



Efficient and Secure Transfer, Synchronization, and Sharing of Big Data

Kyle Chard, Steven Tuecke, and Ian Foster, University of Chicago and Argonne National Laboratory

Globus supports standard data interfaces and common security models for securely accessing, transferring, synchronizing, and sharing large quantities of data.

Cloud computing's unprecedented adoption by commercial and scientific communities is due in part to its elastic computing capability, pay-as-you-go usage model, and inherent scalability. Cloud platforms are proving to be viable alternatives to in-house resources for scholarly applications, with researchers in areas spanning physical and natural sciences through the arts regularly using them.¹ As we enter the era of big data and data-driven research—the “fourth paradigm of science”²—researchers face challenges related to hosting, organizing, transferring, sharing, and analyzing large quantities of data. Many believe that cloud models provide an ideal platform for supporting big data.

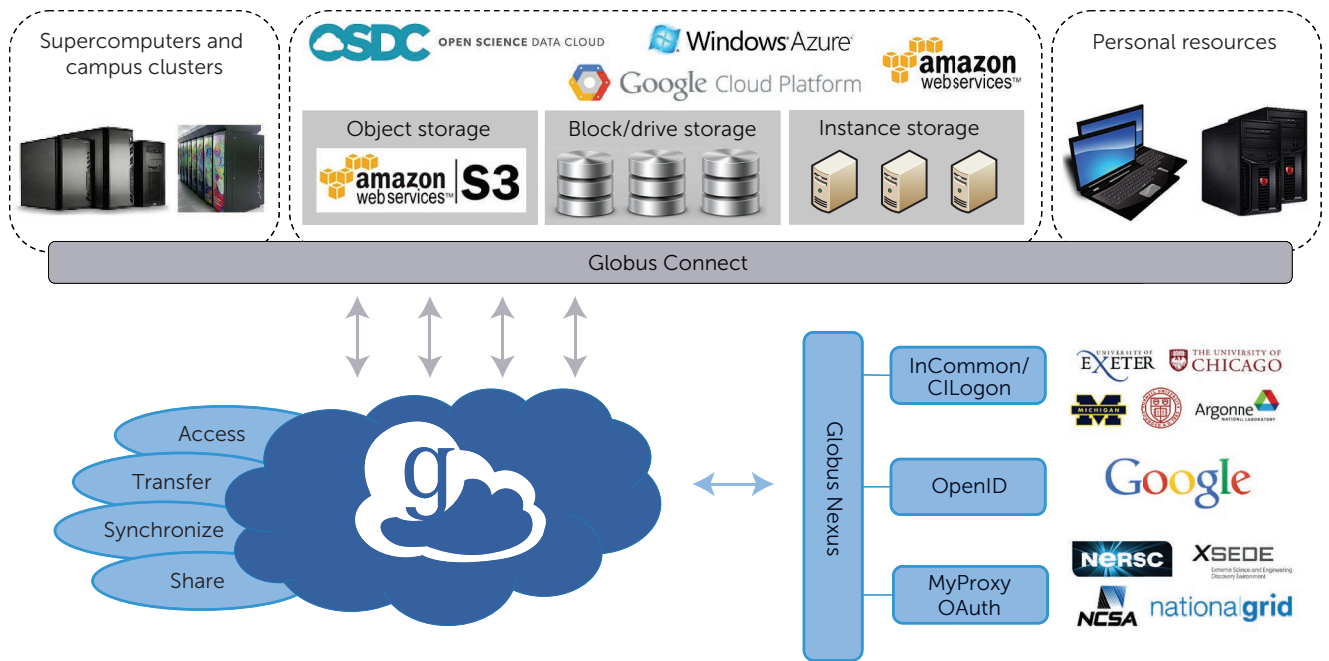


FIGURE 1. Globus provides transfer, synchronization, and sharing of data across a wide variety of storage resources. Globus Nexus provides a security layer through which users can authenticate using a number of linked identities. Globus Connect provides a standard API for accessing storage resources.

Large scientific datasets are increasingly hosted on both public and private clouds. For example, public datasets hosted by Amazon Web Services (AWS) include 20 Tbytes of NASA Earth science data, 500 Tbytes of Web-crawled data, and 200 Tbytes of genomic data from the 1000 Genomes project. Open clouds such as the Open Science Data Cloud (OSDC)³ host many of the same research datasets in their collection of more than 1 Pbyte of open data. Thus, it's frequently convenient, efficient, and cost-effective to work with these datasets on the cloud. In addition to these high-profile public datasets, many researchers store and work with large datasets distributed across a plethora of cloud and local storage systems. For example, researchers might use datasets stored in object stores such as Amazon Simple Storage Service (S3), large mountable block stores such as Amazon Elastic Block Store (EBS), instance storage attached to running cloud virtual machine (VM) instances, and other data stored on their institutional clusters, personal computers, and in super-computing centers.

