

The new EGRID infrastructure

Ezio Corso* Angelo Leto, Antonio Messina, Riccardo Murri, Riccardo Di Meo,
Alessio Terpin

EGRID project

E-mail: {ecorso, aletto, amessina, rmurri, dimeo, aterpin}@ictp.it

Stefano Cozzini

CNR-INFN Democritos and EGRID project

E-mail: cozzini@democritos.it

This article is an update on the status of the EGRID infrastructure for the analysis of economic and financial data, as first described in [4]. A brief review of the original design is given, followed by the problems identified during the operation of its implementation. The new design is then presented together with the solutions introduced with it.

Grid Technology for Financial Modeling and Simulation
3-4 February 2006
Palermo, Italy

*Speaker.

1. Introduction

The aim of the EGRID project is to implement an Italian national facility for processing economic and financial data using grid computing technology. It therefore acts as the underlying fabric on top of which partner projects more strictly focused on research, develop end-user applications.

The first version of the EGRID infrastructure was released on October 2004, it was based on European Data-Grid (EDG, [2]) middleware, and it was hosted as an independent Virtual Organization (VO) within INFN's grid.IT¹. The middleware's data management service did not meet EGRID's privacy and security requirements so a temporary workaround was implemented. The learning curve associated with grid usage was also a concern so specific command line tools were developed to aid end users; UI software installation was also simplified with live-CD technology.

During this time the EDG middleware evolved into the Large Hadron Collider Computing Grid (LCG, [1]), part of which was also used in EGEE's middleware [3].

The new infrastructure was redesigned to address the temporary workarounds, to employ the new middleware, and to use the new tools developed independently within EGRID. Recently, EGRID also joined EGEE [3] as pilot application in the field of finance; the EGRID VO will soon be recognized on the full EGEE computing grid, and possible interoperability problems will be handled when they become better defined.

2. State of the art of the EGRID infrastructure

The most important user requirements follow, together with a brief overview of the original EGRID infrastructure: full details can be found elsewhere [4].

2.1 User requirements

EGRID must be able to store and to supply management services for about 2TB of data coming from various stock exchanges: the New York Stock Exchange, the London Stock Exchange, the Borsa di Milano, etc. The user community must be supplied with a basic set of user-friendly data management commands for uploading/downloading data to/from the grid, for removing and browsing the data, and more generally for all the usual file manipulation operations. Legally binding disclosure policies on the data imply that special attention must be paid to privacy issues. By privacy it is meant that not all users *within* the EGRID VO have indiscriminate *read* access to all grid stored data. For example, some researchers may have an exclusive contract with the London Stock Exchange, while others may have an agreement with the New York Stock Exchange for information pertaining only to specific companies. Two classes of users must be distinguished: those whose only responsibility is to upload data files into the grid and if necessary to remove them, and those that work on that data.

EGRID must also organize a processing facility for transforming, filtering and analyzing the data. Two classes of operations must be distinguished: *raw data pre-processing* and *end-user applications*.

¹The national grid computing infrastructure of INFN (*Istituto Nazionale di Fisica Nucleare*: Italian National Institute for Nuclear Physics).

- *Raw data pre-processing* are operations whose output is used by a large group of users; for example, they can be filters which bring data into a more standard format than the proprietary one offered by the stock exchanges.
- *End-user applications* are custom operations performed during research activity.

2.2 The original EGRID infrastructure

Because of contractual terms the EGRID project could only own a minimum amount of hardware, so all bulk computing power and storage space was rented from an external computing centre, the INFN of Padova. EGRID also decided to employ the same EDG middleware in use there, in order to take full advantage of that infrastructure.

EGRID sites were then classified based on their network bandwidth. Those with limited connectivity were designated “peripheral sites” and were used to provide grid services only to *local* users. They acted mainly as cache areas for large data transfers, and as job execution points for non CPU- intensive local data processing. Sites with good connectivity were designated “central sites” and provided grid services to the whole EGRID community. This organization resulted in a two-tiered topology.

