

A web-based utility for Grid data management

Fattibene Enrico¹

INFN-CNAF

Bologna, Italy

E-mail: enrico.fattibene@cnafe.infn.it

**Bencivenni Marco, Ceccanti Andrea, Cesini Daniele, Di Benedetto Michele,
Michelotto Diego, Misurelli Giuseppe, Venturi Valerio, Veronesi Paolo, Zappi
Riccardo**

INFN-CNAF

Bologna, Italy

Brunetti Riccardo, Gaido Luciano

INFN-Torino

Torino, Italy

Caltroni Andrea

INFN-Padova

Padova, Italy

Large Grid infrastructures deploying the European Middleware Initiative (EMI) software release are currently used by many scientific communities that take advantage of the available distributed computing and storage resources. However, the exploitation of Grid services by new users, especially those belonging to small communities (often with very limited knowledge of the Grid) is not very easy due to the difficulty of handling X.509 digital certificates and to the complexity of the job submission and data management tasks.

¹ Speaker

In order to hide this complexity, the Italian Grid Initiative (IGI), that operates the Italian Grid infrastructure, has developed a web-based portal that provides users with several important services such as job submission, workflow definition and data management.

This paper describes the IGI Portal's Data Management component that allows authorized users to easily manage their data by means of very simple operations within the portal. The Portal allows users to upload and download files on the Grid, browse Grid file catalogues, replicate data on specific storage elements, manage file privileges and perform some other basic operations.

The architecture of the Data Management component has been designed in such a way to avoid the Portal itself to become a bottleneck when large or many concurrent data transfers occur. The adopted solution consists of a service, the Data Mover, that manages the data transfers by means of a dedicated storage service acting as a cache. When users upload data from their desktops/laptops through the Portal's web interface, the Data Mover stores them in its storage cache and then asynchronously copies them to the desired Storage Element and registers them into the appropriate Grid catalogue; in this way the Grid middleware complexity is totally hidden to the end user. The structure of this component is modular in order to make the integration of new features easier.

POS (ISGC 2013) 004

The International Symposium on Grids and Clouds (ISGC) 2013
March 17-22, 2013
Academia Sinica, Taipei, Taiwan

1. Introduction

In the last years, large Grid infrastructures have been deployed worldwide to provide scientific communities with powerful distributed computing and storage facilities. One of the major existing Grid infrastructures is the European Grid Infrastructure (EGI) [1], that federates different National Grid Infrastructures (NGIs) and it is based on the European Middleware Initiative (EMI) [2] software release.

The experience of the EGI's operational and user support activity shows that while these Grid resources are being exploited by different scientific communities that benefit from the capabilities of the infrastructure, the access to the resources is still quite difficult to new user communities. In particular, the usage of X509 personal certificates and the complexity of the job submission and data management tasks discourage them to approach the Grid world, especially when dealing with users not very experienced in computer science.

Some NGIs, among which the Italian Grid Infrastructure (IGI) [3], one of the largest national Grids included in the EGI federation, have already started working on possible solutions to improve the usability of the Grid. Driven by the need to make the computing infrastructure easily available to the largest number of scientific communities as possible, IGI started developing the IGI Portal, an easy-to-use web interface to the Grid.

2. The IGI Portal

The IGI Portal is a general purpose Grid interface based on the Liferay [4] framework. By means of dedicated portlets [5] it provides users with several important services, such as job submission, workflow definition (based on WS-Pgrade engine [6]), data management and accounting services. It is also interfaced with external Infrastructure-as-a-Service (IaaS) frameworks for the dynamic provisioning of computing resources.

User authentication is demanded to an Identity Provider (IDP) federation based on the SAML 2.0 [7] protocol, while the authorization is based on personal X.509 certificates and Virtual Organization (VO) membership. The management of the Grid credentials is handled by a MyProxy server [8], integrated into the Portal, where long term encrypted VOMS [9] proxies are securely stored. Unlike most other existing solutions (in particular the so-called science gateways), the IGI Portal does not leverage robot certificates for the user credentials since it makes of digital certificates issued by an embedded online Certification Authority. This approach has two important advantages with respect to the science gateways. The first one is that user identity is not hidden across the middleware stack thus preserving the functionality and effectiveness of the existing distributed accounting and authorization mechanisms; the second one is that users' activity is not limited to a predefined set of options but they can take advantage of all the Grid services and perform any computational or data-intensive task they need.

3. EMI Data Management utilities

In the EMI middleware, data can be stored on different storage systems based on different technologies. The details of the different storage implementations is hidden by the Storage Resource Manager (SRM) [10], a common interface that takes care of the interaction with the storage systems and provides a common Grid interface to the outside world.