Given the distribution and diversity of storage as well as increasingly huge data sizes, we need standardized, secure, and efficient methods to access data, move it to other systems for analysis, synchronize changing datasets across systems without copying the entire dataset, and share data with collaborators and others for extension and verification. Although high-performance methods are clearly required as data sizes grow, secure methods are equally important, given that these datasets might include medical, personal, financial, government, and intellectual property data. Thus, we need models that provide a standard interface through which users can perform these actions and methods that leverage proven security models to provide a common interface and single-sign-on. These approaches must also be easy to use, scalable, efficient, and independent of storage type.

Globus is a hosted provider of high-performance, reliable, and secure data transfer, synchronization, and sharing.⁴ In essence, it establishes a huge distributed data cloud through a vast network of

Globus-accessible endpoints—storage resources that implement Globus’s data access APIs. Through this cloud, users can access, move, and share large amounts of data remotely, without worrying about performance, reliability, or data integrity.

Globus: Large-Scale Research Data Management as a Service

Figure 1 gives a high-level view of the Globus ecosystem. Core Globus capabilities are split into two services: *Globus Nexus* manages user identities and groups,⁵ whereas the *Globus transfer service* manages transfer, synchronization, and sharing tasks on the user’s behalf.⁶ Both services offer programmatic APIs and clients to access their functionality remotely. They’re also accessible via the Globus Web interface (www.globus.org).

Globus Nexus provides the high-level security fabric that supports authentication and authorization. Its identity management function lets users create and manage a Globus identity; users can create a profile associated with their identity, which they can then use to make authorization decisions. It also acts as an identity hub, where users can link external identities to their Globus identity. Users can authenticate with Globus through these linked external identities using a single-sign-on model. Supported identities include campus identities using InCommon/CILogon via OAuth, Google accounts via OpenID, XSEDE accounts via MyProxy OAuth, an Interoperable Global Trust Federation (IGTF)-certified X.509 certificate authority, and Secure Socket Shell (SSH) key pairs. To support collective authorization decisions (such as when sharing data with collaborators), Globus Nexus also supports the creation and management of user-defined groups.

The Globus transfer service provides core data management capabilities and implements an associated data access security fabric. Globus uses the GridFTP protocol⁷ to transfer data between logical endpoints—a Globus representation of an accessible GridFTP server. GridFTP extends FTP to improve performance, enable third-party transfers, and support enhanced security models. The basic Globus model for accessing and moving data requires deploying a GridFTP server on a computer and registering a corresponding logical endpoint in Globus. The GridFTP server must be configured with an authentication provider that handles the mapping of credentials to user accounts. Often, authentication is provided by a co-located MyProxy credential management system,⁸ which lets users obtain short-term X.509 certificate-based proxy credentials by authenticating with a plug-in authentication module (for

example, local user accounts, Lightweight Directory Access Protocol [LDAP], or InCommon/CILogon).

Globus uses two separate communication channels. The *control* channel is established between Globus and the endpoint to start and manage transfers, retrieve directory listings, and establish the data channel. The *data* channel is established directly between two Globus endpoints (GridFTP servers) and is used for data flowing between systems. The data channel is inaccessible to the Globus service, so no data passes through it.

Several capabilities differentiate Globus from its competitors:

- **High performance.** Globus tunes performance based on heuristics to maximize throughput using techniques such as pipelining and parallel datastreams.
- **Reliable.** Globus manages every stage of data transfer, periodically checks transfer performance, recovers from errors by retrying transfers, and notifies users of various events (such as errors and success). At the conclusion of a transfer, Globus compares checksums to ensure data integrity.
- **Secure.** Globus implements best practices security approaches with respect to user authentication and authorization, securely manages the storage and transmission of credentials to endpoints for authentication, and supports optional data encryption.
- **Third-party transfer.** Unlike most transfer mechanisms (such as SCP [secure copy]) Globus facilitates third-party transfers between two remote endpoints. That is, rather than maintain a persistent connection to an endpoint, users can start a transfer and then let Globus manage it for the duration of the transfer.
- **High availability.** Globus is hosted using a distributed, replicated, and redundant hosting model deployed across several AWS availability zones. In the past year, Globus and its constituent services have achieved 99.96 percent availability.
- **Accessible.** Because Globus is a software-as-a-service (SaaS) provider, users can access its capabilities without installing client software locally, so they can start and manage transfers through their Web browsers, or using the Globus command-line interface or REST API.