A workaround was devised in order to meet the security and privacy requirements, because EDG lacks a robust mechanism. Indeed, the middleware uses the GridFTP daemon to expose local files to the grid, allowing file manipulation operations compatible with the set of permissions present on the underlying Unix filesystem. This becomes problematic when data is spread over *different* SEs, and also for the tools that users must be supplied with in order to manage those permissions. To meet the requirements the SEs were configured to have each grid user mapped to personal local accounts, and an especially crafted nested directory tree was set up (see [6, 7]). Finely tuned group membership of the various users together with a careful replication of that tree, completed the workaround.

Raw data processing operations were supported by providing users with a tailor made wrapper component that encapsulated grid operations: `egrid-transform-dispatcher`. It allowed developers to more easily grid enable their processing operations, and it also allowed users to more easily launch grid processing on specific datasets. After realising that most data resided in plain text files, some common UNIX shell tools for text processing were adapted to operate on grid storage: `cat`, `cut`, `grep`, `join` and `paste`. They were thought of as basic building blocks from which more elaborate operations could be built.

Finally, since user applications were specific to researchers’s interests and lacked general characteristics, programmes and function libraries were developed to help end users port their applications to the grid environment. To facilitate the installation of grid client software on user workstations, live-CD technology was also used.

3. The design of the new release

Several aspects of the original EGRID infrastructure described in the previous section could, and had to, be improved upon.

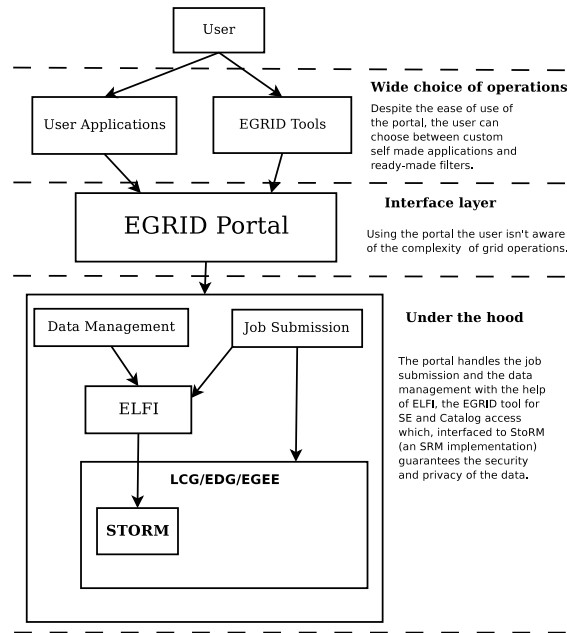


Figure 1: Schematic representation of the EGRID infrastructure

The presence of only one large computing site was clearly insufficient to demonstrate the grid potential for distributed resource allocation. It was therefore decided to expand the hardware infrastructure to include a second large computing centre in Catania, besides the one of Padova.

Unfortunately the EDG middleware and its successor LCG, still lacked support for security and privacy of stored data *within members of the same VO*. The workaround mentioned before was complex and had scalability issues: two aspects were deemed particularly unsatisfactory. First, having a real account for each grid user in each grid SE; and second, having permissions set on the filesystem thereby posing severe replication problems. To overcome them, a middleware intervention was made at a sufficiently low level.

The two tiered topology was also problematic. Installing and maintaining peripheral sites proved to be quite difficult: the local users in charge of the site often were not skilled enough to easily manage the task; and the EGRID team had not got enough manpower to foresee the management. The topology was dropped by giving up on peripheral sites altogether.

Although the command line tools were tailored specifically to the needs of the users, unfortunately only a small part of the community adopted them. Similarly for the UI distributed through live-CD technology; although users had been spared the need to reinstall their workstations, they complained the usage was awkward and that it was interfering with their way of working. A web portal, then, promised to be a very good tool to address these issues.

The new infrastructure, therefore, was designed with various architectural layers as depicted in figure 3. The main focus will be a web portal that will act as the user interface and main entrance point to EGRID. It will be better described in the next section, while brief comments on the other components follow.

