Instant GridFTP

Rajkumar Kettimuthu, Lukasz Lacinski, Mike Link, Karl Pickett, Steve Tuecke, and Ian Foster
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL
Computation Institute, University of Chicago/Argonne National Laboratory, Chicago, IL

***Abstract*— A foundational need in high-performance computing is to move large (multi-gigabyte and even terabyte) datasets between sites. Simple file transfer mechanisms such as FTP and SCP are not sufficient from either a reliability or performance perspective. GridFTP is the de facto standard protocol for transferring large data files in production Grid/HPC environments. GridFTP extends the standard FTP protocol to provide a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks. The Globus GridFTP implementation has become the preeminent high-performance data transfer tool for the Grid community, with large facilities and projects using it to transfer billions of files per year. We report here on a new product, Globus Connect Multi User (GCMU), that greatly streamlines Globus GridFTP installation and configuration. GCMU packages a GridFTP server, MyProxy Online Certificate Authority, and other components in a manner that avoids the need for any end-user or system administrator involvement in security configuration or credential management. We describe the GCMU design and a GridFTP protocol extension that simplifies transfers across security domains. We also explain how GCMU interacts with the Globus Online software-as-a-service solution. By enabling "instant GridFTP," this work makes the powerful Globus GridFTP tool accessible to nonexpert users and to smaller laboratories and projects.**

Keywords – GridFTP, Grid data movement, high-speed transfers, secure WAN transfers, GCMU, Globus Online

I. INTRODUCTION

Research often requires transferring large amounts of data among instruments, Web portals, local servers, HPC clusters, and laptops or desktops. Traditional methods such as FTP [1] and SCP are ill-suited to data movement on this scale because of their poor performance and reliability. GridFTP [2] is a powerful solution that has been developed over the past decade to address these big data transport requirements. The GridFTP protocol extends the standard File Transfer Protocol (FTP) with useful features such as Grid Security Infrastructure (GSI) security [3], increased reliability via restart markers, high-performance data transfer by using striping and parallel streams, and support for third-party transfer between GridFTP servers. GridFTP has been shown to deliver multiple orders of magnitude higher throughput than do other data transfer methods such as secure copy (SCP).

We report here on work aimed at making GridFTP deployment trivial, so that GridFTP transfers can be achieved "instantly" even by nonexperts. This work seeks in particular to automate the process of configuring the public key infrastructure (PKI)-based Grid Security Infrastructure (GSI) [3] that GridFTP uses to secure data transfers. In conventional GridFTP, users are required to obtain a PKI credential from a certificate authority (CA) and then to manage that credential themselves, while the GridFTP system administrator is required to manage mappings from PKI credentials to local user ids. In what we call Globus Connect Multi User (GCMU: a multiuser version of the Globus Connect software previously developed for use with Globus Online), a MyProxy Online CA packaged with a GridFTP server handles the generation and management of user PKI credentials and the mapping between those credentials and local user ids. Thus, we reduce burden on both users and administrators and eliminate frequent sources of errors in GridFTP configuration and use.

The rest of the paper is as follows. In Section II, we provide background on Globus GridFTP architecture and security handling. In Section III, we discuss the GridFTP installation process and prior work on reducing its complexity. We present GCMU in Section IV. In Section V, we describe the new command added to the GridFTP protocol to support data channel authentication in GCMU. In Section VI, we showcase how GCMU is used in Globus Online. We present related work in Section VII. We summarize our work and discuss future plans in Section VIII.

II. BACKGROUND

We discuss here the use of Globus GridFTP as well as its architecture and the way in which it handles security.

A. Globus GridFTP Background

The Globus GridFTP implementation [4] has become the preeminent high-performance data transfer tool for the Grid community. The Globus GridFTP server is deployed on more than 5,000 servers worldwide and is responsible for an average of more than 10 million transfers totaling approximately half a petabyte of data every day (see Figure 1; these numbers are based on reporting from GridFTP servers that choose to enable reporting, presumably a subset of all servers). Its modular architecture enables a standard GridFTP-compliant client access to any storage system that can implement its data storage interface [5], including the HPSS archival storage system [6] and POSIX-compliant [7] file systems. Its eXtensible I/O interface [8] allows GridFTP to target high-performance wide-area communication protocols such as UDT [9] and emerging RDMA-based protocols [10]. Globus GridFTP is optimized to handle various types of datasets—from a single, huge file to datasets comprising lots of small files [11, 12].

