International Journal of HIGH PERFORMANCE COMPUTING APPLICATIONS

Special Issue Paper

The International Journal of High Performance Computing Applications
2026, Vol. 40(3) 421–433

DOI: 10.1177/10943420261441742
journals.sagepub.com/home/hpc
Sage

Automated, reliable, and efficient continental-scale replication of 7.3 petabytes of computational simulation data: A case study

Lukasz Lacinski[1,2], Lee Liming[1,2], Steven Tuscoy[1,2], Cameron Harr[3], Kyle Chard[1,2], Eli Dart[4], Paul J. Durack[3], Sasha Ames[3], Forrest M. Hoffman[5] and Ian T. Foster[1,2]

Abstract

We report on our experiences replicating 7.3 petabytes (PB) of Earth System Grid Federation (ESGF) computational simulation data from Lawrence Livermore National Laboratory (LLNL) in California to Argonne National Laboratory (ANL) in Illinois and Oak Ridge National Laboratory (ORNL) in Tennessee—a task motivated by a need for increased reliability, capacity, and performance. This task presented significant challenges: the need to move 29 million files twice under time pressure from aging storage hardware; a source file system bottleneck limiting throughput to 1.5 GB/s; frequent site maintenance windows; and the need for complete reliability at scale. We addressed these challenges using a simple replication tool that invoked Globus to transfer large bundles of files while tracking progress in a database, dynamically rerouting transfers to work around maintenance periods and file system limitations. Under the covers, Globus organized transfers to make efficient use of the high-speed Energy Sciences network (ESnet) and the data transfer nodes deployed at participating sites, and also addressed security, integrity checking, and recovery from a variety of transient failures. This success demonstrates the considerable benefits that can accrue from the adoption of performant data replication infrastructure. The replication tool is available at https://github.com/esgf2-us/data-replication-tools.

Keywords

data replication, climate data, high-speed network, automation, Globus

1. Introduction

The Earth System Grid Federation (ESGF) is an international collaboration that develops, deploys, and maintains software infrastructure for the management and dissemination of large-scale Earth system and environmental data (Cinquini et al., 2014). Its distributed architecture, which interconnects multiple data centers and storage systems across the globe, allows users to access data from various model intercomparison projects, observational data sets, and reanalysis products.

One important set of datasets made available by ESGF are those associated with the Coupled Model Intercomparison Projects (CMIPs). These datasets have grown steadily in size over the years, from a few terabytes (TB) for CMIP1 in 1995 to a few 100 TB for CMIP5 (Taylor et al. 2012) in 2008, around 10 PB for CMIP6 (Eyring et al., 2016) in 2016, and substantially more for CMIP7, with first Fast Track data expected in late 2025 and individual national contributions projected at multiple petabytes (e.g., 8 PB for Australia alone).

In the US, the Department of Energy (DOE)'s Lawrence Livermore National Laboratory (LLNL) has long operated an ESGF node (and previously, the Earth System Grid (Bernholdt et al., 2005; Williams et al., 2009)). By 2021, that node held a copy of the entire CMIP3 collection, close to 1 PB of CMIP5 and 90% of CMIP6, as well as other related datasets such as Energy Exascale Earth System Model (E3SM) output data, selected input4MIPs forcing datasets (Durack et al., 2017), and several curated obs4MIPs

[1]The University of Chicago, Chicago, IL, USA
[2]Argonne National Laboratory, Lemont, IL, USA
[3]Lawrence Livermore National Laboratory, Livermore, CA, USA
[4]ESnet, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[5]Oak Ridge National Laboratory, Oak Ridge, TN, USA

Corresponding author:
Ian T. Foster, The University of Chicago, 5735 South Ellis Avenue, Chicago, IL 60637, USA; Argonne National Laboratory, Lemont, IL 60439, USA.
Email: foster@anl.gov

(Teixeira et al., 2014) observational sets—for a total of 7.3 PB in 29 million files.

In 2022, the DOE Office of Biological and Environmental Research (BER) program that funds the primary US ESGF operations requested the creation of copies of LLNL data at the Argonne Leadership Computing Facility (ALCF) at Argonne National Laboratory (ANL) and the Oak Ridge Leadership Computing Facility (OLCF) at Oak Ridge National Laboratory (ORNL). The purpose of this replication was both to increase ESGF reliability and to take advantage of large-capacity, high-performance storage and computing at ALCF and OLCF for both data distribution and analysis.

The copying of 29 million files, twice, is a daunting task from three perspectives: (1) *Time*: A September 30, 2022, end of the service contract for LLNL storage hardware provided considerable urgency, demanding both a rapid start to the replication task and high data copying performance. (2) *Reliability*: It was important that replication not lead to data loss. (3) *Effort*: Replicating this large volume of data and number of files required substantial automation in order to keep human effort manageable.

In terms of performance, all three sites are fortunately connected to ESnet (see Figure 1) at 100 Gbps or higher speeds. Thus, data replication over the network was feasible, and indeed would require only two weeks if that peak speed could be realized. However, achieving such high speeds raised the ante on data transfer technology, as it had to be able to make efficient use of both high-speed links and source and destination file systems.

An early obstacle to rapid movement was soon determined to be the LLNL file system, which could source data only at about 1.5 GB/s. At that speed, it would take 58 days to copy data from LLNL → ALCF and then another 58 days for LLNL → OLCF. We therefore decided to copy files first from LLNL to one of the two LCF sites and then from that LCF site to the other. This approach was considerably faster, for three reasons: (1) inter-LCF transfers could proceed at a much higher rate, up to 7.5 GB/s; (2) LLNL-to-LCF and the inter-LCF transfers could proceed concurrently; and (3) if one LCF was unavailable (e.g., due to periodic maintenance), transfers from LLNL could nevertheless proceed to the other LCF. This situation emphasizes the importance of being able to reconfigure transfer paths flexibly.

Having verified that the required transfers could, in principle, be performed in time, the question arose as to how to automate those transfers so that they could be performed with modest or no human intervention. We employed Globus (K. Chard et al. 2016), a set of cloud-hosted services to which users can make requests such as "copy data from storage system A to storage system B." Local Globus Connect agents running at A and B then perform the transfer, under the supervision of the cloud service, which handles details such as authentication and authorization; striping to make efficient use of high-speed networks and parallel file systems; monitoring of progress; verification of file integrity; and, crucially, retries on failure.

This approach proved successful, with replication starting on February 15, 2022, and completing on May 3, 2022: a total of 77 days, not far from the theoretical minimum time possible (based on the rate-limiting LLNL file system) of 58 days. While some difficulties were encountered—notably, persistent failures at LLNL due to a misconfigured file system—the process overall was relatively painless, demonstrating the power of the overall data replication framework. This document is intended to capture lessons learned from this replication task. To that end, we: review briefly CMIP and ESGF, the ESnet network infrastructure, and the Globus platform (Section 2); describe the methodology followed in performing the replication