The old command line interface will be kept for compatibility reasons with existing applica-

tions, and for the few sophisticated users that eagerly adopted them. All new effort, however, will be devoted to development and integration of the portal. This is indicated in the figure with “Wide choice of operations”.

The layer denoted with “Under the hood” addresses several issues. Security was introduced at the data management middleware level: classical SEs will be replaced with an implementation of the SRM (Storage Resource Manager, [5]) protocol that is currently being completed. This allows the workaround mentioned earlier on to be dropped. More generally, SRM is becoming widely used as a storage access interface and it will soon be available on the full EGEE infrastructure. The security solution therefore is in line with mainstream grid trends.

This SRM implementation is the result of an on-going collaboration with INFN- CNAF on the StoRM SRM server [8]; it is an opportunity to make an intervention at the software level, to implement the kind of fine grained access control that EGRID needs.

StoRM adds appropriate permissions using POSIX ACLs to a file being requested by a user, and then removes those permissions when the client finishes working on the file. Since permissions are granted on-the-fly, grid users can be mapped into pool accounts and no special permission sets need to be enforced prior to grid usage. An important role is given to a secure web service (ECAR) purpose built by EGRID to act as a bridge between the (resource-level) StoRM SRM server, and the (grid-level) central LFC logical filename catalogue from EGEE, that replaces the old RLS of EDG. The LFC natively implements POSIX-like ACLs on the logical file names; the StoRM server can thus read (via ECAR) the ACLs on the logical filename corresponding to a given physical file and grant or deny access to the local files, depending on the permissions on the LFC. This provides users with a consistent view of the files in grid storage.

Data management innovation does not stop with StoRM. In order to make more transparent the use of grid stored data, EGRID developed the ELFI tool that allows access through the classic POSIX I/O software interface. ELFI is a FUSE filesystem implementation, so grid resources are seen through a local mount- point. This has two effects: first, all existing tools for file management work immediately without any modifications; and second, developers have not got to learn new APIs when porting applications to the grid. For example not only classic command line tools such as `cat` can be used, but also graphical file browsers such as Konqueror.

ELFI will be installed on all WNs of the farm, so applications will no longer need to explicitly run file transfer commands: the available mount point will let files to be seen as local. ELFI is also designed to communicate with StoRM: this will be very useful when installing the portal because although its APIs allow local file operations, SRM is not yet supported.

4. The EGRID portal

The main entrance point to the new EGRID infrastructure will be a web portal, long proved to be a very effective way for users to interact with an organisation’s information system. Grid services will therefore be integrated into the portal, where users will be able to find all they need; the graphical nature of the portal will also greatly help those not used to terse command line tools. The portal, moreover, will alleviate the EGRID team from the problem of software dissemination: there is no need to install anything on users’ workstations because any interaction will happen through the portal as displayed in a web browser.

The EGRID portal that is currently under development is based on P-grade [9] which seemed sufficiently sophisticated and mature as a starting point from which to meet the user requirements; in fact only small parts need to be enhanced.

In particular P-grade allows:

- Easier job management

The GUI allows the user to easily accomplish what can already be done through the command line, but since it is graphical the impact on the user is limited. For example simple selection boxes allow the user to choose between the Broker that automatically takes care of choosing a submission site, or a specific Computing Element. It is possible to customize selection lists with specific ranking parameters, so the user has only got to choose from a carefully crafted list of options which displays friendly names. The GUI is powerful enough to also allow manual specification of JDL parameters, for the more advanced users.

Job control and monitoring is fairly sophisticated. When using the command line it is very important to save the job identification string, because it is the only way the job output can be retrieved: its handling is cumbersome. When using the portal a specific display keeps track of all launched jobs and their status: job handling is transparent. The interface also allows jobs to be stopped, their output retrieved, etc.

P-grade supports MPI jobs and thanks to a specific API which can optionally be used in the user's application, a detailed graphical report about the communication/status of all processes involved on such jobs can be displayed in real time.

- Enhanced job usability

The presence of a workflow manager greatly boosts job processing: it is possible to graphically specify several jobs, and then define connections among them showing the flow of data from one to the other. The portal will take care of retrieving results from one job and feed them to the graphically connected ones. Monitoring of the workflow is done graphically in real time, showing status progress for each job.