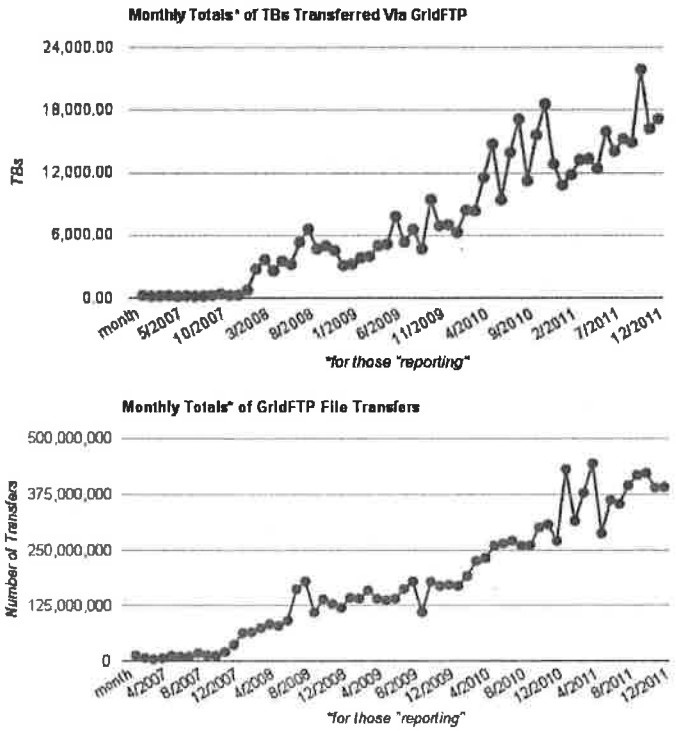

Figure 1: Globus GridFTP Usage Data

Many science communities rely on Globus GridFTP for their production operation. U.S. Department of Energy leadership computing facilities at Argonne and Oak Ridge and the supercomputing facility at Berkeley and the National Science Foundation's advanced cyberinfrastructure XSEDE [13] run dedicated GridFTP servers ("data transfer nodes") to support high-performance, wide-area data movement for their users. The tiered data movement infrastructure for the Large Hadron Collider computing grid [14] is based on GridFTP, as is the Earth System Grid [15], which serves data to the global climate community. The National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, Relativistic Heavy Ion Collider at Brookhaven, Advanced Photon Source at Argonne, and Spallation Neutron Source at Oak Ridge use GridFTP for bulk data movement.

B. Globus GridFTP Architecture

Globus GridFTP comprises three logically distinct components: client and server protocol interpreters (PIs), which handle the control channel protocol (these two functions are distinct because the protocol exchange is asymmetric), and the data transfer process (DTP), which handles access to the actual data and its movement via the data channel protocol. These components can be combined in various ways to create servers with different capabilities. For example, combining the server PI and DTP components in one process creates a conventional FTP server, while a striped server might use one server PI on the head node of a cluster and a DTP on all other nodes.

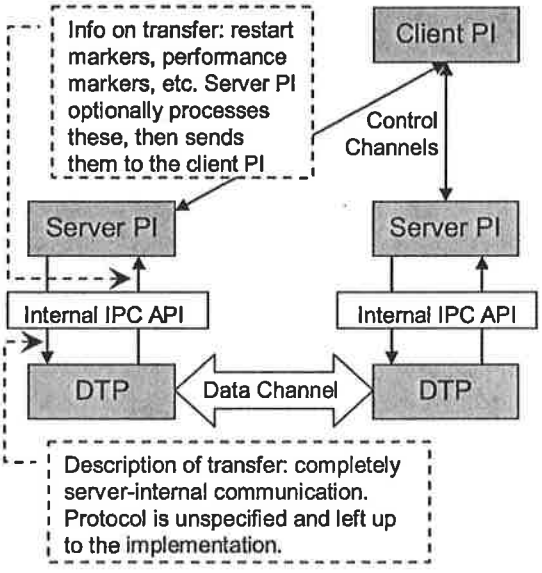

Figure 2: Globus GridFTP Architecture

C. Security in Globus GridFTP

The Globus GridFTP design provides for secure authentication of control channel requests (obligatory) and for data channel integrity and confidentiality

(optional). A session is established when the client initiates a TCP connection to the port on which the server is listening. First, authentication must be done per RFC 2228 [16]. By default, the client presents a delegated proxy certificate [17], and the server must present a certificate issued by a CA trusted by the client. If authentication is not successful, the connection is dropped.

If authentication is successful, an authorization callout is invoked to verify authorization and determine the local user id for which the request should be executed. This callout is linked dynamically; the server does a "setuid" to the local user id as determined by the authorization callout. If authorization succeeds, the control channel is established, and the rest of the control channel protocol exchange can proceed. The control channel is encrypted and integrity protected by default.

