A comprehensive study of wide area data movement at a scientific computing facility

Zhengchun Liu,[‡*] Rajkumar Kettimuthu,[*] Ian Foster,[*†] and Yuanlai Liu[§]
[*] Data Science and Learning Division, Argonne National Laboratory, 9700 Cass Ave., Lemont, IL 60439, USA
{zhengchun.liu, kettimut, foster}@anl.gov, yliu158@ucr.edu
[†] Department of Computer Science, University of Chicago, Chicago, IL 60637, USA
[‡] Computation Institute, University of Chicago, Chicago, IL 60637, USA
[§] University of California, Riverside, 900 University Ave., Riverside, CA 92521, USA

Abstract—**Wide-area data transfer is central to distributed science. Network capacity, data movement infrastructure, and tools in science environments continuously evolve to meet the requirements of distributed-science applications. Research and education (R&E) networks such as the U.S. Department of Energy's Energy Sciences network and Internet2 provide multiple 100 Gbps backbone networks. Large scientific facilities and research institutions have 100 Gbps wide-area network connectivity, and 10 Gbps wide-area network connectivity is common for a lot of R&E institutions. Many of these institutions employ Science DMZs, dedicated data transfer node(s), and high performance data movement tools to improve wide area data transfer performance. Large facilities may use 10 or more dedicated data transfer nodes to meet the needs of their users. In this work, we analyze various logs pertaining to wide area data transfers in and out of a large scientific facility to obtain insights on data transfer characteristics and behavior. We also show some of the inefficiencies in the state-of-the-art data movement tool and discuss approaches to address these inefficiencies.**

I. Introduction

Large data transfers over wide area networks are an inherent part of many scientific workflows [1]. Thus, scientific facilities and research and education institutions deploy dedicated infrastructure and high performance tools to handle such transfers. Tools such as Globus GridFTP [2], XDD [3], bbcp [4], and FDT [5] use parallel streams [6, 7] and other optimizations [8] to provide high performance. Many institutions use Science DMZs [9] to eliminate institutional firewall bottlenecks, deploy high-speed data transfer methods such as GridFTP [2] on dedicated data transfer nodes (DTNs) [10], and provide a 'fire and forget' data movement capability to their users via the Globus transfer service [11].

Yet despite much work on accelerating individual file transfers and scheduling of multiple transfers to improve aggregate performance (e.g., [8, 12–23]), the performance achieved by transfers in practice is usually much lower than line rates [24]. Given the importance of data transfer to science and the large investments that continue to be made in data transfer infrastructure, it is important to identify bottlenecks and explain why transfers achieve the performance that they do.

We have previously used Globus transfer service logs to explain the factors that affect wide area data transfer performance [24]. We also used four years of GridFTP logs collected from over 60,000 servers in conjunction with Globus transfer service logs for the same period of time to characterize file types transferred, transfer performance, and user behavior [25]. These studies yielded unique insights into the aggregate properties of wide area data transfers and the performance achieved "in the wild."

In this work, we zoom in to study the wide area data transfer characteristics of a single (anonymous) scientific computing facility, '*BigSite*.' We examine transfers performed during the year 2017 at three different levels: that of user transfer requests; that of individual file transfers; and that of TCP flows. In general, the processing of a single transfer request can involve the transfer of one or more individual files, the transfer of a single file may involve multiple concurrent TCP flows, and a single TCP flow may be involved in multiple file transfers at different times. Thus, the interactions between these three different levels can be complex. Fortunately, we have access to time-synchronized logs from each level, allowing us to study these interactions in detail. Specifically, we use transfer logs collected by the Globus transfer service, file logs collected by the GridFTP server on each DTN, and TCP logs collected by TSTAT [26]. Using these logs, we present insights on transfer, file, and flow characteristics, and identify areas for improvement in transfer performance and resource utilization. Even though we use logs from a single facility, this study shows that useful insights can be obtained by combined analysis of logs from different layers of the data movement stack. Moreover, the findings on the areas for performance improvement are applicable for wider audiences; and some of the findings on flow, file, and transfer characteristics are applicable to other large facilities. We believe that our study will help resource providers optimize the resources used for data transfer and will help researchers and tool developers optimize data transfer protocols and tools.

The rest of the paper is as follows. In §II, we introduce the data transfer tools and the scientific computing facility that are the subjects of this study, and §III, we present the characteristics of user data transfer requests, files transferred, and network flows in and out of the facility. In §IV, we study how the data sets in the user transfer request are distributed among different data transfer processes and TCP flows. In §V, we present two potential future research opportunities to improve the efficiency of wide-area data transfer infrastructure and tools. We review related work in §VI and conclude in §VII.

II. Background

We briefly describe the facility whose data transfer characteristics we study in this work and then provide background on the tools whose logs we use for this study.

A. *The* BigSite *facility*

The scientific facility that we study here, *BigSite*, is a high-performance computing facility that serves several thousand users researching a wide range of problems in various science disciplines. *BigSite* has a total of 10 DTNs, of which nine are available for file transfers between facility storage and storage at other sites. (The tenth is dedicated to HPSS, and is not considered here.) Each DTN has multiple 10 Gbps Ethernet links for transfers over the network and multiple IB connections to *BigSite* filesystems.

Figure 1 shows the number of TCP flows between this facility and different cities worldwide during the year 2017. These data are obtained from TSAT logs (see the next subsection). We used the MaxMind IP geolocation service [27] to obtain approximate endpoint locations.

B. *TSTAT*

The TCP STatistic and Analysis Tool (TSTAT) [26] analyzes network traffic and stores a complete transport-level log of all measured parameters. It can be used to collect many different statistics for TCP, UDP, and RTP/RTCP traffic. For TCP connections, congestion window size, out-of-sequence segments, duplicated segments, number of bytes and segments retransmitted, and RTT are some of the statistics that it can collect. TSTAT distinguishes between completed and not completed flows, and between clients (hosts that actively open a connection) and servers (hosts that passively listen for connection requests). TSTAT also records UDP messages. However, since UDP communication contributes less than 0.01% of the total bytes moved from/to *BigSite*, we did not consider UDP communications in this study.

