Cross-Geography Scientific Data Transferring Trends and Behavior

Zhengchun Liu
Argonne National Laboratory
Lemont, IL, USA
zhengchun.liu@anl.gov

Rajkumar Kettimuthu
Argonne National Laboratory
Lemont, IL, USA
kettimut@anl.gov

Ian Foster
Argonne National Lab and University of Chicago
Lemont, IL, USA
foster@anl.gov

Nageswara S.V. Rao
Oak Ridge National Laboratory
Oak Ridge, TN, USA
raons@ornl.gov

ABSTRACT

Wide area data transfers play an important role in many science applications but rely on expensive infrastructure that often delivers disappointing performance in practice. In response, we present a systematic examination of a large set of data transfer log data to characterize transfer characteristics, including the nature of the datasets transferred, achieved throughput, user behavior, and resource usage. This analysis yields 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. Our analysis shows that (i) most of the datasets as well as individual files transferred are very small; (ii) data corruption is not negligible for large data transfers; and (iii) the data transfer nodes utilization is low. Insights gained from our analysis suggest directions for further analysis.

CCS CONCEPTS

• **Networks → Network performance analysis**; **Network performance modeling**;

KEYWORDS

GridFTP; Wide area network; File transfer; Usage management

ACM Reference Format:
Zhengchun Liu, Rajkumar Kettimuthu, Ian Foster, and Nageswara S.V. Rao. 2018. Cross-Geography Scientific Data Transferring Trends and Behavior. In *HPDC '18: International Symposium on High-Performance Parallel and Distributed Computing, June 11–15, 2018, Tempe, AZ, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3208040.3208053

1 INTRODUCTION

Many science workflows are distributed in nature and rely on networks to move data to geographically distributed resources for analysis, sharing, and storing [16]. Since 1990, ESnet traffic has grown by a factor of 10 every four years, approximately double the rate of growth in commercial Internet traffic [9]. This growth in demand has motivated considerable investments in wide area science networks and in network and data transfer infrastructure at university campuses and research institutions [23].

Researchers have studied packet-level network traces to get insights into wide area traffic patterns [29, 33] and have used Netflow [35] and SNMP [8] data to analyze the impact of bulk data flows on delay-sensitive flows [15] and to forecast network traffic [12]. File transfer application logs such as GridFTP logs have been used to study the gap between peak and average utilization of network resources [27] and to model transfer throughput [17]. Here, we use file transfer application logs to characterize wide area science data transfers over a four-year period. The resulting insights can help

(1) resource providers optimize the resources used for data transfer;
(2) researchers and tool developers build new (or optimizing the existing) data transfer protocols and tools;
(3) end users organize their datasets to maximize performance; and
(4) funding agencies plan investments.

We analyze approximately 40 billion GridFTP command logs totaling 3.3 exabytes and 4.8 million transfers logs collected by the Globus transfer service from 2014/01/01 to 2018/01/01. The results provide a number of insights in terms of utilization of the data transfer nodes, data corruption in wide area transfers, repeat transfers, file types transferred, transfer performance, and user behavior.

The rest of this paper is organized as follows. First, we introduce (§2) the Globus transfer service and the GridFTP protocol used in the transfers that we consider here, the logs that we study, and the methodology that we use to analyze those logs. Next, we analyze those logs from the perspectives of dataset characteristics and trends (§3), transfer performance and reliability (§4), user behavior (§5), and utilization and sharing behaviors of dedicated data transfer nodes (DTNs) (§6). In §7 we review related work, and in §8 we summarize our conclusions and briefly discuss future work.

2 WIDE AREA DATA TRANSFER

End-to-end wide area file transfers are carried out by tools such as GridFTP, standard FTP, rsync, SCP, Globus transfer [34], BBCP [4], FDT [10], XDD [30], Aspera[14], and others. GridFTP is used by

HPDC '18, June 11–15, 2018, Tempe, AZ, USA

ACM ISBN 978-1-4503-5785-2/18/06…$15.00
https://doi.org/10.1145/3208040.3208053

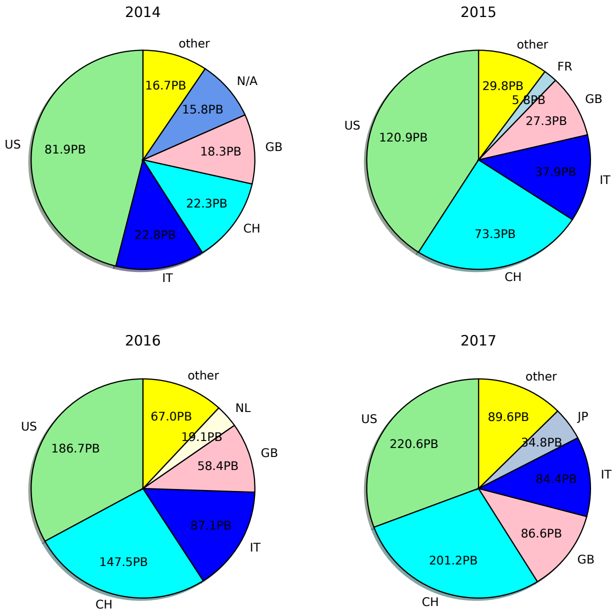


Figure 1: Total bytes transferred by country.

Table 1: Fields found in (anonymized) GridFTP logs. Some fields are omitted because their values are either identical or empty for all logs.

Key	Value	Description
num_streams	4	Parallel TCP streams
appname	globusonline-fxp	Application name
hostname	grid-cr2.desy.de	Server hostname
start_time	1452637794.757349	POSIX time with μsec
ftp_return_code	226	RFC959 completion code
ip_address	212.189.205.173	Host IP address
num_bytes	976526	Bytes transferred (file size)
end_time	1452637794.877834	POSIX time with μsec
trans_type	STOR	FTP command (RFC959)
buffer_size	174758	TCP buffer size

many science domains in many countries (see Figure 1) and has usage logs available [28]. Moreover, our analysis (details provided in the subsection below) of the GridFTP usage logs in conjunction with the ESnet network traffic data showed that GridFTP traffic forms a major portion of the ESnet traffic and can thus serve as a representative set for all wide area science data transfers.

2.1 GridFTP

GridFTP, an extension of the standard FTP protocol for high performance, better security, and reliability, is one of the most widely used protocols for science data transfers. GridFTP was standardized through the Open Grid Forum, and multiple implementations of that standard exist. The Globus [2] and dCache [11] implementation are the most popular.

The Globus implementation of GridFTP reports limited usage information to a usage analytic server by sending a UDP packet for each successful transfer (the usage reporting can be disabled by the user) [28]. Table 1 shows an example record of one transfer (IP address and hostname are anonymized).

By comparing the statistics from these usage reports with the ESnet SNMP statistics [8], we determined that the GridFTP traffic accounts for about 65% of incoming traffic to and about 42% of outgoing traffic from DOE national laboratories in 2017. We note that the total incoming traffic of all DOE laboratories is 114.17 PB, and the total outgoing traffic is 234.20 PB. Since the vast majority of traffic to and from the two Large Hadron Collider (LHC) Tier-1 sites in the United States—namely, Brookhaven National Laboratory and Fermi National Accelerator Laboratory [7]—is dCache GridFTP [11] traffic and since we do not have logs from the dCache GridFTP servers, we exclude these two laboratories from our consideration. For the rest of the laboratories, the incoming and outgoing traffic totals are 51.49 PB and 87.72 PB, respectively. And the incoming(*STOR*) and outgoing(*RETR*) traffic totals of Globus GridFTP servers are 33.36 PB and 36.62 PB, respectively. Thus, GridFTP accounts for 64.79% of incoming traffic and 41.75% of outgoing traffic, and GridFTP transfer characteristics should provide a reasonable approximation of the overall science data transfer characteristics. Also, we do not have access to the logs for other data transfer tools that are not based on Globus GridFTP, such as BBCP [4], FDT [10], XDD [30], Aspera [14], dCache [11], and SCP based on SSH protocol.

We note that the total traffic we obtained from the SNMP logs were collected at the router interface and therefore includes all traffic to and from the laboratories including the protocol headers. In contrast, GridFTP bytes were computed at the application level and thus exclude the protocol headers. The IP addresses in GridFTP logs were mapped to the national labs by using the information from whois [13]. Arguably, since the laboratories rotate IP addresses used for their resources from a pool of IP addresses they own, we may have missed transfers from some of GridFTP servers while computing the total bytes transferred by the GridFTP servers at the laboratories. Therefore, the percentage of GridFTP traffic we compute here is the base line; the actual percentage may be higher.