Thanks to this interface, EGI/EMI users can carry out the main operations (upload, copy, download, etc.) on Grid data even if they have no knowledge about the specific Storage Elements (SEs) implementations. These operations are possible by means of a set of command line utilities included into the EMI User Interface middleware profile: the *LCG utils* and the *lcf-* commands*.

Both these utilities interact with the LCG File Catalogue (LFC) [11], that maps a generic file name, the Logical File Name (LFN), to one or more physical locations of a file on the Grid.

According to the EMI/EGI file name conventions, each file is univocally identified by a Global Unique Identifier (GUID), a non-human-readable fixed-format string that identifies an item of data. Each file can be replicated in different locations and each replica is identified by a Site URL (SURL) or Physical File Name (PFN). The Transport URL (TURL), a string composed by the access protocol and the temporary locator of a replica, provides the necessary information to access a file in a specific SE.

3.1 LCG utils

LCG utils is a suite of client tools for data movement based on the Grid File Access Library (GFAL) [11], which is also included in the EMI middleware distribution. These tools allow users to copy files between SEs, register entries in the LFC and replicate files between SEs. Some commands use LFNs and require a connection to the a LFC; exceptions are file copies and deletions, which take endpoints based on the SURL. Table 1 reports a list of the most important commands.

lgc-cp	Copy a Grid file to a local destination
lgc-cr	Copy a file to a SE and registers the file in the catalogue
lgc-del	Delete a file
lgc-rep	Replicate a file between SE
lgc-gt	Get the TURL for a given SURL
lgc-aa	Add an alias in LFC for a given GUID
lgc-ra	Remove an alias in LFC for a given GUID
lgc-rf	Register in LFC a file stored in a SE
lgc-uf	Unregister in LFC a file stored in a SE
lgc-la	List the alias for a given SURL, GUID or LFN
lgc-lg	Get the GUID for a given LFN or SURL
lgc-lr	List the replicas for a given SURL, GUID or LFN

Table 1 – LCG utils commands

3.2 *lfc-** commands

The *lfc-** commands make use of LFC only and don't operate on data directly. They can be used to list the content of LFC directories, rename or delete files and folders, get or set permissions and ACL, etc. Table 2 shows the more common available *lfc-** commands.

lfc-chmod	Change access mode of the LFC file/directory
lfc-chown	Change owner/group of the LFC file/directory
lfc-setcomment	Add/replace a comment
lfc-delcomment	Delete the comment associated with the file/directory
lfc-ln	Make a symbolic link to a file/directory
lfc-ls	List file/directory entries in a directory
lfc-getacl	Get file/directory access control list
lfc-setacl	Set file/directory access control list
lfc-mkdir	Create a directory
lfc-rename	Rename a file/directory
lfc-rm	Remove a file/directory

Table 2 – *lfc- commands**

4. IGI Portal Data Management

The utilities described above allow users to perform all the desired operations on the data stored on the Grid and provide a set of options that allow them to modify the command behavior. Unfortunately, to fully benefit from these API, users are required to know a set of commands that are often long, syntactically difficult and very far from the “Drag-and-Drop” model implemented in the most common interfaces..

In order to address this issue, the IGI Portal provides a web-based data movement service that offers the possibility to benefit from the capabilities of the Grid data management command line utilities by means of simple web operations.

The data management service is based on the AjaXplorer [12] framework, a PHP-based file management system which provides an intuitive interface and the allows an easy integration of new components. Many plug-ins are natively provided by AjaXplorer but many others can be easily developed and integrated. A new plug-in for the data management within the IGI Portal has been developed to extend the basic functionalities of AjaXplorer in order to manage the local file systems and integrate the *LCG utils* and the *lfc-** commands.

In this way, through the web interface, users can easily browse the content of the file catalogues pertaining to their VO and access their Portal File System, a storage space reserved users on the Portal SEs (see paragraph 4.1).

The Data Management Interface is shown in Figure 1.

