ISTITUTO NAZIONALE DI FISICA NUCLEARE

Progetto INFN-GRID

INFN/TC-02-30
18 Dicembre 2002

The INFN-GRID Testbed

Roberto Alfieri[1], Roberto Barbera[2], Patrizia Belluomo[2], Alessandro Cavalli[3], Roberto Cecchini[4], Andrea Chierici[3], Vincenzo Ciaschini[3], Luca Dell'Agnello[3], Flavia Donno[5], Enrico Ferro[6], Antonio Forte[7], Luciano Gaido[7], Antonia Ghiselli[3], Alberto Gianoli[8], Alessandro Italiano[3], Stefano Lusso[7], Marisa Luvisetto[9], Paolo Mastroserio[10], Mirco Mazzucato[11], Daniele Mura[12], Mario Reale[3], Livio Salconi[5], Marco Serra[13], Fabio Spataro[1], Francesco Taurino[10,14], Gennaro Tortone[10], Luca Vaccarossa[15], Marco Verlato[11], Giulia Vita Finzi[3]

1) *INFN-Sezione di Parma, Dipartimento di Fisica, Parco Area delle Scienze 7/A, I-43100 Parma, Italy*
2) *INFN-Sezione di Catania, Via S.Sofia 64, I-95123 Catania, Italy*
3) *INFN-CNAF, Viale Berti Pichat 6/2, I-00100 Bologna, Italy*
4) *INFN Sezione di Firenze, Largo E. Fermi 2, I-50125 Firenze, Italy*
5) *INFN-Sezione di Pisa, Via Livornese 1291, I-56010 San Piero a Grado (PI), Italy*
6) *INFN-Laboratori Nazionali diLegnaro, Via Romea 4, I-35020 Legnaro (PD), Italy*
7) *INFN-Sezione di Torino, Via Giuria 1, I-10125 Torino, Italy*
8) *INFN-Sezione di Ferrara, Via del Paradiso 12, I-44100 Ferrara, Italy*
9) *INFN-Sezione di Bologna, Via Irnerio 46, I-40126 Bologna, Italy*
10) *INFN-Sezione di Napoli, Complesso Universitario di Monte S. Angelo, Via Cintia, I-80126 Napoli, Italy*
11) *INFN, sezione di Padova, Via Marzolo 8, 35131 Padova - Italy*
12) *INFN-Sezione di Cagliari, Cittadella Universitaria di Monserrato, Strada provinciale per Sestu Km 0.700 Casella Postale 170, I-09042 Monserrato (Cagliari), Italy*
13) *INFN-Sezione di Roma1, Università degli Studi "La Sapienza", P.le A. Moro 5, I-00185 Roma, Italy*
14) *INFM UdR di Napoli, Complesso Universitario di Monte S. Angelo, Via Cintia, I-80126 Napoli, Italy*
15) *INFN-Sezione di Milano, Via Celoria 16, I-20133 Milano, Italy*

Abstract

The INFN-GRID project, in close collaboration with the EU-funded DataGrid project, is facing the challenge of implementing a wide-area computational and data grid for the needs of the INFN experimental activities. The testbed is the test infrastructure where the grid services have been implemented. In this document the goals, the actual implementation and the future evolution of the testbed are described.

Keywords: distributed computing; grid; testbed

Published by ***SIS–Pubblicazioni***
Laboratori Nazionali di Frascati

1 Introduction

As described in the INFN-GRID project proposal, the objectives of the INFN-GRID project [1] are to develop and deploy a computational and data Grid prototype capable to efficiently manage and provide effective usage of the large clusters and supercomputers located in the INFN sites connected through GARR-B [2], the Italian Academic and Research Network.
These geographically distributed resources, normally used by people at a single site, are integrated using the Grid technology to form a coherent high throughput computing facility transparently accessible by all INFN users.

The INFN-GRID testbed is integrated with the European testbed deployed by the DataGrid Project (EDG) [3], aiming at building a computational grid prototype on the basis of the users' requirements in the application fields of High Energy Physics [4], Earth Observations (EO) [5], and Bio-Informatics [6]. Both INFN-GRID and DataGrid are three years projects whose activities are structured in several Work Packages (WP). The goal of WP1-WP5 (also known as "Middleware Work Packages") is to develop the software components needed in the mentioned grid projects. This document describes the activities of Work Package 6, dealing with the INFN-GRID Testbed.
The scale of the computational, storage and networking capacity of the INFN-GRID prototype is determined by the needs of the LHC experiments [7], Virgo [8] and other experiments like CDF [9] and BaBar [10] that recently joined the INFN-GRID project. The activities of the next generation high energy physics experiments are driven by some important, and to a certain extent new, constraints: the computing solutions have to fit with many different applications, with a large amount of data (of the order of Petabytes), with thousands of nodes and huge user collaborations in a distributed environment. The challenge of the INFN-GRID project, in collaboration with DataGrid, DataTAG [11] and LHC Computing Grid (LCG) [12] projects, is to gradually set up growing testbeds to test the scalability and reliability of such distributed computing solutions in EU and USA.

As stated in the LCG project proposal, "the computing facility for LHC will be implemented as a global computational grid, with the goal of integrating large geographically distributed computing fabrics into a virtual computing environment". LCG "will be integrated with several European national computational grid activities (such as GridPP [13] in the United Kingdom and the INFN-GRID in Italy), and it will closely collaborate with other projects involved in advanced grid technology and high performance wide area networking".
LCG foresees to set up a production testbed for the first half of year 2003 while the INFN-GRID activity plan aims at setting up the INFN part of the future LCG testbed in late 2002.
As in many other projects developing distributed systems, in the grid projects it has been necessary to set up several testbed infrastructures. The aim is not only to check the grid middleware and to evaluate its functionalities and performances but also to understand the management problems arising from a in a wide area distributed system and to guarantee satisfactory functionality and reliability.

Some grid specific characteristics make the grid testbed activity fairly new:

- since the farms should be simultaneously used both by local and grid users it is fundamental to optimize the schedulings of jobs to allow both grid and local scheduling;
- the resource descriptions should be available to the grid scheduler in an efficient way through appropriate distributed Information Systems;
- the need for data replication implies an accurate evaluation of where and when replicating files in order to optimize the job execution; the network parameters should also be taken into account;
- computing resources belonging to different virtual organizations require fine-grained authorization mechanisms based on priorities when scheduling jobs for the execution and appropriate tools to manage the information about the members of the Virtual Organizations (VO).

This document describes the different aspects of the creation of a grid software release, the configuration problems of the specific grid services, the functionality verification of a single grid element as well as of all elements as a whole. This activity has been carried out through the setup of different types of testbeds: development, validation and production.
The solutions adopted for the management, control and monitoring of the grid system are also described as well as the authentication, authorization and support issues.

2 The Testbed: goals and characteristics

The main goal of the Testbed Work Package in the INFN-GRID and DataGrid projects is to set up a grid infrastructure in order to demonstrate that the DataGrid software components could be integrated into a functioning computational grid. This will allow the middleware work packages to evaluate the design and performance of their products and to facilitate interaction and feedback between the end-users and developers. The middleware validation process is performed using real applications supplied by the experiments involved in the projects.

The main basic functionalities needed in a grid environment are user authentication and authorization, resource and data management and monitoring of the availability and status of the grid resources.

2.1 Evaluation of the existing grid software

At the very beginning, since the Globus Toolkit [14] (release 1.3) seemed to provide such functionalities, it was decided to focus the preliminary INFN-GRID activities on the evaluation of the Globus "layered bag of services" on a nation-wide Testbed infrastructure constiting of small computing resources at several sites.

Various Globus services have been tested and evaluated: the security service (the Globus Security Infrastructure, GSI), the information service (the Grid Information Service, GIS) based on the Lightweight Directory Access Protocol (LDAP), the resource management service (the Globus Resource Allocation Manager, GRAM) and the data access and migration services (the Globus Access to Secondary Storage, GASS, and GlobusFtp).

The evaluation activity has been very useful to deeply understand the characteristics and the shortcomings of the Globus software. Feedback has been provided to the Globus Team as well as some information about the specific needs of the scientific communities involved in the INFN-GRID project. The results of the evaluation have already been reported in a document [15].

2.2 The first DataGrid Testbed

In order to test the suitability at the European level, after the evaluation of the Globus Toolkit functionalities, the first European Testbed infrastructure (the so-called Testbed 0) has been set up over several sites of the main EDG partners (CERN, PPARC, IN2P3, Nikhef and INFN). INFN joined the Testbed0 with resources located in 5 sites (CNAF/Bologna, Milano, Padova, Pisa, and Torino). The Globus software has been distributed for installation in a special toolkit [16] containing also the INFN customization (e.g. the support for the INFN Certificate Authority). The documentation, including also an installation manual, is available [17].

2.3 The test layouts for the DataGrid releases

As soon as the DataGrid middleware Work Packages started delivering their software, an Integration Team was formed with the aim to integrate the software components into a single DataGrid (EDG) release. The Integration Team consists of people from the Testbed Work Package (WP6), representatives from each of the middleware work packages, and observers from the application work packages (WP8-WP10).

The Testbed activities are continuous: every time the middleware Work Packages deliver their software components, these are integrated by the Integration Team and then a new EDG Release is produced, tested and finally tagged. After this new release has been deployed on the Testbed, it is validated by a small group of initial users, the Validation Team, and finally delivered for the distribution to the entire production grid.
Since it is essential, for the success of the grid projects, that the middleware satisfies the users' initial requirements it was decided that the validation had to be performed using real applications in a large scale environment. Some experiment independent people from WP8 are also part of the Validation Team where they play a fundamental role.

Two different issues are addressed by the test and validation activities:

- test of grid services when all the components are operative;
- evaluation of the "real" ability to run application software (from WP8-WP10) in the grid context ("middleware validation").

By using a testbed it is possible to find (and solve if necessary) middleware specific bugs, and the real interoperability among the grid components can also be verified.
Moreover the grid stability when a problem occurs to one (or more) sites in the infrastructure and the scalability issues when the number of sites (or nodes) increases can also be investigated.

The testbed infrastructure actually is not unique since there is the need for different and separate infrastructures devoted to different tasks. As a matter of fact, developers need a test infrastructure to run the middleware software, both for dedicated WP tests and for more general tests with all the grid components in place. In this context stability is not an issue and frequent services interruptions are possible. For this reason a special, dedicated, testbed infrastructure, the Development Testbed, has been set up.
On the contrary, the application group needs a "stable and robust grid" with all the services running to evaluate the grid performances when real applications are in execution. For this purpose a Validation Testbed where application software has been installed (and in particular HEP applications to perform Montecarlo production and batch analysis) has been deployed.

Three major EDG releases are foreseen before the end of the DataGrid Project. The timescale and characteristics of each EDG software release are described in the EDG Software Release Plan [18]. Intermediate (minor) releases are foreseen as well.

3 Testbed1

In order to test and validate the EDG software Release 1 the Testbed1 infrastructure has been created. This infrastructure has been set up with the participation of various sites geographically spread over different European countries.

A schematic overview of the EDG Release 1 software is represented in a 3 layers model (shown in fig. 1) including the *users* in the top layer (*local layer*), the *fabric* in the bottom layer and all the main *grid services* in the central layer.

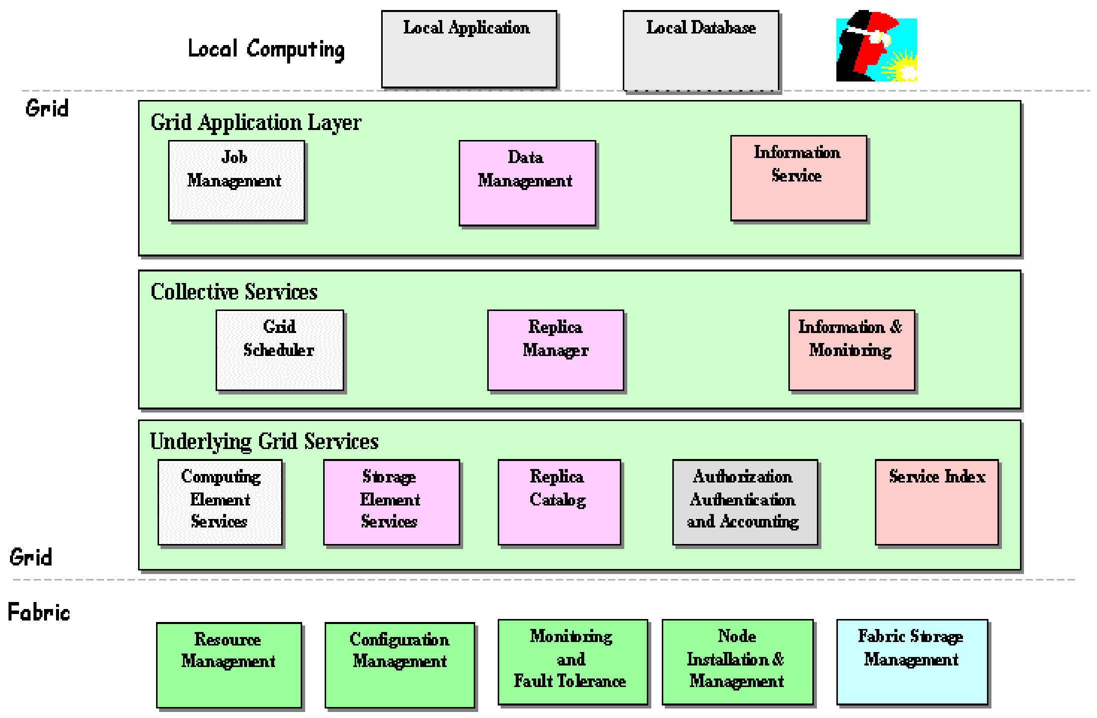


Fig. 1 - Schematic overview of EDG Release 1 software and services

The *fabric layer* represents all the physical infrastructures (pc, network, storage, etc.) and the software running on a computing farm. The *central layer* refers to the middleware, including all the services that provide "grid functionalities", connecting different farms/sites by means of the network. The *top level* represents the grid users, who submit jobs and/or move data to a computational grid for the execution.