2.2 GridFTP clients

Since the GridFTP protocol is standardized, many different implementations of GridFTP clients (more than the number of server implementations) exist. Table 2 lists the statistics (from the server logs) of transfers by the top five heavily used clients and the total transfers. *libglobus_ftp_client* indicates that the client application was built using this library, but the application does not set the application name field while interacting with the server.

From Table 2 we can see that *fts_url_copy* [5] (the service responsible for globally distributing the majority of the LHC data across the Worldwide LHC Computing Grid infrastructure) accounts for almost half of the total bytes, while *globusonline-fxp* (the Globus transfer service) manages about 60% of all files. In other words, *fts_url_copy* is used primarily for transferring LHC data [5], while Globus transfer service users are more diverse.

Table 2: Petabytes and millions of files transferred via GridFTP using different tools over the past four years.

Year	fts_url_copy		libglobus_ftp_client		globusonline-fxp		globus-url-copy		gfal2-util		Total	
	PBytes	MFiles	PBytes	MFiles	PBytes	MFiles	PBytes	MFiles	PBytes	MFiles	PBytes	MFiles
2014	N/A	N/A	111.23	746.59	39.81	1646.10	13.13	816.67	N/A	N/A	176.24	3431.78
2015	48.09	77.29	103.21	841.96	52.89	2424.58	19.27	947.78	0.93	6.70	267.33	4435.13
2016	244.46	295.67	105.75	998.96	88.56	3600.78	14.76	850.76	10.03	74.05	466.91	5922.83
2017	342.12	550.57	40.11	885.65	113.45	3901.27	16.89	898.14	45.93	234.65	585.01	6671.79
Total	634.67	923.53	360.3	3,473.16	294.71	11,572.73	64.05	3,513.35	56.89	315.4	1,495.49	20,461.53

2.3 Limitations in GridFTP Usage Logs

Because of privacy considerations [28], the GridFTP toolkit reports the IP address only of the machine that runs it; in other words, logs for the *STOR* command do not have the IP address of the source endpoint. Similarly there is no IP address of the destination endpoint for *RETR* logs. The total number of endpoints (unique IP address) in the past four years is 63,166. There are 20.5 billion *STOR* logs totaling 1.5 exabytes received and 19.4 billion *RETR* logs totaling 1.8 exabyte transferred. We note that since GridFTP uses unreliable UDP to collect usage and since users can disable the collection, the *STOR* logs and *RETR* logs are different. Considering the large number of logs even in a short time—on average there are more than 25,000 *STOR* and *RETR* logs per minute in 2017—accurately matching a *STOR* log with a *RETR* log is almost impossible. On the other hand, Globus transfer (being a hosted service) logs have this information and many other details about the transfers. Arguably, these logs still have some limitations; for example, they do not have the size of the individual files in a transfer. Nevertheless, these logs are much more comprehensive than the GridFTP logs.

2.4 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 [3]. A transfer request specifies, among other things, a source and destination; the file(s) and/or directory(s) to be transferred; and (optionally) whether to perform integrity checking (enabled by default) and/or to encrypt the data (disabled by default). It provides automatic fault recovery and automatic tuning of optimization parameters to achieve high performance. Globus can transfer data with either the GridFTP or HTTP protocols; we focus here on GridFTP transfers, since HTTP support has been added only recently.

The Globus transfer service distinguishes between the two types of GridFTP server installations: Globus Connect Personal (GCP), a lightweight single-user GridFTP server designed to be deployed on personal computers, and Globus Connect Server (GCS), a multiuser GridFTP server designed to be deployed on high-performance storage systems that may be accessed by many users concurrently.

Globus transfer logs recorded 4,813,091 transfers from 2014/01/01 to 2018/01/01, totaling 13.1 billion files and 305.8 PB. These transfers involved 41,900 unique endpoints and 71,800 unique source-to-destination pairs (edges), and 26,100 users. We used the MaxMind IP geolocation service [25] to obtain approximate endpoint locations. Figure 2 shows the number in each city worldwide. Table 3 shows the total bytes and files transferred per year, both within a single country (nationally) and between countries (internationally). Logs include the unique name of the source and destination endpoints, transfer start and end date and time, the user who submitted the transfer, total bytes, number of files and number of directories, and number of faults and file integrity failures. The logs also have tunable parameters. Therefore, the Globus logs are a good supplement to GridFTP logs in order to characterize wide area data transfer.

Table 3: Data transferred by Globus: petabytes and millions of files.

	National		International		Total	
Year	PBytes	MFiles	PBytes	MFiles	PBytes	MFiles
2014	41.44	1,865	0.78	26.9	42.32	1,892
2015	53.45	2,763	2.55	94.3	56.39	2,873
2016	90.10	3,929	2.84	110.8	93.60	14,042
2017	109.16	4,162	3.23	94.3	113.50	4,264

2.5 Analysis Framework

Four years of raw GridFTP logs were stored in about 100,000 compressed files in json format, for a total of 1.2 TB. We parsed and saved these logs in MongoDB for our analysis. The raw Globus transfer service logs were saved in millions of tiny files in json format. Since Globus logs is much smaller than GridFTP logs, we parsed these tiny json files and saved them as one file by using the Python pickle module (it implements binary protocols for serializing and deserializing a Python object structure). In our analysis, we used the Python pandas library [26] to load the Globus transfer logs. We performed all raw data analysis on a Cray Urika-GX platform (a high-performance big data analytics platform optimized for multiple workflows), with the Apache Spark [37] cluster-computing framework. Anonymized sample data files are available at https://github.com/ramsesproject/wan-dts-log. The GridFTP logs soon will be publicly available for researchers for further analysis via the data-sharing service of Globus.

3 DATASET CHARACTERISTICS

Users' transfers consist of one or more files. GridFTP clients use one or more control channel sessions to the GridFTP server(s) (for third-party server-to-server transfers, clients establish control channel sessions with both the source and destination servers). The GridFTP server handles each control channel session independently and thus does not what files belong to the same transfer. GridFTP logs have statistics for each individual file, which could be a separate transfer in itself or part of a bigger multi-file or directory transfer. On the

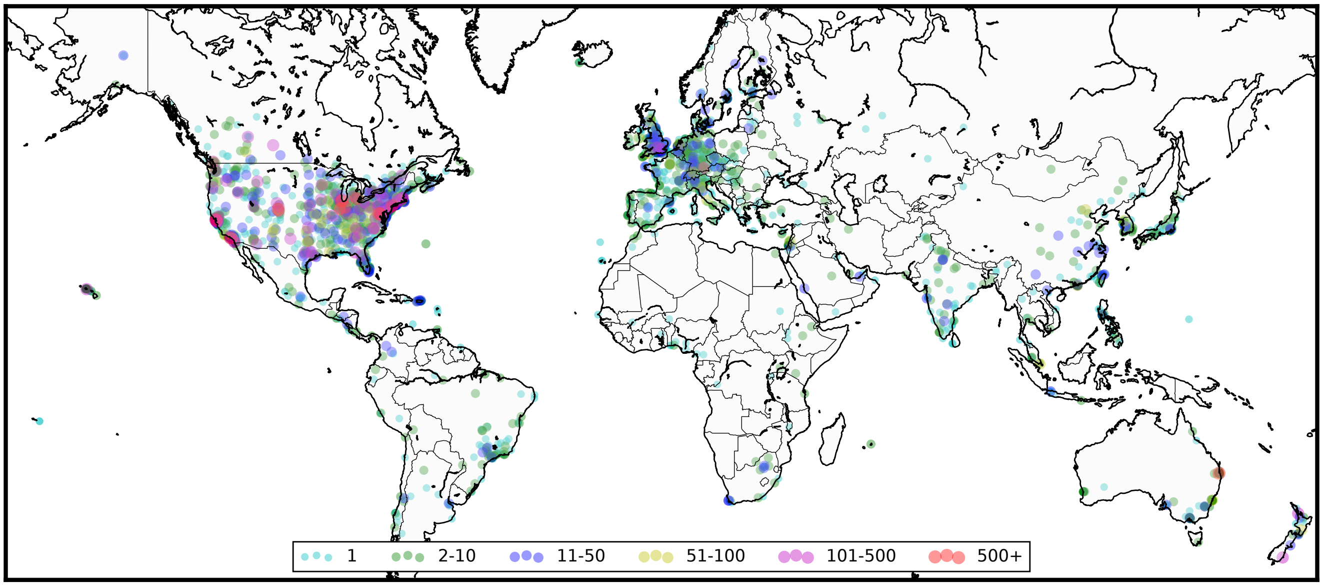


Figure 2: Geographical distribution of Globus endpoints, with color coding used to show number per city.

other hand, Globus transfer logs have information at the transfer (single-file, multi-file, or directory) level including the number of files and total bytes, but they do not have the size of each file. Therefore, for multi-file transfers, we know only the average file size. We note that one cannot correlate the Globus transfer logs and GridFTP logs in order to determine the size of individual files in a multi-file transfer because the GridFTP logs do not have the filename and path information. Instead, we use GridFTP logs to study the trends at the file level and Globus transfer logs to study the trends at the transfer level.