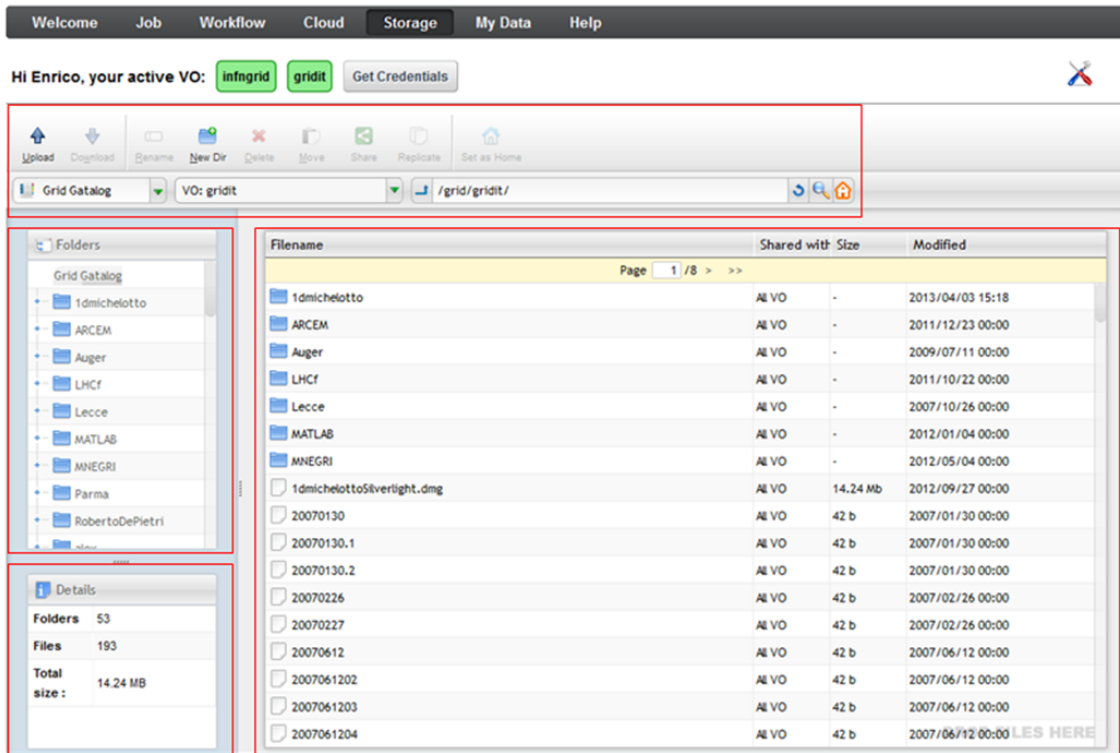


Figure 1 – Data Management Interface

The Data Management interface is composed by four sections.

- *Command Section* (on the top), where users can perform actions on selected files or directories and browse the content of the Grid Catalogue (LFC) or the Portal File System.
- *List Section* (on the right), where all the files and directory are listed. If the number of files is too big, the output is automatically split in pages. *Folders Section* (on the top left), where only the directories are listed: it is useful to quickly navigate the tree.
- *Details Section* (on the bottom left), where detailed information about the selected folder or file is shown.

4.1 Architecture

In order to avoid the Portal to become a bottleneck for data movement operations, files are uploaded/downloaded through an external temporary storage. The architecture of the Data Management service is shown in Figure 2.

POS (ISGC 2013) 004

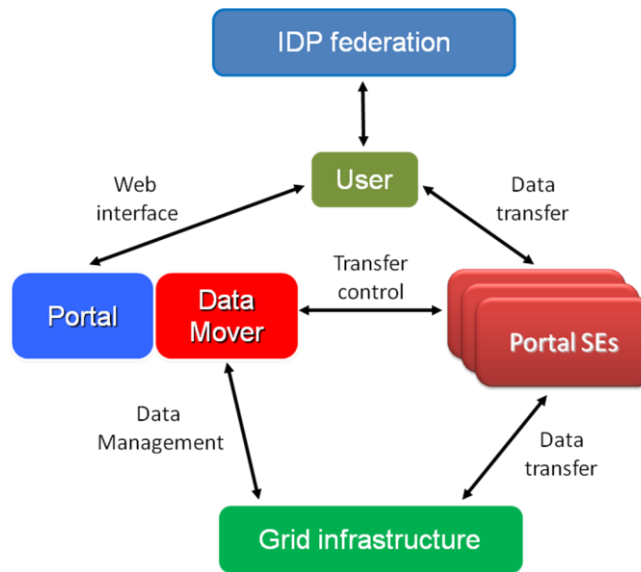


Figure 2 – Data Management Architecture

From the logical point of view, the main components of this architecture are the Data Mover Service and the external temporary storage composed by a set of SEs (Portal SEs). For the data transfer operations (upload and download), the Portal SEs act as cache memory which keeps the files until they are transferred to (upload phase) or downloaded from (download phase) a Grid SE. The Data Mover controls and manages every step of the file transfer. For the other operations on logical and physical data, the Data Mover uses the standard Grid tools and services, i.e. LFC for operations on logical file names and SEs for permanent data archiving..