In order to establish the data channel (the connection over which the actual data of interest will flow), a listening port must be established and the other end informed of this port. The GridFTP protocol requires that the receiver be the listener and that the sender issue the TCP connect. Thus, the client sends a PASV command to the server that is to receive the data. The receiver begins listening on a TCP port and responds to the command indicating the IP address and port of the listener. (If this is a striped transfer, the client sends a striped PASV, or SPAS, command and an array of IP/ports is returned.) The client then sends to the other server a PORT (or SPOR, for striped port) command, which takes the IP/ports as a parameter. This command directs the server to initiate the TCP connect and establish the data channel.

Third-party transfer (client initiating a transfer between two remote GridFTP servers) presents a security issue because the receiving server starts listening on a port but has no way of knowing the IP address of the server that will connect to it. To mitigate this issue, GridFTP defaults to requiring GSI authentication on the data channel as well. In this case, the server performs a delegation, and both ends of the authentication must present the user's proxy certificate. A limitation of current GridFTP protocol implementations is that all parties involved in the transfer must accept the same CA.

Both cryptographic confidentiality and integrity protection are supported on the data channel but are not enabled by default because of cost. (An order of magnitude slowdown is not unusual on high-speed links.)

III. Globus GridFTP Installation

We next discuss the GridFTP installation process and prior work on reducing its complexity.

A. Installation Process

The installation and configuration of a GridFTP server are currently a multistep process:

1. *Installation* involves four steps: (a) download Globus; (b) untar the Globus tar file; (c) run "configure"; and (d) run "make" and "make install." These steps are not too difficult but can be time consuming, and require the target machine to have development tools installed.
2. *Configuring security* involves four further, potentially multistep and time-consuming tasks: (e) obtain an X.509 host certificate from a well-known certificate authority; (f) install the X.509 host certificate; (g) configure the trusted certificates directory with the certificates of CAs that you want to trust; and (h) set up authorization, that is, generate mappings between users' Grid identities (distinguished name in their certificate) to a local user account.
3. Further work is involved to *configure security for each user*, if this has not already been done. This work involves obtaining an X.509 user certificate from a well-known certificate authority, installing that certificate, configuring the trusted certificates directory with the certificates of CAs that you want to trust, and sending the distinguished name in the new certificate to your server admin so that he can map it to your local user account on the server.

This process is too complex for many users and for smaller laboratories and projects, requiring a degree of systems administration expertise that is not required when using tools such as SCP.

B. Prior Work

Two pieces of prior work partially address the usability issues in GridFTP.

1. GridFTP-Lite

GridFTP-Lite uses SSH for user authentication. Specifically, it uses SSH to dynamically start a GridFTP server on a target machine and then uses that SSH session to tunnel the GridFTP control channel.

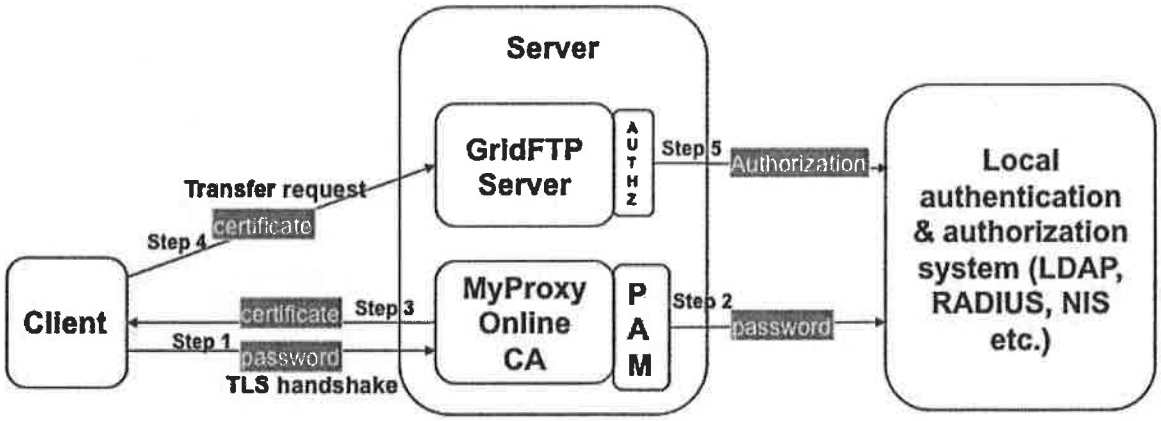

Figure 3: GCMU Workflow

This technology avoids the need for administrators and users to set up the normal X.509 Grid Security Infrastructure (steps 2–3 above). But the approach has three major limitations. First, the data channel has no security. (GridFTP protocol extensions proposed here will help address this issue for non-striped transfers.) Second, since SSH does not support delegation, users cannot hand off SSH-based GridFTP transfers to transfer agents such as Globus Online [18, 19]. Third, no security exists on the communication channel between the control node and the data mover node in the striped GridFTP server.