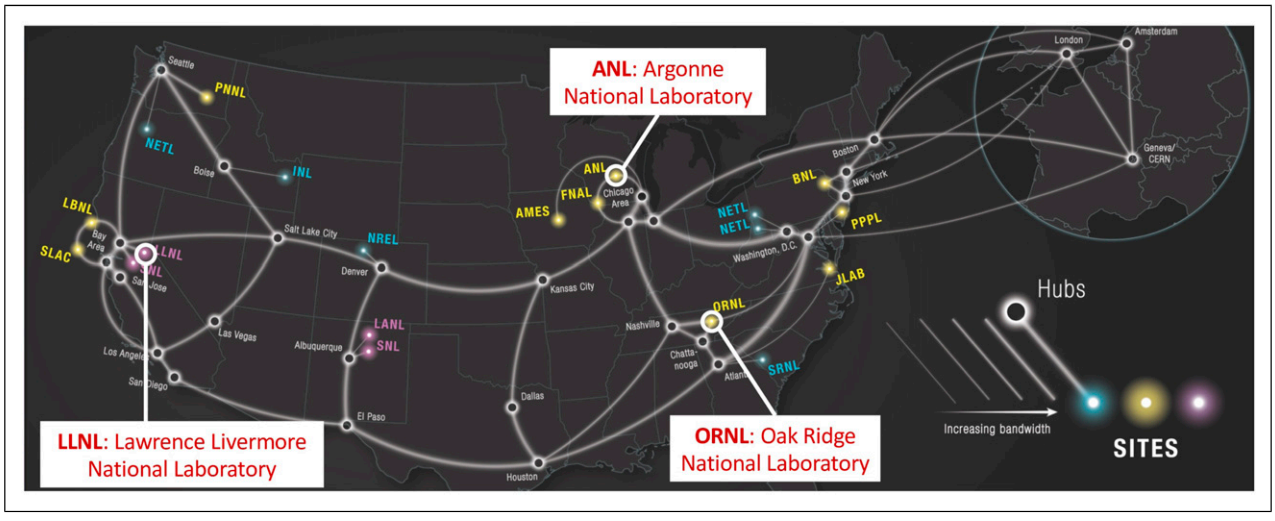


Figure 1. The DOE Energy Sciences network, ESnet, as of 2022, with the three sites involved in this data replication task highlighted. Map from https://www.es.net/about/.

(Section 3); describe the replication task itself (Section 4); discuss lessons learned (Section 5) and implications for data infrastructure (Section 6); and conclude (Section 7).

2. Background

2.1. CMIP and the earth system grid federation

The Coupled Model Intercomparison Projects (CMIPs) were established to facilitate systematic comparison of global Earth system models in order to increase understanding of atmospheric and oceanic variability, the processes that drive the Earth system, and potential future trajectories. In each of a series of "phases" (CMIP1, CMIP2, etc.), a set of standardized experiments are designed for all participating models to simulate. These experiments typically include both historical simulations and future scenario projections based on different forcing trajectories. Modeling groups worldwide then run their models using the prescribed experimental protocols.

ESGF provides the infrastructure and services required to collect these results and make them available to researchers worldwide, who use the data to improve models and perform a wide variety of scientific analyses. ESGF comprises a set of *nodes* worldwide: both *data nodes*, responsible for storing and distributing data, and *index nodes*, which manage metadata for data stored at data nodes and permit search of that metadata. A CMIP Data Node Operations Team (Petrie et al., 2021) coordinates work on ESGF data node architecture and deployment.

As already noted, the size of Coupled Model Intercomparison Project datasets has increased substantially over the years as a result of faster computers and more sophisticated Earth system models. In addition, model downscaling and other model comparison initiatives have been established that create further data that researchers want to access. One consequence of this increase in size is an evolution of how the distribution of Coupled Model Intercomparison Project data to researchers was achieved. For CMIP1 through CMIP3, a single central repository, at LLNL, was employed. Subsequently, recognizing that it made little sense for all data requests to proceed via a single site, ESGF was established as a peer-to-peer federation, in which multiple sites (around 20 at present) maintain copies of some or all datasets. (A small set of Tier 1 sites host all or most of the data, while other sites host subsets.) Researchers can thus choose from where they want to obtain a copy.

2.2. Physical infrastructure

As noted, the replication task involved copying data from LLNL to ALCF and OLCF. This task necessarily engages not only the wide area network connecting those sites but also the network elements that connect file systems at each site to the wide area network, and the file systems themselves: see Figure 2.

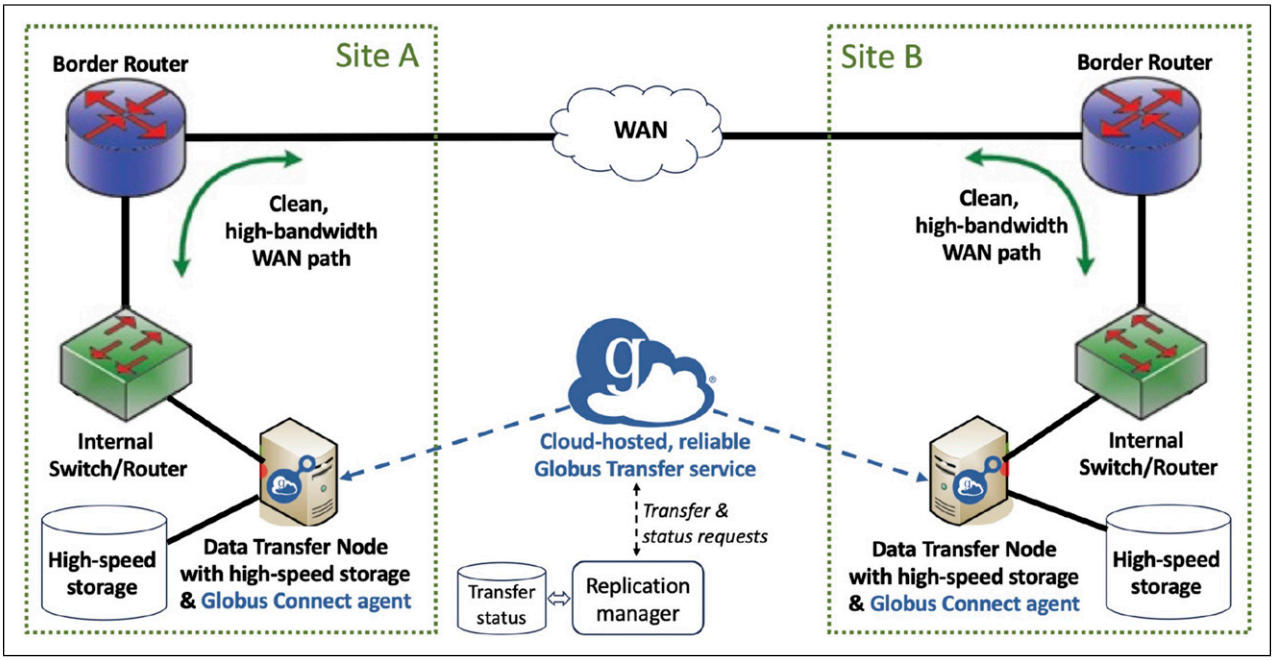


Figure 2. Principal elements of a high-performance data replication framework. High-performance storage systems at two sites, A and B, are connected to Data Transfer Nodes optimized for high-speed data movement and themselves connected to a wide area network (WAN) via a clean, high-bandwidth network path. Globus orchestrates data transfers, negotiating authentication and authorization, configuring transfer parameters for high-speed data movement, checking integrity of transfers, and detecting and responding to failures.

2.3.1. Wide area network. All three sites were connected to ESnet (see Figure 1) at 100 Gbps or higher speeds. ESnet is a science network that is optimized for high-speed, reliable data transfer; it makes it easy for appropriate data transfer protocols, such as the GridFTP (Allcock et al., 2005) used by Globus, to drive transfers at close to line rates.

2.3.2. Data transfer nodes. Each site has deployed one or more Data Transfer Nodes (DTNs)—e.g., four at ALCF—to enable rapid end-to-end data movement over the network (Dart, Allcock, et al., 2021; Dart, Rotman, et al., 2013). DTNs are specialized servers configured specifically for efficient, high-speed data transfers. Specifically, they are equipped with high-performance network interfaces, often 10 Gigabit Ethernet or higher, and are configured to optimize the data path to minimize latency and maximize throughput; are typically connected at or near the site network perimeter to take full advantage of high-speed ESnet connectivity; are also connected directly to high-speed storage; and run Globus Connect servers to handle large datasets, support reliable multi-stream file transfers, and manage security, access control, and logging.

