural networks is amongst the first examples of its use in building recommendation engines [5, 9, 14].

3 APPROACH AND UNIQUENESS

We aim to tackle three separate recommendation problems: location, throughput, and failures. In each case we apply a similar approach: derive a set of heuristics that leverage unique characteristics of the recommendation problem (including information about users, data, and endpoints), evaluate in which situations each heuristic does well, combine the benefits of each heuristic using ensemble models.

Our work is distinguished from previous work by its study of user-managed transfers in wide area networks comprised of user-owned endpoints. Previous work has focused solely on experimental transfers or transfers using dedicated networks between a few powerful computers. In particular, all of the voluminous previous work on throughput prediction focuses on artificial networks, and thus ignores a critical problem of real-world throughput prediction: sparsity and a lack of absolute information about network contention. In addition, as we detail in the following two sections, we use a novel set of powerful features to feed into ensemble methods to predict locations, failures, and throughput.

3.1 Prediction heuristics

Globus stores detailed records regarding users, endpoints, and transfers. This information includes user name, email address, institution, etc; endpoint name, description, owner, location, default performance settings, etc.; and transfer source, destination, user, number of files, errors, transfer time, throughput, etc. In some cases we have derived new features from Globus data such as institution from email addresses, location from ips, and distance from location. In total, we have more than 70 raw and derived user and endpoint features. We use this rich source of information to develop a collection of heuristics for predicting endpoint locations, failures, and throughput.

3.1.1 Endpoint prediction heuristics. When recommending endpoints to users we have developed a set of heuristics that leverage the features described above. These heuristics are: *History:* A baseline heuristic that predicts the most recently used source (S) / destination (D) endpoint. *Markov Chain:* Using a user/endpoint

transition matrix the heuristic predicts the S/D endpoint based on the most likely endpoint transition given that user's previously used S/D endpoint. *Most Unique Users:* The most likely S/D endpoint is the S/D endpoint with the most unique users. *Institution:* The most likely S/D endpoint for a user is the endpoint with the most unique users that is also owned by a user belonging to the same institution. *Endpoint Ownership:* The most likely endpoint is the endpoint most recently created by the user.

3.1.2 Throughout and failure features. In addition to over 70 user and endpoint features described at the start of this section, we have implemented dozens of throughput prediction heuristics from previous work (some of which are referenced in §2). In each case the heuristic predicts throughput for given input parameters (e.g., endpoint, data size, time). We augment the heuristic to also output current accuracy for predicting transfers for that endpoint pair, and, where applicable, a confidence score for the throughput prediction heuristic's prediction. We use this same set of heuristics when predicting file failures based on the intuition that transfers with many file failures are unlikely to have good transfer parameter configurations or network connections and therefore will have lower throughput.

3.2 Ensemble recommendation methods

The various endpoint prediction heuristics model different aspects of the Globus ecosystem and therefore perform well for different classes of users, endpoints, and transfers. To combine the strengths of each heuristic we use ensemble methods based on deep recurrent neural networks (RNNs) [8] and XGoost on the series of prediction heuristic outputs.

In the case of endpoint location prediction the recurrent neural network is given each heuristics' recommendation weightings and some additional user and endpoint information, it re-weights heuristic recommendations and chooses the most highly re-weighted recommendation.

When predicting throughout and failures we use a slightly different model. Here, we attempt to combat the sparsity problem by embedding the transfer history between each pair of endpoints using a recurrent neural network. That is, we use the recurrent neural network to produce a concise summary of past transfers and features of endpoint pairs with the aim that this summary will prove useful to predicting the throughput or file failures of the current transfer. The intuition behind this approach is that the recurrent neural network can learn to quickly recognize different types of endpoints (e.g. personal laptop, supercomputer) and networks that connect them as well as transfer trends (e.g., time-based contention) by training on millions of previous transactions. By quickly producing accurate "guesses" about endpoint pairs, the RNN embedding helps our ensemble method overcome the lack of transfer history between many endpoint pairs. Our RNN embedder was created by training a multilayer GRU outputting to two dense layers on the series of transfers from one endpoint pair to another to predict throughput or file failures, and then taking the output of the GRU as the embedding.

Our use of RNNs to analyze an entire series of historical recommendations to inform current recommendations has been validated by both our results and concurrent work on using recurrent neural networks for collaborative filtering [5, 9, 14]. To the best of our knowledge we are the first to use recurrent neural networks to learn an embedding of past recommendation instances, allowing the RNN's ability to effectively learn series to be combined with models better suited for the final recommendation task, in our case, XGBoost.

