Motivations for the integration of globusonline.org with SRM

Gabriele Garzoglio, Tanya Levshina, Dmitry Litvintsev
May 31, 2011

Abstract

This document presents the use cases for the access of SRM storage through globusonline.org and discusses the scientific communities in OSG that might use the system.

Introduction

Every day the Open Science Grid makes available thousands of computing nodes and TeraBytes of storage for opportunistic usage. Many scientific communities associated with OSG, also called Virtual Organizations (VO), take advantage of these resources to advance their scientific program.

The OSG User Support group helps new and existing communities to use opportunistic resources. For new communities in particular, the group partners with “power” users to define appropriate processes and share best practices for the use of the distributed high-throughput computing paradigm of OSG.

In this paradigm, the availability of abundant computing resources often allows the production of data at an unprecedented scale for these new communities. Handling this volume of data is one of the most challenging tasks for new OSG users. For this reason, the OSG user support group is active in identifying new technologies and processes to make the task of handling data easier for its end users. The globusonline.org (GO) service is a promising technology in this regard. Considering that the OSG users are not necessarily experts in Grid technologies, we find particularly attractive the globusonline.org web interface to select and transfer data and the single binary (globusconnect) to instantiate a GridFTP server at the user PC.

To date, however, the only storage interface supported by globusonline.org is GridFTP. In OSG, the *de facto* standard interface to access storage is the Storage Resource Manager (SRM). SRM provides the means for a site to define local policies for network shaping and resource balance. SRM then delegates the function of data transport to underlying deployments of GridFTP servers.

In this document we present various use cases to use SRM through the globusonline.org service. We also speculate on the potential use of this technology by the OSG communities.

Use Cases

We present step-by-step scenarios in the use of SRM via globusonline.org. Scenarios are roughly ordered from the most urgent and immediately applicable.

1. User retrieves output from an SRM storage (SRM output retrieval case)

1. The user logs to an OSG submission node. She prepares local input data, executable, necessary tools, job description files, etc. User submits jobs to the Workload Management System (Glidein WMS, Condor-G, Panda, …)
2. The WMS dispatches the jobs and their input to OSG worker nodes.
3. The application runs and produces output in a scratch disk local to the worker node.
4. The application uses SRM tools to store the output to a pre-defined SRM storage. The SRM tools are typically pre-installed at the worker node.
5. The WMS successfully terminates the jobs.
6. Periodically, the user inspects the status of her jobs and wants to retrieve the output to a local PC for further studies.
7. The user instantiates a GridFTP server on her PC using globusconnet. She “activates” all SRM endpoints that received the output. She interacts with the GO web interface to select the available output. She initiates the transfer via the web interface, ideally deciding whether to have globusonline.org erase the data from SRM after the transfer.
8. GO interacts with the SRM server to get a GridFTP endpoint. It initiate a 3rd-party data transfer and verifies its integrity.
9. The user has successfully retrieved the data to her PC.

2. User pre-stage data from home institution to target SRM storage (LIGO case)

1. The user logs to a submission node, prepares a list of files that should be transferred via GO to a target SRM storage. The transfer volume varies from 90 GB to 1TB. The source endpoint deploys a GridFTP server.
2. The user may want to compare the checksum of the upload files with the ones at the source.
3. Once the transfer is done, the user submits jobs to analyze the data. The job access the data via one of the mechanisms above or other mechanisms (SRM transfer, shared file system, etc).

3. Job asynchronously stores output to an SRM storage (CEDPS case)

1. The user logs to an OSG Glidein WMS submission node. She prepares local input data, etc. User submits jobs to the WMS.
2. The WMS dispatches the job to the OSG worker node. We assume that this deployment of Glidein WMS supports asynchronous output storage. This feature was recently introduced by the CEDPS project. In this mode, the WMS handles output storage, rather than the job, and starts a new job on the same worker node slot while the output is being transferred. This optimizes compute cycles.
3. The application runs and produces output in a scratch disk local to the worker node.
4. The WMS system interacts with GO to asynchronously store the output to an SRM storage.
5. Once the transfer is done, the WMS successfully terminates the job.
6. The user will periodically retrieve the data from SRM to her PC (see use case above).

4. Back up of multiple home areas to SRM storage (NEES backup case)

1. A user wants to have a backup of her home areas, which contain numerous files with different sizes, with total volume of O(10 TB).
2. A user starts globusconnect on his node and transfers the data to the selected SRM storage.

5. A VO needs to move TBs of data from one SRM to another (Fly's Eye case)

1. A user pre-staged data to an SRM storage using GO. The user is planning to use the same data for a long period of time (1 year).
2. At some point, either the site admin either needs to free up the space or available CPU cycles on this site doesn't satisfy the VO needs any more.
3. The user needs to move a large volume of data from one site to another in a timely manner. She uses GO to transfer data from and SRM storage to another to accomplish that.

6. **User retrieves output from job submission node via SRM (condor I/O case)**
    1. The user logs to an OSG submission node. Prepare local input data, etc. User submits jobs to WMS.
    2. The WMS system dispatches the jobs to the OSG worker nodes.
    3. The application produces its output to a local scratch disk. In this use case, instead of actively storing the output to an SRM storage, the application relies on the WMS to bring it back to the submission site.
    4. The WMS system terminates the job and makes the output available to a file system externally visible through SRM.
    5. The user periodically retrieves the output to her PC from the SRM interface at the submission node using the mechanism above.

7. **Job retrieves input from SRM storage (input retrieval case)**
    1. The user logs to an OSG submission node, prepares a list of files that should be transferred via GO from a source SRM storage to the WN. The user submits the jobs.
    2. The WMS dispatches the job to the WN.
    3. The application starts up globusconnect and uses GO to transfer the files from SRM to the local resource.
    4. The application runs the analysis. Output is handled with GO as above or by other mechanism.

Potential User Communities

The OSG User Support group is investigating the inclusion of globusonline.org in its processes and best practices to move files, according to the use cases described above. The group integrates support for new communities in stages, from a proof-of-principle, to a production-demo, to a production-ready phase. Should globusonline.org become a more integral part of the user support processes after our evaluation, the users could learn the globusonline.org system during the initial integration phase and eventually keep using it in their production activities. A goal of the user support group is to integrate in OSG the operations of two new user communities per year.

In the last year, the group has been supporting these new OSG communities: the Large Synoptic Survey Telescope (LSST), the Network for Earthquake Engineering Simulation (NEES), the Dark Energy Survey (DES), and the Worldwide e-Infrastructure for NMR and structural biology (WeNMR).

Other communities that have worked with OSG User Support in the recent past include Fly's Eye, Geant4 development team, and ITER / TGyro.