From the implementation point of view, the Data Mover service is integrated in AjaXplorer and the external temporary storage (the so-called Portal SEs) are StoRM [13] based Grid SEs in a cluster configuration. In this way it is possible to achieve a complete decoupling of the view interface (Portal), the service control (Data Mover) and the physical storage (Portal SEs). This architecture is scalable since the SEs cluster can grow, also with geographically distributed SEs, and ensures the final user a correct and fast data transfer without bottlenecks.

4.2 Features

Any authenticated user, holding a valid VOMS proxy stored in the Portal, can access the Portal's Data Management Interface (*Storage* button in the web menu) and browse the content of the LFC for his/her own VO. The default home browsing path is always set to the LFC home for that VO but each single user can customize it and make it to point, for example, to a given subdirectory in the catalogue. For each listed item a number of details are given, like the type of item (file or folder) or the size and the last modification time. The information about the level of sharing for a file is also reported: through the web interface users can share a given file either with all the members of their VO or with selected Portal users or with nobody else (i.e. only the owner has the privileges to read and write the file). By clicking on items in the *List Section* or in the *Folders Section*, it is possible to browse the LFC content.

The allowed actions can be grouped in three main categories: operations on the logical data (which affect the catalogues), operations on the physical files (which affect the SE) or both.

The basic permitted operations on the catalogue content with effect only on the logical data are:

- creation of a new folder
- deletion of an empty folder
- renaming or movement of a folder or file (changing the LFN)
- getting detailed information about a file (LFN, GUID, list of replicas, owner, ACL)
- sharing the file with other Portal users

The basic permitted operations with effect only on the physical data are:

- replication of files on other storage elements
- download of files (see paragraph 4.2.2)

The basic permitted operation on catalogue content with effect on both physical and logical data are:

- deletion of files
- upload of files (see paragraph 4.2.1)

4.2.1 File Upload

In order to upload files to the Grid, the Data Mover implements two external tools: Plupload [14] (for Internet Explorer web browser) and jQuery File Upload [15] (for the other web browsers). Both these tools implement the chunking mechanism, that allows the upload of large files, of the order of tens of gigabytes. Moreover, for the browsers that support the HTML 5 function *xmlhttprequest*, jQuery File Upload operates also the Resumable Upload, that permits to resume a previous failed upload restarting rather from the last bit successfully transferred. The system is able to recognize the web browser used by the requester and automatically choose the appropriate tool accordingly.

Through the upload operation the file is copied on a SE (the user can either choose the destination SE or not) and registered into the LFC. In particular, the file is first uploaded on the Portal SEs and then copied on the final SE and registered in the catalogue.

According to the file size and the VO, the system produces a list of possible destination SEs. If the user chooses a specific SE the system tries to upload the file to that destination; in case of failure, the system tries to copy the file on some other SEs of the list following a random order. During the upload operation the user can monitor the transfer status, as shown in Figure 3.

When the transfer has been completed users are informed about the result. There are three possibilities:

- *Transfer ok*: the file has been correctly transferred to the selected SE
- *Transfer ok but not on the selected SE*: the file has been correctly transferred into Grid but not to the selected SE
- *Transfer failed*: the file has not been transferred to any SEs, in this case more details about the failed transfer could be provided.