In three and a half years of operation, Globus has attracted more than 18,000 registered users, of which approximately 200 to 250 are active every

day, and has conducted nearly 1 million transfers, collectively containing more than 2 billion files and 52 Pbytes of data. Figure 2 summarizes the Globus transfers over this period. The graphs include only transfer tasks (that is, they don't include mkdir, delete, and so on) in which data has been transferred (for example, they don't include sync jobs that don't transfer files) between nontesting endpoints (that is, they ignore Globus test endpoints `go#ep1` and `go#ep2`). Figure 2a shows the frequency of the total number of bytes transferred in a single transfer task (note log bins), and Figure 2b shows the frequency of the total number of files and directories transferred in a single transfer task. As Figure 2a shows, the most common transfers are between 100 Mbytes and 1 Gbyte (81,624 total transfers), whereas more than 700 transfers have moved tens of Tbytes of data and 39 have moved hundreds of Tbytes (max 500.415 Tbytes). The most common number of files and directories transferred is less than 10; however, more than 400 transfers have moved more than 1 million files each (max 39,643,018), and 120 transfers have moved more than 100,000 directories (max 7,675,096). Figure 2 highlights the huge scale at which Globus operates in terms of data sizes transferred, number of files and directories moved, and number of transfers conducted.

Extending the Globus Data Cloud

Globus currently supports a network of more than 8,000 active (used within the last year) endpoints distributed across the world and hosted at a variety of locations, from PCs to supercomputers. Users can already access and transfer data from many locations via Globus—supercomputing centers such as the National Center for Supercomputing Applications (NCSA) and the San Diego Supercomputer Center (SDSC); university research computing centers such as those at the University of Chicago; cloud platforms such as Amazon Web Services and the Open Science Data Cloud (OSDC); large user facilities such as CERN and Argonne National Laboratory's Advanced Photon Source; and commercial data providers such as PerkinElmer. This vast collection of accessible endpoints ensures that new Globus users have access to large quantities of data immediately.

As new users join Globus, they often require access to new storage resources (including their own PCs). Thus, an important goal is to provide trivial methods for making resources accessible via Globus. To allow data access via Globus, storage systems must be configured with a GridFTP server and some authentication method. To ease this process, we de-

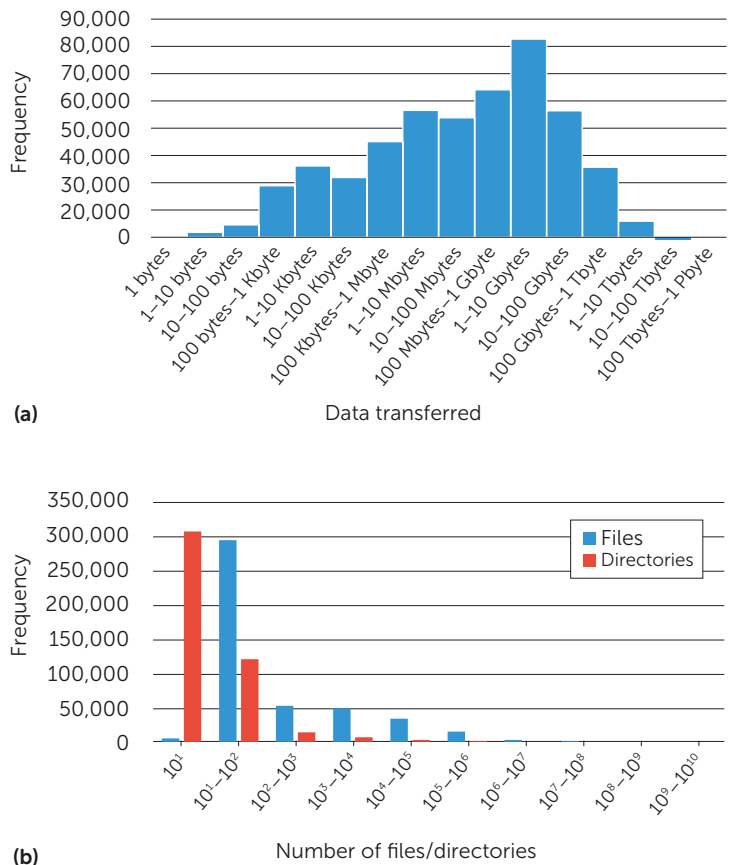


FIGURE 2. Frequency of transfers with given transfer size and number of files and directories. Transfer task frequency for (a) total transfer size, and (b) number of files and directories.

veloped Globus Connect, a software package that can be deployed quickly and easily to make resources accessible to Globus. We developed two versions of Globus Connect for different deployment scenarios.