3.1 Dataset size

Figure 3 shows the cumulative distribution of dataset size (note that a dataset consists of one or more files and zero or more directories). We see in Figure 3 that most transfers are only a few megabytes in size. The average transfer size is 63.5 GB, but the median is only 6.4 MB. This is not to say that there are no large transfers: 17.6% are >1 GB in size and furthermore account for 99.9% of all data transferred; 0.8% are >1 TB in size and account for 80.6% of all data transferred; and 97.4% of the bytes were transferred by the top 5% of transfers. Surprisingly, the average transfer size is becoming smaller, especially the smaller transfers (e.g., transfer size smaller than 1MB). For example, the 20th percentile in 2017 is only about 1% of the 2014 value; the 80th percentile decreased from about 2^{32} bytes in 2014 to about 2^{26} bytes in 2017.

Observation 1. *Most of the datasets moved over the wide area are small. Specifically, the 50th, 75th, and 95th quartiles of dataset size are 6.3 MB, 221.5 MB, and 55.8 GB, respectively. Counterintuitively, the dataset size has decreased year by year from 2014 to 2017.*

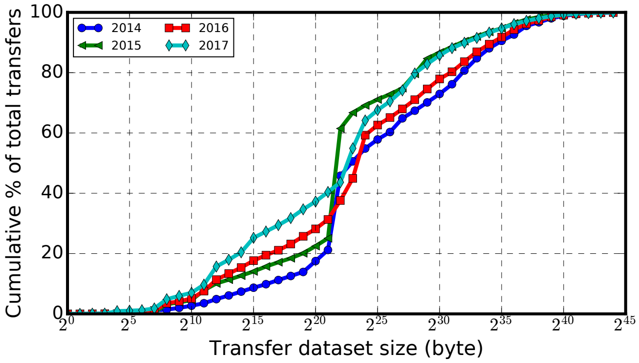


Figure 3: Cumulative distribution of transfer dataset size.

3.2 Number of files

Figure 4 presents the cumulative distribution of the number of files per dataset in each year.

We see in Figure 4 that many transfers—specifically, 2,515,278 (63% of the total)—are involved a single file. However, these transfers account for a relatively small amount of data: only 10.96% of the total bytes .

Observation 2. *Most of the datasets transferred by the Globus transfer service have only one file. And 17.6% of those datasets (or 11% of the total) have a file size of ≥ 100 MB, motivating the need for striping the single-file transfer over multiple servers.*

3.3 File size

We know that file size has a considerable influence on transfer performance [21]. Globus transfer service logs provide the total number of files and total bytes for each transfer (dataset), allowing

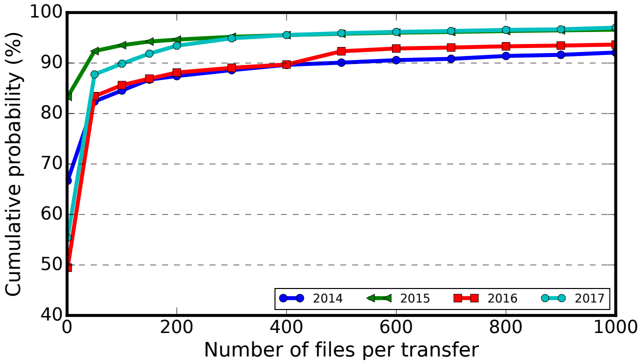


Figure 4: Distribution of number of files per dataset.

us to compute the average file size per transfer, as shown in Figure 5. We see that for most datasets, the average file size is just a few megabytes, with the median average file size being only 3.44 MB. However, variance is high, with a standard deviation of 1.6 TB. We also see that average dataset file size has decreased year by year. For example, the 20th percentile of average file size in 2017 is only about 10% of the 2014 value.

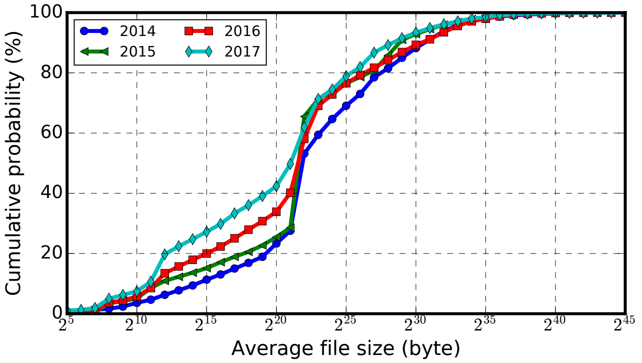


Figure 5: Cumulative distribution of the dataset average file size.

Figure 6 shows the distribution of size of individual files users have transferred, extracted from GridFTP logs. Clearly, most of the files are small. The 50th and 75th percentiles are 2^{16} and 2^{20} bytes, respectively. Not much difference is seen in terms of small size files year by year. However, the difference in the big files (greater than 1 MB) becomes smaller year by year. The 80th percentile in 2017 is about a quarter of that in 2014.

We note some surprising findings. For example, in 2017, users transferred 1.3 million one-byte files, and around 1 billion files were less than 1 KB in size. Large transfers also occurred. For example, in 2017, 3,536 transfers were greater than 1 TB; the largest was 454 TB. However, only four file transfers used the striping [2] feature (i.e., used a cluster of nodes at the source and destination to transfer a large file).

Table 4 lists the average file size by application. The table clearly shows that (fts_url_copy) users tend to transfer big files and that Globus transfer service users are more likely to transfer small files.

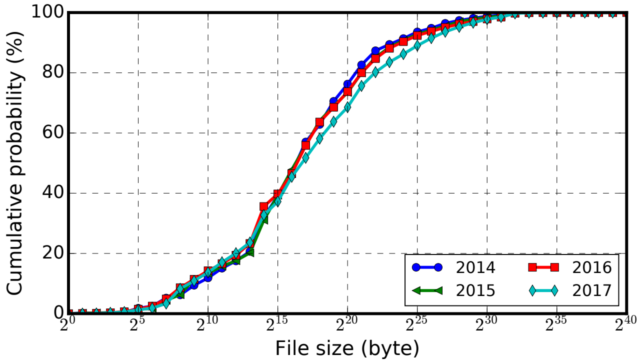


Figure 6: Cumulative distribution of individual file size.

The overall average file shows an increasing trend over the years. However, the average file size for the individual client applications does not show such a trend.

Observation 3. *The average file size of most datasets transferred is small (on the order of few megabytes). Majority of individual file size is less than 1 MB. These results motivate the need for performance optimizations aimed at small file transfers.*

3.4 Directory depth

Figure 7 shows the cumulative distribution function of the directory depth. Most users organize files using a reasonable subdirectory hierarchy (80% of the datasets have a depth less than 9). The number of directories in a dataset also influences the transfer performance [21] because there is a cost to create folders. This analysis is beneficial for transfer tool designers and performance optimization.