2. Native Packaging

We have created source and binary packages of GridFTP using Red Hat Linux RPM and Debian Linux DEB are in the process of creating packages for other operating systems Mac OS X. (For Windows, see below.) This work addresses the installation problem (step 1) but not the security configuration problems (steps 2–3).

IV. GLOBUS CONNECT MULTI USER

We aim to make GridFTP trivial to setup, configure, and use. To achieve that objective, we need to do the following:

- Make it easy for system administrators to install, configure, and set up a GridFTP server that interfaces to their local file system and security domain.
- Make it easy (e.g., as simple as SCP and GridFTP-Lite) for end users to use GridFTP.

Our solution, Globus Connect Multi User, is—as the name suggests—a multiuser version of the one-click install Globus Connect client that we developed previously for use by Globus Online.

Figure 3 shows the GCMU architecture and the associated workflow. GCMU combines a GridFTP server, a MyProxy Online Certificate Authority (CA) [20] server, and a custom authorization callout for GridFTP ("AUTHZ"). The user accesses the MyProxy Online CA on the server machine by providing his username and password for the server (step 1 in the figure). MyProxy Online CA in turn passes the username and password to the local authentication system such as LDAP [21], RADIUS [22], or NIS [23] via a Pluggable Authentication Module (PAM) [24] API to authenticate the user (step 2). If the user is authenticated successfully, the MyProxy Online CA issues a short-lived X.509 certificate to the user (step 3). It embeds the local username in the distinguished name (DN) of the certificate, since this certificate will be used to authenticate with this site only. The user then authenticates to the GridFTP server using this certificate (step 4). Once the authentication is successful and upon knowing that local MyProxy Online CA issued the certificate, AUTHZ is invoked to parse the username from the DN. There is no need to maintain an explicit DN to username mapping (step (h) in Section IIIA). Once the username with which the requested should be executed is determined, the authorization is enforced by the local authorization system (step 5).

Configuring security is the most complex process in GridFTP setup. Obtaining an X.509 certificate from a well-known certificate authority alone is a complex and time-consuming process. It sometimes requires generating a key pair and certificate signing request using OpenSSL [25] and/or exporting the certificate

and key from a browser and then using OpenSSL tools to translate the certificate into a different format [26]. The process also involves out-of-band vetting to ensure that the user is really associated with the organization he claims to be associated with. This is just one step in configuring the security (step (e) in Section IIIA).

A. Myproxy Online CA

One approach to simplifying the management of certificates is to not require the use of a single, common security credential across all endpoints involved in a transfer. Users have many identities for use with many different service providers. Data transfer tools should seamlessly handle transfers across multiple security domains with multiple user identities. One can use tools such as PAM to embed short-term certificates in the existing authentication processes.

MyProxy Online CA adopts this approach. It can be run at a site and tied to the local identity domain via a PAM. It issues short-lived X.509 credentials to authenticated users, which can then be used to authenticate with the GridFTP server. To obtain credentials, users run client software on their local host. The software generates the subscriber's private key locally, authenticates the user to the site's MyProxy Online CA using the user's credentials for the site (username/password, OTP, etc.), and issues a signed certificate request to the CA. If the request is approved, a signed certificate is received from the CA.

GCMU uses MyProxy Online CA to simplify security configuration. As illustrated in Figure 3, GCMU users can access the GridFTP server using their normal site username and password. This is an enormous advance over the complex certificate acquisition and installation process required previously.

Arguably, however, Myproxy Online CA alone does not solve all usability issues, as we now discuss.

B. Data Channel Authentication

The GridFTP protocol [2] defines data channel authentication (DCAU) for a third-party transfer to involve mutual validation of the X.509 credentials that the requesting user provides to the two endpoints involved in the transfer.

Thus, if endpoint A requires that the user supply a credential A (e.g., an X.509 proxy certificate issued by CA-A) and endpoint B requires that the user supply a credential B (e.g., an X.509 proxy certificate issued by CA-B), then endpoint A must, as part of data channel setup, receive and validate credential B, and endpoint B must receive and validate credential A. However, this process fails if CA-A is unknown to endpoint B, or vice versa; see Figure 4.

