Mem. S.A.It. Vol. 80, 493

Memorie della

Grid FITS IO

G. Taffoni[1,2], C. Vuerli[1,2], and F. Pasian[1,2]

[1] Istituto Nazionale di Astrofisica – Information Systems Unit
[2] Istituto Nazionale di Astrofisica – Osservatorio Astronomico di Trieste, Via Tiepolo 11, I-34143 Trieste, Italy, e-mail: [vuerli;taffoni;pasian]@oats.inaf.it

Abstract. One of the main problems in Grid infrastructures is data access and sharing. In the case of EGEE grid infrastructure the access to data is mediated through a grid file catalogue that virtualize the distributed files and it appears to the user as a virtual filesystem. One of the main problem when working with the file catalogue is to implement the file catalogue APIs in to the user code to read/write files directly from/to the grid infrastructure. In the case of Astronomy the standard format for Astronomical data is FITS. The library to manipulate FITS les is the CFITSIO library. We designed and implemented a set of drivers for the CFITSIO libraries that interact directly with the Grid lesystem in this way the Astronomers that want to use the EGEE grid infrastructure should not modify their code but compile it with a CFITSIO library with grid support.

Key words. Grid: applications – Grid, data – EGEE project and infrastructure – CFITSIO

1. Introduction

Grid computing is emerging as a global resource for the next generation of e-Science applications. It is essentially distributed computing over wide-area networks and it involves large scale resource sharing, selection and aggregation. The concept of Grid computing originated as a pro ject to link geographically distributed computing and data resources and to link a number of scientic institutions bringing together communities with common ob jectives. That is the case of the international collaborations that involve the new-generation sci- entic experiments and studies. Commonly, these experiments produce huge amounts of data (in the terabyte and petabyte range), distributed along dif- ferent sites and needing to be processed and analyzed. Some example of these new-generation experiments are the High Energy Physics CERN LHC exper- iments (http://lcg.web.cern.ch/LCG/), the ESA astronomical satellite Planck (Taffoni et al 2006) or Multi-Site Online Simulation Test (MOST) earthquake engineering experiment.

Even if Grid infrastructures originated to fulll the needs of huge interna- tional collaborations, nowadays the concept of Grid computing has grown far beyond its original intent. The Grid infrastructures can be benecial to many applications, including collaborative engineering, distributed computing or data exploration. A number of different scientic communities, ranging from High Energy Physics, Hearth science, Astronomy to Bioinformatics and Biomedicine, are utilizing Grid computing for their everyday computational and data neces- sities. One of the key points in using Grid infrastructures is that scientists do not

Send offprint requests to: G. Taffoni

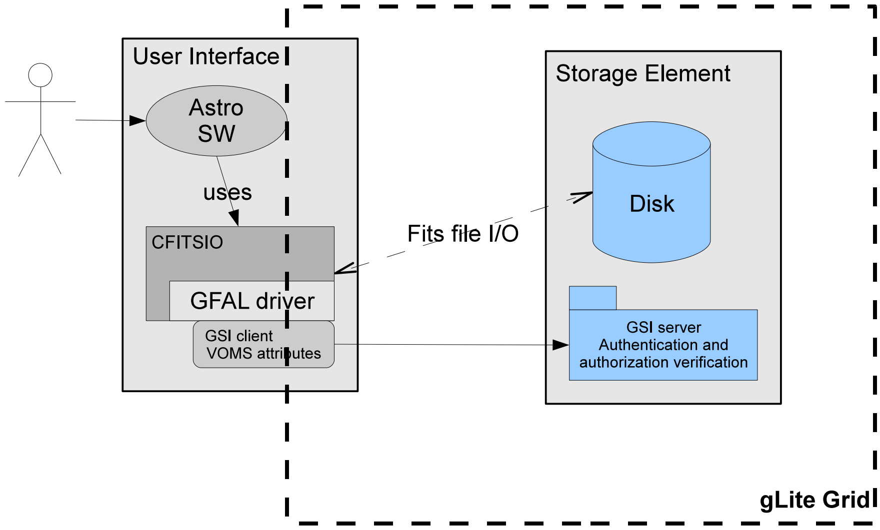

Fig. 1. Structure of FITS data access through the Grid. Any software that uses CFITSIO with GFAL support, may access remote les from a User Interface (i.e. any computer with gLite clients installed). When remote IO is required, CFITSIO uses the gLite APIs to verify the presence of a valid user X509 proxy certicate in the user interface and to transport the limited proxy to the remote resource. These steps initialize the connection. If the user is authenticated and authorized by the Grid Security Infrastructure (GSI) then the real le IO begins.