2.3.3. File systems. The first and final step in storage-to-storage transfers such as those considered here is typically a high-performance parallel file system. At LLNL and OLCF, the file system was a General Parallel File System (GPFS) (Schmuck and Haskin 2002) deployment, while ALCF ran Lustre (Dart, Allcock, et al., 2021).

We note that both ALCF and OLCF, as high-performance computing centers, are configured with specialized equipment to provide the highest possible computational and data I/O performance for the most challenging computations. These characteristics make them excellent places to store enormous datasets on which scientists wish to perform advanced analyses. One less desirable feature is that they undergo both frequent planned, and occasional unplanned, maintenance periods, during which times storage systems are not available. ESGF is designed to compensate for such outages by allowing requests to be directed to other locations, but as we explain below, it is something to consider when performing data replication.

2.3. The Globus platform

The final major element of the infrastructure on which we perform our transfers is Globus. Operated by the University of Chicago, this platform enables secure, reliable, and performant file transfer, sharing, and data management automation throughout the research lifecycle. Globus comprises two primary components: persistent, replicated cloud-hosted management services (e.g., Globus Transfer for data transfers) and tens of thousands of local agents (e.g., Globus Connect Servers for data transfer) deployed at thousands of sites worldwide. Users make requests to a Globus service (e.g., the request, "transfer directory D from LLNL to ALCF", is made to Globus Transfer); after authentication and authorization, the service then issues requests to local agents (in our example, at LLNL and ALCF) and monitors their progress so as to ensure reliable and rapid completion.

Transfers are typically performed via the GridFTP protocol (Allcock et al., 2005), due to its abilities to engage multiple TCP streams and DTNs to accelerate transfers and restart interrupted transfers, and other features important for high-speed movement of large data. The Globus Connect software supports a large and growing selection of file systems, object stores, and hierarchical storage systems, and Globus Auth provides standards-based authentication and authorization to more than 1000 unique applications and services. Recent papers describe the use of Globus services to build data portals (K. Chard et al., 2018)

Table 1. Description of the transfer table used by the replication tool to track the progress of transfers over time.

Field name	Description
Dataset	The directory path to be transferred
Source	LLNL, ALCF, or OLCF
Destination	ALCF or OLCF
Uuid	Globus transfer identifier
Requested	Timestamp for transfer start
Completed	Timestamp for transfer end
Status	ACTIVE, SUCCEEDED, FAILED, QUEUED, or NULL
Directories	Number of directories transferred
Files	Number of files transferred
Rate	Transfer rate in bytes/s
Faults	Number of faults encountered
bytes_transferred	Number of bytes transferred

and analyze properties of Globus transfers (Liu et al., 2018).

3. Methodology

As noted, the purpose of this project was to replicate 7.3 PB of datasets located on disk storage at LLNL to both ALCF and OLCF. LLNL, ALCF, and OLCF all have Globus endpoints that provide access to their mass storage for individuals with accounts, and so performing the replication task required simply making the appropriate Globus transfer requests (Table 1).

ESGF stores datasets in directory hierarchies that can be up to eleven levels deep, with subdirectory names describing various aspects of an experiment. For example, the directory CMIP6.CMIP.MPI-M.MPI-ESM1-2-LR.historical contains multiple subdirectories, each corresponding to a different CMIP6 simulation performed by MPI-M with the MPI-ESM1-2-LR model and the historical scenario. Each of these subdirectories (e.g., r27i1p1f1) then contains output files from one of those simulations, organized in subdirectories according to table_id, variable_id, grid_label, and version: see Table 2. The directory hierarchies used for other sets of experiments (e.g., CMIP5, Obs4MIPS) were different.

In principle, we could have performed our replication task simply by initiating two Globus Transfer requests, each listing the directories at LLNL that were to be transferred, and specifying a destination at either ALCF or OLCF. In practice, things were not quite so simple, because: (1) We wanted to transfer data first to one LCF and then from that LCF to the other, so as to avoid moving files twice from the slower LLNL file system. (2) We wanted to enable transfers to continue even when one site was unavailable. (93) Not every file at LLNL was to be copied: instead, we were provided with listings of several thousand directory names. (4) We needed to detect and address certain failures that were beyond Globus' capabilities to handle, such as persistent failures of certain disk drives at LLNL. (5) When Globus is asked to transfer a set of directories and files, it first scans the source file system to determine the number and size of the files to be transferred, so that it can optimize subsequent data movement. However, the instability of the LLNL GPFS file system meant that it could hang if too many metadata requests were made at once.

For these reasons, we organized the overall replication task as a set of 2 × 2291 transfers, two per ESGF path: one from LLNL to ALCF and one from LLNL to OLCF. We launched these transfers over time programmatically, via a replication tool that we created for this purpose[1] that implements the logic presented in Figure 3. This script makes a series of Globus transfer requests, monitors their progress, and updates the database appropriately. It prioritizes the route LLNL → ALCF, but if any transfer to ALCF is PAUSED, then the script automatically runs instead a LLNL → OLCF transfer. (Transfers to/from a Globus collection can be paused by the collection manager – ALCF uses this mechanism to pause active Globus transfers involving ALCF endpoints before an ALCF maintenance period, to prevent the transfers from failing.) As soon as no transfers to ALCF are PAUSED, the script stops submitting new transfers LLNL → OLCF, while also submitting ALCF → OLCF transfers and OLCF → ALCF as needed to ensure that each dataset transferred to one of these locations is replicated to the other. Note that Globus performs integrity checking automatically for each file that was transmitted, computing and comparing checksums at source and destination, and retransmitting files that are found to be corrupted.

Once this initial replication task was completed, as described in Section 4, additional datasets continued to be published to the LLNL index node. Replication of those datasets is handled by an automated task that checks daily for new datasets and transfers them to ALCF and OLCF.

(1) Create a database table transfer and populate this table with two rows for each of the 2291 ESGF paths, one with source LLNL and destination ALCF and one with source LLNL and destination OLCF. Set the status of each such row to NULL.
(2) Repeatedly:

Table 2. CMIP6 directory hierarchy with, as an example, the directory/css03_data/CMIP6/CMIP/MPI-M/MPI-ESM1-2-LR/historical/r27i1p1f1/EdayZ/hus/gn/v20210901/.

Subdirectory	Example value	Meaning of the example value
Mip_era	CMIP6	CMIP phase 6
Activity_drs	CMIP	CMIP activity under which simulation performed
Institution_id	MPI-M	Max-planck-institute for meteorology
Source_id	MPI-ESM1-2-LR	MPI model used for simulation
Experiment_id	Historical	Simulation run under historical conditions
Member_id	r27i1p1f1	Unique ensemble member identifier
Table_id	EdayZ	Data type, here daily processed atmospheric data
Variable_id	Hus	Specific humidity
Grid_label	Gn	Output grid type: here, "grid native"
Version	v20210901	Version, identified by date

1. Create a database table `transfer` and populate this table with two rows for each of the 2291 ESGF paths, one with source LLNL and destination ALCF and one with source LLNL and destination OLCF. Set the status of each such row to NULL.
2. Repeatedly:
   (a) *Start LLNL→ALCF transfers if possible.* If the database contains fewer than two paths with status=ACTIVE, source=LLNL, destination=ALCF, and there is a path P in the database with destination=ALCF and status NULL or FAILED, then:
      - Request Globus to transfer recursively path P from LLNL to ALCF, and set status for P in the database to ACTIVE.
   (b) *Update database status for transfers that have succeeded or failed.* If there is a path P in the database with status ACTIVE then:
      - Check the transfer associated with path P with Globus; if that transfer has succeeded or failed, set status for P in the database to SUCCEEDED or FAILED, respectively.
   (c) *Start LLNL→OLCF transfers if transfers to ALCF are paused.* If any transfer to ALCF is PAUSED, if the database contains a path P with destination=OLCF and status NULL or FAILED, and if fewer than two LLNL→OLCF transfers are active then:
      - Request Globus to transfer recursively the path P from LLNL to OLCF and set status for P in the database to ACTIVE.
   (d) *Start ALCF→OLCF transfers if possible.* If the database contains a path P that has been successfully transferred to ALCF but not to OLCF, and if fewer than two ALCF→OLCF transfers are active then:
      - Request Globus to transfer recursively the path P from ALCF to OLCF, and set status for P in the database to ACTIVE.
   (e) *Start OLCF→ALCF transfers if possible.* If the database contains a path P that has been successfully transferred to OLCF but not to ALCF, and if fewer than two OLCF→ALCF transfers are active, then:
      - Request Globus to transfer recursively the path P from OLCF to ALCF, and set status for P in the database to ACTIVE.
   (f) *Check for termination.* If no transfers have status NULL, ACTIVE, FAILED, or PAUSED then:
      - Terminate.