The *BigSite* DTNs support data transfer via Globus GridFTP, BBCP, rsync [28], SCP/SFTP [29], and HTTP. As the TSTAT TCP logs include all TCP flows and furthermore indicate the port used for each flow, we can label transfers according to the tool used and calculate the aggregate bytes moved with each tool, as shown in Figure 2. We observe that Globus GridFTP traffic constitutes more than 70% of total traffic and that ∼22% of the GridFTP traffic (∼16% of total traffic) is driven by the cloud-hosted Globus transfer service. (The remainder of the GridFTP traffic is likely associated with specialized applications such as those used in high energy physics.)

C. *The GridFTP protocol*

GridFTP, an extension of the standard FTP protocol for high performance, better security, and improved reliability, is a widely used protocol for science data transfers. The GridFTP protocol was standardized through the Open Grid Forum and multiple implementations exist, of which Globus [2] and dCache [30] are the most popular. *BigSite* has Globus GridFTP servers deployed on its DTNs. Globus GridFTP server logs information about each file it transfers. As shown in Table I, GridFTP transfer log records include information such as file size, transfer duration, number of parallel TCP streams, TCP buffer size, and block size.

D. *The Globus transfer service*

The Globus transfer service is a cloud-hosted software-as-a-service implementation of the logic required to orchestrate file transfers between pairs of storage systems [11, 31]. The service also supports data sharing, publication, and discovery; we focus here on its transfer capabilities. Between 2014/01/01 and 2018/01/01, 26,100 users made 4,813,091 transfers via Globus, totaling 13.1 billion files and 305.8 PB. These transfers involved 41,900 unique endpoints and 71,800 unique source-to-destination pairs [25].

III. Flow, file, and request characteristics

We next examine the characteristics of TCP flows (from TSTAT logs), files transferred (from GridFTP logs), and user transfer requests (from Globus transfer service logs; note that each transfer request may involve multiple files and/or directories).

A. *Flow characteristics*

TSTAT [26] recorded more than 81 million TCP flows between *BigSite* DTNs and 60,201 unique IP address from 2017/01/01 to 2017/12/31. Figures 3 and 4 show the average number of TCP flows and average data moved, respectively, per hour for each day of the week in 2017. Not surprisingly, we see that both the number of flows and data transferred are greater on weekdays than in weekends. Interestingly, we see more activity earlier in the day on weekdays.

Figure 5 shows the cumulative distribution of the duration of all TCP flows. We see that there are many short-lived TCP connections, with a remarkable 75% of all flows lasting less than five seconds. Note, however, that short-lived flows account for little data transport. For example, the duration of 34.7% of the total flows is less than 1 second, but these flows contribute only 0.1% of the total bytes moved to/from the facility DTNs.

B. *Characteristics of the files transferred*

Figure 6 shows the distribution of the size of individual files transferred by GridFTP, inbound and outbound. It is clear that most files are small. The 50th and 75th percentile values for outbound files are 32 KB and 256 KB, respectively. The inbound files are slightly bigger: the 50th and 75th percentiles are about 128 KB and 2 MB, respectively.

C. *Characteristics of Globus transfer requests*

Figure 7 shows the cumulative distribution of Globus transfer request size, separately for inbound and outbound transfer requests. The median dataset sizes of outbound and inbound transfers are 9.2 GB and 2.1 GB respectively.

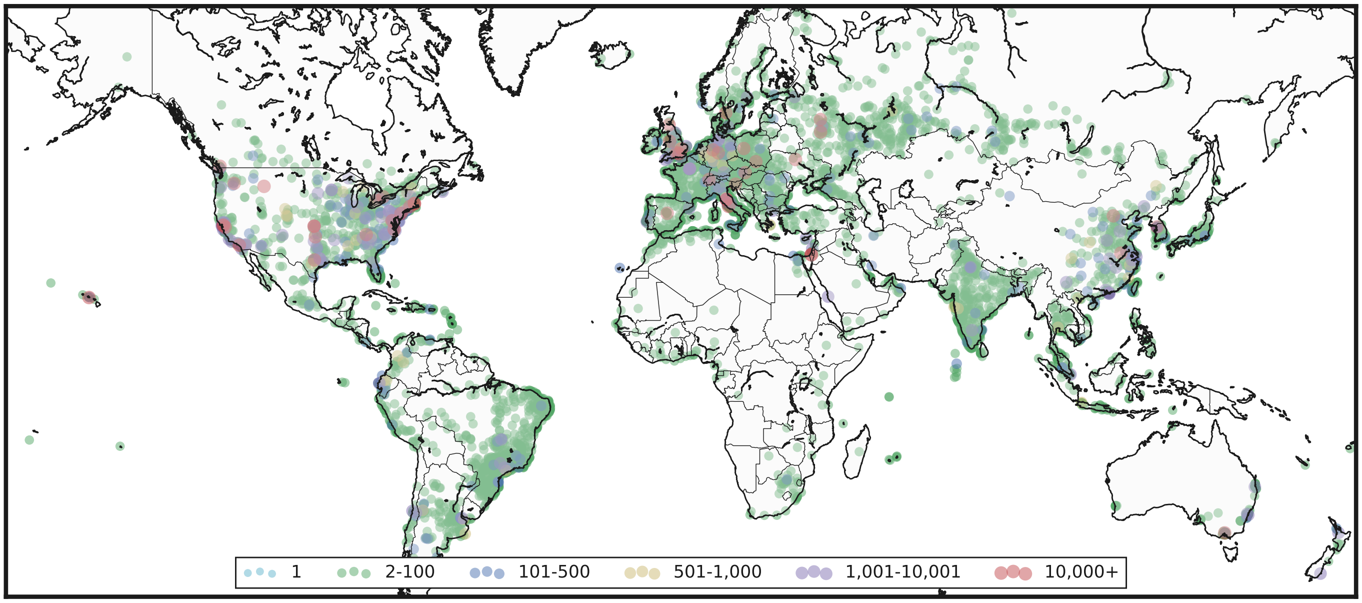

Fig. 1. Geographical distribution of TCP flows to/from *BigSite* DTNs in 2017, with color used to code number per city.