Globus Connect Personal is a lightweight single-user agent that operates in the background much like other SaaS agents (such as Google Drive and Dropbox). A unique key is created for each installation and is used to peer Globus Connect to the user's Globus account, ensuring that the endpoint is only accessible to its owner. Because we designed Globus Connect Personal for installation on PCs, it supports operation on networks behind firewalls and network address translation (NAT) through its use of outbound connections and relay servers (similar to other user agents such as Skype). Because it can run in user space, it doesn't require administrator privileges. Globus Connect Personal is available for Linux, Windows, and MacOS.

Globus Connect Server is a multiuser server installation that supports advanced configuration

options. It includes a full GridFTP server and an optional colocated MyProxy server for authentication. Alternatively, users can configure existing authentication sources upon installation. The installation process requires a one-command setup and completion of a configuration file that defines aspects such as the endpoint name, file system restrictions, network interface, and authentication method. Globus Connect Server also supports multiserver data transfer node configurations to provide increased throughput. Globus Connect Server is available as native Debian and RedHat packages.

With Globus Connect, users can quickly expose any type of storage resource to the Globus cloud. They can use lightweight Globus Connect Personal endpoints on PCs and even short-lived cloud instances. They can even script the download and configuration of these endpoints for programmatic execution. For more frequently used resources with multiple users (such as data transfer nodes, clusters,

sure data integrity). Users can also leverage Globus's synchronization and sharing capabilities directly from S3 endpoints.

Globus S3 endpoints support transfers directly from existing endpoints, so don't require data staging via a Globus Connect deployment hosted on Amazon's cloud. This approach differs from GreenButton WarpDrive (www.greenbutton.com), which, although it also uses GridFTP, relies on a pool of GridFTP servers hosted on cloud instances. Globus's S3 support builds upon extensions to GridFTP to support communication directly between S3 and GridFTP servers. Globus enables user-controlled registration of logical S3 endpoints requiring only details identifying the storage location (that is, the S3 bucket) and appropriate information required to connect to the S3 endpoint. To provide secure access to data stored in S3, while also enabling user-controlled sharing via Globus, we leverage Amazon's Identity and Access Management (IAM) service to delegate control of an S3 bucket to a trusted Globus user. We peer this Globus IAM user with the Globus transfer service via trusted credentials. Thus, when delegating access of an S3 bucket, Globus can base authorization decisions on internal policies (such as sharing permissions) to allow transfers between other Globus endpoints and the S3 endpoint.

One of the most common requirements associated with big data is the ability to share data with collaborators.

storage providers, long-term and high-performance storage such as High Performance Storage System [HPSS]), they can deploy Globus Connect Server and leverage institutional identity providers. They can then scale deployments over time by adding Globus Connect Server nodes to load balance transfers. Both versions support all Globus features including access, transfer, synchronization, and sharing.

Supporting Cloud Object Stores

To allow users to access a variety of cloud storage systems, Globus supports the creation of endpoints directly on Amazon S3 object storage. Users can thus access, transfer, and share data between S3 and existing Globus endpoints as they do between any other Globus endpoints. To access S3, users must create an S3-backed endpoint that maps to a specific S3 bucket to which they have access. With this model, users can expose access to large datasets stored in S3 and benefit from Globus's advanced features, including high performance and reliable transfer, rather than relying on standard HTTP support (which doesn't scale to large datasets and doesn't en-

Providing Scalable In-Place Data Sharing

One of the most common requirements associated with big data (and scientific data in general) is the ability to share data with collaborators. Current models for data sharing are limited in many ways, especially as data sizes increase. For example, cloud-based mechanisms such as Dropbox require that users first move (replicate) their data to the cloud, which is both costly and time consuming. Ad hoc models, such as directly sharing from institutional storage, require manual configuration, creation, and management of remote user accounts, making them difficult to manage and audit. These difficulties become insurmountable when data is large and when dynamic sharing changes are required. Rather than implement yet another storage service, we focus on enabling in-place data sharing. That is, shared data does not reside on Globus; rather, Globus lets users control who can access their data directly on their existing endpoints.