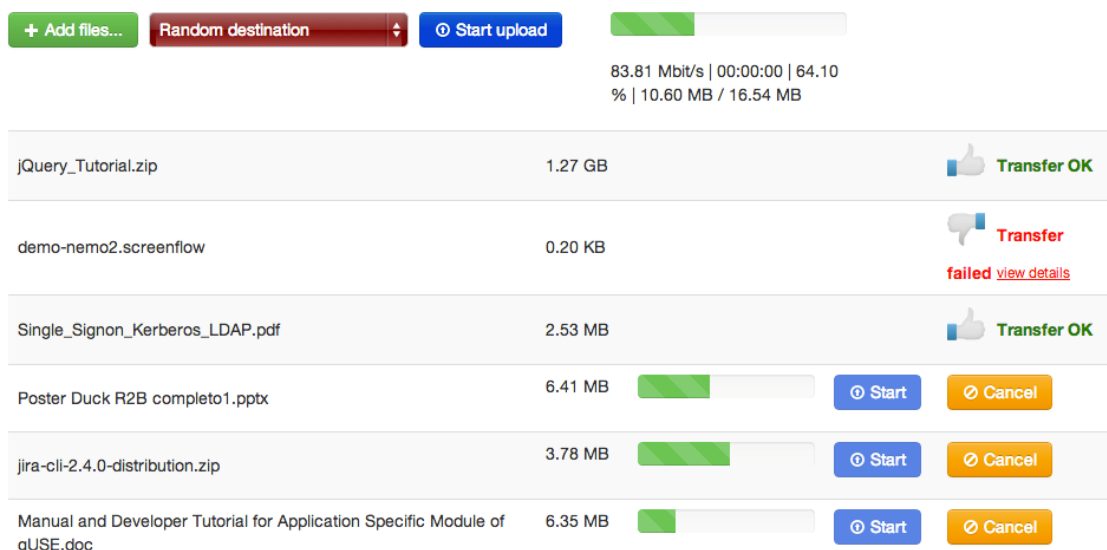


Figure 3 – Upload Interface: monitoring the transfer status

4.2.2 File Download

Another feature is the possibility to download files from the Grid infrastructure to other destinations. At the moment it is possible to copy a Grid file either to a local destination or to an external server. To avoid transferring big files to the user’s local space, it is not allowed to download a file or a bunch of files whose overall size is greater than 5 gigabytes on the user’s personal computer.

For the transfer to external servers various protocols are supported: FTP, SFTP and WEBDav. Users has to set at least the server hostname, the username and the password (if needed); port and path are optional fields.

The file transfer from the Grid infrastructure to user’s local space or an external server is completely transparent to the user and, like the upload phase, it consists of two steps. The file is first retrieved from Grid then temporarily stored on the Portal SEs and finally transferred to the final destination. For file transfers to the local space users can explicitly choose to save a copy of selected items on the Portal File System (Portal SEs) (otherwise the file will be removed when the transfer has been completed. It is also possible to save more than one file in a single operation; in this case the system waits until all the files are retrieved from the Grid and then generates a *tar* file and transfers it to the chosen destination.

5. Conclusions

This paper has described the Data Management service integrated in the IGI Portal that leverages data management tools and command line utilities, hiding their complexity to users and providing some additional functionalities.

It is a real benefit for the user communities because they can perform all the common Grid data management tasks through an intuitive web interface, without the need of learning

command line commands whose syntax is often very complex. It is therefore a powerful tool to attract new user communities. Some additional features, including the integration of an online Certificate Authority and the interface to cloud computing and storage services, will make the IGI Portal an enabling tool to provide services to a wider and wider user base. Moreover the possibility to transfer data also among non-Grid SEs is currently under investigation.

References

- [1] EGI website (<http://www.egi.eu/>)
- [2] EMI website (<http://www.eu-emi.eu/>)
- [3] IGI website (<http://www.italiagrid.org/>)
- [4] R. Sezov, *Liferay in Action - The Official Guide to Liferay Portal Development*, September 2011, ISBN 9781935182825
- [5] S. Hepper, *JSR 286: Java Portlet Specification Version 2.0*, Java Community Process (2008)
- [6] P. Kacsuk, K. Karoczkai, G. Hermann, G. Sipos, J. Kovacs, *WS-PGRADE: Supporting parameter sweep applications in workflows*, Workflows in Support of Large-Scale Science, 2008. WORKS 2008. Third Workshop on, vol., no., pp.1,10, 17-17 Nov. 2008
- [7] S. Cantor, et al., eds., *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS Standard, March 2005
- [8] R. Alfieri, R. Cecchini, V. Ciaschini, L. Dell'Agnello, et al., (2004), *VOMS, an authorization system for virtual organizations*, Grid Computing, Springer Berlin Heidelberg, 2004
- [9] D. Koufil, J. Basney, *A credential renewal service for long-running jobs*, Grid Computing, 2005. The 6th IEEE/ACM International Workshop on. IEEE, 2005.
- [10] A. Sim, et al., *The storage resource manager interface specification version 2.2*, OGF Grid Final Document 129 (2008)
- [11] Á. Frohner, et al., *Data management in EGEE*, Journal of Physics: Conference Series. Vol. 219. No. 6. IOP Publishing, 2010
- [12] Ajaxplorer website (<http://ajaxplorer.info/>)
- [13] R. Zappi, et al., *StoRM: grid middleware for disk resource management*, Proceedings of Computing in High-Energy Physics. Vol. 27. 2004.
- [14] Plupload website (www.plupload.com)
- [15] jQuery File Upload website (<http://blueimp.github.io/jQuery-File-Upload/>)