Figure 3. The logic used by the data replication script.

(a) *Start LLNL → ALCF transfers if possible.* If the database contains fewer than two paths with status = ACTIVE, source = LLNL, destination = ALCF, and there is a path P in the database with destination = ALCF and status NULL or FAILED, then:

- Request Globus to transfer recursively path P from LLNL to ALCF, and set status for P in the database to ACTIVE.

(b) *Update database status for transfers that have succeeded or failed.* If there is a path P in the database with status ACTIVE then:

- Check the transfer associated with path P with Globus; if that transfer has succeeded or failed, set status for P in the database to SUCCEEDED or FAILED, respectively.

(c) S*tart LLNL → OLCF transfers if transfers to ALCF are paused.* If any transfer to ALCF is PAUSED, if the database contains a path P with destination = OLCF and status NULL or FAILED, and if fewer than two LLNL → OLCF transfers are active then:

- Request Globus to transfer recursively the path P from LLNL to OLCF and set status for P in the database to ACTIVE.

(d) S*tart ALCF → OLCF transfers if possible.* If the database contains a path P that has been successfully

transferred to ALCF but not to OLCF, and if fewer than two ALCF → OLCF transfers are active then:

- Request Globus to transfer recursively the path *P* from ALCF to OLCF, and set status for *P* in the database to ACTIVE.

(e) S*tart OLCF → ALCF transfers if possible.* If the database contains a path *P* that has been successfully transferred to OLCF but not to ALCF, and if fewer than two OLCF → ALCF transfers are active, then:

- Request Globus to transfer recursively the path *P* from OLCF to ALCF, and set status for *P* in the database to ACTIVE.

(f) C*heck for termination.* If no transfers have status NULL, ACTIVE, FAILED, or PAUSED then:

- Terminate.

4. The Data replication task

The transfer of ESGF data from LLNL to ALCF and OLCF was performed over the period from February 15 to May 4, 2022. This transfer involved 8 18 26 44 44 83 59 330 B (7.3 PB) at LLNL in 17,347,671 directories and 28,907,532 files, for a total of 15 PB in 35 million directories and 58 million files.

We show in Figure 4 two views over time of the overall replication task. We highlight in the figure the following primary phases:

① After some initial tests, the transfer began on February 15. In this first phase, the OLCF transfer node was not online and thus we only saw traffic from LLNL to ALCF. Transfer rate was around 1.5 GB/s, the speed of the LLNL file system interface.

② On February 20, ALCF started an extended maintenance period (a weekly occurrence) and thus LLNL → ALCF transfers ceased for the moment. Fortunately, OLCF came online and thus LLNL → OLCF transfers began. Speed remained about 1.5 GB/s.

③ On February 25, ALCF was operating again and thus from that point onwards we saw both steady LLNL → ALCF traffic and burstier ALCF → OLCF traffic, with the latter occurring as complete files arrived at ALCF. Note how, from February 27–28, files transferred LLNL → OLCF prior to February 25 were transferred OLCF → ALCF, and thus during that period transfers proceeded simultaneously in three directions. Similar phenomena were also seen later as a result of ALCF maintenance periods (e.g., March 22–23) and OLCF maintenance periods.

④ On April 16, transfer of the larger CMIP6 files completed and transfer of CMIP5 files began. Transfers were halted for a while due to permissions issues ("unreadable" files) that prevented Globus from retrieving files, and some associated instabilities in the LLNL GPFS file system: performance tuning of GPFS to improve transfer speed was attempted. However, such tuning was done experimentally (by trial-and-error) and backfired, causing the noted instabilities. (The tuning was removed, but performance remained much less than for CMIP6.)

⑤ By April 26, permissions and GPFS configuration problems had been corrected. From that date until the completion of the entire transfer on May 3, we saw sustained LLNL → ALCF traffic followed by burstier ALCF → OLCF traffic as files arrived at ALCF.

Overall files were transferred at an aggregate rate of about 1.5 GB/s to each of ALCF and OLCF, for a total of 3 GB/s. Instantaneous transfer rates varied according to the source-destination pair, the nature of the files being transferred, and perhaps, on occasion, competing traffic. We show average transfer rates in Table 3. Two transfers were generally in progress over each source-destination pair at any one time, and thus the average achieved rate was roughly twice these numbers (Transfer rates are in gigabytes per second, i.e., 2^{30} B/s.). The highest single-link speed observed was more than 7.5 GB/s from OLCF to ALCF.

As noted earlier, data were, whenever possible, copied first from LLNL to ALCF and then from ALCF to OLCF; thus the quantity available at ALCF generally exceeded slightly that at OLCF until the end of the transfer. However, the order was sometimes reversed to LLNL → OLCF → ALCF when ALCF was unavailable due to maintenance: e.g., see February 25–27.

While many data sets transferred without faults, transient faults were not infrequently encountered due to network and file system errors. As noted in Table 3, we we recorded a total of 4086 errors, for an average of 1.05 per transfer for the 3881 transfers for which error metadata were recorded. Errors were not distributed uniformly across transfers, with just 1069 transfers having any fault recorded, and a small number having many: see Figure 5. Our use of Globus meant that these errors did not interrupt the overall replication process—a nice illustration of the high importance of automated fault recovery in a large-scale data management system. *Once data sets become very large, even low-probability failures occur regularly.* By reliably detecting and automatically recovering from these low-probability failures, Globus allowed the overall system to remain