4 RESULTS AND CONTRIBUTIONS

We evaluate our approach by splitting historical Globus usage data into training and evaluation subsets and measuring the accuracy of our techniques.

4.1 Datasets

We use two different datasets to evaluate our prediction models.

User transfers: To model user interactions with Globus we created a dataset that includes only transfers orchestrated via the Globus web interface. This dataset aims to remove automated transfers that have been executed programmatically. The entire dataset comprises more than 1 million transfers up until May 2015.

Data size cuts: Our second dataset aims to segment Globus transfers into various sets based on transfer size. The reason for doing so is that larger transfers are intuitively easier to predict as they are more likely to use high performance computers and dedicated networks—they can therefore be more accurately modeled using our heuristics. Smaller sized transfers are intuitively harder to predict as they are affected by short term network dynamics to a greater extent. Specifically, our sets are: transfers of more than 10GB, 1GB, 100MB, and 10MB.

4.2 Endpoint Prediction Results

We explore endpoint recommendation accuracy by evaluating the historical *user transfers* dataset. We train our endpoint recommendation RNN on the first 500,000 transfers and evaluate accuracy using the remaining 450,000 transfers. We measure performance using two metrics. Transfer accuracy (Figure 3): the average number of endpoints predicted correctly for all transfers; and user accuracy (Figure 4): the average accuracy per user, where a user's accuracy is the percentage of that user's endpoints that are correctly predicted. We compare accuracy for individual heuristics and the RNN when predicting top-1 and top-3 recommendations.

Our results show that the ensemble RNN outperforms any single heuristic for each prediction task. We also include the *theoretical maximum* which represents the best possible result had an oracle selected from all heuristic predictions. The RNN achieves close to the theoretical maximum in all cases. Looking at individual heuristics, the *most unique users*, *institution*, and *owned endpoints* heuristics perform significantly worse than the other heuristics. When heuristic accuracy is compared with user history size (Figure 5), we see that the *owned endpoints* heuristic performs best when no previous information is known. By combining heuristics, the RNN is able to outperform all individual heuristics, even with little history. This is because the RNN increases the weighting of the *unique users* and *owned* heuristics.