- Grid resources browsing and file management

An embedded tool allows graphical browsing of the grid information system: the user has no need to learn text based LDAP search utilities, nor to have an external GUI for browsing. Similarly for grid file management in general: operations such as creation and deletion of files and directories, uploading of files to the grid, etc. Portal users will benefit from not having to deal with several separate tools, as they will only need to log in to the portal for complete and independent grid access and management.

To completely meet EGRID requirements some enhancements are needed: there is currently a collaboration with the P-grade team in order make the necessary new development and integration.

- Improved proxy management

In P-grade the private key of the user gets uploaded into the portal, which then sends it to the MyProxy server. For EGRID it would be better for the key to be uploaded directly from the

user workstation to the MyProxy server, without passing through the portal. This is needed to lower security risks, so a Java WebStart application was developed and seamlessly integrated into P-grade, by making minor modifications to the existing credentials portlet.

- Data management portlet that uses ELFI

P-grade does not support the SRM protocol, and does not support browsing of files present in the machine hosting the portal itself. ELFI allows access to grid disk resources in general and to those managed by StoRM in particular. A specific portlet was, therefore, written to browse and manipulate the file system present in the portal server itself. Since ELFI allows grid resources to be seen through local mount points it is easier to modify the portal for local operations rather than for some other grid protocol.

This new portlet allows manual transfer of files between different directories of the portal host, but since some of these directories are ELFI mount points then automatically a grid operation takes place behind the scenes. Overall, the portlet allows data movement between the user workstation, the machine hosting the portal, and the grid storage resources.

- File management and job submission interaction

During Workflow definition, all files needed by a single job get specified together with the output files that will be produced. Each file specification also requires the file location: P-grade currently allows only classic SEs, and the user workstation from which the portal was accessed. SRM is not supported, so similarly to the previous point what is required is a file location that is local to the portal server. By leveraging ELFI, access to StoRM or any other SRM server follows transparently. This is an important point, because P-grade sees a single job as a Workflow consisting of one job: input and output files must still be specified as described. There is an on going collaboration with P-grade developers to better define our requirements and study their feasibility.

EGRID also needs a special version of job submission closely related to Workflow jobs, so called *swarm jobs*. The application remains the same while the input data changes parametrically over several possible values; then a final job collects all results and carries out aggregate computations on them. P-grade currently allows manual job parameter specification, so an automatic mechanism is required. This feature is already present in the release schedule of P-grade, where it is called *parametric-jobs*.

We finally note that a possible drawback consists on having Java technology used extensively also on the client side. Applets and Java Web Start technology is used for certain operations, so the end user must have a Java Virtual Machine installed. Yet, given the ubiquitous nature of Java we don't consider it a strong liability.

References

- [1] <http://lcg.web.cern.ch/LCG/>
- [2] <http://eu-datagrid.web.cern.ch/eu-datagrid/>
- [3] <http://public.eu-egee.org/>

- [4] Angelo Leto, Ezio Corso, Riccardo Murri, Alessio Terpin, Cristian Zoicas, Stefano Cozzini. "EGRID Project: Experience Report on the Implementation of a Grid Infrastructure for the Analysis of Financial Data," itcc, pp. 268-273, International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I, 2005.
- [5] Arie Shoshani, Alexander Sim, and Junmin Gu. Grid Resource Management, chapter 20 "Storage Resource Manager "Essential Components for the Grid". Kluwer Academic Publishers, 2003.5
- [6] Riccardo Murri, Angelo Leto: "Sistema di utenti per realizzare l'accesso separato ai dati", <http://www.egrid.it/sysadm/rfc/1>
- [7] Riccardo Murri: "Struttura PFN per realizzare l'accesso separato ai dati: come organizzare la struttura delle directory ed assegnare permessi sullo SE assegnato a EGRID.", <http://www.egrid.it/sysadm/rfc/2>
- [8] <http://grid-it.cnaf.infn.it/storm/index.php>
- [9] <http://www.lpds.sztaki.hu/pgrade>