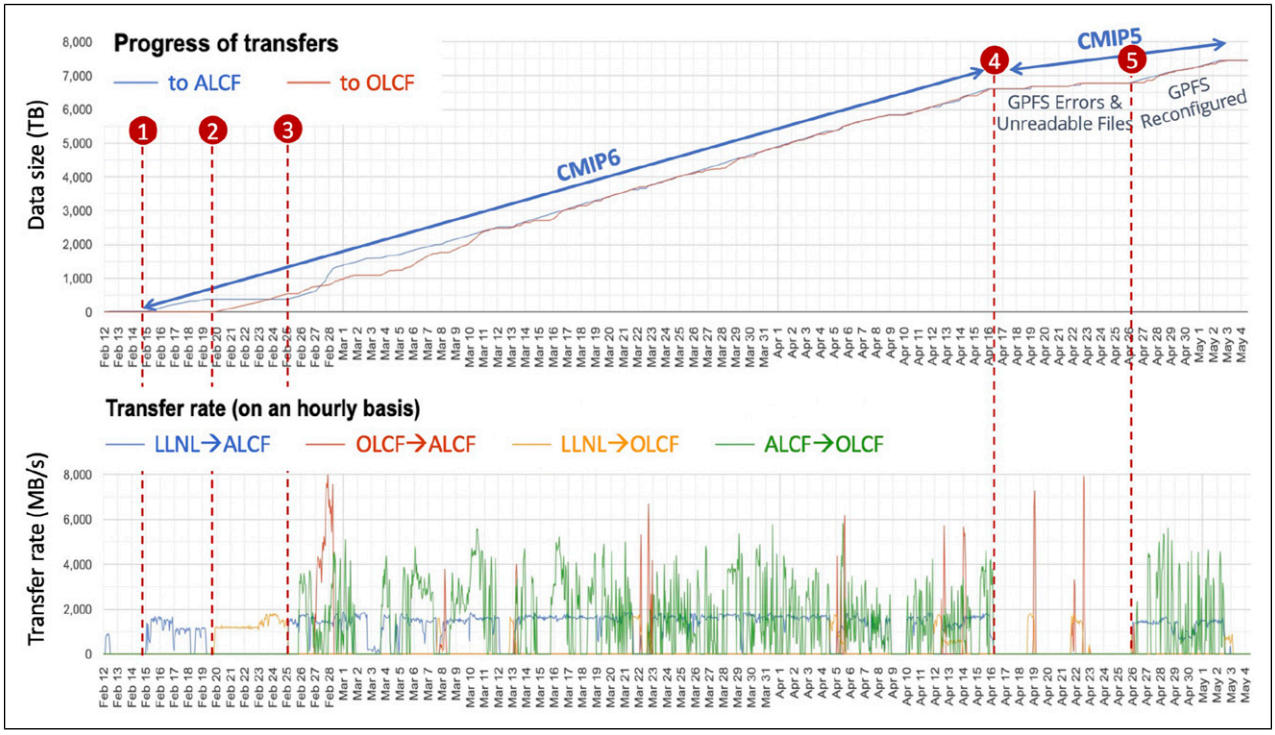


Figure 4. Two views of the replication task. Above: Cumulative bytes received at ALCF and OLCF, shown as separate lines, with some significant phases labeled. Below: Instantaneous transfer rates for the four source-destination pairs, each depicted with a different color. See text for further discussion.

reliable and robust even as the volumes of data to be managed scaled by orders of magnitude.

5. Lessons learned

Based on our experiences, we offer the following recommendations for others undertaking large-scale data replication tasks. These lessons are framed as actionable guidance that can be applied to similar projects.

- Plan for automated fault recovery from the start. As already noted in Section 4, we observed numerous "failures" (events that prevent data movement) during the replication task, of a variety of types (e.g., bad permissions, system maintenance periods, packet corruption). These errors are not typically frequent if measured in terms of errors per unit data transferred, but they occur far too often in large replication tasks to be handled manually. Fortunately, they are all, to varying degrees, transient: most can be corrected by retrying, others by notifying an appropriate person of a need for corrective action. Globus, by both retrying on failures and notifying upon repeated failures, provides a means for recovery from such transient

Table 3. Aggregate data on the 2 × 2291 = 4582 transfers performed for the two CMIP phases over different paths during the replication task. For transfers, Missing counts transfers that lack metadata, due to a change in recording policies after replication started. Rate is faults/transfer and Max is maximum number of faults observed for any transfer for that path and phase.

CMIP	Path	Average	Transfers		Faults/transfer	
Phase	Source → destination	GB/s	Total	Missing	Mean	Max
CMIP5	LLNL → ALCF	0.633	70		1.86	26
	LLNL → OLCF	0.666	60		3.17	123
	ALCF → OLCF	2.853	68		2.65	56
	OLCF → ALCF	3.500	58		0.14	4
CMIP6	LLNL → ALCF	0.648	2015	591	1.84	410
	LLNL → OLCF	0.662	148	110	6.24	42
	ALCF → OLCF	1.706	2015		0.20	55
	OLCF → ALCF	2.352	148		2.13	111

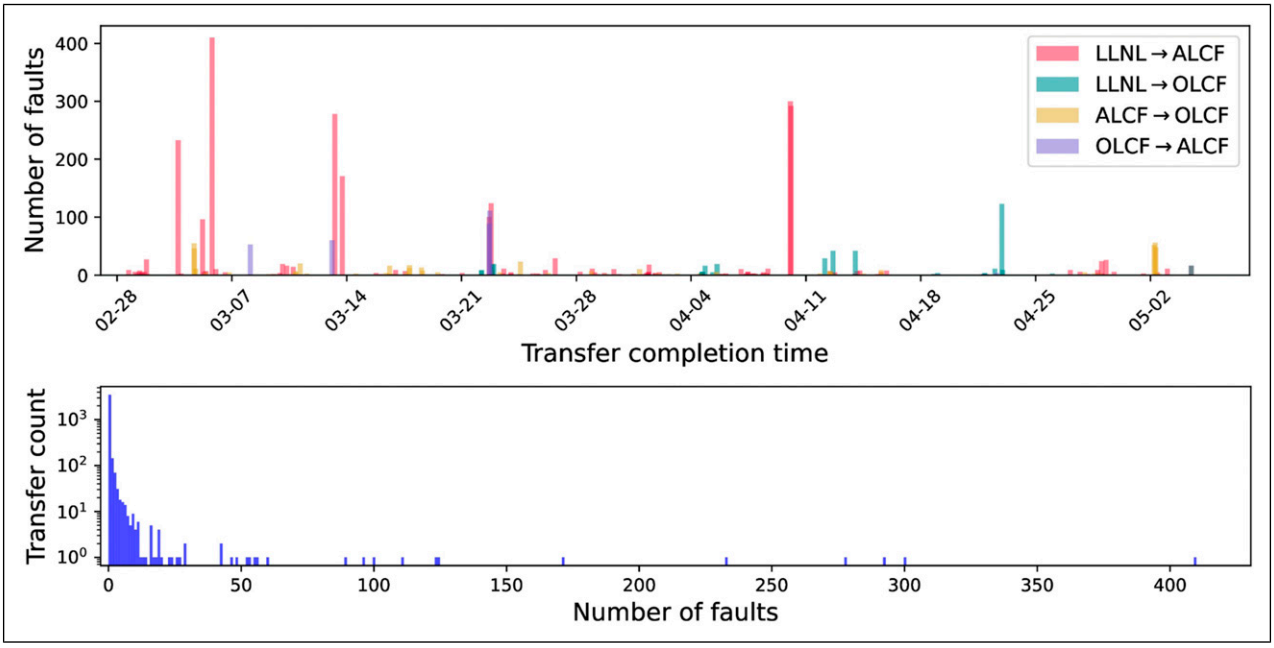


Figure 5. Above: Number of faults per transfer versus date/time at which transfers completed. Below: Frequencies (note log scale) for different numbers of faults per transfer. Overall, we see that while many transfers had faults, most faults were associated with a relatively small number of transfers.

errors. Recommendation: Any large-scale replication system must include automatic retry logic and failure notification. Do not assume manual intervention will be feasible—at PB scale, even rare errors (e.g., 1 per million files) will occur thousands of times.