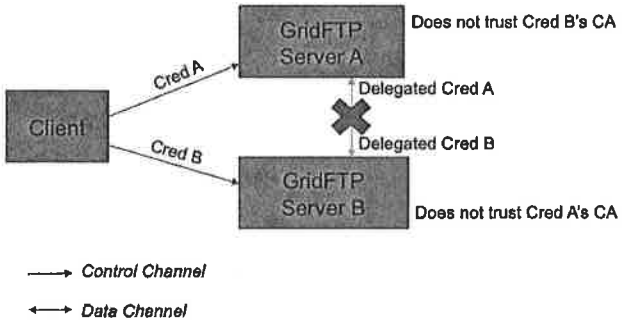

Figure 4: Data Channel Authentication Problem

To address this limitation, we introduced a new Data Channel Security Context (DCSC) command in GridFTP. A GridFTP client can use this command to tell a DCSC-enabled GridFTP endpoint to both accept and present to the other endpoint a credential different from that used to authenticate the control channel. For example (see Figure 5), it can use DCSC to pass credential A to site B, for subsequent presentation to site A. Note that this works even if one endpoint is a legacy GridFTP server that knows nothing about DCSC. We plan to introduce the DCSC command to the GridFTP standardization process through the definition of a GridFTP v3 protocol specification.

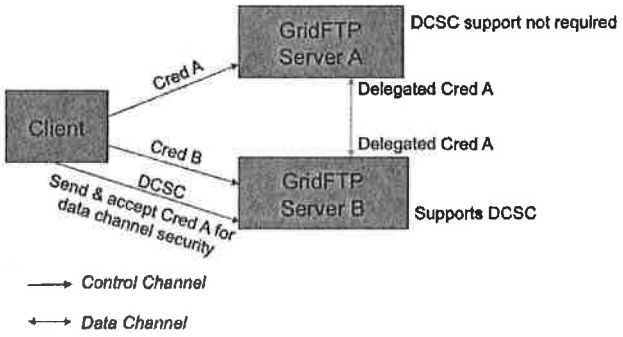

Figure 5: Solving the Data Channel Authentication Problem Using DCSC

C. Certificate Subject to User ID Mapping

As mentioned in Section IIC, once the user authenticates with the GridFTP server successfully, the server has to determine the local user id as which the request should be executed. This mapping is typically done by looking at a Gridmap file—a list of

certificates subject to user id mapping—maintained by the server administrator. This file is, however, a frequent source of errors and complaints, because of the difficulties inherent in keeping it up to date.

In GCMU, we eliminate the need for a Gridmap file; instead, user certificates are issued by the local MyProxy Online CA. We configure the MyProxy Online CA to include the local username in the certificate's subject. In addition, we have developed a custom authorization callout in GridFTP that picks up the local user id from the certificate subject if the certificate is signed by the local MyProxy Online CA.

D. Server Installation

We package and configure the GridFTP and MyProxy servers in a manner that makes GCMU server installation trivial. On the server machine, the following four commands are required to download the tarball, untar, and run the install script to get the GridFTP server and MyProxy CA running.

```
wget https://s3.amazonaws.com/connect.g
  lobusonline.org/linux/stable/globusco
  nnect-multiuser-latest.tgz
tar -xvzf globusconnect-multiuser-
  latest.tgz
cd gcmu*
sudo ./install
```

E. Client Installation

A client wanting to install and use GridFTP follows a similar process to a server to download and install the GCMU code, then interacts with the server to obtain a certificate, and can then perform file transfers. (As we describe in Section VI, a client also has the option of using Globus Online to transfer files, in which case the even simpler Globus Connect software can be installed and used.)

First, the client downloads the tarball and runs the install script to get the client tools installed.

```
wget https://s3.amazonaws.com/connect.g
  lobusonline.org/linux/stable/globusco
  nnect-multiuser-latest.tgz
tar -xvzf globusconnect-multiuser-
  latest.tgz
cd gcmu*
sudo ./install-client
```

Second, the client runs a command to get a short-term credential from the MyProxy CA on the server. This credential is used to authenticate with the GridFTP server when moving data:

```
myproxy-logon -b -T -s <server-name>
Supply user's username/password for
server
```

Data can now be transferred to and from the GCMU server (and any other GridFTP server) using a client such as globus-url-copy:

```
globus-url-copy
  gsiftp://<server>/<path> file:/<path>
```

V. Data Channel Security Context Command

The current DCAU protocol uses an SSL context that contains the logged-in user's credential. If two servers have different user credentials and do not have each other's CA certificates, a client cannot perform a secure third-party transfer between them. If one of the servers supports DCSC, a client can tell it to both send and accept the user credential used by the other server, thus enabling DCAU where it previously was not possible. If both servers support DCSC, clients that desire higher security may specify a random, self-signed certificate as the DCAU context.