TABLE I
FIELDS IN (ANONYMIZED) GRIDFTP LOGS. NOT SHOWN ARE FIELDS WITH VALUES THAT ARE IDENTICAL OR EMPTY IN ALL LOGS.

Key	**Example value**	**Description**
DATE	20180422105055.986780	Date and time (with micro-seconds) when the file transfer completed.
HOST	dtn.example.org	The IP of the server that runs GridFTP.
PROG	globus-gridftp-server	Application name of the GridFTP.
START	20180422102035.460756	Date and time when starting file transfer.
USER	zcl	Username of the user when initiating the transfer.
FILE	/homes/zcl/test.dat	Name of the transferred file.
BUFFER	332800	TCP buffer size (if 0 system defaults were used).
BLOCK	4194304	Size of the data block read from the disk and posted to the network.
NBYTES	3790026	Size of the transferred file in bytes.
STREAMS	4	Number of parallel TCP streams.
STRIPES	1	Number of stripes used on this end of the transfer.
DEST	140.221.11.138	IP of remote server.
TYPE	RETR	Transfer type (FTP RFC959 commands); e.g., RETR is a send, and STOR is receive.
CODE	226	RFC959 completion code. 226 indicates success, 5xx or 4xx are failure codes.
TASKID	59e55c52-461b-11e8-8e5a-0a6d4e044368	Globus-generated taskid (none if it is not a Globus transfer).
RETRANS	2,0,4,7	Number of retransmitted TCP packets per stream.

Figure 8 shows the cumulative distributions of the throughput for Globus transfers, separately for inbound and outbound transfers. We see that overall performance is low: 50% of incoming transfers have throughput $\leq$ 16 Mbps and 50% of outgoing transfers have throughput $\leq$ 512 Mbps. We note that there are usually multiple concurrent transfers from different users, thus the throughput of any one of the transfers may not represent the overall performance of the DTNs.

IV. A TOP-DOWN VIEW OF WORKLOAD DISTRIBUTION

The Globus transfer service uses one or more GridFTP server processes at the source and destination to transfer the file(s) listed in a transfer request. Based on the number and sizes of files in a request, it uses a heuristic to determine the number of GridFTP server processes to use, a number that is referred to as *concurrency* (C). These processes may be on different DTNs, if a site has more than one DTN. The files in the transfer request are distributed among these server processes. The number of files assigned to each server process is influenced by another optimization parameter called pipeline depth, D. Pipeline depth specifies the number of files to be queued in each GridFTP server process in advance, without waiting for the previously queued requests to finish. Pipelining speeds up transfers involving many small files.

Each GridFTP server process, upon receiving a file transfer request, splits the file into multiple chunks and transfers the chunks in parallel over a specified number of TCP connections. The number of parallel TCP connections to use is specified by the client (Globus, in this case). Note that each server process may have to transfer multiple files (depending on the number of files in the Globus transfer request) and uses the same TCP connections to transfer all files requested in a single client session (a client session corresponds to a Globus transfer request).

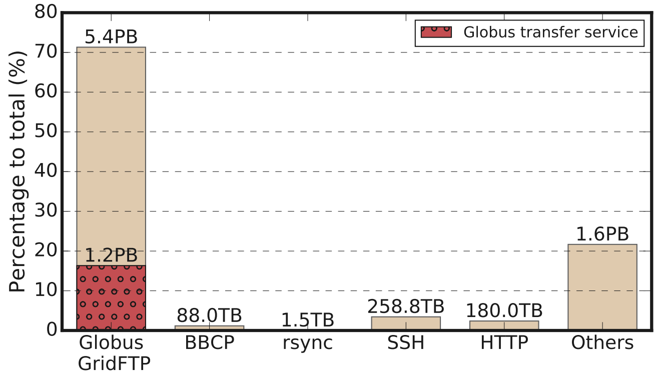

Fig. 2. Data volumes transferred with different tools on *BigSite* during the five-month period 2017/08/01–12/31. Here the GridFTP means the Globus implementation [2].

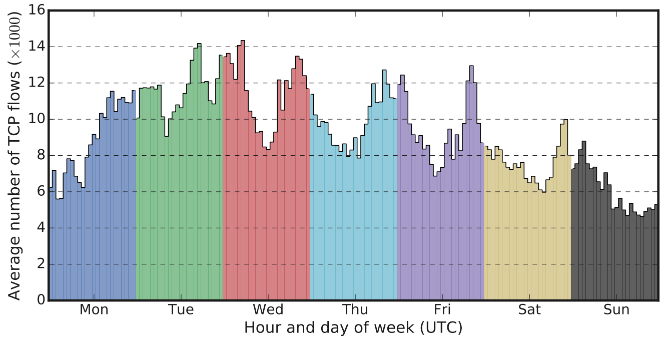

Fig. 3. Average number of TCP flows, to/from all DTNs, per hour and day of the week in 2017. X axis is UTC time.

A Globus transfer that moves N files with concurrency C and parallelism P will involve N GridFTP log entries (one for each file) and $C \times P$ TSTAT log entries (one for each TCP flow). First, we describe how Globus distributes the N files (i.e., the N GridFTP transfers) among the C GridFTP processes. Then we look at how each GridFTP process distributes the chunks of a file into P TCP streams.