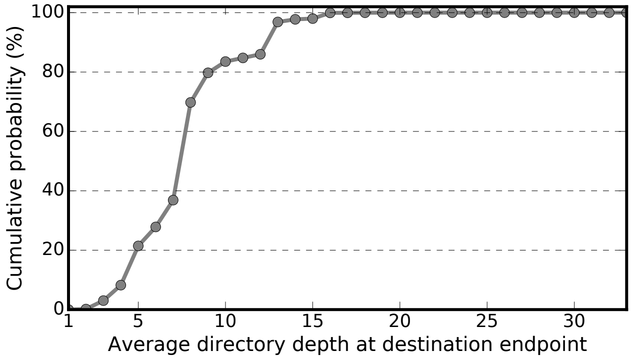


Figure 7: Cumulative distribution of average directory depth

3.5 File type

Researchers have long adopted or designed specific data formats that best represent datasets for different domains. We investigated the popularity of file format by looking at the file extension. Figure 8 shows the distribution of file extension in which 6.8% of files have no extension (marked as *no-ext*). Surprisingly, the three most commonly transferred extensions are images. However, many scientific

Table 4: Average file size (in MB) by application and year.

Year	fts_url_copy [5]	libglobus_ftp_client	globusonline-fxp [34]	globus-url-copy [2]	gfal2-util [6]	Overall
2014	–	142.96	27.31	8.86	–	53.89
2015	652.44	133.78	23.89	18.41	32.72	69.18
2016	856.98	193.83	26.28	45.20	252.22	105.28
2017	719.65	153.42	30.78	29.18	182.29	111.81

applications and researchers use a domain-specific data format that may be suppressed by common file types.

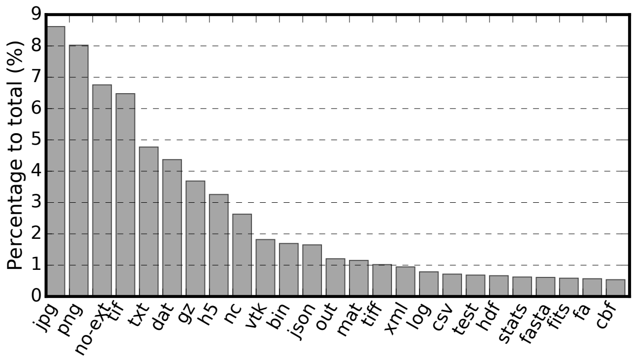


Figure 8: The 25 most-transferred file types: 61.8% of all files.

Observation 4. *Image files are the most common file type transferred, followed by raw text files. Scientific formats such as* .h5 *(hierarchical data format) and* .nc *(NetCDF) are in the top 10.*

3.6 Repeated transfers

We are interested in whether the same datasets are transferred repeatedly, either from a single source or from different sources, since this information can indicate whether multicast and/or caching schemes have value. Lacking checksum data for all files, we approximate this *sharing* phenomenon by computing a fingerprint for each dataset in the Globus logs by combining file names (exclude path, sort, concatenate as one string) and total dataset size (individual file size is not available in Globus logs). This fingerprint is approximate in two respects: first, it does not capture equivalence if files are renamed but contents are not changed; and second, two datasets with the same file names and size may have different content. We ignored single-file datasets because they are likely to have the second mismatching. Nevertheless, we believe that the analysis reveals useful information.

Having computed fingerprints, we can then count the number of times that each fingerprint is transferred via Globus. Table 5 lists the 15 datasets that were transferred most often.

Observation 5. *Repeated transfers are not common, less than 7.7% of the datasets are transferred more than once. When they do occur, the datasets in question are distributed mostly from one (or a few) endpoints to multiple destinations (i.e., $N_{usr} < N_{dst}$). We also observe multiple users transferring the same data to the same destination.*

Table 5: Dataset sharing behavior for the 15 most-transferred datasets. N_{src} and N_{dst} represent the number of unique source and destination endpoints, respectively; N_{usr} and N_{trs} denote the number of users and times transferred, respectively.

N_{src}	N_{dst}	N_{usr}	N_{trs}	Size
1	120	111	131	10.2GB
3	26	24	73	5.0MB
7	8	3	72	14.7GB
1	58	57	64	9.1GB
9	7	6	53	170.4MB
3	12	33	52	3.1GB
1	4	30	51	3.1GB
1	44	43	51	9.3GB
1	47	47	49	8.3GB
1	4	32	42	365.0MB
2	39	39	40	7.4GB
1	5	4	33	3.7GB
2	6	6	31	17.7GB
1	17	17	25	13.3MB
1	4	17	25	0.3MB

4 TRANSFER CHARACTERISTICS

Here we present our analysis of transfer performance, duration, and failures and the usage of tuning parameters.

4.1 Checksum, encryption, and reliability

Wide area data transfers involve more than just data movement: both integrity checking (via a checksum) and encryption can be applied to the data that is transferred.

Because of well-known limitations of the 16-bit TCP checksum [32], transfer tools (including GridFTP) support verifying the integrity of data transferred by using a 32-bit checksum. For example, to verify the integrity of the data transferred, the Globus transfer service rereads the file(s) at the source and at the destination, computes a checksum at each location, and compares the two resulting checksums. The importance of these checksums is revealed by the fact that 27,251 of the 3,312,102 Globus transfers with integrity checking enabled had at least one checksum error (i.e., one in 121 transfers had at least one checksum error).

Checksums are applied by default but can be disabled by the user via a transfer flag. In our dataset, 83.2% of transfers had integrity checking enabled. Transfer tools also support encrypted data transfer, but this feature is not turned by default in most tools because of performance overhead. Of the transfers performed by the Globus transfer service, 2% had encryption enabled.

Figure 9 presents the average number of integrity checking failure per terabyte transferred by month. We can see that no clear burst failures occur in one month besides September 2014. We note that if a user changes a file during a transfer, this action can be reported as an integrity failure. We cannot distinguish this from an actual failure. Data corruption and faults decrease year by year (Figure 10a,10b).

Figure 10b shows the average number of faults per terabyte transferred. Faults include network faults, data transfer node failures, and file integrity check failures. Overall, the service is becoming increasingly reliable.

Observation 6. *At least one checksum failure occurs per 1.26 TB. Although integrity checking adds extra load to storage and CPU on the source and destination endpoints, it is worthwhile. The failures are decreasing year by year. Only 1.9% of transfers used encryption.*

4.2 Transfer direction

As mentioned in subsection 2.4, Globus Connect Personal (GCP) is a lightweight single-user GridFTP server designed to be deployed on personal computers, and Globus Connect Server (GCS), is a multiuser GridFTP server designed to be deployed on high-performance storage systems that may be accessed by many users concurrently. Figure 11 shows the trend in terms of number of transfers and bytes transferred for server-to-server (GCS→GCS) transfers, downloads from servers to personal machines (GCS→GCP), uploads from personal machines to servers (GCP→GCS), and personal machines to personal machines (GCP→GCP). One can see that server-to-server transfers are dominant in terms of bytes transferred and that downloads are equivalent to server-to-server transfers in terms of the number of transfers.

Table 6: Transfer characteristics of different source and destination types: number of transfers, median performance, median size, and average file size (in MB) per transfer (F$_{avg}$).

			Median		
Year	Type	1000s	Mbps	MB	F$_{avg}$
2014	GCS→GCS	168.77	46.62	104.86	20.97
	GCS→GCP	199.44	5.58	3.43	3.31
	GCP→GCS	61.05	2.93	6.95	1.12
	GCP→GCP	1.54	10.68	2,061.18	4.78
2015	GCS→GCS	513.05	30.16	15.40	7.93
	GCS→GCP	678.56	4.92	3.44	3.39
	GCP→GCS	109.33	3.87	5.06	1.05
	GCP→GCP	2.88	186.52	38,091.87	8.17
2016	GCS→GCS	488.95	27.24	35.51	2.58
	GCS→GCP	494.89	13.55	13.00	3.72
	GCP→GCS	156.70	4.95	9.54	1.37
	GCP→GCP	6.15	26.92	530.29	9.59
2017	GCS→GCS	1,019.14	14.50	7.68	1.64
	GCS→GCP	691.56	7.95	8.15	3.55
	GCP→GCS	189.48	0.48	0.45	0.07
	GCP→GCP	5.24	4.11	24.95	0.94

Observation 7. *Transfers involve many more downloads (GCS to GCP) than uploads (GCP to GCS).*

4.3 Performance

Boxplots in Figure 12 show the trend of per dataset transfer performance by the type of source and destination endpoints. No consistent trend across different years is observed for several reasons.