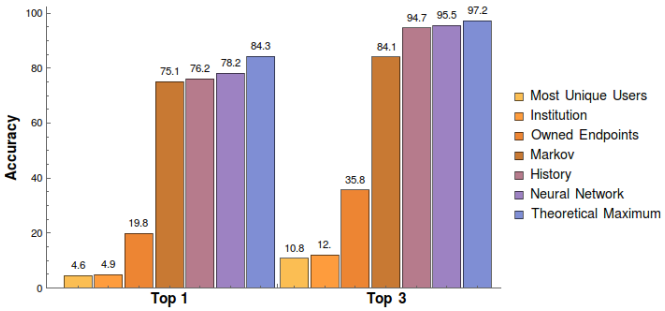

Figure 3: Transfer recommendation accuracy

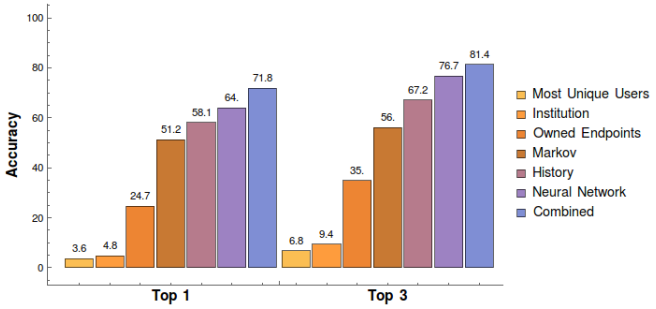

Figure 4: User recommendation accuracy

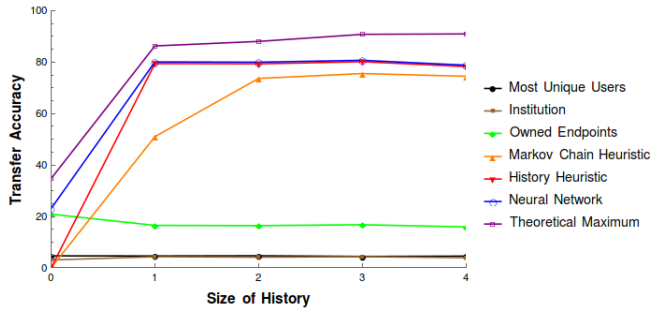

Figure 5: Top-1 transfer accuracy vs. history size

4.3 File Failure Prediction

We now turn our attention to predicting the percentage of file failures in a transfer. For this experiment we use the *data size cuts* datasets. We train two ensemble models to predict failures. The first uses raw user and endpoint features as well as several throughput prediction heuristics (called XGBoost). The second model combines these same raw features and heuristics with the RNN history embedding (called XGBoost + RNN Embedding). Both models use the XGBoost [4] method to combine predictions. To evaluate the performance of the different models, we use absolute error between the predicted and the actual percentage of failed files in a transfer. Figure 6 shows the results. In all cases our methods are ale to achieve high accuracy with errors lower than 0.016. The model that includes RNN embeddings outperforms the XGBoost models on all but one cut. Surprisingly, our models are able to more accurately predict failures in the much larger cuts. We are actively investigating this behavior.

Figure 6: File failure prediction error

4.4 Throughput Prediction

We now evaluate the performance of our approach on the hardest prediction task: predicting throughput performance. In the following experiments we use the *data size cuts* datasets. We use the same two models as described above, with the RNN history embedder trained to predict throughput rather than failures. As is common in previous work [11], we configure our ensemble models to predict the log of the throughput, exponentiating to obtain the actual throughput.

To evaluate the performance of the different models, we use the factor our prediction differs from the actual value by:

$$\max\left(\frac{predicted}{actual}, \frac{actual}{predicted}\right)$$

. This metric is intuitively similar to relative error, but, unlike factor error, relative error suffers from a bias towards lower predictions, which becomes significant with errors of the magnitude we encountered while modeling throughput.

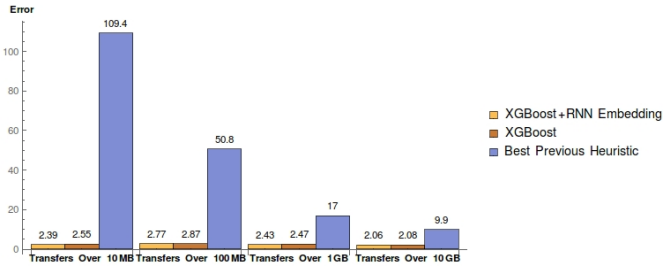

Figure 7: Throughput prediction error

Figure 7 shows the results of our two models and also includes the best performing single heuristic for each dataset. These results show that XGBoost performs remarkably well compared to previous heuristics. Including the RNN transfer history embedding further improves accuracy. As expected, accuracy improves for the larger cuts that include fewer (but larger) transfers. For all cuts, our best model is off by a factor of approximately two on average, whereas the best previous model is off by as much as 100. We attribute this significant increase in accuracy to the sophistication of XGBoost compared to the previous model, which were generally variants of

moving averages or polynomial fitting, and to the log transform, which takes the prediction problem from one where the dependent variable is exponential in the inputs to one where the dependent variable is closer to linear, allowing models designed to predict approximately linear relations to perform much better.

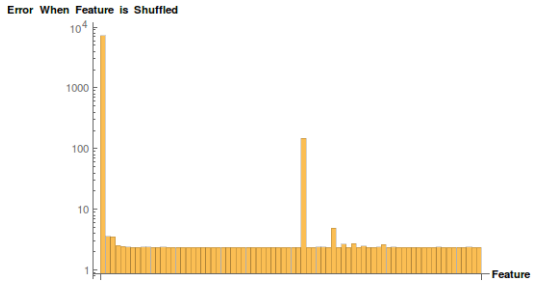

Figure 8: Errors when most important features are shuffled

Finally, we now explore which features are most important for determining throughput. For each feature we randomly shuffle the values of that feature among samples, and then evaluate our model on the data with the shuffled feature column. The results are shown in Figure 8. Our results show that two features affect accuracy more significantly than all others: transfer amount (increasing error to approximately 7000) and the user-supplied transfer duration deadline (increasing error to approximately 150). The correlation of transfer amount is expected (as seen in Figure 2); however, the importance of the transfer deadline suggests that the Globus system has considerable influence on transfer throughput. This validates our approach, and shows that our approaches could be used to further optimize transfer performance.

5 SUMMARY AND FUTURE WORK

We have developed several powerful and general techniques for understanding big data transfers in real-world networks. Using XGBoost, RNN Embeddings, and an ensemble of previous heuristics, we can predict the throughput of transfers to within a factor of approximately two, and the percentage of files that will fail to within an absolute error of roughly 1%. We have also developed and evaluated a collection of heuristics for recommending endpoints to users. By combining these heuristics using an RNN, we correctly predict endpoints with 78.2% and 95.5% accuracy for top-1 and top-3, respectively. In future work we aim to integrate these online models within Globus to improve user experience, automate and optimize transfer parameter settings, diagnose failure conditions, and prioritize transfers based on expected performance and failures. Finally, our analysis of throughput feature importance shows that Globus has considerable control over transfer throughput. We would like to determine which transfers are being prioritized, whether this bandwidth distribution is good from a user perspective, and, if not, how to improve it.