A. Load imbalance among concurrent GridFTP server processes of a single Globus transfer request

Even though pipelining reduces latency between file transfers in a single GridFTP server process, it can create load imbalance among the concurrent GridFTP server processes used to serve a single Globus transfer request. The degree of imbalance depends on the pipelining depth. For example, Figure 9 shows how files were mapped to servers in the case of a transfer that involving identically sized 28 files with total volume 97 GB when using a concurrency of 4. This transfer used a pipeline depth of 10 (calculated by the Globus transfer service's autotuning heuristics), which means that Globus may send up to 10 transfer commands (for 10 files) to each GridFTP process before and file transfer completes. Since there are only 28 files in total, Globus distributed seven files each to the four GridFTP processes at the beginning. Despite the fact that each server process was given the same amount of data to transfer, some processes were fast and some were slow (due to external factors). The result is a significant tail effect as the complete transfer waits for server 4 to complete its allocated files.

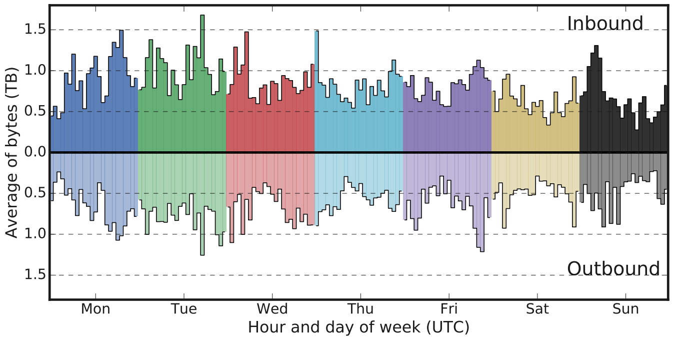

Fig. 4. Average number of bytes moved, to/from all DTNs, per hour of day of week in 2017. X axis is UTC time.

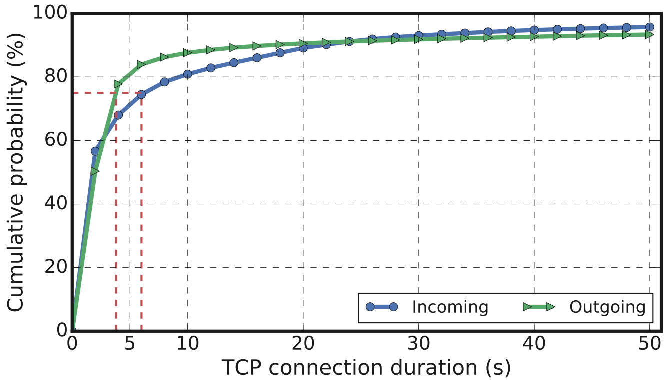

Fig. 5. Cumulative distributions of TCP flow duration, with 75th percentiles indicated by dashed red lines.

Consider C GridFTP processes (possibly assigned to different DTNs) that belong to a Globus transfer T. Let G_{st}^{i} and G_{et}^{i} represent respectively the start and end timestamp of GridFTP process i $(i \in [C] = \{1, 2, 3, \cdots, C\})$. We define the absolute imbalance time of a Globus transfer T as:

$$T_{imb}^{A} = \underset{i \in [C]}{max} \left(G_{et}^{i} \right) - \underset{i \in [C]}{min} \left(G_{et}^{i} \right), \tag{1}$$

where the max and min represent maximum and minimum over all the C GridFTP processes. Thus, we define the relative imbalance of a Globus transfer, T_{imb}^{R}, as the ratio of absolute imbalance time to Globus transfer duration:

$$T_{imb}^{R} = \frac{T_{imb}^{A}}{\underset{i \in [C]}{max} \left(G_{et}^{i} \right) - \underset{i \in [C]}{min} \left(G_{st}^{i} \right)}. \tag{2}$$

Figure 10 shows the cumulative distribution of both the absolute imbalance time and relative imbalance of Globus transfers. As one can see, 50% of the transfers have an absolute imbalance time $\geq$ 43 seconds. In terms of relative imbalance, 50% of the transfers have a relative imbalance $\geq$ 11%.

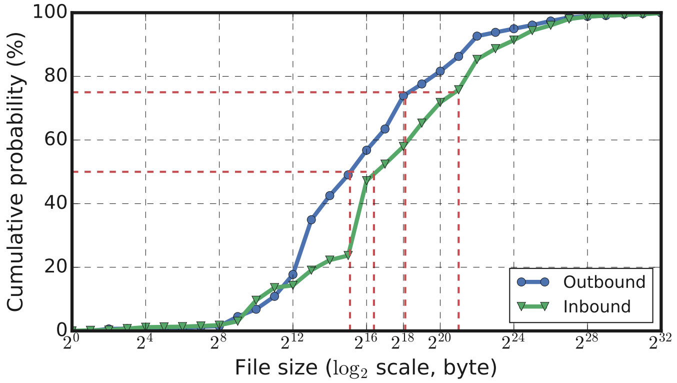

Fig. 6. Cumulative distributions of GridFTP transfer file sizes, with 50th and 75th percentiles highlighted.

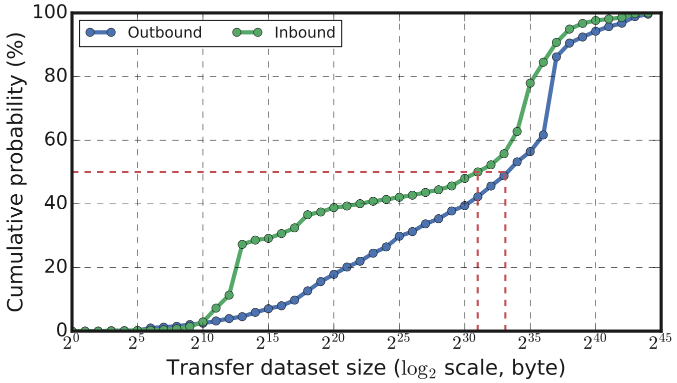

Fig. 7. Cumulative distributions of Globus transfer sizes, with 50th percentiles highlighted.