To share data in Globus, a user selects a file system location and creates a shared endpoint—that is, a virtual endpoint rooted at the shared location on his or her file system. The user can then select

other users, or groups of users, who can access the shared endpoint—or parts thereof—by specifying fine-grained read and write permissions. One advantage of this model is that permission changes are reflected immediately, so users can revoke access to a shared dataset instantly.

Globus's sharing capabilities are extensions built onto the GridFTP server, which, when enabled, let the GridFTP server delegate authorization decisions to Globus. Specifically, two new GridFTP `site` commands let Globus check that sharing is enabled on an endpoint and create a new shared endpoint. We also extended the GridFTP access protocol to allow access by a predefined trusted Globus user. The access request includes additional parameters such as the shared owner, shared user, and access control list (ACL) for the shared endpoint, which Globus maintains. When accessing the endpoint, this information is passed to the GridFTP server to enable delegated authorization decisions from the requesting user to the local user account of the shared endpoint owner. Using this approach, the GridFTP server can perform an authorization check to ensure that the shared user can access the requested path before following the normal access protocol, which requires changing to the shared endpoint owner's local user account and performing the requested action.

Secure Data Access, Transfer, and Sharing

There are a wide range of potential security implications when accessing distributed data, hosted by different providers, across security domains, and using different security protocols. Globus's multilayered architecture leverages standard security protocols to manage authentication and authorization, and avoid unnecessary storage of (or access to) users' credentials and data. Most importantly, data does not pass through Globus; rather, it acts as a mediator, allowing endpoints to establish secure connections between one another.

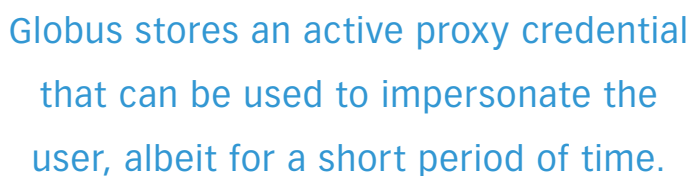
Authentication and Authorization

At the heart of the Globus security model is Globus Nexus, which facilitates the complex security protocols required to access the Globus service and endpoints using Globus identities as well as linked external identities.

Globus stores identities (and groups) in a connected graph. For Globus identities, it stores hashed and salted passwords for comparison when authenticating. For the linked identities (SSH public keys,

X509 certificates, OpenID identities, InCommon/CILogon OAuth, and so on) used to provide single-sign-on, it stores only public information, such as SSH public keys, X509 certificates, OpenID identity URLs and usernames, and OAuth provider servers, certificates, and usernames. Thus, when authenticating, Globus can validate a user's identity by following the private authentication process using cryptographic techniques rather than comparing passwords. Consider, for example, authenticating using a campus identity. Here, Globus leverages the InCommon/CILogon system and the OAuth protocol to let users enter their username/password via a trusted campus website. Globus passes a request token with the user authentication and receives an OAuth token and signature in return, which it exchanges for an OAuth access token (and later a certificate) from the campus identity provider.

Linked identities, such as XSEDE identities, are also used for single-sign-on access to endpoints.



Globus stores an active proxy credential that can be used to impersonate the user, albeit for a short period of time.

Rather than require users to authenticate multiple times for every action and to allow Globus to manage transfers on a user's behalf, Globus stores short-term proxy credentials. This allows Globus to perform important transfer-management tasks such as restarting transfers upon error. Here, Globus stores an active proxy credential that can be used to impersonate the user, albeit for a short period of time. To do so securely, Globus only caches the active credential and encrypts it using a private key owned by Globus Nexus. When the active credential is required (for example, to compute a file checksum on an endpoint), the credential is decrypted and passed to the specific GridFTP server over the encrypted control channel.

Endpoint Data Access and Transfer

GridFTP uses the Grid Security Infrastructure (GSI), a specification that allows secure and delegated communication between services in distributed computing environments. GridFTP relies on external services to authenticate users and provide trusted signed certificates (typically from a MyProxy

server) used to access the server. These certificates are often hidden from users by the use of an online certificate authority (CA), such as MyProxy. The GridFTP service has a certificate containing the hostname and host information that it uses to identify itself. (This certificate is created automatically when users install Globus Connect or it can be issued by a CA.) In Globus Connect, the MyProxy server can be optionally installed to issue short-term certificates on demand. Globus Connect can also be configured to use external MyProxy servers. Globus, GridFTP, and MyProxy servers are configured to trust the certificates exchanged between each other.