- As shown in Table 6 and Figure 11, the transfer size and average file size change inconsistently, and these two characteristics have a big influence on transfer performance [21].
- The number of active users increases year by year but with much variance.
- The number of GCPs increases year by year. The performance capability and network environment of these PC-based endpoints are not stable and vary a lot from one to another.
- The number of active GCS endpoints are 3,095, 2,166, 1,773, and 1,883, respectively, for the years 2014 to 2017. The number of transfers increases consistently, meaning that the load of GCS changes year by year inconsistently.

Figure 13 shows the distribution of per file transfer performance. The majority of the files achieve about 64 Mbps throughput, and the overall transfer performance has not changed much over time.

Observation 8. *Although some server-to-server transfers achieve high performance (dozens of Gbps), most transfer throughput is low. For example, the median throughput is only tens of Mbps. There is no clear increasing trend in terms of transfer performance over time.*

4.4 Duration

The transfer time distribution and trend are shown in Figure 14. More than half of all the transfers finished in less than 10 seconds. The longest-running transfer to date ran for six months; this was a large transfer from one tape archive to another. Of all the transfers, 0.004% ran for more than a month, 0.09% for more than a week, 1.2% for more than a day, and 8% for more than an hour.

4.5 Transfer parameters

Regular FTP sends a file over a single TCP stream; with **Parallelism**, a file's data blocks are distributed over a specified number (P) of TCP streams. All TCP streams have the same source and destination GridFTP server process. Large files over high-latency links can benefit from higher parallelism, since the multiple streams devoted to a single file can in effect increase the TCP window size and in addition can provide increased resilience to packet losses. Beside P, the Globus transfer service has two other application-level tuning parameters: **Concurrency** C and **Pipelining** D.

Concurrency involves starting C independent GridFTP processes at the source and destination file systems. Each of the C resulting process pairs can then work on the transfer of a separate file, which provides for concurrency at the file system I/O, CPU core, data transfer nodes (each transfer can involve multiple servers), and network levels. In general, concurrency is good for multi-file transfers because it can drive more filesystem processes, CPU cores, and even machine nodes, in addition to opening more network data streams. Since striping feature is not enabled in Globus, single file transfers cannot have $C > 1$.

Pipelining, D, speeds transfers involving many small files by dispatching up to D FTP commands over the same control channel,

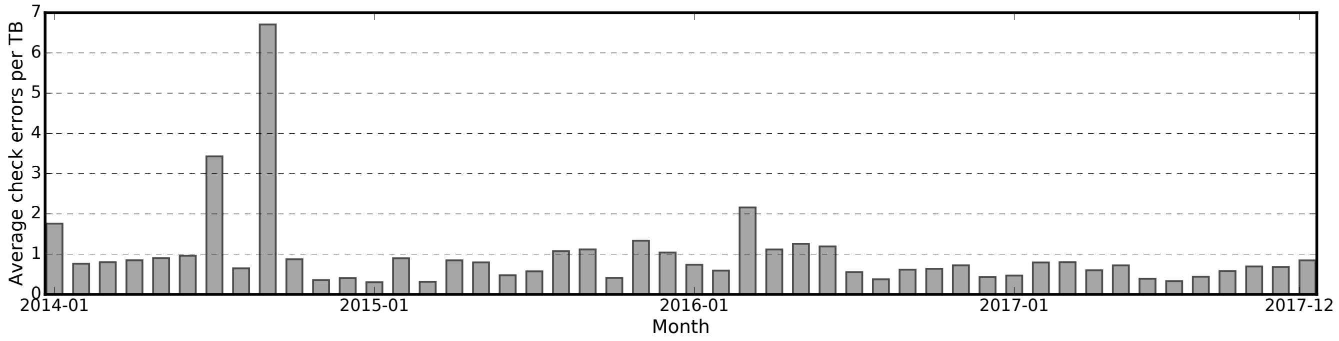
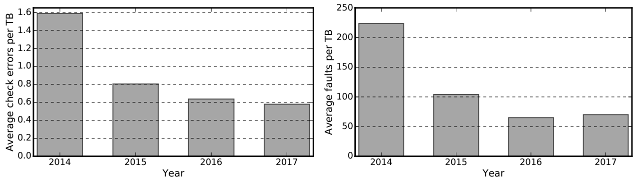


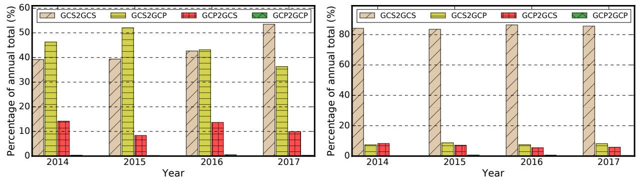
Figure 9: Monthly average number of integrity check failures per TB transferred.



(a) Integrity check failures.

(b) Number of faults.

Figure 10: Annual average number of integrity check failures and faults per TB transferred.



(a) By number of transfers.

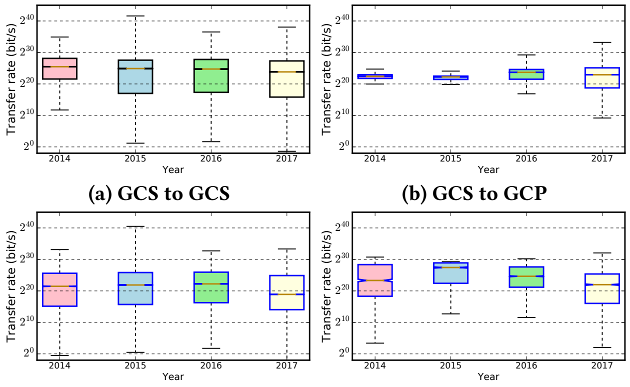
(b) By total bytes.

Figure 11: Transfer numbers and volume vs. endpoint type.

back to back, without waiting for the first command's response. This method reduces latency and keeps the GridFTP server constantly busy; it is never idle waiting for the next command.

In general, transfer performance is improved by increased P when sending a large file over high-latency (and lossy) links and by increased C when transferring many files. When sending many large files, increased P and C can both be beneficial [2, 24, 36]. The Globus transfer service thus sets C and P parameters according to simple heuristics based on the number and sizes of files in a request, subject to site-specific limits and policies specified by endpoint administrators.

Figure 15 shows the distribution of parameters values used for transfers from the Globus transfer service. More than 60% of transfers used $C = 1$ because more than half of the Globus transfers are single file transfers (Figure 4). Most users let Globus choose the proper parameters. But the best choice is not necessarily the one that maximizes the performance of a single transfer; other considerations can also come into play, such as the need to moderate bandwidth usage by individual flows for purposes of fairness and/or flow prioritization, a desire to manage performance-energy tradeoffs [1], or the desire to orchestrate transfers from/to the same data transfer nodes(DTN) to reduce resource contention cost [22].



(a) GCS to GCS

(b) GCS to GCP

(c) GCP to GCS

(d) GCP to GCP

Figure 12: Distribution of per dataset transfer performance trend extracted from Globus logs.

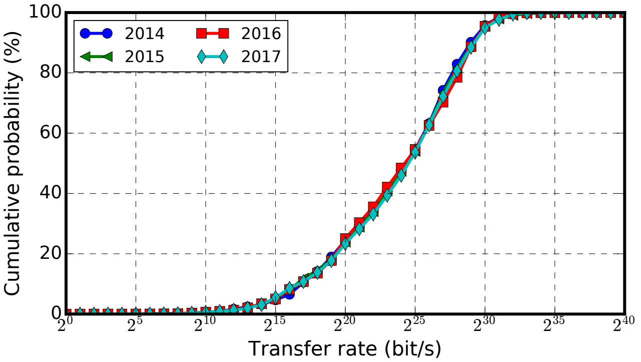


Figure 13: Cumulative distribution of per file transfer performance extracted from GridFTP logs.

In GridFTP logs, 94.6% of the transfers by *globus-url-copy* use the default 1 TCP stream (i.e., $P = 1$). Similarly, 93.4% of the *fts_url_copy* transfers also use 1 TCP stream. *gfal2-util* almost never uses more than 1 TCP stream.