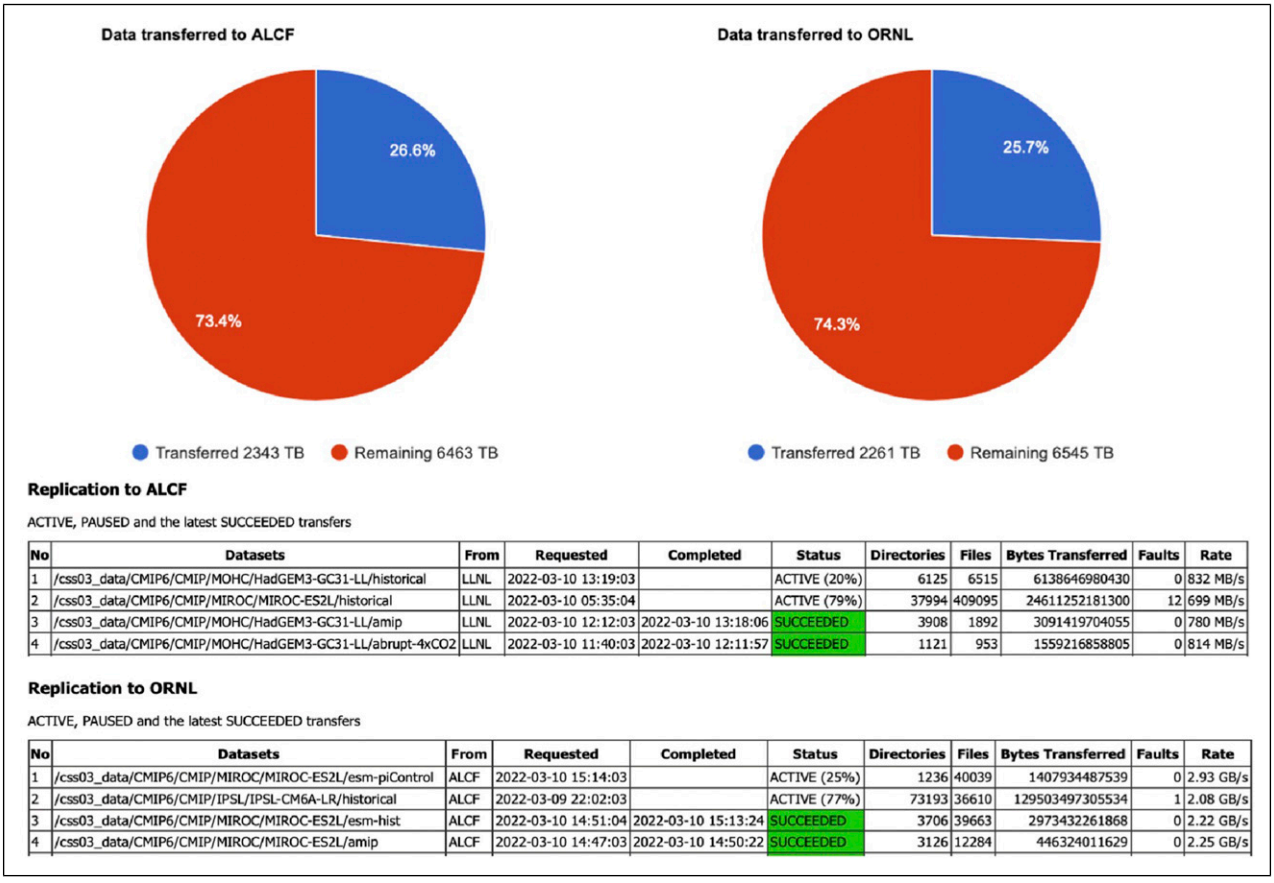
- Design for scheduled downtime at HPC centers. One "failure" case that bears special consideration in the case of large HPC centers is the sometimes long maintenance periods necessitated by cutting edge systems and small staffs. Our replication tool, with its special handling of PAUSED tasks, is an example of how one can respond to such situations. Recommendation: When replicating to/from HPC facilities, build in logic to detect maintenance windows (e.g., via PAUSED transfer status) and automatically reroute transfers to alternate paths. Having multiple destination sites enables continued progress even when one site is unavailable.
- Measure and exploit asymmetric transfer performance. The wide area performance achieved between different file systems can vary significantly in ways that are not symmetric (i.e., speed(A → B) ≠ speed(B → A)). Overall replication performance can be improved significantly by considering such differences. Recommendation: Before large transfers, benchmark all potential paths between sites. In our case, LLNL → ALCF ran at 0.65 GB/s while ALCF → OLCF achieved 1.7–3.5 GB/s. We exploited this by routing data through the faster intermediate site rather than transferring twice from the slower source.
- Identify the true bottleneck—it is often the file system, not the network. Particularly in the case of well-provisioned science networks such as ESnet and facilities such as ALCF and OLCF, file systems are often the bottleneck for data transfers. It is important to: measure end-to-end performance; engage file system administrators to adjust file system configurations when required; and consider the performance achieved for real transfers, not advertised network bandwidth, when planning large transfers. An example of such an effort, and the resultant performance increases that can be achieved, is described in Dart, Allcock, et al. (2021). Recommendation: Run test transfers early to measure actual throughput. Our 100 Gbps network links could have supported ∼12 GB/s, but LLNL's file system limited us to 1.5 GB/s—a factor of 8× below network capacity. Use measured rates, not theoretical bandwidth, for project planning and timeline estimation.
- Be aware that directory scanning can be expensive. A Globus transfer request can either pass a list of files to transfer or, alternatively, request recursive transfer of specified directories; in the latter case, Globus then traverses those directories to determine their contents. (Globus uses the resulting information for, among other things, configuring transfer parameters such as number of concurrent transfers and pipeline depth.) In the work presented here, we adopted the latter approach. We found at one point that scanning of an extremely large directory led to out-of-memory errors

on a LLNL computer, a problem that we addressed by performing multiple smaller subdirectory transfers. Recommendation: For very large directory trees, break transfers into smaller units to avoid memory exhaustion during the scanning phase. We found that organizing our 7.3 PB transfer into ∼2300 separate transfer requests (averaging ∼3 TB each) avoided scanning-related failures.

- The "sync" option is not always the fastest recovery method. Globus provides a "sync" option for transfers that compares files and directories at the source endpoint with those at the destination endpoint, checking (in a user-configurable manner) factors such as file names, sizes, checksums, and last modified timestamps, and transferring only files that are new or that have changed. We initially sought to use this option to recover from certain failed transfers, but found that it was generally faster (due to slow scanning on the LLNL file system) to transfer all files again than to ask endpoints to scan directories/paths and compare hashes. Presumably this observation might not hold in other situations. Recommendation: When recovering from partial failures, compare the cost of sync (which requires scanning both endpoints) versus simply re-transferring. On file systems with slow metadata operations, re-transfer may be faster. Test both approaches on representative data before committing to a recovery strategy.
- Run multiple concurrent transfers to overlap scanning with data movement. As noted above, we found that transfer requests that involve too many files could cause problems on the LLNL GPFS file system, we believe due to the memory demands of the scanning step. Our solution was to organize the replication task into more (∼3000) requests. In addition, we generally ran two transfer requests concurrently from each source to each destination, so that scanning by one could overlap with transfers by a second. Recommendation: Run 2–4 concurrent transfer requests per source-destination pair, possibly staggering the start of each request to avoid competition for the initial file system scan at the source. While one transfer scans directories, another can be actively moving data, improving overall throughput. This also provides natural checkpointing—if a transfer fails, only that portion needs to be retried.
- Build a monitoring dashboard for visibility and early problem detection. We established a replication dashboard to enable real-time tracking of progress. This dashboard is still accessible at https://dashboard.globus.org/esgf, although now that the replication task is over it is somewhat lacking in interest. We show in Figure 6 a picture of this dashboard on March 10, 2022, when the replication task was about 25% complete. We found this dashboard to be useful for communicating the progress of the replication task to management and to collaborators, and on occasion for spotting failures (e.g., out-of-memory errors at LLNL mentioned above). This dashboard was relatively easy to create, but is not a standard Globus feature. Recommendation: Invest early in building a simple monitoring dashboard that shows transfer progress, active/failed/completed counts, and current throughput. This pays dividends in stakeholder communication, helps identify stalls or failures quickly, and provides data for post-hoc analysis. The Globus API makes extracting this information straightforward.
- Structure transfers as a set of independent units for flexibility. We implemented our replication script to operate on a set of directory transfers, a property that it leverages to redirect transfers during maintenance periods. Globus or the replication script could be adapted to work on a single large transfer, but the "set of directories" approach seems to work well. Recommendation: Organize large replication tasks as many independent transfer units (e.g., one per top-level directory or dataset). This enables: (a) flexible routing around unavailable sites, (b) parallel transfers via multiple paths, (c) fine-grained progress tracking, and (d) easier recovery from failures. Store transfer state in a database for persistence across script restarts.
- A simple script suffices, but consider a persistent service for ongoing replication. The replication script that we implemented to manage transfers was not complicated, but it would be useful to turn it into a persistent service in the future. Recommendation: For one-time bulk transfers, a straightforward script (like ours, available at https://github.com/esgf2-us/data-replication-tools) with database-backed state tracking is sufficient. For ongoing replication needs, consider wrapping this logic in a persistent service with scheduling, alerting, and automatic handling of new datasets. Globus Flows (R. Chard et al., 2023) provides a natural platform for implementing such a service, enabling automated, event-driven orchestration of transfers, notifications of new data availability, automated visualization or analysis tasks, and other actions.