MyProxy servers let users obtain short-term credentials that a GridFTP server uses to assert user access to the file system. Administrators can configure MyProxy servers to use various mechanisms for authentication through pluggable authentication

As part of shared endpoint creation, a unique token is created on the GridFTP server for each shared endpoint.

modules (PAMs). Usually, these PAMs support local system credentials or institutional LDAP credentials. There are two basic models in which Globus uses a MyProxy server to obtain a credential. In the first, Globus passes the user's username and password to the MyProxy server and receives a credential in response. Thus, users must trust Globus not to store their passwords and to transfer them securely. In the second and preferred model, Globus uses the OAuth protocol to redirect the user to the MyProxy server to authenticate directly (that is, Globus doesn't see the username and password), and the server returns a credential in the OAuth redirection workflow.

When accessing data on an endpoint, Globus uses SSL/TLS to authenticate with the registered GridFTP server using the user's certificate. The GridFTP server validates the user's certificate, retrieves a mapping to a local user account from a predefined mechanism (such as a GridMap file), and changes the local user account (used to access the file system) to the requesting user's local account. Subsequent file system access occurs as the authenticated user's local account. To provide an additional layer of security, endpoint administrators can configure path restrictions (`restrict_paths`) that

limit GridFTP access to particular parts of the file system. For instance, administrators might allow access only to users' home directories or to specialized locations on the file system.

The flow of data between endpoints (including S3-backed endpoints and shared endpoints) is another potential area of vulnerability because data can travel on the general Internet. To provide secure data transfer, Globus supports data encryption based on secure sockets layer (SSL) connections between endpoints. In the case of S3 endpoints, the connection uses HTTPS. To avoid unnecessary overhead of less sensitive data, encryption is not a default setting and must be explicitly selected for individual transfers. The control channel used to start and manage transfers is always encrypted to avoid potential visibility of credential, transfer, and file system information.

Secure Sharing

Globus sharing creates several new security considerations, such as requiring secure peering of shared endpoints and Globus, authorizing access to shared data, and ensuring that file system information is not disclosed outside of the restricted shared endpoint.

The Globus sharing model requires the GridFTP server to be explicitly configured to allow sharing. As part of this process, the GridFTP server is configured to allow a trusted Globus user to access the server (and to later change the local user account to the shared endpoint owner's local user account). A unique distinguished name (DN) obtained from a Globus CA operated for this purpose identifies the user. The GridFTP server is configured to trust both this special Globus user and the Globus CA via the registered DN. During configuration, administrators can set restrictions (`sharing_rp`) defining what files and paths may be shared on the file system and which users may create shared endpoints. For example, administrators could limit sharing to a particular path (analogous to a `public_html` directory) and a subset of administrative users.

As part of shared endpoint creation, a unique token is created on the GridFTP server for each shared endpoint. This token is used to safeguard against redirection and man-in-the-middle attacks. For instance, an attacker who gains control of a compromised Globus account might change the physical GridFTP server associated with a trusted endpoint (for example, an XSEDE endpoint) to a malicious endpoint under the attacker's control. In this case, the attacker can create a shared end-

point and can then change the physical server back to the trusted server. Because the unique token is created on the malicious server, it won't be present on the trusted (XSEDE) server, so the attacker won't be able to exploit the shared endpoint to access the trusted server.

Accessing data on a shared endpoint using the extended GridFTP protocol lets Globus access the GridFTP server (as the trusted Globus account). The extended access request specifies data location, shared endpoint owner, the user accessing the shared endpoint, and current ACLs for that shared endpoint. To ensure that data is accessed only within the boundaries of what has been shared and within restrictions placed by the server administrator, the GridFTP server checks restricted paths, shared paths, and Globus ACLs (in that order). Assuming nothing negates the access, the GridFTP server changes the local user account, with which it accesses the file system, to the shared endpoint owner's local user account and satisfies the request.


Finally, because potentially sensitive path information could be included in a shared file path, Globus hides the root path from users accessing the shared endpoint. For example, if a user shares the directory "/kyle/secret/," it will appear simply as "/~/ " through the shared endpoint. Globus translates paths before sending requests to the GridFTP server.