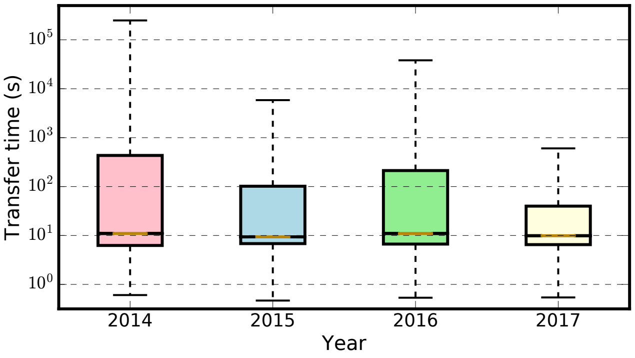


Figure 14: Transfer duration.

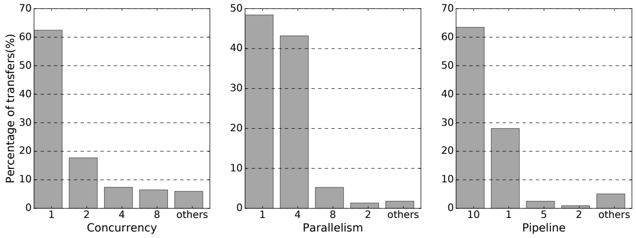


Figure 15: Parameter values used in Globus transfers.

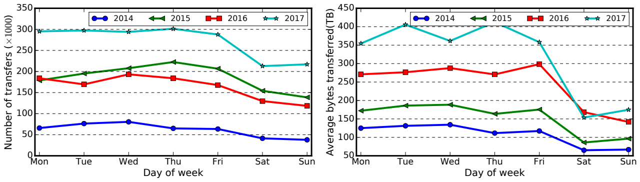
Observation 9. *Most users do not manually tune the transfer parameters (e.g., 94.6% of the transfers use $P = 1$). Transfer tools should be smart enough to choose the best parameters for each transfer in order to achieve maximum performance.*

5 USER BEHAVIORS

Users who perform at least one transfer during a given year are considered *active*. The number of active users from 2014 to 2017 was 4,602, 6,985, 10,234, and 13,321, respectively.

5.1 Transfer frequency

User behavior is hard to predict, but the statistics can help users better plan their own transfer. The statistics about user behavior can also help resource providers schedule maintenance and plan resource allocation. Figure 16 shows user transfer behavior by day of week.



(a) Average number of transfers. **(b) Average bytes transferred.**

Figure 16: Average number of transfers and bytes transferred, by day of week.

The figure shows a clear drop in usage on weekends in terms of both total bytes and number of transfers.

5.2 Transfer volume

Figure 17 shows the distribution of bytes transferred by percentage of users.

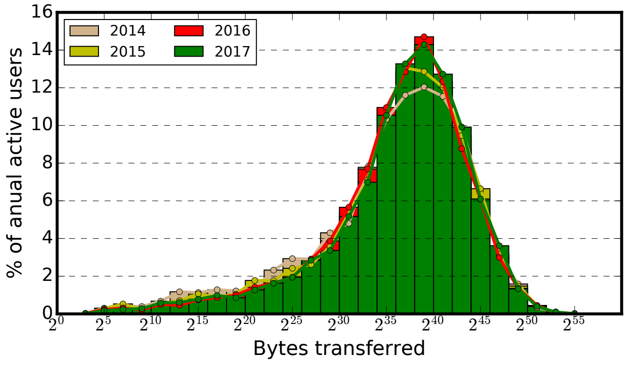


Figure 17: Distribution of bytes transferred per user.

The figure shows most users transferred dozens of gigabytes. The few users who transferred hundreds of terabytes accounted for the majority of total bytes moved. Figure 18 shows the cumulative distribution of bytes moved by percentage of active users in each year.

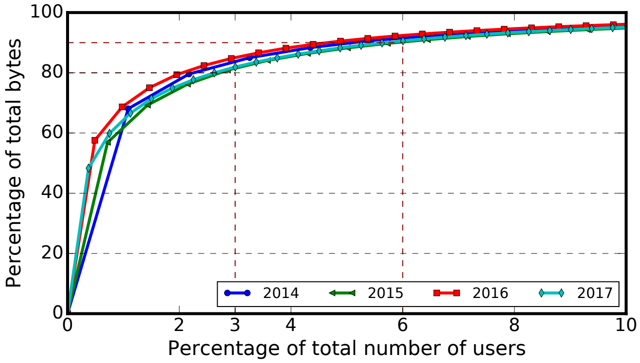


Figure 18: Cumulative distribution of the percent of annual active users versus the total bytes transferred.

Observation 10. *Of all the bytes transferred, 80% are by just 3% of all users; 10% of the users transferred 95% of the data.*

5.3 Degree of connection

Similarly to a person in the social network, we define an endpoint's degree of connection as the number of unique endpoints with which it has engaged in one or more transfers. The degree is a measure of the endpoint's popularity.

We compute the degree of each endpoint annually. In 2017, 81% of the endpoints connected to only one other endpoint, 11% to two other endpoints, and only 8% to three or more. This is not to say that there is no widely connected endpoints. For example, the Blue Waters DTN at the National Center for Supercomputing

Applications had a degree of connection of 855, 706, and 2,092 for 2016, 2017, and 2014–2017, respectively.

Figure 19 shows the degree of connection for the 100 most-connected endpoints for each of the four years. Clearly, some endpoints are highly connected, and the degree of connection is increasing over time.

5.4 User access to endpoints

The number of active users in Globus transfer service has increased steadily year by year, with 4,652, 7,025, 10,313, and 13,433 active users annually from 2014 to 2017. Here we analyze the number of endpoints accessed by individual users, in order to understand the trend of data sharing and collaboration.

As shown in Figure 20, slightly more than half of the users accessed two or fewer endpoints. Specifically, only 41.76% of the users accessed three or more endpoints; and 1.5% of the users accessed only one endpoint, which means that they used the Globus transfer service to copy files locally in a fire-and-forget manner. Specifically, we found 71,000 transfers for which the source and destination were the same. These transfers, totaling 17 PB, were done by 2,868 users over 2,090 unique endpoints; 0.34% (90 users) users accessed more than 20 endpoints.

6 ENDPOINT CHARACTERISTICS

We call an endpoint active in a given year if there is at least one transfer to/from this given endpoint. The number of active endpoints in 2014 to 2017 was 8,620, 10,478, 13,482, and 16,826, respectively. Among them, 5,820, 8,592, 12,008, and 15,251 were GCP, respectively; and 2,800, 1,887, 1,474, and 1,575 were GCS, respectively.

6.1 Degree of sharing

Here we study the number of users who have access to an endpoint. This analysis describes how the endpoints are shared. We focus on GCS endpoints because a GCP endpoints can be accessed only by the user who set it up. For a given endpoint, the number of users accessed represent the degree of sharing. Figure 21 presents the number of users per endpoint for the top 1000 most-shared GCS endpoints (the 100th endpoint has 4 users).

We observe a descending linear slope in the log-log plot in Figure 21, suggesting that the edge (user to endpoint) degree distribution of vertices (endpoint) follows a power law, which is common in many real-world networks [19]. Lim et al. [20] observed a similar distribution for the number of files generated by a user in a different project on a petascale file system.

Observation 11. *The degree distribution of the number of users per endpoint follows a power-law distribution, similar to other real-world social network graphs.*

6.2 Utilization

DTNs are compute systems dedicated for wide area data transfers in distributed science environments. DTNs typically have GCS deployed on them. In this section, we study the utilization of those DTNs. For each minute in 2017, we mark a given DTN as active if there is at least one transfer over the DTN; otherwise we marked it as idle. We found that, on average, DTNs are completely idle (i.e., there is no transfers) for 94.3% of the time. Figure 22 shows the cumulative distribution of the time that DTNs are active. The percentage of active time clearly is low. For example, 80% of the endpoints are active less than 6% of the time.

However, some endpoints are heavily used. For the top 100 most heavily used endpoints, Figure 23a shows the percentge of time that at least one transfer was happening over the endpoints. To investigate how busy the endpoint is when there is at least one transfer, we assume that the endpoint resource utilization is 100% when it gets the maximum aggregated throughput (incoming and outgoing), and we compute the utilization at a given instant as the ratio of the aggregate throughput at the instant to the maximum aggregate throughput observed at the endpoint in the entire year. Figure 23b shows the different percentile values of the utilization of the top 100 most heavily used endpoints. Clearly, their utilization is very low.