The specific implementations of the fabric (or users) services are hidden to the Grid by means of dedicated middleware services, that we can call *underlying grid services* (for the fabric interface) and *grid application services* (for the user interface).

In this layered model it is possible to single out some software components (specific for the Release 1) that can be grouped in:

- *collective services*. The most important are: "Resource Broker", "Job Submission Service" and "Logging & Bookeeping Service" (developed by WP1), "Replica Catalog" and "Replica Manager" (WP2), "Grid Information Service and Monitoring" (WP3);
- *farm services*. They are: "Fabric Installation and Management System" (WP4), "Massive Storage Management System" (WP5);
- *authorization and monitoring services* (developed by the the security group and WP7)

The *farm services* are the tools used to install, configure and manage computing clusters or farms in the testbed infrastructure. In the farms the underlying grid services provide information to all the other middleware services. These usually describe the farm status (CPU availability, storage, etc) or data availability.

The farm services might be different for different sites or farms. On the contrary, the *collective services* run on a shared set of nodes, that all the grid nodes can contact. For example a job is submitted to the grid by means of the Job Submission Service that is running on the User Interface and on the nodes hosting the collective services. The Resource Broker performs the next step sending the job to a free resource after querying the Information Service to get information about the available grid resources. If the job needs to move data among different grid elements the Replica Manager Service is activated. The data location is managed by a Replica Catalog.

In the Testbed1 all the collective services are centralized i.e. they run on single grid nodes. In the next testbeds, based on the next EDG software releases, a distributed layout for the collective services is going to be implemented.

To simplify the installation and configuration of the EDG software release on the distributed resources some node types (grid elements) have been identified and a specific profile (list of rpm's and configuration instructions) has been created. Each grid element has a different role in the grid.

The grid elements defined so far are:

- User Interface (UI): the node used to access the grid and submit jobs;
- Resource broker (RB): the core component of the Workload Management system. Its duty is to find a resource matching the user's requirements;
- Logging and Bookkeeping element (LB): repository for the events occurred in the lifespan of a job;
- Information Index (II): caching information index, based on MDS-2, directly connected to the RB;

- Computing Element (CE): the gateway to the grid farm nodes. A job dispatched to a CE is then passed to a local scheduler for the execution;
- Worker Node (WN): a single node of a farm;
- Storage Element (SE): the node providing services to store, locate and replicate the data and to publish information about data availability. It is the interface to both disk and tape servers;
- Replica Catalog (RC): stores information about the physical files on all the SE's;

When the Integration Team delivered the first EDG software release (Release 1), the grid administrators (the so-called *site managers*) on eight European sites installed it on their computers creating the first DataGrid Testbed infrastructure.

INFN has contributed to the DataGrid Testbed with a subset of the INFN-GRID Testbed (4 sites: Catania, CNAF/Bologna, Padova and Torino). The Testbed infrastructure based on the DataGrid Release 1 (officially released in December 2001) is known as Testbed 1.

More than thirty software components have been delivered and integrated in Testbed 1.

The deployment of the Testbed 1 required a huge effort in testing the integrated software, checking the software dependencies and debugging the automatic installation tools (based on LCFG [19], a tool developed by the Edimburgh University) adopted to simplify the installation and management of the grid elements for the local administrators.

A detailed description of LCFG is reported in Chapter 4.

Concerning the specific INFN-GRID Testbed1 infrastructure, the INFN resources are distributed in small clusters made up of about 3-5 nodes at thirteen sites (Bari, Bologna, Cagliari, Ferrara, Genova, Lecce, Legnaro, Napoli, Parma, Pavia, Roma1, Roma3 and Trieste) and medium clusters (10-12 nodes) at six sites (Catania, CNAF/Bologna, Milano, Padova, Pisa, and Torino). As already mentioned four sites joined the DataGrid Testbed1 and therefore they were accessible through the CNAF and CERN Resource Brokers, while the other sites have been connected only to the INFN-GRID Testbed1 infrastructure and so they were accessible only through the INFN Resource Broker running at CNAF. The access to the grid resources is allowed through the User Interface that can connect to whatever Resource Broker in the Testbed. A User Interface has been configured at each site.
Initially, for Testbed1, the grid elements have been manually registered in the Information Index (an MDS with a flat structure and no automatic registration mechanism) used by the Resource Broker for both at CERN and CNAF since the sites and the grid resources were not too many. But as more sites and resources join the Testbed a dynamical resource registration mechanism is needed. In the next EDG release a fully operational MDS will be used. A more detailed description is reported in chapter 4.4.
Fig. 2 shows the Testbed1 layout.

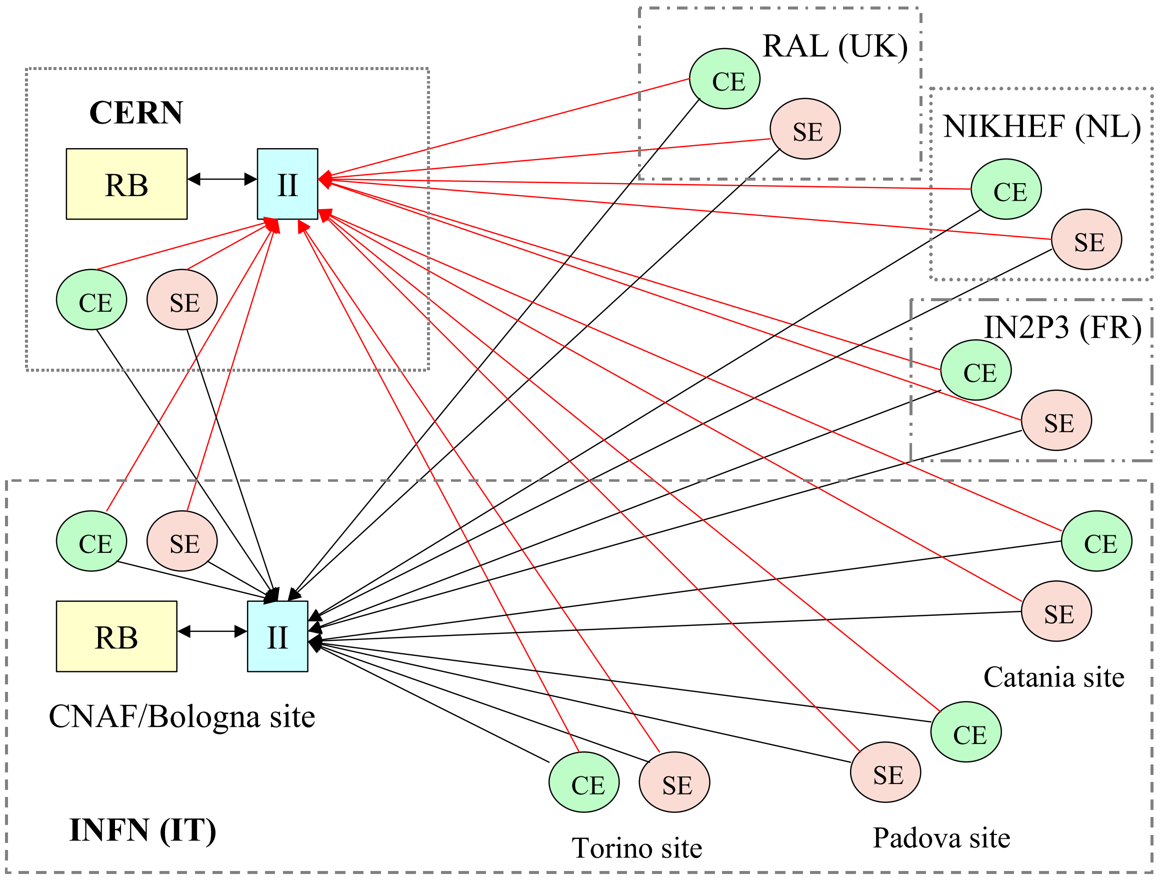


Fig. 2 - Testbed1 layout

The evaluation of the Release 1 and of the Testbed 1 environment [20] provided a valuable feedback to the development work packages in terms of bug fixes, usability, and direction for future improvements. A global evaluation of the testbed operation is described in [21]. As a further consequence of the validation activity the application work packages have also started the design of a common application layer based on the Testbed 1 experience [22].

4 Implementation and management of the INFN-GRID Testbed

The INFN sites are geographically distributed and the resources are continuously growing. In such a scenario an appropriate management needs to be carefully investigated.
As described in chapter 2, after the early evaluation of Globus Toolkit, some sites installed the first middleware prototype (EDG release) at the end of 2001 while the other sites joined the INFN-GRID Testbed during the following year.
To install and maintain such a large amount of resources, the existence of an automatic tool has been very useful. In the next section LCFG, the tool adopted for this purpose, is described in detail.
Since the DataGrid middleware is evolving through frequent major and minor releases, all sites have to migrate to the last stable release as soon as possible to let the experiments' people perform the validation of the software. At the same time the Testbed must be always running. For this reason the upgrade of the testbed has been managed through a precise migration plan that defines when and how each site has to migrate to the new EDG release in order to guarantee the complete operation of both old and new testbeds.
In principle the Testbed resources can be used by every grid users at any time. This is one of the most important characteristics of the grids, although the management of such infrastructure is quite complex. As already mentioned the grid resources can be devoted to various different tasks; these activities are concurrent and can involve many sites. To manage the "access" to the resources for a restricted group of grid user in a very simple way the concept of Virtual Organization (VO) was introduced. As explained in sec. 4.3, the authorization is granted via the so-called grid-mapfile, which is automatically updated. Users are grouped into VOs and their certificate's subjects are stored in an appropriate LDAP database. The access to the grid resources is controlled simply by enabling the appropriate VO's. For each specific grid activity the resource layout can be planned in order to satisfy the users' requirements.
Such an amount of resources, geographically dispersed, requires also an adequate monitoring system. Either the Grid Services (RB, II, RC, L&B, VO LDAP servers) or the distribuited Grid Elements (UI, SE, CE, WN) are continuously monitored. The adopted monitoring system is described in sec. 4.5. Accurate information about the resource usage is very useful to avoid overusage as well as underusage problems.
Although grids are quite new the number of people involved is continuously increasing; so support is becoming a very important issue. The support infrastructure that has been implemented within INFN-GRID is described in sec 4.6.

4.1 Installation and management of the grid resources

As already mentioned the installation of the Testbed was done by means of LCFG [19], a tool developed at the Department of Informatics of Edinburgh University for the automatic and centralized management of Linux systems. The EDG *ad interim* installation system developed by DataGrid WP4 [23] is based on LCFG because its architecture is extremely modular, flexible and customizable and matches the main EDG requirements.

The tool is based on a client-server architecture, where a server holds the configuration files for all the nodes to be installed. On each client node a daemon gets the node configuration from the server and updates the node's state so that it matches with the node's profile stored in the server. The update of the node's state is done by means of a collection of *objects*. Each object is a script that is responsible to completely manage a particular software component (e.g. inetd); thus there must be an object for each service or application that has to be automatically managed.

4.1.1 Installation from scratch

LCFG provides the capability of installing a node from scratch. It requires that the software packages and a basic Linux filesystem (*installroot*) are available via NFS; it also makes use of a DHCP server and a HTTP server. These services are normally installed on the LCFG server, but it is possible to use some already existing DHCP and NFS servers.
The installation steps are:

- a kernel is loaded via floppy disk
- the kernel is configured to get the network configuration via DHCP. This service is used only at this stage (the IP address is assigned statically)
- the kernel mounts the root filesystem via NFS; this is used only during the installation process (*installroot*). Normally this filesystem is exported by the LCFG server
- a script (*dcsrc*) that performs the installation process is started. It downloads the XML node's profile from the LCFG server. Once loaded, some LCFG objects whose tasks are to partition the disks, to install all the required packages and so on, are started.

After a reboot the client daemon and the set of objects are started to complete the configuration of the installed package; then the node becomes operational.

4.1.2 LCFG configuration update

The configuration of each node of a fabric is described in a set of text files (*source files*) located in an appropriate directory on the LCFG server. In the current LCFG prototype this information consists in simple name-value pairs. For the next future a new language (HLDL, High Level Definition Language [24]) has been developed to improve and simplify the definition of these data; currently it is under testing by WP4. Typically the configuration information is stored in various different text files containing a set of properties describing the node's overall configuration. It is possible to combine these files by inclusion; the information in these files can be extended and overrided in a similar way as it happens with Object Oriented (OO) languages.