need to deeply modify their codes, which can commonly run on the Grid with only some minor changes. Commonly, those changes are related to the program IO that needs to be adapted to use the powerful distributed data management system (DM) of the Grid. In this paper we present a new set of drivers that we implemented in the CFITSIO library (Pence 1998) in order to facilitate as much as possible the use of the Grid environment for Astronomers.

2. The EGEE grid infrastructure

The whole software that underlies the fundamental Grid services, such as infor- mation services, resource discovery and monitoring, job submission and man- agement, brokering and DM and resource management represents the Grid mid- dleware. The middleware is the core of the Grid, just as the operating system is the core of a computer. More specically, the middleware is an evolving layer of services residing between the network and more traditional applica- tions for managing security, access and information exchange. It builds upon standard open source solutions as the Globus Toolkit (GT, `http://www.globus.org`) (Foster et al 2001). Commonly, Grid infrastructures use GT as foundation and they build on top of it high level services that direct the whole infrastructure. This is the case of the Enabling Grids for E-science (EGEE) project. EGEE (`http://egee-intranet.web.cern.ch/`) aims at providing academic and industry researchers with access to ma jor computing

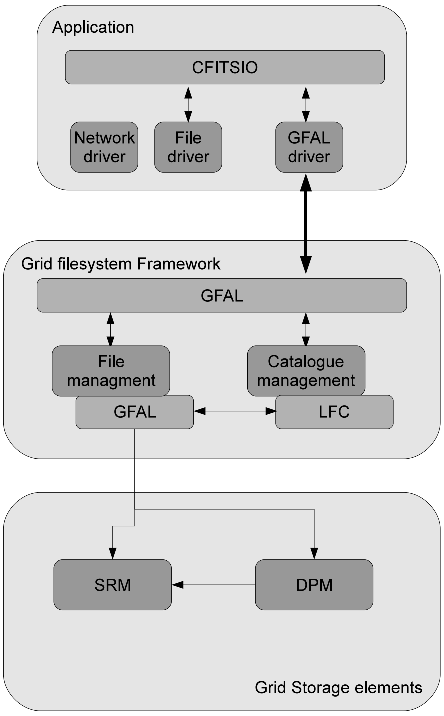

Fig. 2. Layered architecture of CFITSIO grid implementation. The GFAL driver is used to access the grid filesystem. When a user specify a LFN, the Grid filesystem framework translate the LFN into a Storage URL and then proceed to access file. The user can also specify the Storage URL. The GFAL APIs mediate the access over the SRM interface and DPM interface of Storage Elements.

resources, independently of their geographic location (Jones 2007). The EGEE middleware is gLite (`http://www.glite.org`) (Laure et al. 2006).

3. Data Managment in EGEE: the file catalogue

In gLite architecture, data management is build on top of GT gridFTP protocol and a file catalogue. The lowest granularity of the data is on the file level. Files are stored in *Storage Elements* (SE), the storage resources that provides SRM (Storage Resource Manager) and gridFTP interfaces.

On top of the SEs, a grid file catalogue is build: it virtualizes the single SEs on a virtual distributed "filesystem".

Users identify files by logical file names (LFNs). The LFN namespace is hierarchical, just like a conventional filesystem. The semantics of the LFN namespace is also almost exactly like that of a Unix filesystem. The LFN is not the only name/identifier that is associated with a file in the Grid, although the average user may never use any other filename and is given the benefit of a single global namespace. gLite provides a catalog, named LFC, to store the location(s) of their files and replicas. LFC will map LFNs to Storage URLs. It is a high performance file catalogue that builds on the experiences gathered from the EGEE user communities.