6. Discussion

A review of networking requirements for Earth system research notes that storage limitations at ESGF partner sites mean that no one site has a complete copy of the CMIP6 archive (Zurawski et al., 2023). Thus a scientist wishing to access a specific dataset must first determine the site(s) that hold the data, and then either perform in situ analysis, if supported by a site; copy it to another site that supports in situ analysis; or download the data for local analysis. As datasets become larger, replicating to another



Replication to ALCF

ACTIVE, PAUSED and the latest SUCCEEDED transfers

No	Datasets	From	Requested	Completed	Status	Directories	Files	Bytes Transferred	Faults	Rate
1	/css03_data/CMIP6/CMIP/MOHC/HadGEM3-GC31-LL/historical	LLNL	2022-03-10 13:19:03		ACTIVE (20%)	6125	6515	6138646980430	0	832 MB/s
2	/css03_data/CMIP6/CMIP/MIROC/MIROC-ES2L/historical	LLNL	2022-03-10 05:35:04		ACTIVE (79%)	37994	409095	24611252181300	12	699 MB/s
3	/css03_data/CMIP6/CMIP/MOHC/HadGEM3-GC31-LL/amip	LLNL	2022-03-10 12:12:03	2022-03-10 13:18:06	SUCCEEDED	3908	1892	3091419704055	0	780 MB/s
4	/css03_data/CMIP6/CMIP/MOHC/HadGEM3-GC31-LL/abrupt-4xCO2	LLNL	2022-03-10 11:40:03	2022-03-10 12:11:57	SUCCEEDED	1121	953	1559216858805	0	814 MB/s

Replication to ORNL

ACTIVE, PAUSED and the latest SUCCEEDED transfers

No	Datasets	From	Requested	Completed	Status	Directories	Files	Bytes Transferred	Faults	Rate
1	/css03_data/CMIP6/CMIP/MIROC/MIROC-ES2L/esm-piControl	ALCF	2022-03-10 15:14:03		ACTIVE (25%)	1236	40039	1407934487539	0	2.93 GB/s
2	/css03_data/CMIP6/CMIP/IPSL/IPSL-CM6A-LR/historical	ALCF	2022-03-09 22:02:03		ACTIVE (77%)	73193	36610	129503497305534	1	2.08 GB/s
3	/css03_data/CMIP6/CMIP/MIROC/MIROC-ES2L/esm-hist	ALCF	2022-03-10 14:51:04	2022-03-10 15:13:24	SUCCEEDED	3706	39663	2973432261868	0	2.22 GB/s
4	/css03_data/CMIP6/CMIP/MIROC/MIROC-ES2L/amip	ALCF	2022-03-10 14:47:03	2022-03-10 14:50:22	SUCCEEDED	3126	12284	446324011629	0	2.25 GB/s

Figure 6. A March 10, 2022, view of a dashboard created to show progress of the data replication task. We see the instantaneous quantities and fraction of the total data copied successfully to each destination, and both currently active and completed transfers (eight are shown here). "Remaining" numbers below the circles are incorrect because they were based on a higher initial estimate of totals.

ESGF site is likely increasingly to be the preferred option, making the ability to replicate large datasets rapidly and reliably among peer centers increasingly important. Similar large-scale data replication challenges arise in other domains; for example, the Rucio system (Barisits et al., 2019) manages exabyte-scale data distribution for CERN's particle physics experiments across globally distributed sites, while Google's Effingo (Pápay et al., 2024) transfers over an exabyte of data daily across dozens of clusters worldwide.

High-speed networks make the replication of large datasets feasible, but such replication tasks are nevertheless more challenging than in the past due to the larger quantities of data and numbers of files and the need for specialized methods to make effective use of faster networks. Fortunately, reliable and rapid data replication can be treated as a solved problem if supported by the right infrastructure. Specifically, as illustrated in Figure 2, the source and destination sites need: access to a high-speed wide area network (WAN); a clean, high-bandwidth path from WAN to Data Transfer Nodes associated with their storage; and Globus Connect agents deployed on the Data Transfer Nodes. The Globus Transfer service can then be used to drive high-speed, reliable, and secure data transfers.

Such a data replication infrastructure is deployed across the three US DOE ESGF sites, and as we have demonstrated in this project, enables large data replication tasks to be run largely automatically and at close to line network speeds for extended periods. Especially given the much larger data volumes expected for CMIP7, it would seem advantageous to deploy such data replication also at other major ESGF sites.

7. Conclusions

As science data become ever larger and networks more capable we see increasing needs to copy large datasets among sites. This copying may be for preservation, to be closer to computation, for merging with other data, or for numerous other reasons; once copied, the data may be used and discarded, or alternatively preserved for extended periods. Regardless of the reasons why and intended use, the ability to transfer data rapidly (at close to line speeds), reliably (without data loss or corruption), securely (e.g., without exposing sensitive data), and automatically (with little or no human intervention) is of growing importance.

Here we have described the methodology and technology that we employed to replicate 7.3 petabytes of computational simulation data from Lawrence Livermore National Laboratory, previously the only Tier 1 Earth System Grid Federation data node in the U.S., to Argonne and Oak Ridge National Laboratories, in order to provide faster and more resilient access. We describe this task not because it is in any way remarkable (indeed, it was performed quasi-automatically) but because we expect the methods employed to perform this transfer to be of interest to workers in many disciplines.

That we could perform the replication task easily, reliably, and rapidly is thanks to the exemplary data replication infrastructure provided across LLNL, ANL, and ORNL by ESnet and Globus. We described the important components of this infrastructure in Sections 2.2 and 2.3. ESnet and major computer centers provide high-performance physical infrastructure, while Globus enables large data replication tasks to be run largely automatically and at high speeds for extended periods. As noted in the ESGF Future Architecture Report (Kershaw et al., 2020), such data replication infrastructure is essential for ESGF and the Coupled Model Intercomparison Projects. It is also important for many other applications. As such, it also represents, in its sustained high performance and large degree of automation, a success story for DOE's Integrated Research Infrastructure (IRI) program (Miller et al., 2023), for which *long-term campaigns* and *data-integration-intensive* applications are important use cases (Dart et al., 2023). The lessons learned here are also relevant to emerging initiatives such as the American Science Cloud, which aims to integrate DOE computing facilities and data resources across twelve national laboratories.

We hope that this report will encourage broader deployment and use of such infrastructure. One area of obvious need in the context of ESGF is the next Coupled Model Intercomparison Project phase, CMIP7, which will feature datasets much larger than in CMIP6. Deployment of the data replication infrastructure described here across all major ESGF sites will do much to accelerate research.

Acknowledgments

We thank Dr. Justin Jay Hnilo at DOE BER for his stewardship of the ESGF2-US Project. We thank Andrew Cherry of the Argonne Leadership Computing Facility (ALCF) for his assistance, and the ALCF and Oak Ridge Leadership Computing Facility (OLCF), DOE Office of Science User Facilities supported under Contracts DE-AC02-06CH11357 and DE-AC05-00OR22725, respectively, for access to computing resources used in experiments. ESnet is operated by Lawrence Berkeley National Laboratory (Berkeley Lab), which is operated by the University of California for the US Department of Energy under contract DE-AC02-05CH11231.