The set of the configuration files is compiled by a tool (*mkxprof*) to produce an individual *profile* for each fabric node: the output of this process is a XML document containing the entire description of a node.

When the profiles are updated from the source files stored in the LCFG server, a notification is sent to the clients via the UDP protocol; the profiles are available to the client via a HTTP server. The client's update process is shown in fig. 3.

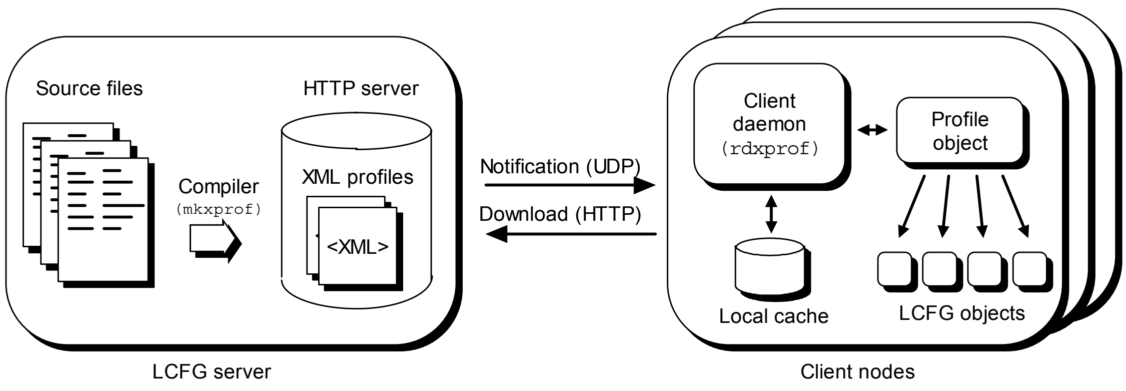


Fig 3 –

At boot time a daemon (*rdxprof*) is started on the client nodes managed by LCFG; when the configuration changes it receives the notification from the LCFG server, downloads the updated XML profile via HTTP, converts it from XML into a simpler file format, and stores it on a local disk cache. Then the LCFG objects update the client's configuration.
The rdxprof daemon is also able to detect which objects changed their configuration. Every object whose configuration has been changed on the server, gets the update of its own configuration files on the client (e.g. overwrites its /etc/inetd.conf), restarts the daemons and so on. Of course this behaviour is not applicable to the LCFG components that are active only at install time (e.g. the object that changes the disk partition table, etc.).

4.1.3 Software packages management

Currently the software packages are managed by LCFG via the RPM standard, the RedHat Package Management system [25]. As for other services (e.g. NFS server) there is an LCFG object responsible for this task. It gets as input a list of packages and a list of directories where these packages are present (normally these directories are imported via NFS); then it tries to sync the packages installed on the client with the list obtained from the server by adding, removing, upgrading and downgrading the packages as necessary.

4.1.4 Documentation and new components

An installation guide was produced to simplify the installation of a new testbed site [26]. A first standard set of LCFG source files needed to configure all the grid elements in a site has been produced; this has been possible thanks to the strong collaborative effort of many INFN testbed site managers. It took many validation tests to solve the dependency problems among the different software packages, due to their frequent changes and updates.
Another important contribution to the production of the *ad interim* installation system has been the development of two new LCFG objects (*obj-globus* and *obj-gdmp*). These objects

manage the configuration of some grid-related software, not supported by the already existing components:

- the Globus Toolkit configuration file
- the GRIS (Grid Resource Information Service)
- the UserInterface package delivered by WP1
- the InformationProviders package delivered by WP4
- the GDMP (Grid Data Mirroring Package)

Currently WP4 is deploying a new version of LCFG called LCFGng; it provides many improvements to the existing components and supports Linux RedHat 6.2 and 7.x; in addiction it is possible to install the grid elements via network and get some information about the node status through a web interface.

4.2 Testing the grid elements and services

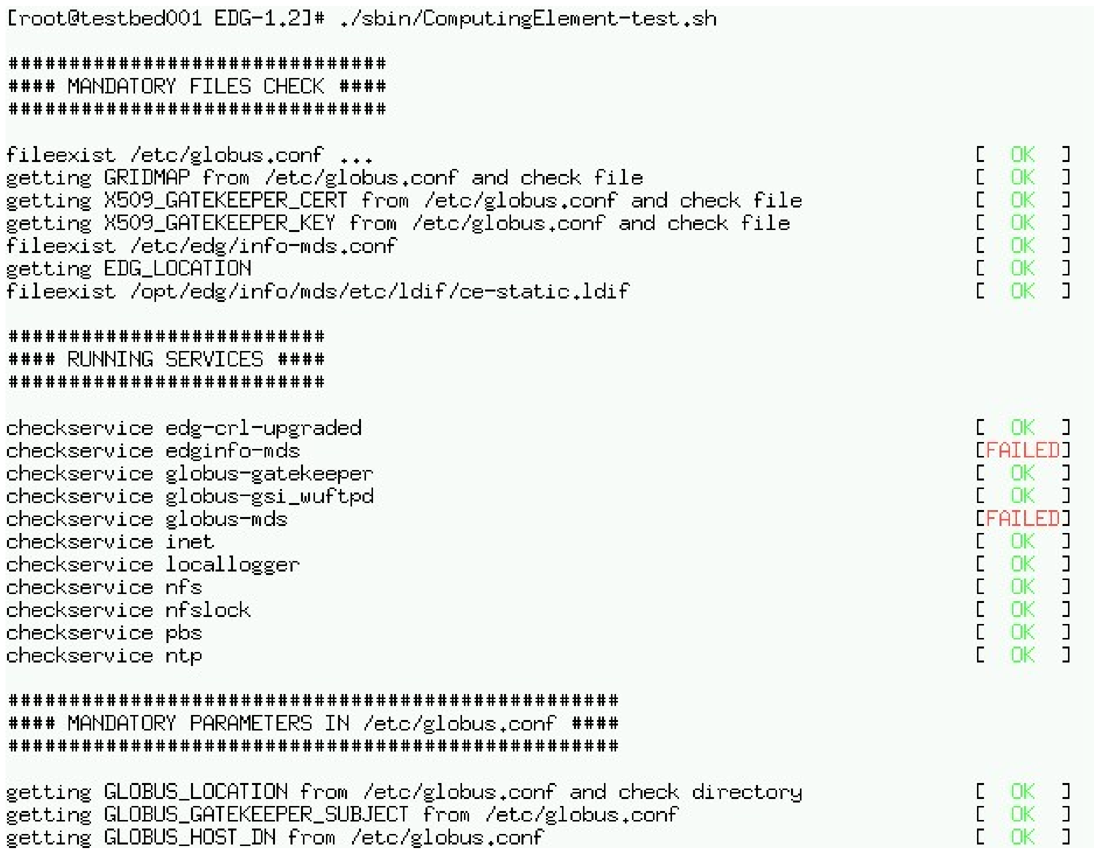
A thorough test of the grid testbed is of utmost importance in all phases of the wide deployment and use of a grid middleware like the one delivered within the DataGrid Project. In fact, detailed tests must be performed in order to check at least that:

1) all the grid-elements at all sites are configured in the same way for all the supported Virtual Organizations;
2) user's jobs can be successfully submitted to all Computing Elements which the user is allowed to submit jobs to through the appropriate Resource Brokers;
3) the user's job environment is properly defined at each worker node of each Computing Element not only at run-time but also at compilation and building times.

On top of the evident complication implied in points 1)-3), there is also the need of automatic tools to test both the middleware and the applications deployed on the testbed due to the usually limited human resources devoted to management tasks and to the large number of sites.
In order to cope with the stringent needs described, the INFN-GRID project has recently formed a testing working group with the goal of setting up a series of procedures (basically shell scripts) to automatically test the various grid elements at a given site and the basic middleware services.
The first version of this test suite has been released and it is available at a CVS server [27].
Using these scripts the site managers can:

a) verify, for each grid element (UI, WN, CE, SE, RB, RC, etc.), the correct installation and functionality;
b) verify operative interfaces between grid elements;
c) monitor testbed's element performances.

The test suite is composed by a set of core functions which are independent of the particular middleware release so it can be used with all the next software releases without any changes in the code.
The system administrators can run the tests interactively, with the possibility to exit the procedure and fix the problems at their first occurrence, or in batch mode where the results of the tests are written in a detailed report file. In both cases, at the end of each testing phase, a score (number of tests passed/ total number of tests) is returned and a threshold can be set on this score to define the level of acceptance of the test.
If the result of the test is below the threshold, the system manager is asked to fix immediately the problem(s) or contact the testbed support.
If the threshold is passed but not with full score, the system manager is advised in the actions to take.
An example of the output provided by the automatic tool is shown in figure 4.

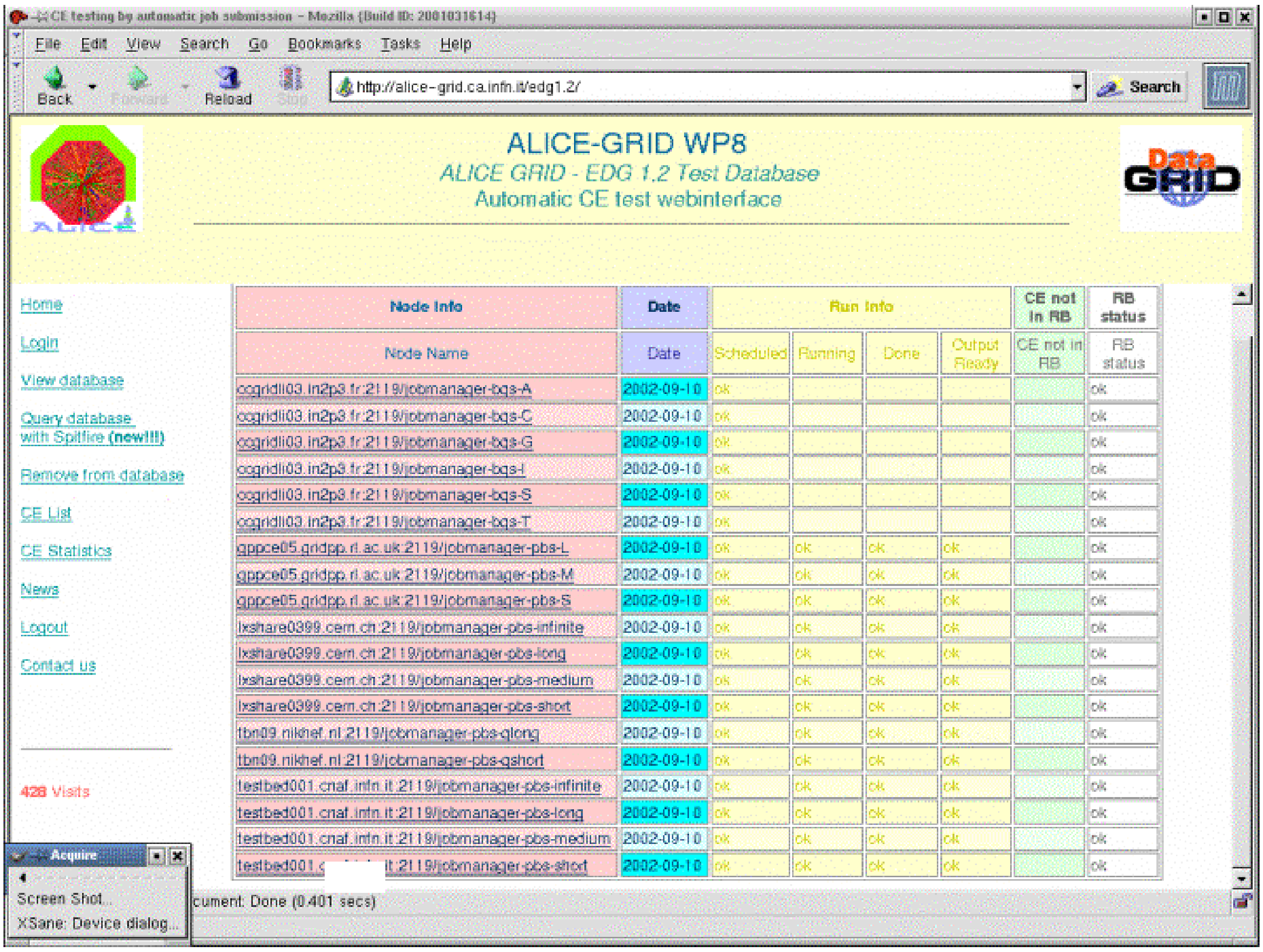


```
[root@testbed001 EDG-1.2]# ./sbin/ComputingElement-test.sh

################################
#### MANDATORY FILES CHECK ####
################################

fileexist /etc/globus.conf ...                                                    [  OK  ]
getting GRIDMAP from /etc/globus.conf and check file                              [  OK  ]
getting X509_GATEKEEPER_CERT from /etc/globus.conf and check file                 [  OK  ]
getting X509_GATEKEEPER_KEY from /etc/globus.conf and check file                  [  OK  ]
fileexist /etc/edg/info-mds.conf                                                  [  OK  ]
getting EDG_LOCATION                                                              [  OK  ]
fileexist /opt/edg/info/mds/etc/ldif/ce-static.ldif                               [  OK  ]

##########################
#### RUNNING SERVICES ####
##########################

checkservice edg-crl-upgraded                                                     [  OK  ]
checkservice edginfo-mds                                                          [FAILED]
checkservice globus-gatekeeper                                                    [  OK  ]
checkservice globus-gsi_wuftpd                                                    [  OK  ]
checkservice globus-mds                                                           [FAILED]
checkservice inet                                                                 [  OK  ]
checkservice locallogger                                                          [  OK  ]
checkservice nfs                                                                  [  OK  ]
checkservice nfslock                                                              [  OK  ]
checkservice pbs                                                                  [  OK  ]
checkservice ntp                                                                  [  OK  ]

##################################################
#### MANDATORY PARAMETERS IN /etc/globus.conf ####
##################################################

getting GLOBUS_LOCATION from /etc/globus.conf and check directory                 [  OK  ]
getting GLOBUS_GATEKEEPER_SUBJECT from /etc/globus.conf                           [  OK  ]
getting GLOBUS_HOST_DN from /etc/globus.conf                                      [  OK  ]
```

Figure 4 – Output of the automatic test procedure.