The general format for DCSC is

DCSC context-type context-specific-blob,

where context-type is a case-insensitive string and the blob is a string composed of only printable ASCII (32–126) characters, such as base64 encoding would produce. Context type can be "P" or "D."

A. DCSC P

A "P" context type (short for proxy/PEM) message is

DCSC P base64-encoded-blob.

The base64-encoded blob comprises three parts.

1. An X.509 certificate in PEM format
2. A private key in PEM format
3. Additional X.509 certificates in PEM format, unordered (optional)

A DCSC "P" command will overwrite any previous request.

The certificate in (1) must be self-signed or verifiable by using only intermediate and/or CA certificates in (3). If the certificate in (1) is not self-signed, clients must send its full certificate chain, including the CA certificate, in (3).

A server validates the remote party's certificate using a combination of the following.

- The server's default CA certificates and signing policies
- All self-signed certificates given in (1) and (3)

Servers do not require signing policy files for any CA certificates in (3). If signing policies do exist for any CA certificates in (3), the server will still use and enforce them. The DCSC command does not provide a way to specify signing policies; the server's default CA certificates are expected to already be protected by signing policies.

B. DCSC D

The "D" context type stands for "default context." The command "DCSC D" will revert the context to whatever it was immediately after login.

VI. Use Case: Globus Online

We next explain how GCMU is used in the context of Globus Online.

A. Globus Online

Globus Online [18, 19] is a software-as-a-service (SaaS) client for GridFTP. The Globus team operates this hosted service as a third-party mediator/facilitator of file transfers between GridFTP servers. This robust, reliable, secure, and highly monitored file transfer environment has powerful yet easy-to-use Web 2.0 interfaces.

The SaaS approach allows Globus Online to exploit economies of scale, since a single hosted service serves many individuals and communities. With SaaS, new features are immediately available to all users, and service operators can intervene and troubleshoot on the user's behalf to deal with more complex faults.

Globus Online can be accessed via different interfaces depending on the user and the application. A simple web GUI serves the needs of ad hoc and less technical users. A command line interface via SSH exposes more advanced capabilities and enables scripting for use in automated workflows. A REST API facilitates integration for system builders who do not want to re-engineer file transfer solutions for their end users.

Globus Online also has the ability to automatically tune GridFTP transfer options for high performance.

Globus Online has made the data movement task simpler for end users by eliminating the need to monitor transfers. Since Globus Online requires GridFTP servers at both ends of the transfer, GCMU makes it easier for small and medium compute resources to be part of the Globus Online ecosystem.

B. GCMU Usage in Globus Online

Figure 6 shows how a user interacts with GCMU via Globus Online. GCMU has an option in the installation to make the server available as an endpoint on Globus Online, meaning that the GCMU GridFTP server is visible to Globus Online users. Let us say a cluster administration installs GCMU on a cluster and makes it available as an endpoint on Globus Online. Now the user can login to Globus Online and access this endpoint. The user then will be prompted for a username and password. Once the user provides this information, Globus Online passes it to GCMU running on the cluster. Globus Online does not store the password. GCMU then interacts with the local authentication system to authenticate the user. If the user is authenticated successfully, GCMU returns a short-term certificate to Globus Online that it can use to authenticate with the GridFTP server running on the cluster and access the data. A user can follow the same procedure to access the other endpoint for this transfer on Globus Online and submit a transfer request. If any failure occurs during the transfer, Globus Online will use the short-term certificate to reauthenticate with the endpoints on the user's behalf and restart the transfer from the last checkpoint.

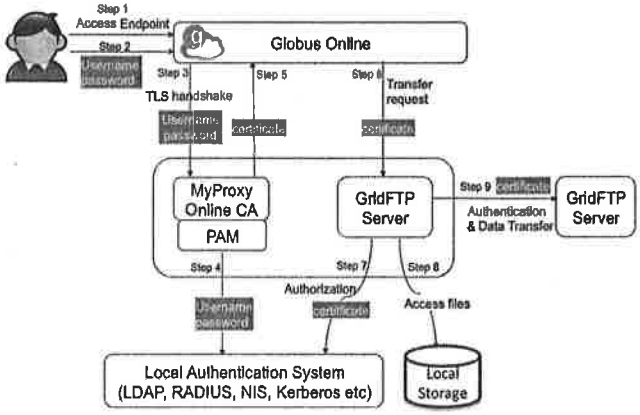

Figure 6: Globus Online / GCMU Interaction

To mitigate the security concerns associated with passing the username/password through a third-party site, OAuth server [27] can be used in conjunction with GCMU. Figure 7 shows the end-to-end workflow in the presence of an OAuth server. With an OAuth server on GCMU endpoint and the OAuth support that already exists in Globus Online, users do not have to enter a username or password on Globus Online. Instead, when users access a GCMU

endpoint, they will be redirected to a web page running on the endpoint; when they enter the username/password on that site, Globus Online will get a short-term certificate from the endpoint via the OAuth protocol.