Hosting and Security Policies


All Globus services are hosted on AWS. Although this environment has many advantages, such as high availability and elastic scalability, as with all hosting options, it also has inherent risks. We mitigate these risks by following best practices with respect to deployment and management of instances. These practices include storing all sensitive state encrypted, isolating data stores from the general Internet so they're only accessible to Globus service nodes (by AWS security groups), performing active intrusion detection and log monitoring to discover threats, auditing public-facing services and using strict firewalls to restrict access to predefined ports, and establishing backup processes to ensure that all data is encrypted before it's put in cloud storage. To ensure that these practices are followed, we conducted an external security review,⁹ and resolved the identified issues.

One important security aspect relates to policies for responding to security breaches and vulnerabilities. The recent HeartBleed bug is an example of a security vulnerability that affected a huge number

of websites across the world. Although Globus uses custom data transfer protocols that are unlikely targets of such an attack, exploits via the website, endpoints, and linked identity providers are still possible. In this particular case, we followed pre-defined internal security policies to determine if the vulnerability impacted our services, patched the issue for all Globus services and Globus-managed endpoints, and generated new private keys. We then followed internal processes for responding to potentially compromised user access by revoking user access tokens (invalidating all user sessions) and analyzing access logs. Finally, because of the exploit's nature, we analyzed all user endpoints to identify potentially vulnerable endpoints. We then contacted administrators of these endpoints and recommended that they take specific measures to patch the systems.



One important security aspect relates to policies for responding to security breaches and vulnerabilities.



As data sizes increase, researchers must look toward more efficient ways of storing, organizing, accessing, sharing and analyzing data. Although Globus's capabilities make it easy to access, transfer, and share large amounts of data across an ever-increasing ecosystem of active data endpoints, it also provides a framework on which new approaches for efficiently managing and interacting with big data can be explored.

The predominant use of file-based data is often inefficient because the data required for analysis doesn't always match the model used to store it. Researchers typically slice climate data in different ways depending on the analysis—for example, geographically, temporally, or based on a specific type of data such as rainfall or temperature. Accessing entire datasets when only small subsets of it are of interest is both impractical and inefficient. Although some data protocols, such as the Open source Project for a Network Data Access Protocol (OpenDAP), provide methods for accessing data subsets within files, no standard model for accessing a wide range of data formats currently exists. Recently, researchers have proposed more sophisticated data access models within GridFTP that use dynamic query and subsetting operations to retrieve (or transfer) data

subsets.¹⁰ Although this work presents a potential model for providing such capabilities, further work is needed to generalize the approach across data types and to develop a flexible and usable language to express such restrictions.

Files typically contain valuable metadata that can be used for organization, browsing, and discovery. However, accessing this metadata is often difficult because it's stored in various science-specific formats, often encoded in proprietary binary formats, and typically unstructured (or at least doesn't follow standard conventions). Moreover, even when the metadata is accessible, few high-level methods exist for browsing it across many files or across storage systems. Often, the line between metadata and data is blurred, and, whereas metadata might be unnecessary for some analyses, it can be valuable for others. Thus, we need methods that enable structured access to both data and metadata using common formats. Given that metadata can describe data or contain other sensitive information (for example, patient names), it's equally important to provide secure access methods. We therefore need models that expose such metadata to users and let them query over it to find relevant data for analysis or share it in a scalable and secure manner.

Often, data sharing occurs for the purpose of publishing to the wider community or as part of a publication. Considerable research has explored current data publishing practices.^{11,12} In many cases, researchers found that data wasn't published with papers and that original datasets couldn't be located. This affects one of the core principles of scientific discovery: that research is reproducible and verifiable. In response, funding agencies and publishers are increasingly placing strict requirements on data availability associated with grants and publications, although these requirements are often disregarded.¹² Even when researchers do publish data, they often do so poorly, in an ad hoc manner that makes the data difficult to find and understand (due to a lack of metadata), and with little guarantee that the data is unchanged or complete. We need new systems that let researchers publish data, easily associate persistent identifiers (such as DOIs) with that data, provide guarantees that the data is immutable and consistent with what was published, provide common interfaces for discovering and accessing published data, and do so at scales that correspond to the growth of big data.

Although these three areas represent different research endeavors, they all require a framework that supports efficient and secure data access. Globus provides a model on which we can continue to

innovate in these areas to provide enhanced capabilities directly through the existing network of Globus endpoints. We benefit from using Globus's transfer and sharing capabilities and from leveraging the same structured approaches toward authentication and authorization.