ORCID iD

Ian T. Foster https://orcid.org/0000-0003-2129-5269

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of his article: This work was supported by the Biological and Environmental Research; DE-AC02-06CH11357, Work at Argonne National Laboratory; DE-AC02-06CH11357, Work at Lawrence Livermore National; DE-AC52-07NA27344, Work at Oak Ridge National Laboratory; DE-AC05-00OR22725.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Note

1. ESGF2-US Data Replication Tool https://github.com/esgf2-us/data-replication-tools.

References

Allcock W, Bresnahan J, Kettimuthu R, et al. (2005) The globus striped GridFTP framework and server ACM/IEEE Conference on Supercomputing. *November*. IEEE Computer Society, 11-14, 54.

Barisits M, Beermann T, Berghaus F, et al. (2019) Rucio: scientific data management. *Computing and Software for Big Science* 3(1): 11. https://doi.org/10.1007/s41781-019-0026-3

Bernholdt D, Bharathi S, Brown D, et al. (2005) The Earth system grid. *Proceedings of the IEEE* 93.3: 485–495. https://arxiv.org/abs/0712.2262

Chard K, Tuecke S and Foster I (2016) Globus: recent enhancements and future plans. *XSEDE16 Conference on Diversity, Big Data, and Science at Scale, July 16-21*. ACM, 1–8. Available at: https://doi.org/10.1145/2949550.2949554

Chard K, Dart E, Foster I, et al. (2018) "The modern research data portal: a design pattern for networked, data-intensive science". *In: PeerJ Computer Science* 4: e144. https://doi.org/10.7717/peerj-cs.144

Chard R, Pruyne J, McKee K, et al. (2023) Globus automation services: research process automation across the space–time continuum. *Future Generation Computer Systems* 142: 393–409. https://doi.org/10.1016/j.future.2023.01.010

Cinquini L, Crichton D, Mattmann C, et al. (2014) The Earth system grid Federation: an open infrastructure for access to distributed geospatial data. *Future Generation Computer Systems* 36: 400–417. https://doi.org/10.1016/j.future.2013.07.002

Dart E, Allcock W, Bhimji W, et al. (2021) "The petascale DTN project: high performance data transfer for HPC facilities". In: Preprint arXiv:2105. 12880.

Dart E, Rotman L, Tierney B, et al. (2013) The science DMZ: a network design pattern for data-intensive science, In: International Conference on High Performance Computing, Networking, Storage and Analysis, November 17-21, pp. 1–10. ACM. https://doi.org/10.1145/2503210.2503245

Dart E, Zurawski J, Hawk C, et al. (2023) *ESnet Requirements Review Program Through the IRI Lens: A Meta-Analysis of*

Workflow Patterns Across DOE Office of Science Programs. Tech. Rep. Lawrence Berkeley National Laboratory (LBNL). Available at: https://escholarship.org/uc/item/9fg8k5xh.

Durack P, Taylor K, Eyring V, et al. (2017) *input4MIPs: Making CMIP Model Forcing More Transparent. Tech. Rep*. Lawrence Livermore National Laboratory. Available at: https://doi.org/10.2172/1463030

Eyring V, Bony S, Meehl GA, et al. (2016) Overview of the coupled model intercomparison project phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development* 9(5): 1937–1958. https://doi.org/10.5194/gmd-9-1937-2016

Kershaw P, Abdulla G, Ames S, et al. (2020) *ESGF Future Architecture Report (V.1.1). Tech. Rep. LLNL-TR-812915*. Lawrence Livermore National Laboratory. Available at: https://www.osti.gov/servlets/purl/1643759.

Liu Z, Kettimuthu R, Foster I, et al. (2018). "*Cross-geography scientific data transferring trends and behavior*". In: *27th International Symposium on High-Performance Parallel and Distributed Computing*, July 15-18, pp. 267–278. ACM. https://www.osti.gov/servlets/purl/1468117.Tempe,Arizona:ACM. https://doi.org/10.1145/3208040.3208053. isbn: 9781450357852.

Miller W, Bard D, Boehnlein A, et al. (2023) *Integrated Research Infrastructure architecture blueprint activity (final report 2023). Tech. rep. US Department of Energy, Office of Science.* Available at: https://doi.org/10.2172/1984466

Pápay L, Pustelnik J, Rzadca K, et al. (2024) *An exabyte a day: throughput-oriented, large scale, managed data transfers with Effingo*, In: Proceedings of the ACM SIGCOMM 2024 Conference, August 4-8, pp. 459–473. ACM. DOI: 10.1145/3651890.3672262.

Petrie R, Denvil S, Ames S, et al. (2021) "Coordinating an operational data distribution network for CMIP6 data". *In: Geoscientific Model Development* 14(1): 629–644. https://doi.org/10.5194/gmd-14-629-2021

Schmuck F and Haskin R (2002) GPFS: a shared-disk file system for large computing clusters, In: Conference on File and Storage Technologies, January 28-30. USENIX.

Taylor KE, Stouffer RJ and Meehl GA (2012) An overview of CMIP5 and the experiment design. *Bulletin of the American Meteorological Society* 93(4): 485–498. https://doi.org/10.1175/bams-d-11-00094.1

Teixeira J, Waliser D, Ferraro R, et al. (2014) Satellite observations for CMIP5: the genesis of Obs4MIPs. *Bulletin of the American Meteorological Society* 95(9): 1329–1334. https://doi.org/10.1175/bams-d-12-00204.1

Williams DN, Ananthakrishnan R, Bernholdt D, et al. (2009) The Earth System Grid. *Bulletin of the American Meteorological Society* 90.2: 195–206.

Zurawski J, Dart E, Harlan Z, et al. (2023) *Biological and Environmental Research Network Requirements Review Final Report. Tech. Rep*. Lawrence Berkeley National Laboratory (LBNL). Available at: https://escholarship.org/uc/item/3mz7h3mm.

Author biographies

Lukasz Lacinski is a software developer at the University of Chicago and Argonne National Laboratory, working with the Globus team.

Lee Liming is Director of Professional Services in the Globus team at Argonne National Laboratory. He holds a BSE from the University of Michigan.

Steven Turoscy is Professional Services Manager at Globus.org and Argonne National Laboratory.

Cameron Harr served as the Livermore Computing I/O strategist and team lead for Lustre operations at Lawrence Livermore National Laboratory. Heholds a B.S. from Brigham Young University.

Kyle Chard is a Research Associate Professor in the Department of Computer Science at the University of Chicago with a joint appointment at Argonne National Laboratory. He co-leads Globus Labs and received his Ph.D. from Victoria University of Wellington, New Zealand.

Eli Dart is a Network Engineer at ESnet (Energy Sciences Network), Lawrence Berkeley National Laboratory. He is the creator of the Science DMZ network architecture.

Paul J. Durack is a Senior Manager at Western Australia's Department of Water and Environmental Regulation. He received the World Climate Research Programme 2018 Data Prize and was a contributing author to the IPCC Sixth Assessment Report.

Sasha Ames is a computational scientist at Lawrence Livermore National Laboratory, where he leads ESGF development and operations. He was educated at UC Santa Cruz.

Forrest M. Hoffman is a Corporate Fellow and Group Leader for the Integrated Computational Earth Sciences Group at Oak Ridge National Laboratory. He is a Fellow of AAAS and Joint Faculty Member at the University of Tennessee.

Ian T. Foster is the Arthur Holly Compton Distinguished Service Professor of Computer Science at the University of Chicago and Senior Scientist and Distinguished Fellow at Argonne National Laboratory. His honors include the Gordon Bell Prize, the Lovelace Medal, and the IEEE Internet Award.