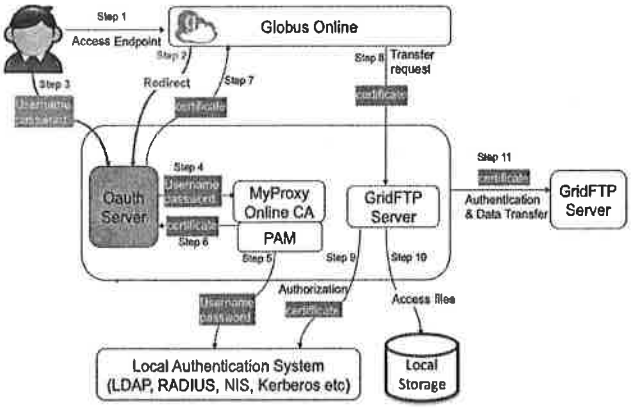

Figure 7: Globus Online / GCMU Interaction in the Presence of a MyProxy OAuth Server

VII. Related work

Tools such as SCP and rsync are ubiquitously available and easy to use, but they provide only modest performance and no fault recovery. Legacy FTP, SFTP, and HTTP also suffer from low performance. Although these tools can be used to move small quantities of data, larger datasets can take hours or even days to transfer with these tools and can require frequent user intervention.

Scientists often need to stage data from a remote data store to a remote compute cluster by issuing commands from their local machine; thus, third-party transfers are frequently involved in science data movement. HTTP and rsync do not support third-party transfers. SCP routes data through the client for transfers between two remote hosts; but often, the two remote hosts are connected by a high-speed link whereas the client and remote hosts are connected by low-bandwidth links.

GridFTP provides significantly higher performance compared with these tools, through various optimizations such as parallelism [4], pipelining [11], concurrency [12], and striping [4]. It also supports third-party transfers in which a user or application at one site can initiate a data transfer operation between two other sites and the data flows directly between two remote sites.

Some high energy physics experiments use Xrootd [28], a system that does storage aggregation and provides POSIX-like file access to data. Distributed file systems [29, 30] provide access to remote data while maintaining file system semantics. GridFTP is intended to be used in less tightly coupled environments in which storage aggregation and file system semantics are not feasible.

VIII. Summary and Future work

We have presented GCMU, which allows users and administrators to quickly and easily set up and use GridFTP for secure, reliable, high-performance data movement. This work eliminates the time-consuming and complex steps otherwise associated with running and using a secure GridFTP server—complexities that have previously hindered the use of GridFTP by nonexperts and by smaller research facilities and projects with limited system administration resources.

An important feature of GCMU is that it does not require the use of a common security credential across all endpoints. Users can use a certificate issued by one CA to authenticate with a GridFTP server at one site and a certificate issued by another CA to authenticate with the GridFTP server at another site and then perform a secure third-party transfer between the two sites without either site needing to have the other CA in its trust roots. This feature is particularly important when GCMU is used via Globus Online, since all the transfers done by Globus Online are third-party transfers.

An OAuth server can be installed in conjunction with GCMU so that the user's credential for an endpoint does not have to flow through third-party agents like Globus Online. (Globus Online already provides support for OAuth.)

In future work, we plan to package an OAuth server in GCMU so that this feature (no need to enter user's credential on third party agents) is available automatically. We will also create a "virtual appliance" consisting of a virtual machine image that includes GCMU and a simple web-based (and command line) administrative console for configuring the virtual appliance.