B. Load imbalance among the parallel TCP streams of a single GridFTP server process

Parallel streams are widely used in wide-area data transfer tools [6, 7] to provide high performance. Consider P TCP streams that belong to a transfer tool (e.g., GridFTP) process G. Let S^i_{st} and S^i_{et} represent respectively the timestamps of the first and last packet (obtained from TSTAT [26] records) of TCP stream i ($i \in [P] = \{1, 2, 3, \cdots, P\}$).

We define the absolute imbalance time of a GridFTP server process, G^A_{imb}, as:

$$G^A_{imb} = \underset{i \in [P]}{max} \left(S^i_{et} \right) - \underset{i \in [P]}{min} \left(S^i_{et} \right) . \tag{3}$$

where the max and min represent maximum and minimum over all the P TCP streams. Thus, the relative imbalance of GridFTP server process, G^R_{imb}, is defined as the ratio of absolute imbalance time to GridFTP transfer duration:

$$G^R_{imb} = \frac{G^A_{imb}}{\underset{i \in [P]}{max} \left(S^i_{et} \right) - \underset{i \in [P]}{min} \left(S^i_{st} \right)} \tag{4}$$

Figure 11 shows the cumulative distribution of both absolute imbalance time and relative imbalance of GridFTP server processes. As one can see, the 70th percentile values for G^A_{imb} and G^R_{imb} are less than 0.3 second and 0.4% respectively,

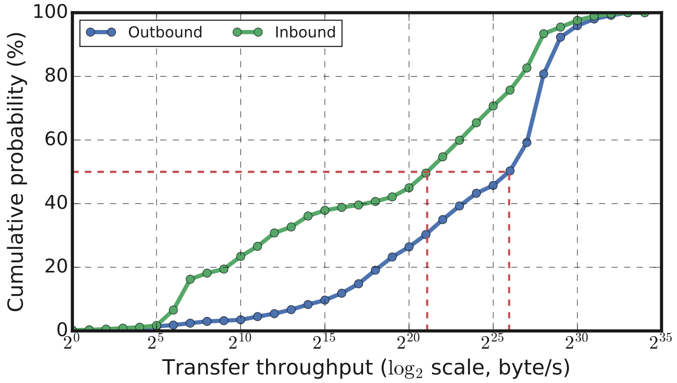

Fig. 8. Cumulative distributions of Globus transfer rate, with 50th percentiles highlighted.

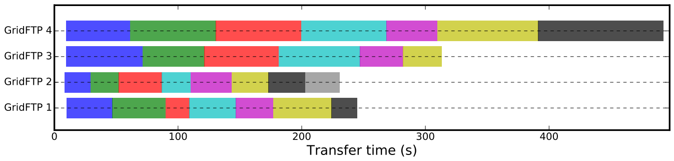

Fig. 9. Imbalanced GridFTP load due to pipelining. Each line represents activity at one of four GridFTP servers, with each rectangle corresponding to a single equi-sized file.

which means that the parallel TCP streams in 70% of the server processes had little imbalance. However, parallel TCP streams of 20% of GridFTP server processes experienced an absolute imbalance time between 1 and 2 seconds.

C. Globus transfer service optimization

In January 2018, Globus deployed an optimization that improves load balancing and reduces long tails, to the transfer service. Specifically, this optimization sorts the files in descending order of their size before assigning them to the concurrent GridFTP server processes. For the last $10 \times C$ (where C is the concurrency or number of GridFTP server processes) files, this optimization forces a pipeline depth of 1.

Figure 12 compares the imbalance (both absolute and relative) in transfers before and after this optimization, using transfers in a two month period before the optimization was put in place and the transfers in a two-month period after the optimization was put in place. We see that both absolute and relative imbalance have decreased. However, about 20% of the Globus transfers still experience an absolute imbalance of more than 20 seconds and an equal percentage of transfers experience a relative imbalance of 25%.

V. Opportunities

Here we discuss opportunities for improvement in wide-area data movement.

A. Load balancing

As discussed in §IV, despite the recent optimization in the Globus transfer service to reduce long tails, room for improvement remains. The lowest granularity of data distributed by the

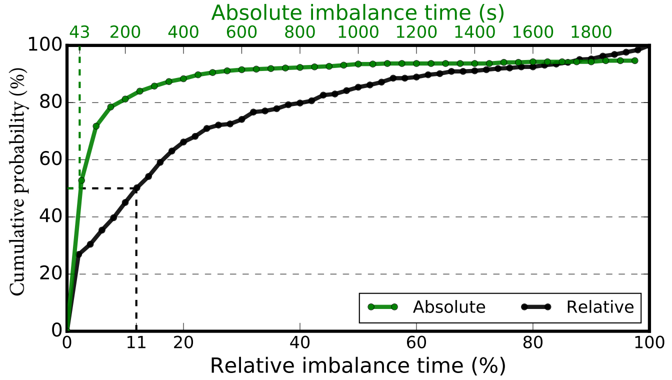

Fig. 10. Cumulative distribution of imbalance (in concurrent GridFTP server processes) of Globus transfers.