The tools developed by the INFN-GRID testing group contribute to the effort on testing carried out by the DataGrid TSTG group [28].
After the successful completion of the installation and configuration test suite, reference jobs have been submitted to the whole grid. The results of the tests can also be stored in a relational database and inspected via a secure web interface. This activity was originally

developed within the grid activities of the ALICE Experiment [29] and will soon be extended to the general tests of the DataGrid testbed. The status of various ALICE reference jobs, run on a daily cron-job basis on the DataGrid testbed, is recorded in a MySQL database together with the information on the job submission time, the Resource Broker and the Computing Element the job has been submitted to. Users can then login to the database web interface using both normal http connection and secure connection, select a Resource Broker and a time interval and inspect the statistics of the tests. The results of the tests can be displayed per Computing Element, per Resource Broker (figure 5), and globally (figure 6).


CE testing by automatic job submission – Mozilla {Build ID: 2001031614}
File Edit View Search Go Bookmarks Tasks Help
Back Forward Reload Stop
http://alice-grid.ca.infn.it/edg1.2/
Search
Data GRID
Home
Login
View database
Query database with Spitfire (new!!!)
Remove from database
CE List
CE Statistics
News
Logout
Contact us
428 Visits
Acquire
Screen Shot...
XSane: Device dialog...
cument: Done (0.401 secs)

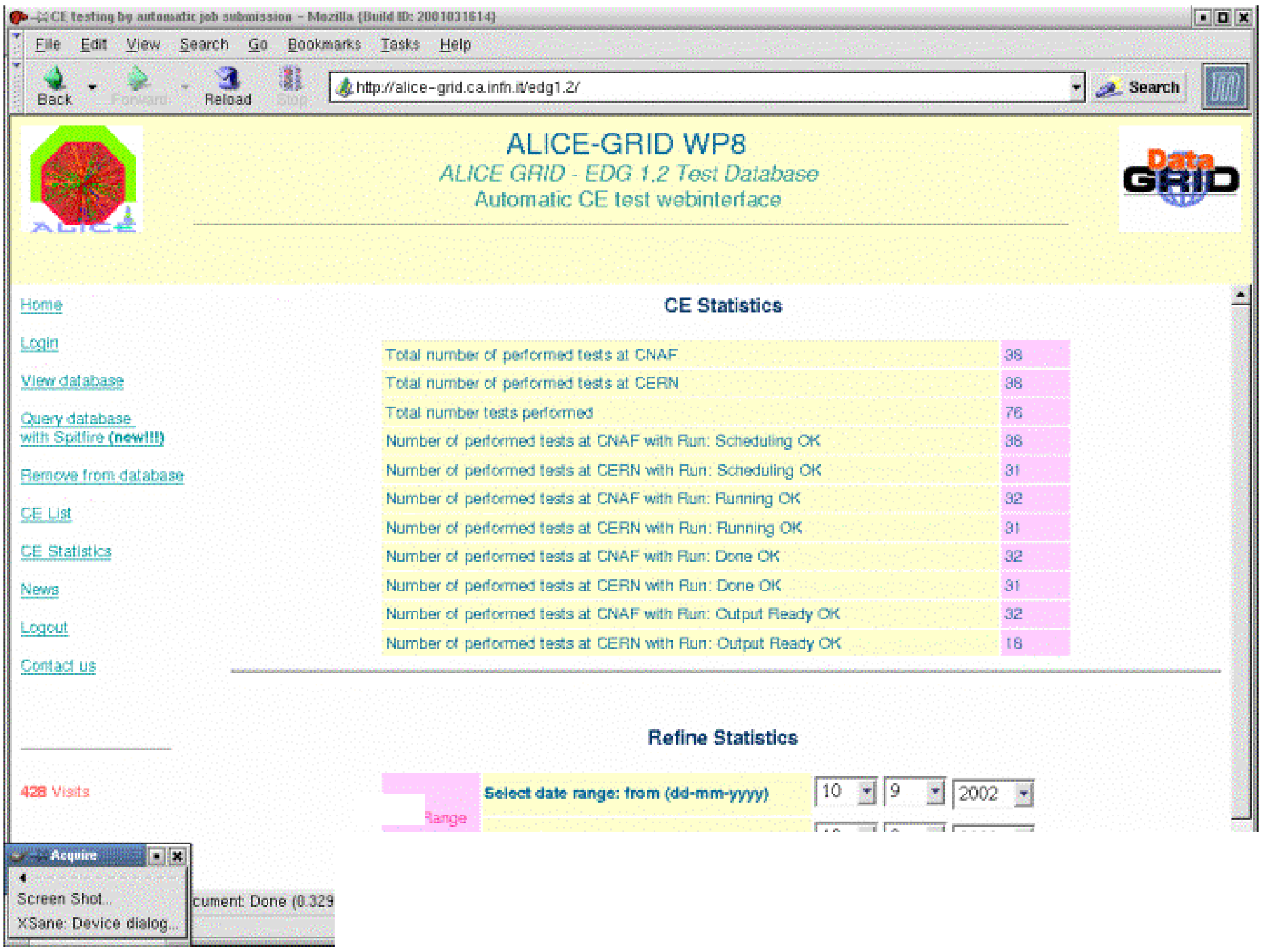
ALICE-GRID WP8

ALICE GRID - EDG 1.2 Test Database

Automatic CE test webinterface

Node Info	Date	Run Info				CE not in RB	RB status
Node Name	Date	Scheduled	Running	Done	Output Ready	CE not in RB	RB status
ccgridli03.in2p3.fr:2119/jobmanager-bqs-A	2002-09-10	ok					ok
ccgridli03.in2p3.fr:2119/jobmanager-bqs-C	2002-09-10	ok					ok
ccgridli03.in2p3.fr:2119/jobmanager-bqs-G	2002-09-10	ok					ok
ccgridli03.in2p3.fr:2119/jobmanager-bqs-I	2002-09-10	ok					ok
ccgridli03.in2p3.fr:2119/jobmanager-bqs-S	2002-09-10	ok					ok
ccgridli03.in2p3.fr:2119/jobmanager-bqs-T	2002-09-10	ok					ok
gppce05.gridpp.rl.ac.uk:2119/jobmanager-pbs-L	2002-09-10	ok	ok	ok	ok		ok
gppce05.gridpp.rl.ac.uk:2119/jobmanager-pbs-M	2002-09-10	ok	ok	ok	ok		ok
gppce05.gridpp.rl.ac.uk:2119/jobmanager-pbs-S	2002-09-10	ok	ok	ok	ok		ok
lxshare0399.cern.ch:2119/jobmanager-pbs-infinite	2002-09-10	ok	ok	ok	ok		ok
lxshare0399.cern.ch:2119/jobmanager-pbs-long	2002-09-10	ok	ok	ok	ok		ok
lxshare0399.cern.ch:2119/jobmanager-pbs-medium	2002-09-10	ok	ok	ok	ok		ok
lxshare0399.cern.ch:2119/jobmanager-pbs-short	2002-09-10	ok	ok	ok	ok		ok
tbn09.nikhef.nl:2119/jobmanager-pbs-qlong	2002-09-10	ok	ok	ok	ok		ok
tbn09.nikhef.nl:2119/jobmanager-pbs-qshort	2002-09-10	ok	ok	ok	ok		ok
testbed001.cnaf.infn.it:2119/jobmanager-pbs-infinite	2002-09-10	ok	ok	ok	ok		ok
testbed001.cnaf.infn.it:2119/jobmanager-pbs-long	2002-09-10	ok	ok	ok	ok		ok
testbed001.cnaf.infn.it:2119/jobmanager-pbs-medium	2002-09-10	ok	ok	ok	ok		ok
testbed001.c [illegible] it:2119/jobmanager-pbs-short	2002-09-10	ok	ok	ok	ok		ok

Figure 5 – Global view of the results for all Computing Elements attached to a given Resource Broker.


CE testing by automatic job submission – Mozilla {Build ID: 2001031614}
File Edit View Search Go Bookmarks Tasks Help
Back Forward Reload Stop
http://alice-grid.ca.infn.it/edg1.2/
Search
ALICE-GRID WP8
ALICE GRID - EDG 1.2 Test Database
Automatic CE test webinterface
Home
Login
View database
Query database with Spitfire (new!!!)
Remove from database
CE List
CE Statistics
News
Logout
Contact us
428 Visits
Acquire
Screen Shot..
XSane: Device dialog...
cument: Done (0.329

CE Statistics

Total number of performed tests at CNAF	38
Total number of performed tests at CERN	38
Total number tests performed	76
Number of performed tests at CNAF with Run: Scheduling OK	38
Number of performed tests at CERN with Run: Scheduling OK	31
Number of performed tests at CNAF with Run: Running OK	32
Number of performed tests at CERN with Run: Running OK	31
Number of performed tests at CNAF with Run: Done OK	32
Number of performed tests at CERN with Run: Done OK	31
Number of performed tests at CNAF with Run: Output Ready OK	32
Number of performed tests at CERN with Run: Output Ready OK	18

Refine Statistics

Select date range: from (dd-mm-yyyy) 10 9 2002

Figure 6 – Global statistics summary.

The system can (and will) be generalized to send different test jobs for the various Virtual Organizations and/or even more refined procedures to check the interface between the various grid-elements available on the testbed.

4.3 Resource access

The Authentication and Authorization method adopted by the INFN-GRID Project and by the European DataGrid Project is based on X.509 Certificates and *grid-mapfiles* [30].
Each INFN-GRID/EDG user owns an Identity Certificate issued by a Certification Authority (CA). When (s)he submits a request to a server, (s)he presents a certificate “derived” from his/her Identity Certificate (*proxy certificate*). The server consults a local

table, named *grid-mapfile*, for the mapping between the user certificate (the Subject field) and local usernames.
Each user belongs to a Virtual Organization (VO) [31]. A VO is a collection of individuals and institutions that are defined according to a set of resource sharing rules. A typical VO is an High Energy Physics (HEP) experiment.
In the first phase of the Testbed, named Testbed0 (TB0), the *grid-mapfile* was generated manually by the local administrator of the resources, according to the VO's agreements. That architecture doesn't scale: as the number of users increases, the manual management of the *grid-mapfile* becomes very difficult.
A VO-based central coordination is necessary to manage the information regarding the relationship between the user and the VO which (s)he belongs to, like group membership, user's roles and any other useful authorization information that has to be sent to the local sites at the job submission stage.
For this purpose the INFN has created a Working Group aimed at the definition, assembly, testing and delivering of a new VO Authorization Mechanism. The Working Group has implemented a system [32] for the automatic generation of the *grid-mapfile's* from the information stored in the VO servers. The local site managers have the complete control of the access to their resources.
This system has been adopted by the INFN-GRID and EDG Testbed1.

4.3.1 The INFN Certification Authority

A Certification Authority was set up by INFN in 1998 (INFN CA). Up to now it has issued about 700 user and server certificates.
To obtain a certificate the INFN users have to fill a WEB form [33] using a standard Web browser. A certificate for a server must be required by the server manager by sending an e-mail signed by his/her personal certificate[2]. The certificate request can be generated using the OpenSSL toolkit and the configuration file provided by the INFN CA. The issued certificates are stored in the INFN CA LDAP server and published on the web. INFN CA satisfies the requirements specified by the EDG Testbed.

4.3.2 Virtual Organization Directories

Authorization is the granting or denial of permission to carry out a given action. In both INFN-GRID and DataGrid projects the Authorization information for a Virtual Organization is kept in dedicated repositories and used by the local site managers to generate the *grid-mapfiles*. The VO repositories are implemented as LDAP servers that must contain at least two branches: a '**People**' branch and one or more 'group' branches.
All the users belonging to the VO are listed under the 'ou=People' branch. Each user entry contains at least:

- the URI of the certificate on the CA LDAP server;
- the subject of the user's certificate (to speed up the generation of the grid-mapfile and reduce the network load).