Acknowledgment

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

References

[1] Postel, J. and Reynolds, J. File Transfer Protocol. Internet Engineering Task Force, RFC 959, 1985.
[2] Allcock, W. GridFTP: Protocol Extensions to FTP for the Grid. Global Grid ForumGFD-R-P.020, 2003.
[3] Foster, I. Kesselman, C. Tsudik, G. and Tuecke, S. A Security Architecture for Computational Grids. 5th ACM Conference on Computer and Communications Security Conference, 1998, pp. 83-92.
[4] Allcock, W. Bresnahan, J. Kettimuthu, R. Link, M. Dumitrescu, C. Raicu, I. and Foster, I. The Globus Striped GridFTP Framework and Server. SC'05, ACM Press, 2005
[5] Kettimuthu, R. Link, M. Bresnahan, J. and Allcock, W. Globus Data Storage Interface (DSI) - Enabling Easy Access to Grid Datasets. 1st DIALOGUE Workshop: Applications-Driven Issues in Data Grids, Aug. 2005.
[6] Watson, R.W. and Coyne, R.A. The Parallel I/O Architecture of the High-Performance Storage System (HPSS). IEEE MSS Symposium, 1995.
[7] POSIX 1003.1e draft specification http://www.suse.de/~agruen/acl/posix/posix_1003.1e-990310.pdf.
[8] Allcock, W. Bresnahan, J. Kettimuthu, R. and Link, J. The Globus eXtensible Input/Output System (XIO): A Protocol-Independent I/O System for the Grid. Joint Workshop on High-Performance Grid Computing and High-Level Parallel Programming Models held in conjunction with International Parallel and Distributed Processing Symposium, 2005.
[9] Gu, Y. and Grossman, R.L., UDT: An Application Level Transport Protocol for Grid Computing. Second International Workshop on Protocols for Fast Long-Distance Networks, 2003.
[10] Subramoni, H. Lai, P. Kettimuthu, R. and Panda, D. K. High Performance Data Transfer in Grid Environment Using GridFTP over InfiniBand. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGRID '10), 2010. IEEE Computer Society, Washington, DC, 557-564. DOI=10.1109/CCGRID.2010.115.
[11] Bresnahan, J. Link, M. Kettimuthu, R. Fraser, D. and Foster, I. GridFTP Pipelining. 2007 TeraGrid Conference, Madison, WI, 2007.
[12] Kettimuthu, R. et al. Lessons Learned from Moving Earth System Grid Data Sets over a 20 Gbps Wide-Area Network. 19th ACM International Symposium on High Performance Distributed Computing (HPDC), 2010.
[13] https://www.xsede.org/
[14] Lamanna, M. The LHC computing grid project at CERN. Nuclear Instruments and Methods in Physics Research, 534:1–6, 2004.
[15] Middleton, D. E. et al Enabling Worldwide Access to Climate Simulation Data: the Earth System Grid (ESG). Scientific Discovery through Advanced Computing. J Phys 46(2006):510–514
[16] Horowitz, M. and Lunt, S. FTP Security Extensions. Internet Engineering Task Force, RFC 2228, 1997.
[17] Tuecke, S. Welch, V. Engert, D. Pearlman, L. and Thompson, M. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. IETF, RFC 3820, 2004
[18] Foster, I. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. IEEE Internet Computing, pp. 70–73, 2011.
[19] Allen, B. et al Software as a Service for Data Scientists. Communications of the ACM 55, pp. 81–88, 2012.
[20] Novotny, J. et al An Online Credential Repository for the Grid: MyProxy. 10th IEEE International Symposium on High Performance Distributed Computing, San Francisco, 2001.
[21] Koutsonikola, V. and Vakali, A. LDAP: Framework, Practices, and Trends. IEEE Internet Computing 8, 5 September 2004, 66-72. DOI=10.1109/MIC.2004.44 http://dx.doi.org/10.1109/MIC.2004.44
[22] Rigney, C. Willens, S. Rubens, A. and Simpson, W. Remote Authentication Dial In User Service (RADIUS). Internet Engineering Task Force, RFC 2865, 2000.
[23] Stern, H. Managing NFS and NIS (2nd ed.). Edited by Mike Loukides. O'Reilly & Associates, 2001. Sebastopol, CA.
[24] Samar, V. and Schemers, R. Unified Login with Pluggable Authentication Modules (PAM). OSF RFC 86.0, October 1995.
[25] http://www.openssl.org/
[26] http://www.doegrids.org/pages/cert-request.html
[27] Basney, J. and Gaynor, J. An OAuth Service for Issuing Certificates to Science Gateways for TeraGrid Users. TeraGrid Conference, July 18–21, 2011, Salt Lake City, UT. http://dx.doi.org/10.1145/2016741.2016776.
[28] http://xrootd.slac.stanford.edu/
[29] Howard, J.H. Kazar, M.L. Menees, S.G. Nichols, D.A. Satyanarayanan, M. Sidebotham, R.N. and West, M.J. Scale and Performance in a Distributed File System. ACM Transactions on Computer Systems, 6 (1). 51-81. 1988.
[30] Popek, G.J. Guy, R.G. Thomas, W. Page, J. and Heidemann, J.S. Replication in Ficus Distributed File Systems. Workshop on Management of Replicated Data, 1990, IEEE, 20-25.