Integration of Globus Online with the ATLAS PanDA Workload Management System

C.Contreras¹, W.Deng², T.Maeno², P.Nilsson³, M.Potekhin²
for the ATLAS Collaboration

¹Departamento de Fisica-Univ. Tecnica Federico Santa Maria, Chile
²Brookhaven National Laboratory, Upton, NY 11973, USA
³University of Texas at Arlington, TX 76019, USA

E-mail: potekhin@bnl.gov

Abstract. The PanDA Workload Management System is the basis for distributed production and analysis for the ATLAS experiment at the LHC. In this role, it relies on sophisticated dynamic data movement facilities developed in ATLAS. In certain scenarios, such as small research teams in ATLAS Tier-3 sites and non-ATLAS Virtual Organizations, the overhead of installation and operation of these components makes their use not very cost effective. Globus Online is an emerging new tool from the Globus Alliance, which already proved popular within the research community. It provides the users with fast and robust file transfer capabilities that can also be managed from a Web interface, and in addition to grid sites, can have individual workstations and laptops serving as data transmission endpoints. We will describe the integration of the Globus Online functionality into the PanDA suite of software, in order to give more flexibility in choosing the method of data transfer to ATLAS Tier-3 and OSG users.

1. Introduction
PanDA is a Workload Management System utilized in the ATLAS experiment on the LHC [1]. It is built around the concept of Pilot Frameworks. In this approach, workload is assigned to successfully activated and validated Pilot Jobs, which are processes that probe the environment and act as a ‘smart wrapper’ for the payload. PanDA is using a sophisticated system called “DDM” (for Dynamic Data Management) to manage reliable transfer and staging of input and output data. PanDA also implements optimal brokering strategies based on data specific location when distributing jobs across sites [2, 3].

With all its advantages, using DDM incurs certain deployment and operational costs. While these are acceptable for large Grid sites and research organizations, the system is less suited for ad-hoc analysis streams implemented by smaller research teams, e.g. in ATLAS Tier-3 or in life sciences projects.

2. Globus Online
Globus Online is a system that provides robust file transfer capabilities to its users without the need for expensive hardware or extensive software installation. Globus Online manages many aspects of file transfer, monitoring performance, retrying failures, auto-tuning and recovering from faults automatically where possible, and reporting status [4]. It provides remarkable flexibility in establishing sources and destinations for data transfer, whether at facility level or an individual workstation or laptop.

The main component of the Globus Online system is a Web Service that automates data movement across sites and/or systems, called “endpoints” in this context. It is equipped with a user-friendly Web interface that allows the users to initiate and manage data transfer and access logs. In addition, it provides facilities for CLI and REST APIs. The former has syntax similar to scp and can be scripted for automation purposes if needed. The latter presents a foundation for integration of Globus Online into workflow distributed over Grid sites or in Cloud Computing environment. This “RESTful” interface has Python bindings, which makes it especially well suited for use in the PanDA framework, which is itself Python-based.

The actual data transfer layer is currently implemented using the GridFTP protocol. The security level is provided by GSI mechanism using X.509 certificates.

To initiate Globus Online data transfers, the user must first register the endpoints that will be used (such as GridFTP servers with their specific URIs). In this process, the GridFTP server acquires a user-defined symbolic name which can be conveniently used in the CLI or any other interface without the need to spell the full URI in each individual case. Once the user posts a request for data transfer, the Web Service interacts with the chosen endpoints in order to manage data transfer and error handling. In this stage, there is no user involvement as Globus Online will attempt to optimize transfer and will handle retries if necessary. The progress of data transfer can be monitored on the Globus Online web site.

3. Globus Connect
Globus Connect is a feature that enables users and developers to use their personal computing resources (such as laptops or workstations) or individual worker nodes (in the Grid context) as endpoints for Globus Online data transfer. Globus Connect is deployed by the user as a lightweight client that is available for download from the Globus Online servers, and exists in versions for Windows, Mac OS X and Linux. Once installed, the Globus Connect client must be registered with Globus Online Web service by requesting an authentication token.