In order to shield the user from the differences the current storage systems expose in their posix-like access libraries, gLite provides a Grid File Access Library (GFAL), a C API posix-like interface that provides methods such as *gfal_open, gfal_read*, etc. LFC is integrated with the GFAL interface.

4. gLite driver for CFITSIO

CFITIO library mediates read and write operations on FITS les through a set of IO drivers. The IO drivers provide access to FITS les stored locally or remotely. Remote access to HTTP locations or anonymous FTP locations is permitted in read-only mode, while access to a root server is allowed in read/write mode. Taffoni et al. (2006) designed and implemented an I/O driver to access les stored on a remote Gridftp server in read/write mode. They used the GT (>2.4) API to implement a driver that reads FITS les from the server, creates a local buffer and operates on the local buffer as soon as le close/registration is requested. If a new le is created all the operations are performed locally on the le buffer. Only when the le is closed a remote write operation is performed and the local buffer is cleaned.

The drawback of this implementation is that each operation on the fits file requires to transfer the whole file locally, even if just a user needs to read the header, that consists of a few bits of data.

To overcome this limitation, we design and implement a new CFITSIO data driver based on GFAL C API (Fig. 1). The GFAL offers the POSIX I/O client library for direct access of data, when file open operation is requested there is not actual data movement, saving network bandwidth and time. This implementation reduces as much as possible data transmission through wide area network. One of the characteristics of the GFAL protocol is the multiple data channels for parallel transfers, which increases transmission performances. We allow the user to set the number of transfer channels using an environment variable to tune the transfer mode according to his needs (Fig. 2).

To access a file, it is necessary to specify its LFN or Storage URL and the access protocol (srm, gridftp, rfio, etc). Of course a valid user certificate must be provided and a VOMS proxy certificate must be initialized. VOMS (Virtual Organization Membership Service) is the gLite authorization service (Alfieri et al. 2005). Using VOMS it is possible to assign VO based group membership, roles, and capabilities entitlements to a user. Those additional attributes are included in the user proxy certificate and presented to the Grid resource which uses them to define a specific user profile.

The CFITSIO driver extracts the Virtual Organization membership from the VOMS attributes of the certificate and use it to authorize the user on the SE and on the LFC.

5. Conclusions

The use of a Grid IO driver for CFITSIO allows users to access Grid resources with no modications to their codes. The implementation of the IO drivers has been done so that :

- the implementation is multi-platform and it is designed to work in any Grid infrastructure;
- high-level drivers are implemented to work on the LFC;
- the implementation has been explicitly tuned to optimize data tranfer;
- astronomers will benet of all the Grid computing and data resources with no additional modications on their codes due to the peculiar Grid data management.

Using the combination of GFAL API and the ACL of the SE and LFC it is possible to store data on the grid environment that none except the owner (identified through her X509 certificate) can read. That guarantees a sufficient level of security so that data centers may use the grid infrastructure to distribute their data.

Acknowledgements. A special acknowledgement is due all colleagues that collaborates to the development and to the GILDA team for its documentation and examples on GFAL APIs.

References

Alfieri, R., et al. 2005. . Future Generation Comp. Syst. 21(4): 549558.

Foster, I., Kesselman, C.and Tuecke, S., The International Journal of High Per- formance Computing Applications, 15(3):200222, 2001

Jones, R. 2007, Enabling Grids for E-sciencE, 1, 228.

Laure, E., et al., 2006, Comp.Meth.Scie.Tech, 12, 33

Pence, W.1998, in ASP Conf. Ser., Vol. 172, ADASS VIII, ed. D. M. Mehringer, R. L. Plante, & D. A. Roberts (San Francisco: ASP), 487

Taffoni G., et al. 2005, in ASP Conf. Ser., Vol. 347, ADASS XIV, ed. P. Shopbell, M. Britton, & R. Ebert (San Francisco: ASP), 320

Taffoni G., et al. 10.1016/j.future.2006.04.020

Taffoni G., et al. 2006, in ASP Conference Series, Vol. 376, R. Shaw, F. Hil l and D. Bel l, eds. , p. 89