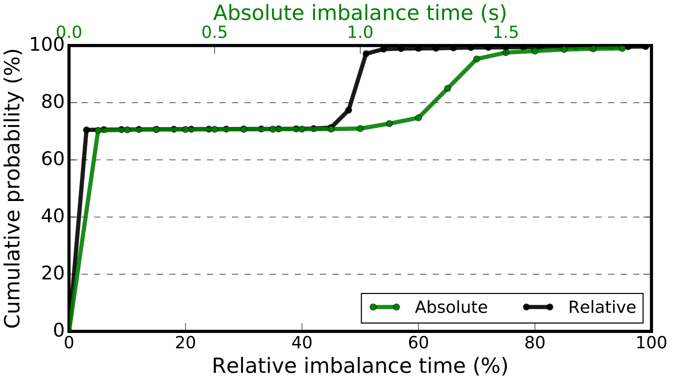

Fig. 11. Cumulative distribution of the imbalance (in parallel TCP streams) of GridFTP server processes.

Globus transfer service to the GridFTP server processes is a file. For datasets with a certain configuration (e.g., $C-1$ large files, where C is concurrency), it is impossible to balance the load among the GridFTP server processes with the file as the lowest granularity of workload. Since GridFTP supports partial file transfers, the Globus transfer service's reducing the lowest workload granularity to a portion of a file will help improve the load balancing for all datasets.

B. Resource allocation

Figure 13 shows the number of bytes moved per day in the year 2017. The peaks are 170 TB and 295 TB for outbound and inbound transfers, respectively. However, the daily averages are only 15.0 TB and 19.6 TB for outbound and inbound, respectively. 75% of days have outbound and inbound volumes less than 18.7 TB and 22.0 TB, respectively.

The number of DTNs operated by *BigSite* for file transfers increased from three to nine in May, 2017. Table II compares the average and maximum hourly and daily data traffic volumes before and after this upgrade. We see that the total quantity of data moved increased substantially after this upgrade, However, there is little change in the ratio between maximum and average. We note that there is large difference between peak and average data movement rates. Traffic at peak load may include both flows that are time-critical and flows that are less time-critical. Utilization of data transfer infrastructure can be increased by adopting measures to spread the load, for example, by treating transfers requiring instant service (on-demand) differently than the transfers that can be delayed by a certain amount (best-effort).

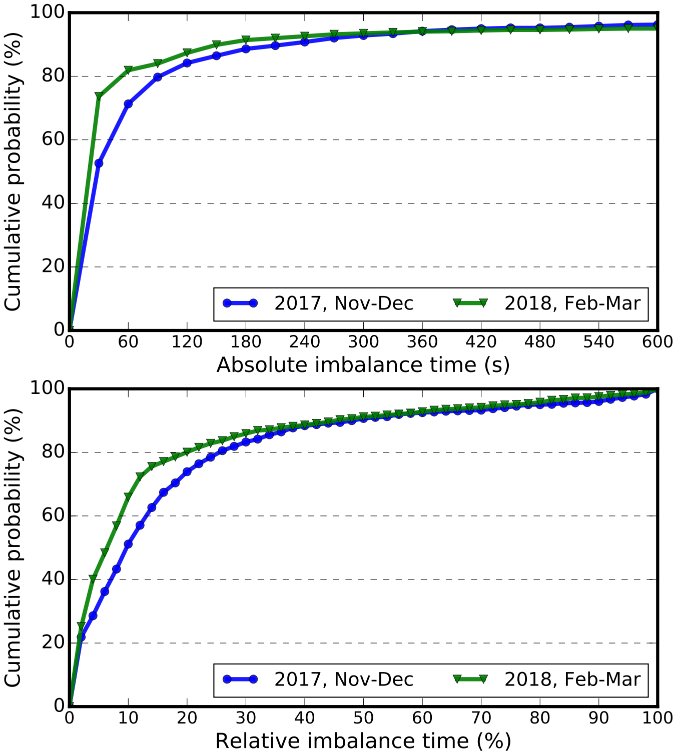

Fig. 12. Cumulative distributions of absolute (above) and relative imbalance (below), before and after the Globus transfer service improvement.

TABLE II

STATISTICS SHOWING MAXIMUM AND AVERAGE OF DAILY DATA VOLUME BEFORE AND AFTER UPGRADE. DATA VOLUME UNIT ARE TB/DAY FOR DAILY BASIS (D) AND TB/HOUR FOR HOURLY BASIS (H). X/M DENOTES MAXIMUM/MEAN. Q75 REPRESENTS 75TH% QUANTILE.

	Outbound				Inbound			
	Q75	Max	Avg	X/M	Q75	Max	Avg	X/M
Before (D)	10.0	56.7	6.8	8.3	11.0	53.7	8.2	6.5
After (D)	23.1	170.4	20.6	8.3	30.6	295.1	27.5	10.7
Before (H)	0.2	12.4	0.3	43.4	0.2	14.6	0.3	42.6
After (H)	0.9	34.3	0.9	39.9	1.1	37.6	1.1	32.8

VI. Related work

Over the past 30 years, numerous efforts have been devoted to modeling and improving wide-area network traffic performance. Some researchers made the first step by providing statistical measurements of wide-area traffic [32–36]. Following these efforts, a wide variety of models were built with the aim of predicting and improving wide-area data transfer. Yildirim et al. [37] built a model to determine the optimal parallelism level at the application layer to optimize end-to-end data transfer performance. Although increasing the number of concurrent streams can improve the transfer speed dramatically