[2] the requests must be addressed to infn-ca@fi.infn.it

Each group entry consists in a list of pointers to a subset of the entries in the ou=People branch.
The INFN-GRID tree layout, already in use, is shown in figure 7.

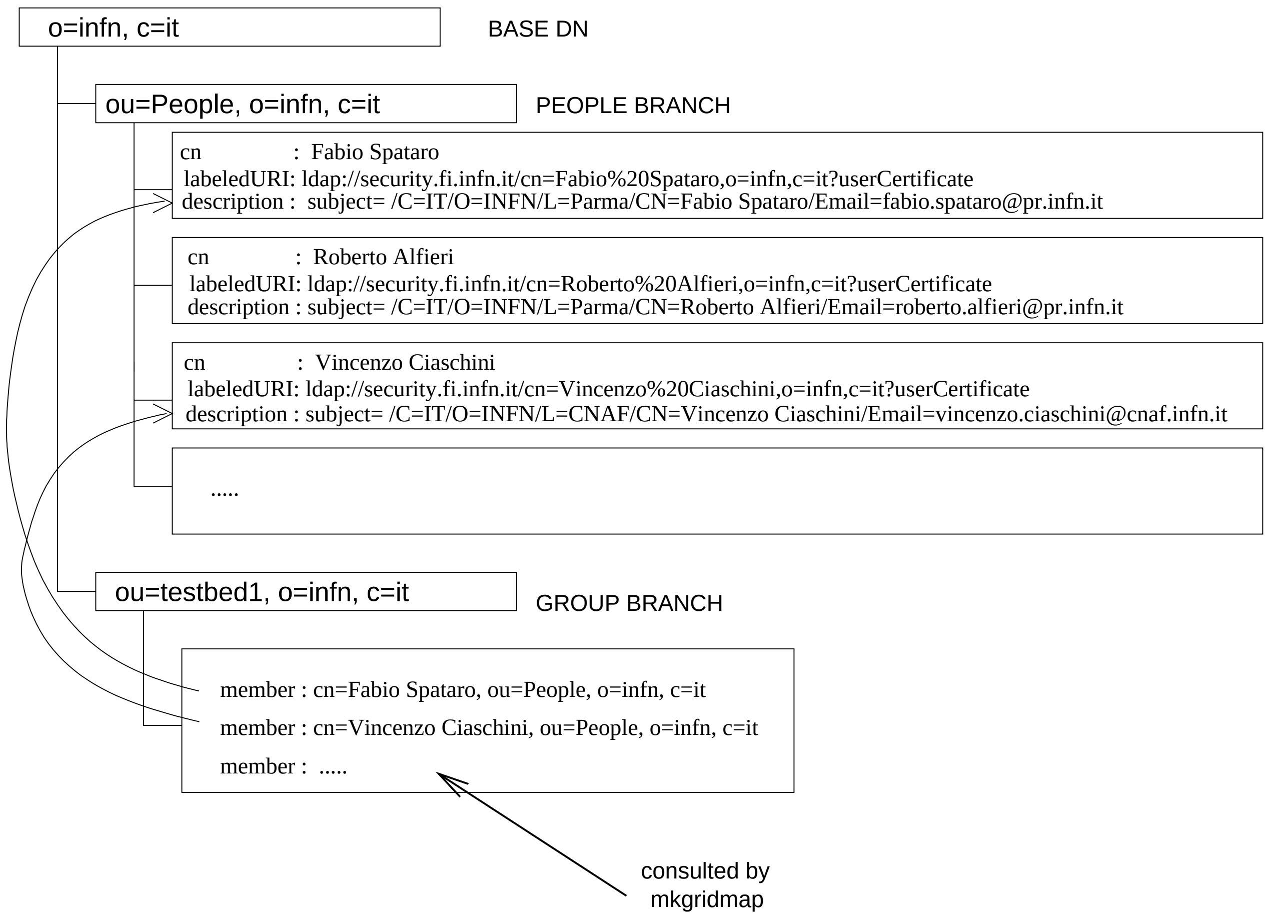

o=infn, c=it
BASE DN
ou=People, o=infn, c=it
PEOPLE BRANCH
cn : Fabio Spataro
labeledURI: ldap://security.fi.infn.it/cn=Fabio%20Spataro,o=infn,c=it?userCertificate
description : subject= /C=IT/O=INFN/L=Parma/CN=Fabio Spataro/Email=fabio.spataro@pr.infn.it
cn : Roberto Alfieri
labeledURI: ldap://security.fi.infn.it/cn=Roberto%20Alfieri,o=infn,c=it?userCertificate
description : subject= /C=IT/O=INFN/L=Parma/CN=Roberto Alfieri/Email=roberto.alfieri@pr.infn.it
cn : Vincenzo Ciaschini
labeledURI: ldap://security.fi.infn.it/cn=Vincenzo%20Ciaschini,o=infn,c=it?userCertificate
description : subject= /C=IT/O=INFN/L=CNAF/CN=Vincenzo Ciaschini/Email=vincenzo.ciaschini@cnaf.infn.it
.....
ou=testbed1, o=infn, c=it
GROUP BRANCH
member : cn=Fabio Spataro, ou=People, o=infn, c=it
member : cn=Vincenzo Ciaschini, ou=People, o=infn, c=it
member : .....
consulted by mkgridmap

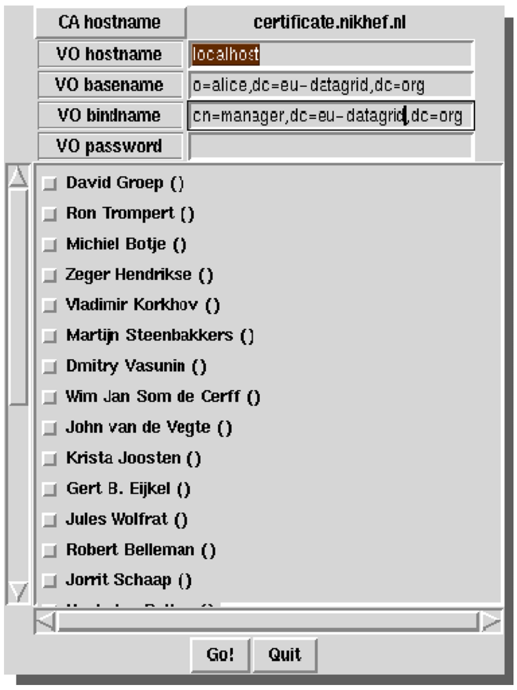
Figure 7 - INFN-GRID VO Architecture

The VO LDAP server is configured by the VO administrator. (s)he may enable SSL access, thus allowing mutual authentication between the grid resource generating the *grid-mapfile* and the server (for Testbed1 this authentication is not usually carried out.)
A set of graphical utilities that allow the VO administrator to populate the VO directory is available:

- vop.pl: the VO manager can select from the CA LDAP servers the proper users to be inserted in the People branch;
- group.pl: using this script the VO manager can browse and select the users in the People branch and insert them into a group;
- chkusers.pl: this script checks all user certificates in the VO Directory against

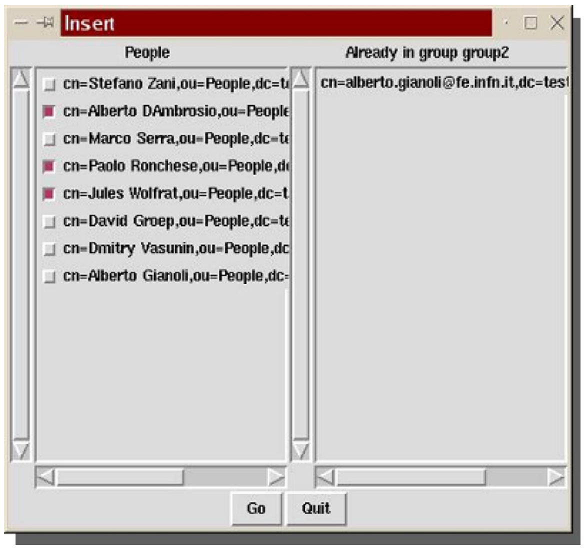
the CA LDAP server, looking for expired or revoked certificates. It must be executed periodically.

The vop.pl and group.pl graphical interfaces are shown in fig. 8.



CA hostname	certificate.nikhef.nl
VO hostname	localhost
VO basename	o=alice,dc=eu-datagrid,dc=org
VO bindname	cn=manager,dc=eu-datagrid,dc=org
VO password	

☐ David Groep ()
☐ Ron Trompert ()
☐ Michiel Botje ()
☐ Zeger Hendrikse ()
☐ Vladimir Korkhov ()
☐ Martijn Steenbakkers ()
☐ Dmitry Vasunin ()
☐ Wim Jan Som de Cerff ()
☐ John van de Vegte ()
☐ Krista Joosten ()
☐ Gert B. Eijkel ()
☐ Jules Wolfrat ()
☐ Robert Belleman ()
☐ Jorrit Schaap ()



Insert

People	Already in group group2
☐ cn=Stefano Zani,ou=People,dc=t	cn=alberto.gianoli@fe.infn.it,dc=test
☑ cn=Alberto DAmbrosio,ou=People	
☐ cn=Marco Serra,ou=People,dc=te	
☑ cn=Paolo Ronchese,ou=People,d	
☑ cn=Jules Wolfrat,ou=People,dc=t	
☐ cn=David Groep,ou=People,dc=te	
☐ cn=Dmitry Vasunin,ou=People,dc	
☐ cn=Alberto Gianoli,ou=People,dc=	

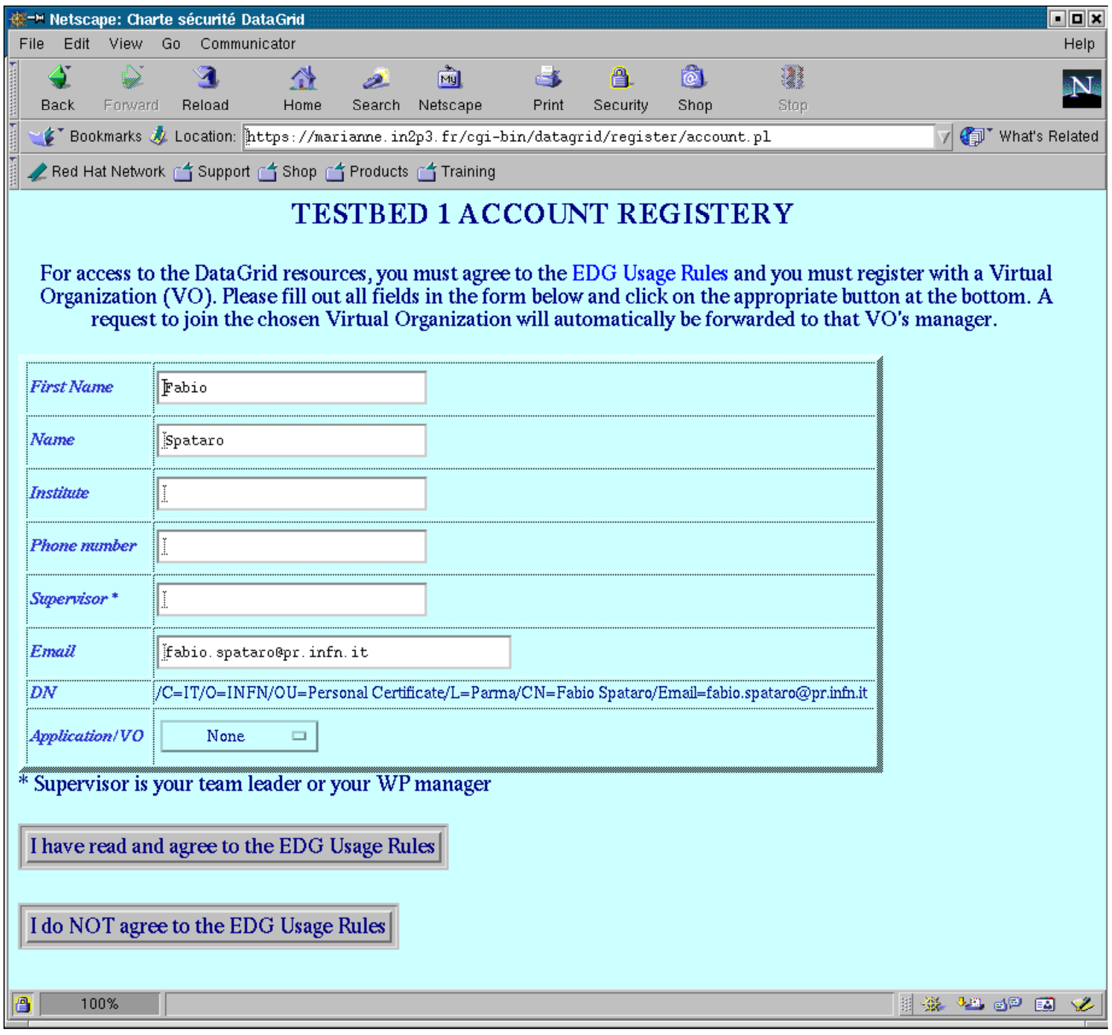
Fig 8: vop.pl and group.pl utilities

4.3.3 Web based VO Registration

The "*EDG Usage Guidelines*" document [34] describes the conditions that must be agreed upon in order to use the Testbed. Users have to sign the EDG Usage Rules to get access to the Testbed resources. This is achieved by means of a web-based tool: when a new user signs the Rules, his/her certificate subject is stored in an appropriate "Authorization Directory" in an LDAP server. The procedure that is used to generate the *grid-mapfile* (see next section) checks for the existence of the user's certificate subject in the "Authorization Directory".
Since a unique authorization directory for the whole Testbed would be quite impossible to manage (an agreement between a VO and its members is needed) as well as a single point of failure, a VO-based approach has been choosen.
Users are inserted in their VO Directory if they sign the Usage Rules through the Web form and after the VO manager's approval. In this way the presence of a user in a VO Directory implies the VO authorization and no other checks are needed. When registering

users have to provide their X.509 certificate (that guarantees their identity) and some other personal information such as phone numbers and e-mail addresses. On the server side a script checks against the existence of the user in the LDAP tree and then sends the request for the user's insertion in the VO repository to the VO manager.
The actual implementation is based on standard software (Apache, mod_ssl, PHP and MySQL) with special attention to security features. For example all the transactions are encrypted and the VO manager's password is useless without the VO manager's certificate.