4. General use cases for Globus Online in PanDA framework
As mentioned above, payload jobs are actuated in PanDA from within the wrapper/agent script known as the Pilot. One of its functions is interfacing the data transfer mechanisms chosen for the job. Discussion of the ATLAS-specific data movement systems is beyond the scope of this discussion, so we shall concentrate on Globus Online as it is applied in the PanDA Pilot framework.

First, let us consider a generic case where the payload has no dependency on ATLAS software or data, such as when the user is involved in research in a different discipline. This is one possible scenario:

- The input data is located on the user’s desktop
- A remote Grid site managed by PanDA will be used for computation
- The output data must arrive back to the user’s desktop

In this case, the following needs to happen:

- The user activates a Globus Connect endpoint on their desktop
- The user defines the GridFTP server present at the processing site as another endpoint via the Globus Online web interface (alternatively, such endpoint is initialized and declared public by a separate process)
Jobs are submitted to PanDA via a command line utility script, which accepts a set of parameters related to the job configuration, destination and location of the input and output data, i.e. the names of endpoints and the paths.

In the above, there are other possibilities for placing the input and output data, such as on GridFTP servers in locations different from both the user and the processing site. General logic and sequence of events remains roughly the same.

In case of ATLAS jobs where utilization of Globus Online is desirable, the situation is more complicated because of the architecture of distributed data storage and cataloging used in ATLAS. One example is grouping of files into “datasets” which are used as tags to define potentially large file collections for purposes of accounting and processing. Currently there is no way to catalog data in Globus Online that would be consistent with what’s used in ATLAS and simultaneously maintain referential integrity. Use cases, therefore, will be more restricted in scope, but useful nevertheless. For example:

- The input data is located on a storage element on an ATLAS-managed site, and is properly accounted for in the file catalog
- A remote Grid site managed by PanDA will be used for computation
- In addition to the storage element normally used in this particular analysis workflow, the output data must arrive to an additional location outside of DDM sphere of management, such as the user’s desktop or a private GridFTP server run by the research team.

5. Role of X.509 credentials
Client software used by Globus Online requires the use of X.509 credentials, such as the end-user’s proxy, in order to get authenticated to the Web Service and be authorized to initiate data transfer. It is necessary therefore for the Pilot process running on the remote Grid site to obtain the user’s proxy at runtime. We opted to utilize an instance of the MyProxy service [5] deployed at CERN in order to securely cache and retrieve the users’ X.509 proxies. When caching the proxy, an option is used to authorize its later retrieval by a trusted agent, in this case the PanDA Pilot, whose identity is recorded in an encrypted form in the DN of its own proxy, which is used to authenticate and get authorized by MyProxy.

6. Summary of software enhancements in the PanDA framework
Payload jobs for generic (non-ATLAS) users are typically run in PanDA using a simplified and lightweight version of the Pilot, whereas the ATLAS jobs are managed by more sophisticated ATLAS Pilot code. Likewise, jobs are submitted using a simplified command-line utility for generic users, and a more specialized tool for ATLAS jobs. We made enhancements in both sets of software to enable the functionality to support use cases presented above. The Python binding of the Globus Online REST interface was used to add requisite functionality to both versions of the Pilot.

7. Results of testing
We completed the initial round of testing with software enhanced as described above, which utilizes the Globus Online mechanism to manage data transfer in one or more stages of the job’s life cycle. Actual ATLAS analysis jobs were successfully run in PanDA resulting in Globus Online transmission of the output data to the endpoint defined by the user (a remote GridFTP site or the user’s desktop).

8. Conclusions
We identified Globus Online Web Service as a useful tool to enhance and expand functionality of the PanDA Workload Management system in two different ways:

- By providing additional flexibility to ATLAS users when running ad-hoc analysis workflows
- By enabling users external to ATLAS to manage input and output of their jobs being run on PanDA in a robust and convenient manner.
The *MyProxy* credential-caching service was utilized to provide PanDA Pilot jobs with correct credentials in order to authenticate to Globus Online and manipulate user-defined data endpoints. Initial testing done with realistic ATLAS analysis jobs was successful. Work is currently underway to expand testing to new Grid sites and to roll out these tools to end users.

**References**