REFERENCES

[1] W. Agnew, M. Fischer, I. Foster, and K. Chard. 2016. An Ensemble-Based Recommendation Engine for Scientific Data Transfers. In *Seventh International Workshop on Data-Intensive Computing in the Clouds (DataCloud)*. 9–16. DOI: http://dx.doi.org/10.1109/DataCloud.2016.005

[2] Asim Ansari, Skander Essegaier, and Rajeev Kohli. 2000. Internet recommendation systems. *Journal of Marketing research* 37, 3 (2000), 363–375.

[3] Engin Arslan, Kemal Guner, and Tevfik Kosar. 2016. HARP: predictive transfer optimization based on historical analysis and real-time probing. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press, 25.

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.

[5] Robin Devooght and Hugues Bersini. 2016. Collaborative filtering with recurrent neural networks. *arXiv preprint arXiv:1608.07400* (2016).

[6] Shyamala Doraimani and Adriana Iamnitchi. 2008. File grouping for scientific data management: lessons from experimenting with real traces. In *17th international symposium on High performance distributed computing*. ACM, 153–164.

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

[8] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[9] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 241–248.

[10] Eun-Sung Jung, Rajkumar Kettimuthu, and Venkatram Vishwanath. 2013. Toward optimizing disk-to-disk transfer on 100G networks. In *IEEE International Conference on Advanced Networks and Telecommunications Systems*. IEEE, 1–6.

[11] Rajkumar Kettimuthu, Gayane Vardoyan, Gagan Agrawal, and P Sadayappan. 2014. Modeling and optimizing large-scale wide-area data transfers. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 196–205.

[12] Rajkumar Kettimuthu, Gayane Vardoyan, Gagan Agrawal, P Sadayappan, and Ian Foster. 2015. An elegant sufficiency: load-aware differentiated scheduling of data transfers. In *International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 46.

[13] JangYoung Kim, Esma Yildirim, and Tevfik Kosar. 2015. A highly-accurate and low-overhead prediction model for transfer throughput optimization. *Cluster Computing* 18, 1 (2015), 41–59.

[14] Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. 2016. Collaborative recurrent neural networks for dynamic recommender systems. *Journal of Machine Learning Research* (2016), 1–16.

[15] MD Nine, Kemal Guner, and Tevfik Kosar. 2015. Hysteresis-based optimization of data transfer throughput. In *Proceedings of the Fifth International Workshop on Network-Aware Data Management*. ACM, 5.

[16] Paul Resnick and Hal R. Varian. 1997. Recommender Systems. *Commun. ACM* 40, 3 (March 1997), 56–58. DOI: http://dx.doi.org/10.1145/245108.245121

[17] A. Singhal, R. Kasturi, V. Sivakumar, and J. Srivastava. 2013. Leveraging Web Intelligence for Finding Interesting Research Datasets. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1. 321–328. DOI: http://dx.doi.org/10.1109/WI-IAT.2013.46

[18] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. 2011. Co-author relationship prediction in heterogeneous bibliographic networks. In *IEEE International Conference on Advances in Social Networks Analysis and Mining*. 121–128.

[19] Martin Swany and Rich Wolski. 2002. Multivariate Resource Performance Forecasting in the Network Weather Service. In *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing (SC '02)*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1–10. http://dl.acm.org/citation.cfm?id=762761.762812

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

[21] Sudharshan Vazhkudai and Jennifer M Schopf. 2002. Predicting sporadic grid data transfers. In *High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium on*. IEEE, 188–196.

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

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

[24] Daqing Yun, Chase Q Wu, Nageswara SV Rao, Bradley W Settlemyer, Josh Lothian, Rajkumar Kettimuthu, and Venkatram Vishwanath. 2015. Profiling transport performance for big data transfer over dedicated channels. In *International Conference on Computing, Networking and Communications*. IEEE, 858–862.

[25] J. Zhang, W. Tan, J. Alexander, I. Foster, and R. Madduri. 2011. Recommend-As-You-Go: A Novel Approach Supporting Services-Oriented Scientific Workflow Reuse. In *IEEE International Conference on Services Computing*. 48–55.