The Web form to register a user's certificate in the VO LDAP servers for ALICE, ATLAS, CMS, LHCb, Earth Observation, Biomedical applications, Integration Team and EDG WP6 is accessible through the EDG Testbed web (fig. 9) while the form for INFN-GRID, VIRGO, DataTAG and CDF is accessible through the INFN-GRID Testbed web page (fig. 10).


Netscape: Charte sécurité DataGrid
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: https://marianne.in2p3.fr/cgi-bin/datagrid/register/account.pl What's Related
Red Hat Network Support Shop Products Training

TESTBED 1 ACCOUNT REGISTERY

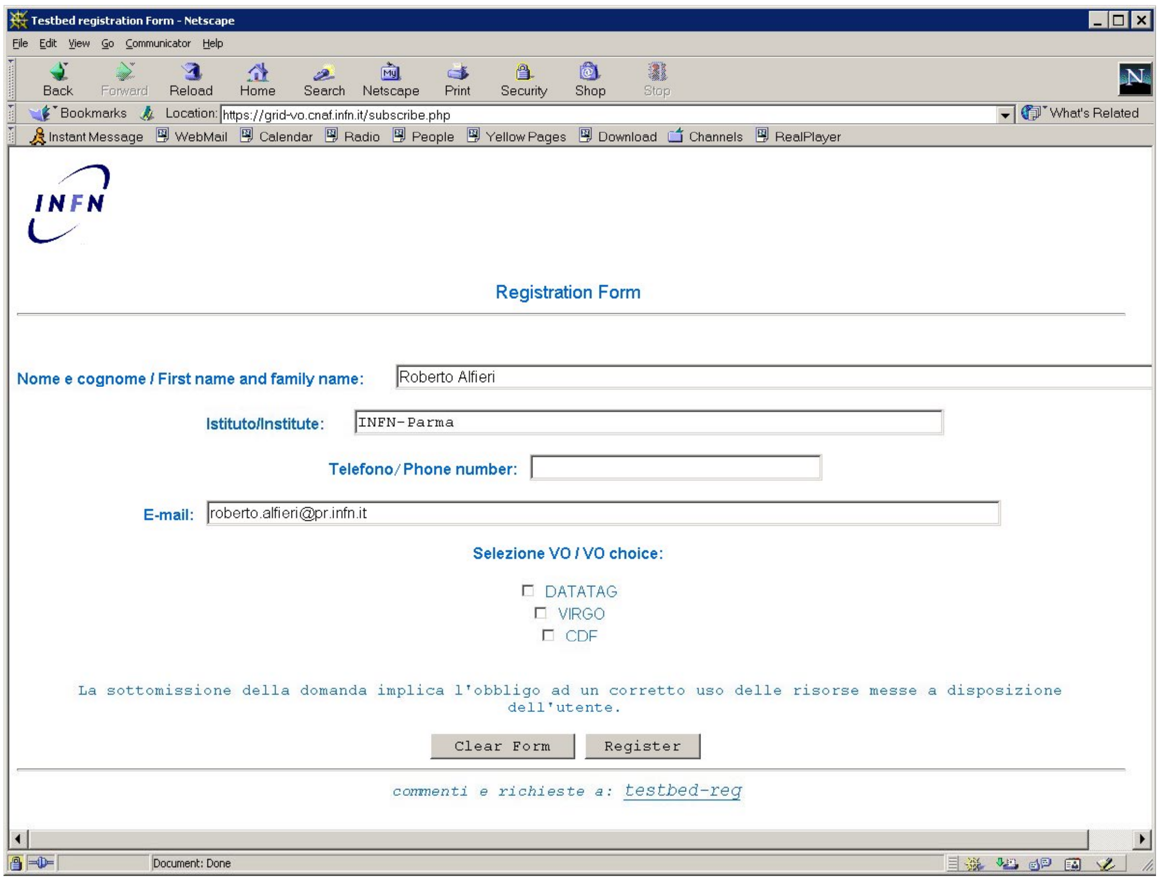
For access to the DataGrid resources, you must agree to the EDG Usage Rules and you must register with a Virtual Organization (VO). Please fill out all fields in the form below and click on the appropriate button at the bottom. A request to join the chosen Virtual Organization will automatically be forwarded to that VO's manager.

First Name	Fabio
Name	Spataro
Institute	
Phone number	
*Supervisor **	
Email	fabio.spataro@pr.infn.it
DN	/C=IT/O=INFN/OU=Personal Certificate/L=Parma/CN=Fabio Spataro/Email=fabio.spataro@pr.infn.it
Application/VO	None

* Supervisor is your team leader or your WP manager

I have read and agree to the EDG Usage Rules
I do NOT agree to the EDG Usage Rules
100%

Fig. 9: EDG user registration form


Testbed registration Form - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: https://grid-vo.cnaf.infn.it/subscribe.php What's Related
Instant Message WebMail Calendar Radio People Yellow Pages Download Channels RealPlayer
INFN
Document: Done

Registration Form

Nome e cognome / First name and family name:	Roberto Alfieri
Istituto/Institute:	INFN-Parma
Telefono/Phone number:	
E-mail:	roberto.alfieri@pr.infn.it

Selezione VO / VO choice:

☐ DATATAG
☐ VIRGO
☐ CDF

La sottomissione della domanda implica l'obbligo ad un corretto uso delle risorse messe a disposizione dell'utente.

Clear Form   Register

commenti e richieste a: testbed-req

Figure 10 – INFN-GRID user registration form

4.3.4 Grid-mapfile generation: mkgridmap

On the client side the *grid-mapfile* is automatically generated from an highly customizable perl script named *mkgridmap*. This script produces, on the standard-output, a *grid-mapfile* from the entries in the VO Directories, according to the directives specified in a configuration file (*mkgridmap.conf)*.
On each grid resource (Computing and Storage Elements) the local site managers should run *mkgridmap* at appropriate time intervals. This task is performed by the EDG daemon *edg-gridmafile-upgrade.*
The mapping between certificate subjects and local accounts is customizable by the local site managers by means of an external program (*subject2user*) or by the *gridmapdir* patch [35] to the Globus code.
The configuration file allows several directives for customization purposes.
The most important directive is

```
group <group_URI> [<lcluser>]
```

The group_URI argument refers to group servers. Each group belongs to a VO. By performing a search on these servers, *mkgridmap* retrieves the informations needed during the *grid-mapfile* generation process. The optional parameter <*lcluser*> is the local user-name associated to each member of the group.

It is possible to specify a default value for "lcuser" with the directive

    default_lcluser <*lcluser*>

If no "lcuser" is specified in the group directive and if the default_lcluser has not been set, then "lcuser" defaults to ".".

The directive:

    allow|deny <*pattern*>

can be repeated several times in order to create an access control list that manages the insertion of the certificate_subject - local_user pairs into the *grid-mapfile*. The <*pattern*> argument may contain wildcards (*). The pattern matching is done on the user certificate subject and the access control process stops after the first match. If at least an *allow* directive is specified then a "*deny **" directive is implicit at the end of the access control list. Otherwise an "*allow **" directive is implicit.

The directive:

    gmf_local <*filename*>

is useful to source entries from one or more pre-existent external files.

Another important directive is:

    auth <auth_URI>

where the auth_URI argument refers to authorization servers. By performing a search on these servers, *mkgridmap* creates a list of authorized users. In this case only the certificate subjects of authorized users will be inserted in the *grid-mapfile*.
It is possible to specify special user-names as <*lcluser*> in order to customize the mapping between the user's certificate subject and the local user-name. If the "AUTO" value is used then an external program (*subject2user)* will be invoked. This program "generates" the local accounts that will be associated to the certificate subject in the grid-mapfile by means of the user's name.
For instance, the generated local account tha will be generated for the user Roberto Alfieri will be "ralfieri". This feature is useful expecially when it is important that the same local account be used at each site.
If the <*lcluser*> is a string whoe first character is a dot (e.g. "*.infngrid"*) then the *gridmapdir* patch will be activated. In this case a pool of accounts (e.g. infngrid001 to infngrid0n) will be used.

4.3.5 Gridmapdir support: mkgridpool

gridmapdir [35] is a patch to the Globus code, developed by Andrew McNab (PPARC, UK), that enables a dynamic allocation of a pool of users when, in the *grid-mapfile*, the local username associated to a given certificate subject starts with a dot (e.g. *.infngrid*). This mechanism avoids the need for the creation of user accounts for every potential user. When a user sends a job to a grid resource for execution a local account is leased to him/her for the entire duration of the job and then it will be released. The patch requires

the existence of one or more pools of accounts on the local system and a reserved directory (*/etc/grid-security/gridmapdir)* to store the account information.
In the EDG Release 1 a perl script (*mkgridpool)* used to administrate the pools is provided. Four options allow different actions:

-add: initializes the *gridmapdir* creating an empty file for each pool of users found in */etc/passwd*;

-list: creates a report about the allocated accounts;

-del: deletes the content of *gridmapdir*. Any information about allocated and free accounts will be lost;

-unlink: all the pools will be made available for new dynamic allocations. Any information about allocated accounts will be lost.

4.3.6 Multiple VO's support

In the grid sometimes users may belong to different VO's. So, for billing or accounting reasons, it is important that the user can choose the appropriate local account associated to the proper VO when executing a job on the grid.
Since this capacity is not present in the standard Globus code the *grid-proxy-init* and *mkgridmap* commands have been patched to allow the user to specify a VO during the creation of his/her proxy certificate.
When the user executes the *grid-proxy-init* to create a new proxy certificate, the DN of the new certificate is the same DN of the original certificate, but a CN=proxy (or CN=limited proxy) is added at the end. Then the proxy certificate is sent to the gatekeeper where its subject, after the CN field has been stripped off, is matched against the *grid-mapfile* to authorize the user to execute jobs into the farm and associate him/her to a local account.
In order to maintain the backward compatibility, it was decided to continue using the subject–grid-mapfile mechanism to authorize users. So the VO membership information has been coded into the DN of the proxy certificate. A new field, D=<*VO*>, has been added just before the CN=proxy during the proxy creation, and this is reflected also in a change to the *grid-mapfile* generated by *mkgridmap.* In this file each DN contains the D field at the end.
The patched *grid-proxy-init* command has two new options:

-vo <VO> lets a user specify the VO (s)he wants to use

-novocreates an old-style certificate

If these options are both specified or are specified multiple times, only the last one is considered. Furthermore, since many applications call *grid-proxy-init* internally, a new environment variable, *VO*, has been created; if defined, it contains the default VO value used by *grid-proxy-init* if a value is not set by the –vo option. It should be noted that no difference is made between an undefined variable and an empty variable.

4.3.7 Future developments

The Authorization architecture designed and implemented so far is not able to satisfy all the requirements specified into the EDG WP7 document on security [31]. Moreover the *grid-mapfile* mapping system allows a very coarse-grained authorization.
A more general authorization architecture should take in account roles, groups, VO based Access Control Lists (ACLs) and any other attribute related to the VO Membership. This task can be achieved by migrating from a *grid-mapfile* based system to an infrastructure based on Authorization Certificates (AC). This kind of Certificates, that are going to be standardized by the PKIX Working Group [36], are issued by an Authorization Authority (AA) and contain Authorization Information. To date, there are at least two projects developing an infrastructure based on AC (or AC-like): the Community Authorization Service (CAS) [37], by the Globus Team and the VO Membership Service (VOMS) [38] under development by the EDG Authorization Group.