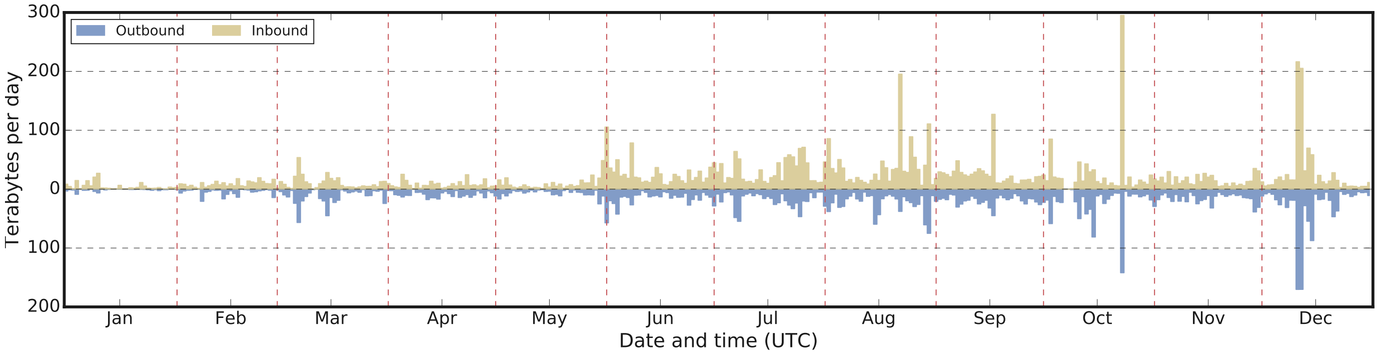

Fig. 13. Terabytes per day in 2017. The daily average of outbound and inbound are 15.0 TB and 19.6 TB respectively.

when the network is uncrowded, too many streams may cause congestion, which can deteriorate data transfer performance extensively. In [38], Yildirim et al. analyzed various factors that affect end-to-end data transfer throughput and developed a network-aware model and algorithm to tune end point resources according to network resources. Kettimuthu et al. [39] used modeling to explore how GridFTP transfer performance is influenced by parallelism and concurrency. The knowledge they gained has been applied to control bandwidth allocation on large-scale data transfers.

Liu et al. [24] analyzed millions of Globus data transfers involving thousands of DTNs and showed that DTN performance has a nonlinear relationship with load. Although their work can explain the performance of wide-area data transfer, the explanation is coarse grained on the subsystem level, and no insights are provided. In another study, Liu et al. [25] conducted a systematic examination of a large set of data transfer logs to characterize transfer characteristics, including the nature of the datasets transferred, achieved throughput, user behavior, and resource usage. Their analysis yielded new insights that can help design better data transfer tools, optimize networking and edge resources used for transfers, and improve the performance and experience for end users. Because of the limited information logged (mostly because of the privacy policy), however, they were not able to find the corresponding GridFTP transfers of a given Globus transfer log. Thus, they were unable to provide insights into the Globus transfer through the underlying GridFTP transfers.

Rao et al. [40] studied the performance of TCP variants and their parameters for high-performance transfers over dedicated connections by collecting systematic measurements using physical and emulated dedicated connections. These experiments revealed important properties such as concave regions and relationships between dynamics and throughput profiles. Their analyses enable the selection of a high-throughput transport method and corresponding parameters for a given connection based on round-trip time.

To quantify the impact of the bulk data transfer across wide-area networks with high performance, Anvari and Lu [41] performed an empirical analysis of how the bulk-data transfer tools perform when competing with a nonsynthetic, application-based workload. Their characterization showed that the network file system performance drops significantly when competing with bulk-data transfers on a shared network.

VII. Conclusion

In this paper, we characterized the network traffic of a computer facility's DTNs at multiple levels, from user transfer requests down to TCP flows. Combining the logs from different layers allowed us to identify load imbalances and opportunities for improvement in wide area data movement. We believe that this facility case study provides valuable insights into the design, operation, and management of data transfer nodes and data transfer tools. We hope that it will encourage other computing facilities to undertake similar efforts. We believe that combining logs of multiple subsystem logs (e.g., wide area network logs that shows the external load, storage monitoring data that represents overall load of the storage system) will enable better understanding of data transfer in shared environments such as the one we considered here, and we plan to undertake such a study. These insights are useful not only for optimizing existing systems and tools but also for planning system upgrades and future investments.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.

References

[1] Z. Liu, R. Kettimuthu *et al.*, "A mathematical programming- and simulation-based framework to evaluate cyberinfrastructure design choices," in *IEEE 13th International Conference on e-Science*, Oct 2017, pp. 148–157. [Online]. Available: http://doi.org/10.1109/eScience.2017.27

[2] W. Allcock, J. Bresnahan *et al.*, "The Globus striped GridFTP framework and server," in *ACM/IEEE Conference on Supercomputing*, ser. SC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 54–. [Online]. Available: https://doi.org/10.1109/SC.2005.72

[3] B. W. Settlemyer, J. D. Dobson *et al.*, "A technique for moving large data sets over high-performance long distance networks," in *27th Symp. on Mass Storage Systems and Technologies*, May 2011, pp. 1–6.

[4] "BaBar Copy," http://slac.stanford.edu/~abh/bbcp/.

[5] "Fast Data Transfer - FDT," http://monalisa.cern.ch/FDT/.

[6] J. Crowcroft and P. Oechslin, "Differentiated end-to-end internet services using a weighted proportional fair sharing TCP," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 3, pp. 53–69, Jul. 1998. [Online]. Available: http://doi.acm.org/10.1145/293927.293930

[7] T. J. Hacker, B. D. Athey *et al.*, "The end-to-end performance effects of parallel TCP sockets on a lossy wide-area network," in *16th Intl Parallel and Distributed Processing Symp.*, 2002, p. 314. [Online]. Available: http://dl.acm.org/citation.cfm?id=645610.661894

[8] J. Bresnahan, M. Link *et al.*, "GridFTP pipelining," in *TeraGrid'2007*.

[9] E. Dart, L. Rotman *et al.*, "The Science DMZ: A network design pattern for data-intensive science," *Scientific Programming*, vol. 22, no. 2, pp. 173–185, 2014.

[10] "Data Transfer Nodes," http://fasterdata.es.net/science-dmz/DTN/.

[11] B. Allen, J. Bresnahan *et al.*, "Software as a service for data scientists," *Communications of the ACM*, vol. 55, no. 2, pp. 81–88, 2012.