Users may use other data transfer tools, such as BBCP [4], FDT [10], XDD [30], or Aspera [14], which may add more utilization. We therefore used port scanning to determine the installation of other data transfer tools and found that less than 1% of the endpoints had other tools installed. This percentage implies that the utilization reported here is accurate for 99% of the endpoints.

Observation 12. *DTN utilization is surprisingly low. Since the DTN requirement is high for high-throughput DTNs, some good topics for research would be the use of these computing resource (1) for other purposes; (2) for complex encoding to deal with data corruption and; (3) to compress data to reduce the network bandwidth consumption.*

6.3 Edge

Figure 24 shows the number of transfers per edge (between source and destination, unidirectional). Most edges have few transfers: indeed, a quarter of all edges are involved in just one transfer. This sparse communication makes performance analysis for such transfers hard.

7 RELATED WORK

We previously used Globus logs to explain performance of wide area data transfers [21]. That work focused on explaining the performance of individual transfers. Here, because we analyze the whole logs in aggregate, our analysis provides deeper insights into the temporal evolution of scientific datasets transferred over wide area networks.

As we have seen in this analysis, sometimes truth hidden in the data is counterintuitive. Rishi et al. [31] studied packet size distributions in Internet traffic and observed that the trimodal packet sizes are around 40, 576, and 1500 Bytes—a change from common wisdom. Lan et al. [18] looked at the Internet traffic data recorded from two different operational networks and found that a small percentage of flows consume most network bandwidth. These observations are important for traffic monitoring and modeling purposes.

Lim et al. [20] analyzed 500 days of metadata snapshots of the Spider parallel file system (PFS) at the Oak Ridge Leadership Computing Facility to characterize user behavior and data-sharing trends on the petascale file system. Their analysis provided deep insights into the temporal evolution of a heavily used petascale PFS of a leading supercomputing center. Our work provides a somewhat similar analysis for wide area data transfers.

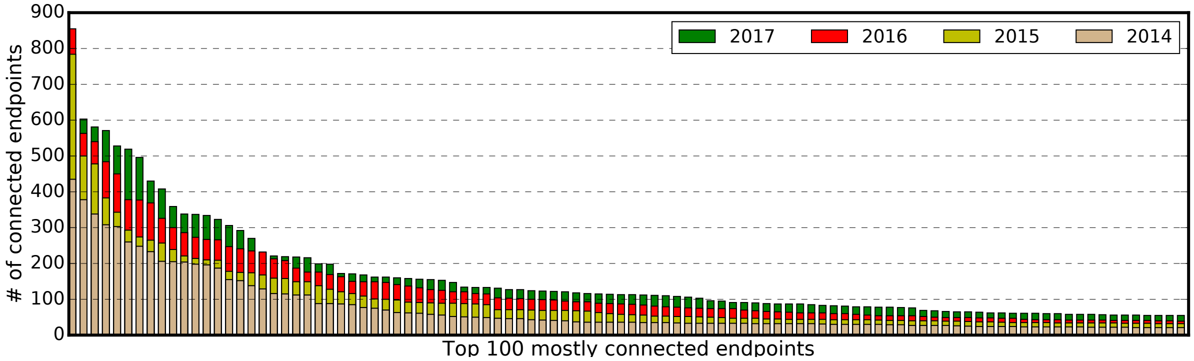


Figure 19: Degree of connection for the 100 most-connected endpoints.

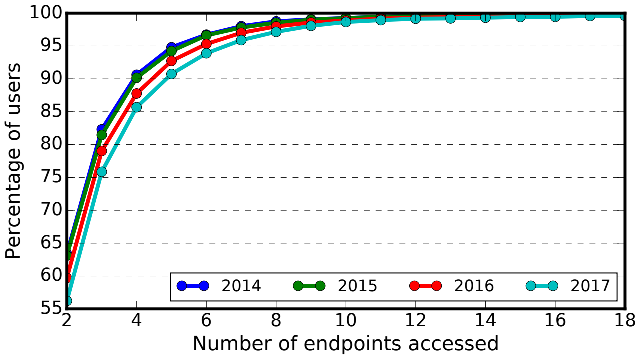


Figure 20: Cumulative distribution of the number of endpoints users have accessed.

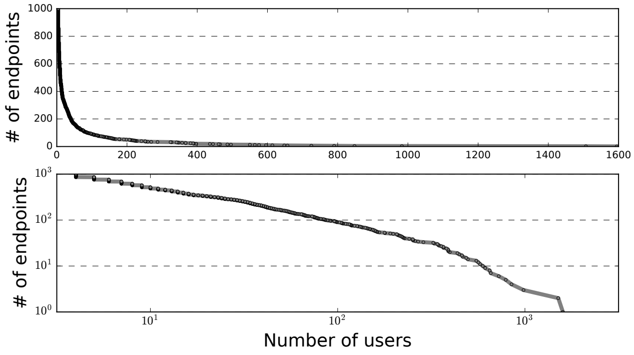


Figure 21: Number of users per endpoint. The log-scaled plot (bottom) shows that the distribution follows a power law.

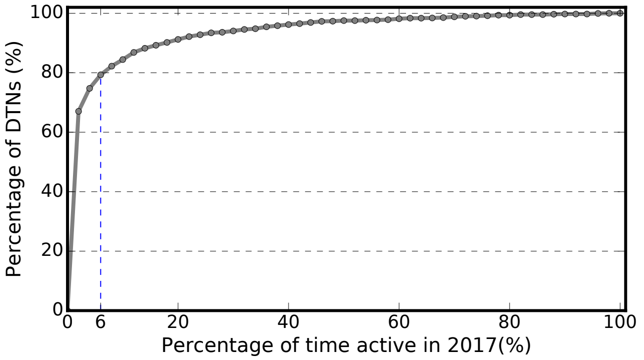
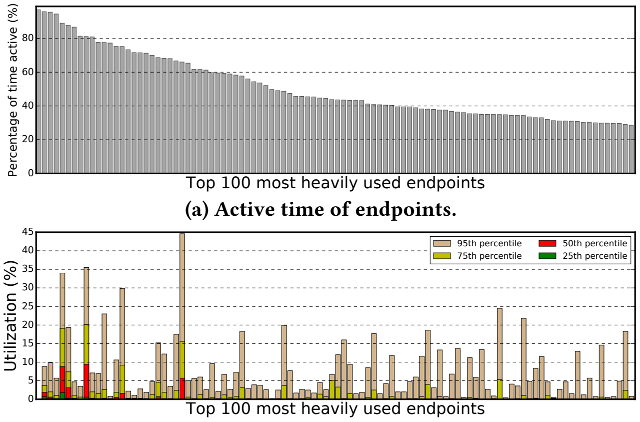


Figure 22: Cumulative distribution of idle time percentage; 80% of endpoints were active less than 6% of the time.



(a) Active time of endpoints.

(b) Endpoint utilization.

Figure 23: Utilization of 100 most heavily used endpoints.

8 CONCLUSIONS

To systematically characterize the wide area transfers for a general understanding, we analyzed 20.5 billion GridFTP *STOR* command logs totaling 1.5 exabytes received and 19.4 billion GridFTP *RETR* command logs totaling 1.8 exabytes transmitted, by a total of 63,166 GridFTP servers distributed all over the world in the past four years. To address the limitations in GridFTP logs, we supplemented our analysis with 4.8 million transfers logs collected by the Globus transfer service from 2014/01/01 to 2018/01/01. These transfers, totaling 13.1 billion files and 305.8 PB, involved 41,900 unique endpoints, 71,800 unique source-to-destination pairs, and 26,100 users. To the best of our knowledge, this is the first study of its kind to systematically characterize the wide area transfers from real logs. Our analysis revealed a number of insights in terms of the utilization

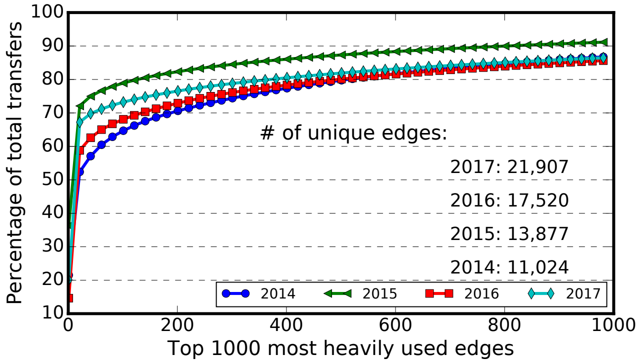


Figure 24: Cumulative distribution of number of transfers over the top most heavily used edges.

of the data transfer nodes, data corruption in wide area transfers, repeat transfers, file types transferred, transfer performance, and user behavior. We believe our analysis can help researchers, tool developers, resource providers, end users, and funding agencies from different perspectives.