4.4 Grid Services layout and configuration

The INFN-GRID production testbed is composed by six main sites, where all the grid elements have been setup (UI, CE, SE, WN), and another group of minor sites where a subset of the grid elements has been configured (generally UI, CE and WNs). In the INFN-GRID Testbed infrastructure the CNAF site (located in Bologna) plays a fundamental role because it also hosts the nodes providing the common services needed for the grid functionality. For each Testbed infrastructure (currently three infrastructure exist: EDG Development Testbed, EDG Production Testbed and INFN-GRID Production Testbed) there are:

- one RB (Resource Broker) which each UI submits jobs to;
- one LB (Logging & Bookkeeping) node that stores the information about every job sent to the grid resources managed by the RB;
- one II (Information Index) that is used by the RB to find out the best CE to submit a job to. The II periodically queries the MDS server (GRIS) installed on each registered CE to retrieve static and dynamic information about the resource status; in the first EDG software releases each CE must be manually registered in the II.

Other services are shared among the different Testbeds:

- four RC's (Replica Catalogues), one for each supported VO (CMS, ATLAS, VIRGO and CDF) that contain the information about the location of the data replicas; the actual RC software implementation is not secure since each user can only access the RC by means of the RcManager password, that therefore must be public. In the next software releases the access will be certificate-based.
- One VO LDAP server containing the users' certificate subject for three different virtual organizations (INFN-GRID, VIRGO and CDF).
- A CVS server managing the software produced by INFN-GRID.

The common grid services are shown in fig. 11, that also includes the grid resources located at CNAF.

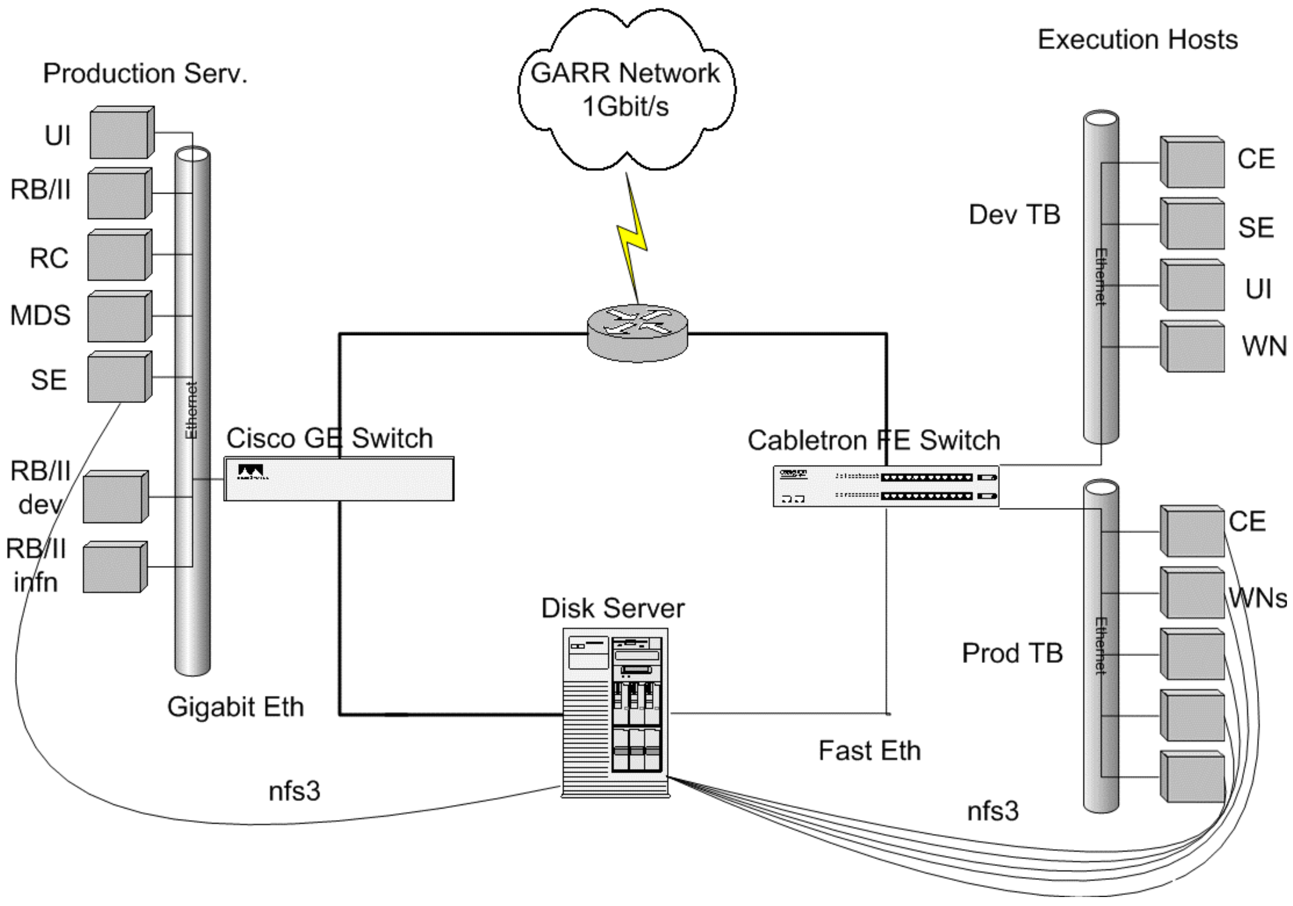


Fig. 11 INFN-GRID services layout

At each testbed site the site managers have first installed all the needed grid elements and created the accounts for their local users on the local UI and then, after some functionality tests, have registered their grid resources in the INFN-GRID Information Index.
After these operations the users are able to submit jobs to each grid resource known to the II by means of their local UI and the common RB.
In the future, as long as the grid resources will remarkably increase, more Resource Brokers will be configured in different VO domains.

The grid resources available at the main INFN-GRID testbed sites have also been registered in the Information Index located at CERN (serving the CERN Resource Broker) in which resources from CERN, INFN, In2p3, Ral and Nikhef are registered. In this way they are available to non-INFN-GRID users too.

4.4.1 Information Service layout

The Information Service is used by the RB to find the available resources. The current implementation involves two services, both based on Globus MDS 2.1: the standard GIIS/GRIS hierarchy and the Information Index (II).
For each CE or SE the GRIS publishes the information provided by some appropriate scripts (information providers) describing the resource into the GIIS; the GIIS is an index service that collects information from GRISes.
The II is a special version of a GIIS, modified by DataGrid WP1, that statically contains all the available grid resources in an appropriate configuration file. When new grid resources join the grid the II configuration file has to be manually updated. For this reason the II should be considered an *ad interim* solution only since it has obvious scalability problems.

In the second half of year 2001 the Globus Team introduced some new features into MDS 2.1 and into the new version of the Globus toolkit (Globus 2.0). The new system was soon tested and after some interactions between the EDG Integration Team and the Globus Team some further improvements were developed. The most important of them is the support for a hierarchical structure in a multi-VO's environment. Then some work has been done by the Integration Team to centralize and automate the setup of these new MDS features for the EDG distribution.
As already mentioned the EDG WP1 developers have modified the configuration of the GIIS, distributing the II as a WP1 package, with the main goal of manually controlling the list of resources known by the Broker.

The II layout used in the Testbed is shown in fig. 12.

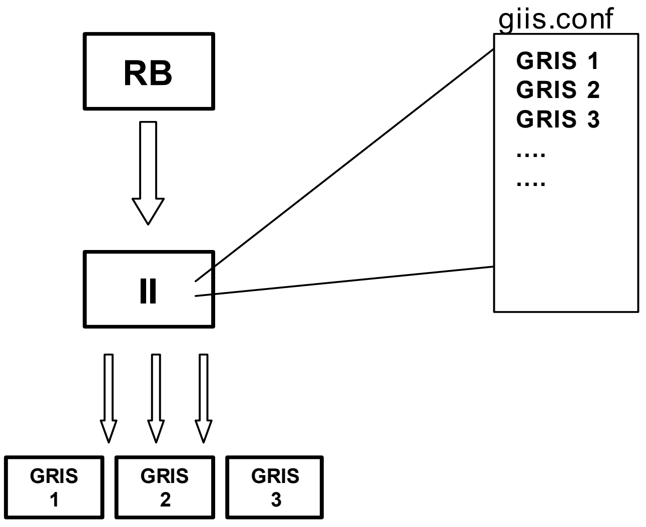


Fig. 12 - The WP1 Information Index layout

Since the Testbed is continuously growing, and also the problems and resource failures could strongly increase, the listing of the existing GRISes (grid resources) in the II has become almost impossible to maintain. Moreover, an annoying deficiency of the OpenLDAP and of the Globus backend when managing network timeouts has been discovered. The problem is really serious because when a single resource becomes unreachable or unavailable, a very frequent situation in a widely deployed Testbed, this can block the Information System, and consequently the job submission. The problem has been deeply investigated and two patches have been produced. The patch produced by NorduGrid people, that solved the problem by means of some modifications to the GIIS backend, has been chosen.
In addiction, to reduce the administration workload of the Information Index , it was decided to use the MDS hierarchical capability. Therefore only some GIIS's (one for each VO at the moment) are listed in the II, that contain the information about all the available grid resources (through the automatic registration of the GRIS's into the GIIS). Such a layout is shown in fig. 13.

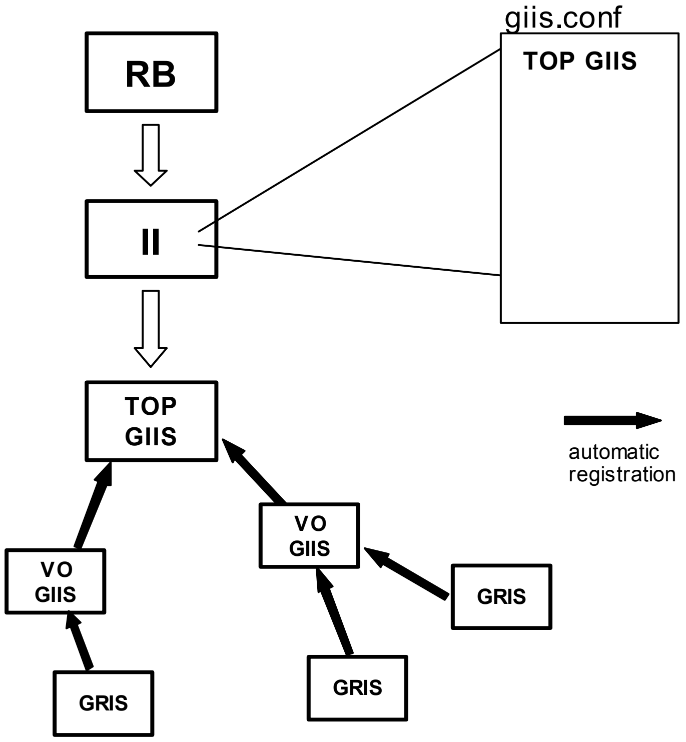


Fig. 13 - MDS configuration

The advantages of this layout are:

- The administrators of the Information Index don't need to frequently add/remove resources and restart services;
- the VO GIISes, and consequently also the resources, register themselves periodically into the TOP GIIS, so if a resource becomes unavailable its registration expires (and disappears) after few minutes.

The global layout for Testbed configuration is based on a TOP GIIS located at CERN (whose name and scope are "Edg"), and a "country" GIIS level. The Italian country GIIS collects all the INFN resources that participate to the European testbed, while a separate INFN GIIS collects all the INFN resources participating to the INFN-GRID Testbed. The node hosting the INFN GIISes is located at CNAF.
The LDAP addresses of the top level INFN GIISes are:

- for the INFN-GRID production Testbed
  ldap://mds.cnaf.infn.it:2135/mds-vo-name=infn,o=grid

- for the INFN part of the EDG production Testbed
  ldap://mds.cnaf.infn.it:2135/mds-vo-name=infn-edg,o=grid