We intend to continue to develop support for other cloud storage and cloud providers, such as persistent long-term storage like Amazon Glacier and storage models used by other cloud providers (Microsoft Azure Storage, for example), with the goal of developing an increasingly broad data cloud. ●●

Acknowledgments

We thank the Globus team for implementing and operating Globus services. This work was supported in part by the US National Institutes of Health through NIGMS grant U24 GM104203, the Bio-Informatics Research Network Coordinating Center (BIRN-CC), the US Department of Energy through grant DE-AC02-06CH11357, and the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by US National Science Foundation grant number ACI-1053575.

References

1. D. Lifka et al., *XSEDE Cloud Survey Report*, tech. report 20130919-XSEDE-Reports-CloudSurvey-v1.0, XSEDE, 2013.
2. T. Hey, S. Tansley, and K. Tolle, eds., *The Fourth Paradigm: Data-Intensive Scientific Discovery*, Microsoft Research, 2009.
3. R.L. Grossman et al., "The Design of a Community Science Cloud: The Open Science Data Cloud Perspective," *Proc. 2012 SC Companion: High Performance Computing, Networking Storage and Analysis (SCC 12)*, 2012, pp. 1051–1057.
4. I. Foster, "Globus Online: Accelerating and Democratizing Science through Cloud-Based Services," *IEEE Internet Computing*, vol. 15, no. 3, 2011, pp. 70–73.
5. R. Ananthakrishnan et al., "Globus Nexus: An Identity, Profile, and Group Management Platform for Science Gateways and Other Collaborative Science Applications," *Proc. IEEE Int'l Conf. Cluster Computing (CLUSTER)*, 2013, pp. 1–3.
6. B. Allen et al., "Software as a Service for Data Scientists," *Comm. ACM*, vol. 55, no. 2, 2012, pp. 81–88.
7. W. Allcock et al., "The Globus Striped GridFTP Framework and Server," *Proc. 2005 ACM/IEEE Conf. Supercomputing (SC 05)*, pp. 54–64.
8. J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy,"


Proc. 10th IEEE Int'l Symp. High Performance Distributed Computing, 2001, pp. 104–111.

9. V. Welch, *Globus Online Security Review*, tech. report, Indiana Univ., 2012; <https://scholarworks.iu.edu/dspace/handle/2022/14147>.
10. Y. Su et al., “SDQuery DSI: Integrating Data Management Support with a Wide Area Data Transfer Protocol,” *Proc. Int'l Conf. High Performance Computing, Networking, Storage and Analysis (SC 13)*, 2013, article 47.
11. T.H. Vines et al., “The Availability of Research Data Declines Rapidly with Article Age,” *Current Biology*, vol. 24, no. 1, 2014, pp. 94–97.
12. A.A. Alsheikh-Ali et al., “Public Availability of Published Research Data in High-Impact Journals,” *PLoS ONE*, vol. 6, no. 9, 2011, e24357.

KYLE CHARD is a senior researcher at the Computation Institute, a joint venture between the University of Chicago and Argonne National Laboratory. His research interests include distributed meta-scheduling, grid and cloud computing, economic resource allocation, social computing, and services computing. Chard received a PhD in computer science from Victoria University of Wellington, New Zealand. Contact him at chard@uchicago.edu.

STEVEN TUECKE is deputy director at the University of Chicago's Computation Institute, where he's responsible for leading and contributing to projects in computational science, high-performance and distributed computing, and biomedical informatics. Tuecke received a BA in mathematics and computer science from St Olaf College. Contact him at tuecke@uchicago.edu.

IAN FOSTER is director of the Computation Institute, a joint institute of the University of Chicago and Argonne National Laboratory. He is also an Argonne senior scientist and distinguished fellow, and the Arthur Holly Compton Distinguished Service Professor of Computer Science. His research interests include distributed, parallel, and data-intensive computing technologies, and innovative applications of those technologies to scientific problems in such domains as climate change and biomedicine. Foster received a PhD in computer science from Imperial College, United Kingdom. Contact him at foster@mcs.anl.gov.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



IEEE Computer Society | Software Engineering Institute

Watts S. Humphrey Software Process Achievement Award

Nomination Deadline: January 15, 2015

Do you know a person or team that deserves recognition for their process improvement activities?

The IEEE Computer Society/Software Engineering Institute Watts S. Humphrey Software Process Achievement Award is presented to recognize outstanding achievements in improving the ability of a target organization to create and evolve software.

The award may be presented to an individual or a group, and the achievements can be the result of any type of process improvement activity.

To nominate an individual or group for a Humphrey SPA Award, please visit <http://www.computer.org/portal/web/awards/spa>