[12] Y. Gu and R. L. Grossman, "UDT: UDP-based data transfer for high-speed wide area networks," *Comput. Netw.*, vol. 51, no. 7, pp. 1777–1799, May 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2006.11.009

[13] L. Ramakrishnan, C. Guok *et al.*, "On-demand overlay networks for large scientific data transfers," in *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 359–367. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2010.82

[14] Y. Ren, T. Li *et al.*, "Protocols for wide-area data-intensive applications: Design and performance issues," in *International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC'12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012, pp. 34:1–34:11. [Online]. Available: http://dl.acm.org/citation.cfm?id=2388996.2389043

[15] S. Thulasidasan, W. chun Feng *et al.*, "Optimizing GridFTP through dynamic right-sizing," in *12th IEEE International Symposium on High Performance Distributed Computing*, June 2003, pp. 14–23.

[16] E. Yildirim and T. Kosar, "End-to-end data-flow parallelism for throughput optimization in high-speed networks," *Journal of Grid Computing*, vol. 10, no. 3, pp. 395–418, Sep 2012. [Online]. Available: https://doi.org/10.1007/s10723-012-9220-9

[17] E.-S. Jung, R. Kettimuthu *et al.*, "Toward optimizing disk-to-disk transfer on 100G networks," in *7th IEEE Intl Conf. on Advanced Networks and Telecommunications Systems*, 2013.

[18] E. Kissel, M. Swany *et al.*, "Efficient wide area data transfer protocols for 100 Gbps networks and beyond," in *3rd Intl Workshop on Network-Aware Data Management*. ACM, 2013, p. 3.

[19] I. Foster, M. Fidler *et al.*, "End-to-end quality of service for high-end applications," *Computer Communications*, vol. 27, no. 14, pp. 1375–1388, 2004.

[20] R. Kettimuthu, G. Vardoyan *et al.*, "An elegant sufficiency: Load-aware differentiated scheduling of data transfers," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '15. New York, NY, USA: ACM, 2015, pp. 46:1–46:12. [Online]. Available: http://doi.acm.org/10.1145/2807591.2807660

[21] R. Kettimuthu, G. Agrawal *et al.*, "Differentiated scheduling of response-critical and best-effort wide-area data transfers," in *IEEE International Parallel and Distributed Processing Symposium*, May 2016, pp. 1113–1122.

[22] Z. Liu, R. Kettimuthu *et al.*, "Towards a smart data transfer node," in *4th International Workshop on Innovating the Network for Data Intensive Science*, November 2017, p. 10.

[23] R. Kettimuthu, Z. Liu *et al.*, "Transferring a petabyte in a day," in *4th International Workshop on Innovating the Network for Data Intensive Science*, November 2017, p. 10.

[24] Z. Liu, P. Balaprakash *et al.*, "Explaining wide area data transfer performance," in *26th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '17. New York, NY, USA: ACM, 2017, pp. 167–178. [Online]. Available: http://doi.acm.org/10.1145/3078597.3078605

[25] Z. Liu, R. Kettimuthu *et al.*, "Cross-geography scientific data transferring trends and behavior," in *Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '18. New York, NY, USA: ACM, 2018, pp. 267–278. [Online]. Available: http://doi.acm.org/10.1145/3208040.3208053

[26] Tstat, *TCP STatistic and Analysis Tool*, http://tstat.tlc.polito.it.

[27] MaxMind, *IP Geolocation and Online Fraud Prevention*, 2018 (accessed January 3, 2018), https://www.maxmind.com.

[28] Linux, *rsync: a fast, versatile, remote (and local) file-copying tool*, https://linux.die.net/man/1/rsync.

[29] ——, *scp: secure copy (remote file copy program)*, https://linux.die.net/man/1/scp.

[30] P. Fuhrmann and V. Gülzow, "dCache, storage system for the future," in *European Conference on Parallel Processing*. Springer, 2006, pp. 1106–1113.

[31] K. Chard, S. Tuecke *et al.*, "Globus: Recent enhancements and future plans," in *XSEDE'16*, 2016.

[32] R. Caceres, "Measurements of wide area internet traffic," University of California at Berkeley, Tech. Rep., 1989.

[33] R. Cáceres, P. B. Danzig *et al.*, "Characteristics of wide-area TCP/IP conversations," in *ACM SIGCOMM Computer Communication Review*, vol. 21, no. 4. ACM, 1991, pp. 101–112.

[34] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," in *1st ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 69–73.

[35] J. Chen, "Enterprise networks: modern techniques for analysis, measurement and performance improvement," Ph.D. dissertation, Télécom ParisTech, 2012.

[36] J. Chen, W. Zhang *et al.*, "Traffic profiling for modern enterprise networks: A case study," in *IEEE 20th International Workshop on Local & Metropolitan Area Networks*, 2014, pp. 1–6.

[37] E. Yildirim, D. Yin *et al.*, "Prediction of optimal parallelism level in wide area data transfers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2033–2045, 2011.

[38] E. Yildirim and T. Kosar, "Network-aware end-to-end data throughput optimization," in *1st International Workshop on Network-aware Data Management*. ACM, 2011, pp. 21–30.

[39] R. Kettimuthu, G. Vardoyan *et al.*, "Modeling and optimizing large-scale wide-area data transfers," in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 2014, pp. 196–205.

[40] N. S. Rao, Q. Liu *et al.*, "TCP throughput profiles using measurements over dedicated connections," in *26th International Symposium on High-Performance Parallel and Distributed Computing*. ACM, 2017, pp. 193–204.

[41] H. Anvari and P. Lu, "The impact of large-data transfers in shared wide-area networks: An empirical study," *Procedia Computer Science*, vol. 108, pp. 1702 – 1711, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917308049