ACKNOWLEDGMENT

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357. We also would like to thank the five anonymous reviewers for their helpful comments.

REFERENCES

[1] Ismail Alan, Engin Arslan, and Tevfik Kosar. 2015. Energy-aware data transfer algorithms. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 44.

[2] William Allcock, John Bresnahan, Rajkumar Kettimuthu, Michael Link, Catalin Dumitrescu, Ioan Raicu, and Ian Foster. 2005. The Globus Striped GridFTP Framework and Server. In *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing (SC '05)*. IEEE Computer Society, Washington, DC, USA, 54–. https://doi.org/10.1109/SC.2005.72

[3] Bryce Allen, John Bresnahan, Lisa Childers, Ian Foster, Gopi Kandaswamy, Raj Kettimuthu, Jack Kordas, Mike Link, Stuart Martin, Karl Pickett, and Steven Tuecke. 2012. Software as a Service for Data Scientists. *Commun. ACM* 55, 2 (2012), 81–88.

[4] BBCP [n. d.]. BBCP. ([n. d.]). http://www.slac.stanford.edu/~abh/bbcp/.

[5] CERN. 2018 (accessed January 3, 2018). *FTS3: Robust, simplified and high-performance data movement service for WLCG*. http://fts3-service.web.cern.ch.

[6] CERN. 2018 (accessed January 3, 2018). *Grid File Access Library: A C library providing an abstraction layer of the grid storage system complexity*. https://dmc.web.cern.ch/projects/gfal-2.

[7] CERN. 2018 (accessed January 3, 2018). *Worldwide LHC Computing Grid*. https://wlcg-rebus.cern.ch/apps/topology/.

[8] United States ESnet, DoE. 2018 (accessed January 3, 2018). *ESnet SNMP data*. https://graphite.es.net/west/.

[9] ESnet-plan [n. d.]. ESnet Strategic Plan. http://www.es.net/assets/Uploads/ESnet-Strategic-Plan-March-2-2013.pdf. ([n. d.]).

[10] FDT. 2018 (accessed January 3, 2018). *FDT - Fast Data Transfer*. http://monalisa.cern.ch/FDT/.

[11] Patrick Fuhrmann and Volker Gülzow. 2006. dCache, storage system for the future. In *European Conference on Parallel Processing*. Springer, 1106–1113.

[12] Kejia Hu, Alex Sim, Demetris Antoniades, and Constantine Dovrolis. 2013. Estimating and Forecasting Network Traffic Performance Based on Statistical Patterns Observed in SNMP Data. In *Machine Learning and Data Mining in Pattern Recognition*, Petra Perner (Ed.). 601–615.

[13] ICANN. 2018 (accessed January 3, 2018). *ICANN WHOIS Search*. https://whois.icann.org.

[14] Aspera Inc. 2018 (accessed January 3, 2018). *Aspera High-Speed File Transfer Software*. http://asperasoft.com.

[15] Tian Jin, C. Tracy, M. Veeraraghavan, and Z. Yan. 2013. Traffic engineering of high-rate large-sized flows. In *2013 IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*. 128–135. https://doi.org/10.1109/HPSR.2013.6602302

[16] Rajkumar Kettimuthu, Zhengchun Liu, David Wheelerd, Ian Foster, Katrin Heitmann, and Franck Cappello. 2017. Transferring a Petabyte in a Day. In *4th International Workshop on Innovating the Network for Data Intensive Science*. 10.

[17] Rajkumar Kettimuthu, Gayane Vardoyan, Gagan Agrawal, and P. Sadayappan. 2014. Modeling and Optimizing Large-Scale Wide-Area Data Transfers. *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* 0 (2014), 196–205. https://doi.org/10.1109/CCGrid.2014.114

[18] Kun-chan Lan and John Heidemann. 2006. A Measurement Study of Correlation of Internet Flow Characteristics. *Computer Networks* 50, 1 (Jan. 2006), 46–62.

[19] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (June 2014).

[20] Seung-Hwan Lim, Hyogi Sim, Raghul Gunasekaran, and Sudharshan S. Vazhkudai. 2017. Scientific User Behavior and Data-sharing Trends in a Petascale File System. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '17)*. ACM, New York, NY, USA, Article 46, 12 pages. https://doi.org/10.1145/3126908.3126924

[21] Zhengchun Liu, Prasanna Balaprakash, Rajkumar Kettimuthu, and Ian Foster. 2017. Explaining Wide Area Data Transfer Performance. In *26th ACM Symposium on High-Performance Parallel and Distributed Computing*. https://doi.org/10.1145/3078597.3078605

[22] Zhengchun Liu, Rajkumar Kettimuthu, Ian Foster, and Peter H. Beckman. 2017. Towards a Smart Data Transfer Node. In *4th International Workshop on Innovating the Network for Data Intensive Science*. 10.

[23] Zhengchun Liu, Rajkumar Kettimuthu, Sven Leyffer, Prashant Palkar, and Ian Foster. 2017. A Mathematical Programming- and Simulation-Based Framework to Evaluate Cyberinfrastructure Design Choices. In *2017 IEEE 13th International Conference on e-Science*. 148–157. https://doi.org/10.1109/eScience.2017.27

[24] Dong Lu, Yi Qiao, Peter A. Dinda, and Fabian E. Bustamante. 2005. Modeling and Taming Parallel TCP on the Wide Area Network. In *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*. IEEE Computer Society, Washington, DC, USA, 68.2–. https://doi.org/10.1109/IPDPS.2005.291

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

[26] Wes McKinney et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, Vol. 445. SciPy Austin, TX, 51–56.

[27] Sam Nickolay, Eun-Sung Jung, Rajkumar Kettimuthu, and Ian Foster. 2018. Bridging the gap between peak and average loads on science networks. *Future Generation Computer Systems* 79 (2018), 169 – 179. https://doi.org/10.1016/j.future.2017.05.012

[28] Globus org. 2018 (accessed January 3, 2018). *Usage Statistics Collection by the Globus Alliance*. http://toolkit.globus.org/toolkit/docs/4.0/Usage_Stats.html.

[29] V. Paxson and S. Floyd. 1995. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking* 3, 3 (Jun 1995), 226–244. https://doi.org/10.1109/90.392383

[30] B. W. Settlemyer, J. D. Dobson, S. W. Hodson, J. A. Kuehn, S. W. Poole, and T. M. Ruwart. 2011. A technique for moving large data sets over high-performance long distance networks. In *27th Symp. on Mass Storage Systems and Technologies*. 1–6. https://doi.org/10.1109/MSST.2011.5937236

[31] Rishi Sinha, Christos Papadopoulos, and John Heidemann. 2007. *Internet Packet Size Distributions: Some Observations*. Technical Report ISI-TR-2007-643.

[32] Jonathan Stone and Craig Partridge. 2000. When The CRC and TCP Checksum Disagree. *ACM SIGCOMM Computer Communications Review* 30, 4 (2000).

[33] Walter Willinger, Vern Paxson, and Murad S. Taqqu. 1998. A Practical Guide to Heavy Tails. Birkhauser Boston Inc., Cambridge, MA, USA, Chapter Self-similarity and Heavy Tails: Structural Modeling of Network Traffic, 27–53. http://dl.acm.org/citation.cfm?id=292595.292597

[34] www.globus.org. 2018 (accessed January 3, 2018). *globus*. https://www.globus.org.

[35] Zhenzhen Yan, Malathi Veeraraghavan, Chris Tracy, and Chin Guok. 2013. On how to provision Quality of Service (QoS) for large dataset transfers. In *Proceedings of the sixth international conference on communication theory, reliability, and quality of service (CTRQ)*. 21–26.

[36] Esma Yildirim, JangYoung Kim, and Tevfik Kosar. 2012. How GridFTP pipelining, parallelism and concurrency work: A guide for optimizing large dataset transfers. In *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*. IEEE, 506–515.

[37] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* 59, 11 (Oct. 2016), 56–65. https://doi.org/10.1145/2934664