The MDS layout for EDG and INFN-GRID Testbeds is shown in fig. 14

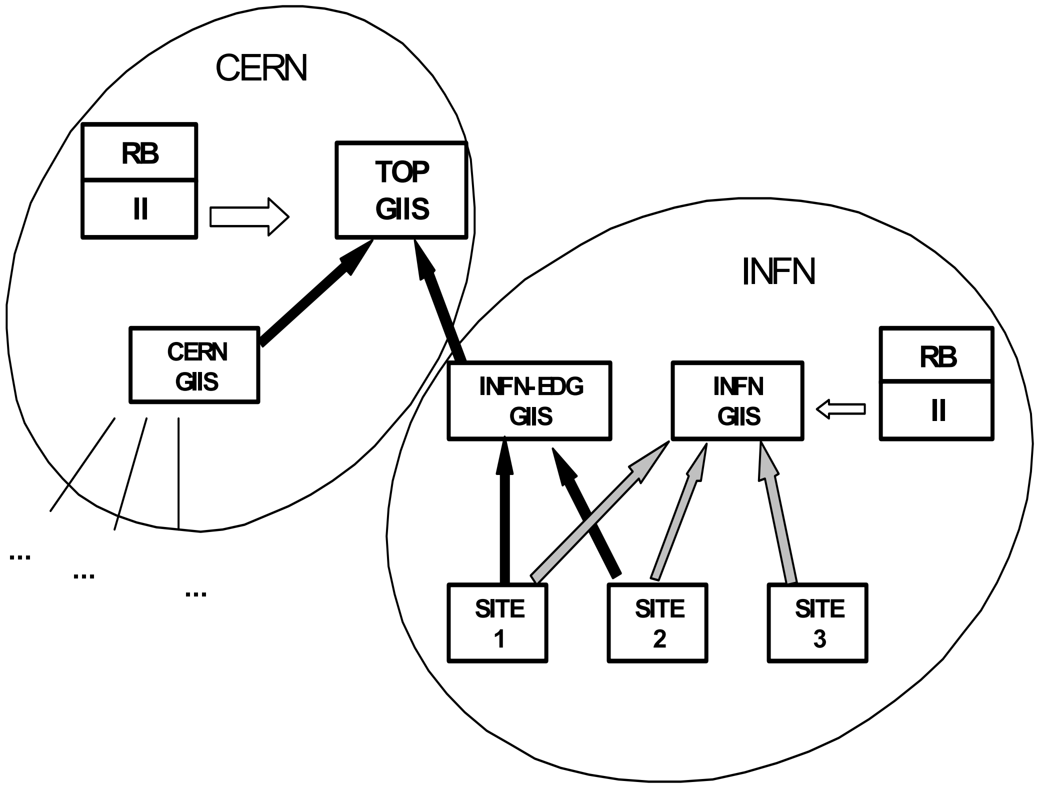


Fig. 14 - MDS layout for DataGrid and INFN Testbeds

This hierarchical layout has been tested with a small set of nodes and it worked well. But when it has been deployed in a wide environment it has shown unacceptable instabilities, with resources appearing and disappearing apparently without a reason. Moreover the multiple registration of the resources is not still available with the EDG setup in use. Some work has to be done to solve this problem since this feature is really important for the implementation of a shared widely deployed grid where most resources participate to various different testbed infrastructures.
In the meantime this has recently forced the testbed to fall-back to the old manual layout while a test of the stability of the new 2.2 release of MDS is in progress in the Integration Team.
The hierarchy described in this chapter reflects a geographical approach. This is sensible in this phase of the testbed activities, but in the next phases (a real production environment) a VO-based approach is foreseen.

4.5 Resources status

Grid is a set of geographically distributed resources that can be used for a particular goal [39]. These resources (CPUs, disks, tapes, network availability and bandwidth, etc.) should be monitored in order to allow the testbed managers both to determine the "health" of the grid at any time and follow its evolution in time.

These monitoring information is collected in grid "snapshots" that should include specific information for each kind of resource:

RESOURCE	**INFORMATION**
network devices (routers, switches, etc.)	CPU load, memory, IN bandwith, OUT bandwith, temperature,…
computing element	CPU load, memory, number of users, number of processes, …
storage element	free space, number of I/O interrupts, …
network services	DNS, rsh/ssh, telnet, NFS, …
"Grid" services	gatekeeper, job-manager, replica catalog, GSIftp …

For the monitoring of the INFN-GRID resources the Nagios system [40] has been chosen.

4.5.1 The Nagios package

Nagios is an OpenSource application for network monitoring designed to run under Linux (with various ports for other OS'es). The main purpose of Nagios is to monitor hosts, services and devices both on LAN's and WAN's. This monitoring system has a friendly web interface to inspect current network status, problem history, availability reports, etc.

and for running commands on each service (disable check, delay next check, disable notifications, etc.).

4.5.2 Architecture and features

The Nagios package is based on a *Plug-in model* [41]**.** When there is a need to check a service or a host, Nagios will execute a sensor plug-in that performs the check and returns the results to the control logic. Each plug-in, in order to check a service, accepts two thresholds (warning and critical) that are compared with the values returned from execution. Typical plug-in return values are: OK, WARNING, CRITICAL, UNKNOWN.

Nagios assigns different *status levels* to services and hosts. Service status levels are PENDING, OK, RECOVERED, WARNING, UNKNOWN, CRITICAL, UNREACHABLE, while host status levels are PENDING, UP, DOWN, UNREACHABLE.

Service checks are scheduled through several configuration options: check period, check interval and retry interval in which the service can be checked. This feature is called *Service check scheduling.*

Another important feature of the Nagios package is represented by the *Event handlers***.** Event handlers are commands that can be executed whenever a host or a service state change occurs. They can be used to proactively fix problems before anyone is notified. There are two types of event handlers that can be defined: service event handlers and host event handlers. Moreover, it is possible to specify global event handlers that should be run for every host or service state change.
When a host or service remains in a non-OK state and the time specified in the host or service definition has passed since the last notification was sent out, Nagios sends out a notification to the contact group defined for that particular service. There are different ways to do notifications and there are some packages that handle notifications for pagers and cellular phones.
The *Indirect host/service checks* feature allows the monitoring of resources that are not directly accessible on the net. Such resources (i.e. hosts behind firewalls) can be checked by means of the use of an agent. NRPE (Nagios Remote Plug-in Executor) is designed to provide a way to execute plug-ins on a remote host, to send plug-in output and to return code back to the Nagios core logic. The NRPE agent authenticates plug-in execution requests by doing a comparison of the IP address of the calling host against a list of allowed IP addresses in a configuration file.

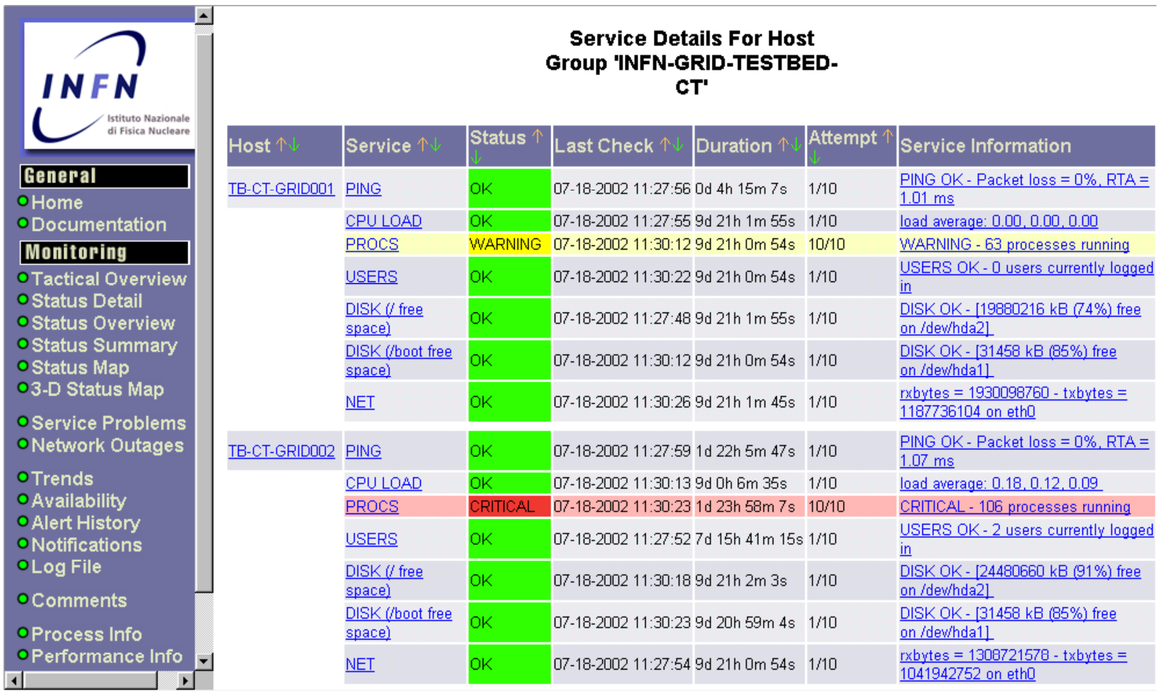
Also the *Passive service checks* feature is very interesting. Nagios can process service check results that are submitted by external applications; these types of checks are called passive checks and are useful for monitoring remote resources, services and hosts behind firewalls; services that are asynchronous (e.g. SNMP traps, security alerts, etc.). The NSCA add-on (Nagios Service Check Acceptor) allows to send passive service check results from remote hosts to a central monitoring host that runs Nagios.

Last but not least, the *Distributed monitoring* feature is really interesting for a distributed computing environment like the grid. The goal in the distributed monitoring environment is to off-load the overhead and single point of failure when performing service checks, from a central server onto one or more distributed servers. The function of a distributed server is to check all the defined services for a set of hosts; it is usually a bare-bones installation of Nagios – it does not have the web interface installed but executes only service checks. The purpose of the central server is simply the listening for service check results from one or more distributed servers. Since the central server is obtaining passive service check results from one or more distributed servers, it serves as the main point for all monitoring logic (i.e. it sends out notifications, runs event handler scripts, determines host states, has the web interface installed, etc.) [42].

4.5.3 Monitoring the testbed: a " real life" use case

In order to test the concepts and capabilities of Nagios in a real GRID environment, the monitoring of the INFN-GRID testbed constituted by tens of computing resources (CPU's, disks, tapes) located in several sites has been defined as a use case. For this purpose, Nagios has been successfully configured to monitor all the grid resources. Configuration takes just a couple of hours and the system is up and running since October 2001 without any failure [43].

The success obtained in this real life application has also triggered the adoption of Nagios as the basic monitoring tool for the INFN-GRID testbed. In fig. 15 the Nagios information about the service status for two hosts is shown: warnings and/or critical status are highlighted. Useful information is reported to the site managers in order to help them to fix the detected problems.



Service Details For Host Group 'INFN-GRID-TESTBED-CT'

Host	Service	Status	Last Check	Duration	Attempt	Service Information
TB-CT-GRID001	PING	OK	07-18-2002 11:27:56	0d 4h 15m 7s	1/10	PING OK - Packet loss = 0%, RTA = 1.01 ms
	CPU LOAD	OK	07-18-2002 11:27:55	9d 21h 1m 55s	1/10	load average: 0.00, 0.00, 0.00
	PROCS	WARNING	07-18-2002 11:30:12	9d 21h 0m 54s	10/10	WARNING - 63 processes running
	USERS	OK	07-18-2002 11:30:22	9d 21h 0m 54s	1/10	USERS OK - 0 users currently logged in
	DISK (/ free space)	OK	07-18-2002 11:27:48	9d 21h 1m 55s	1/10	DISK OK - [19880216 kB (74%) free on /dev/hda2]
	DISK (/boot free space)	OK	07-18-2002 11:30:12	9d 21h 0m 54s	1/10	DISK OK - [31458 kB (85%) free on /dev/hda1]
	NET	OK	07-18-2002 11:30:26	9d 21h 1m 45s	1/10	rxbytes = 1930098760 - txbytes = 1187736104 on eth0
TB-CT-GRID002	PING	OK	07-18-2002 11:27:59	1d 22h 5m 47s	1/10	PING OK - Packet loss = 0%, RTA = 1.07 ms
	CPU LOAD	OK	07-18-2002 11:30:13	9d 0h 6m 35s	1/10	load average: 0.18, 0.12, 0.09
	PROCS	CRITICAL	07-18-2002 11:30:23	1d 23h 58m 7s	10/10	CRITICAL - 106 processes running
	USERS	OK	07-18-2002 11:27:52	7d 15h 41m 15s	1/10	USERS OK - 2 users currently logged in
	DISK (/ free space)	OK	07-18-2002 11:30:18	9d 21h 2m 3s	1/10	DISK OK - [24480660 kB (91%) free on /dev/hda2]
	DISK (/boot free space)	OK	07-18-2002 11:30:23	9d 20h 59m 4s	1/10	DISK OK - [31458 kB (85%) free on /dev/hda1]
	NET	OK	07-18-2002 11:27:54	9d 21h 0m 54s	1/10	rxbytes = 1308721578 - txbytes = 1041942752 on eth0

Fig. 15 – Nagios information about the status of the services

4.6 The INFN-GRID Testbed support

The INFN-GRID support infrastructure has been implemented in order to offer help and answer to various types of requests and questions. Since there are no dedicated human resources to the support activities right now, the service has been implemented on a best effort basis, involving the managers of all the INFN-GRID sites. Currently the service is under test. In order to have an efficient and stable service a dedicated support team is indispensable; the planning of such an operational service for the next year is under discussion. This service is offered to all the members of the INFN-GRID community (site managers and end users) for installing, configuring and using the GRID. The support team should take advisory input from people working in the middleware workpackages and in the Testbed services groups, and should also collaborate to all the INFN-GRID acivities. Users can access the support service through an e-mail address (gridsupport@infn.it) or (preferred method) through a web portal system. By means of the web portal users can reach the support resources and get documents through a single point of access: http://gridsupport.na.infn.it
A web server hosts a central repository for documents about the INFN-GRID Testbed and also keeps track about the problems and their solutions.
The support model consists in two steps:

- a user submits a request to the support system and waits for a reply from the support team from other users;
- if the problem has not been solved by the support group then the request is forwarded to the other grid support services and/or mailing lists (e.g. Bugzilla [44]), or to the appropriate middleware workpackage.

PHP-Nuke [45], an open source portal system, has been choosen for the INFN-GRID testbed. This package is written in PHP and works in combination with the Apache web server and a *mysql* database, where all documents, requests and answers are stored.

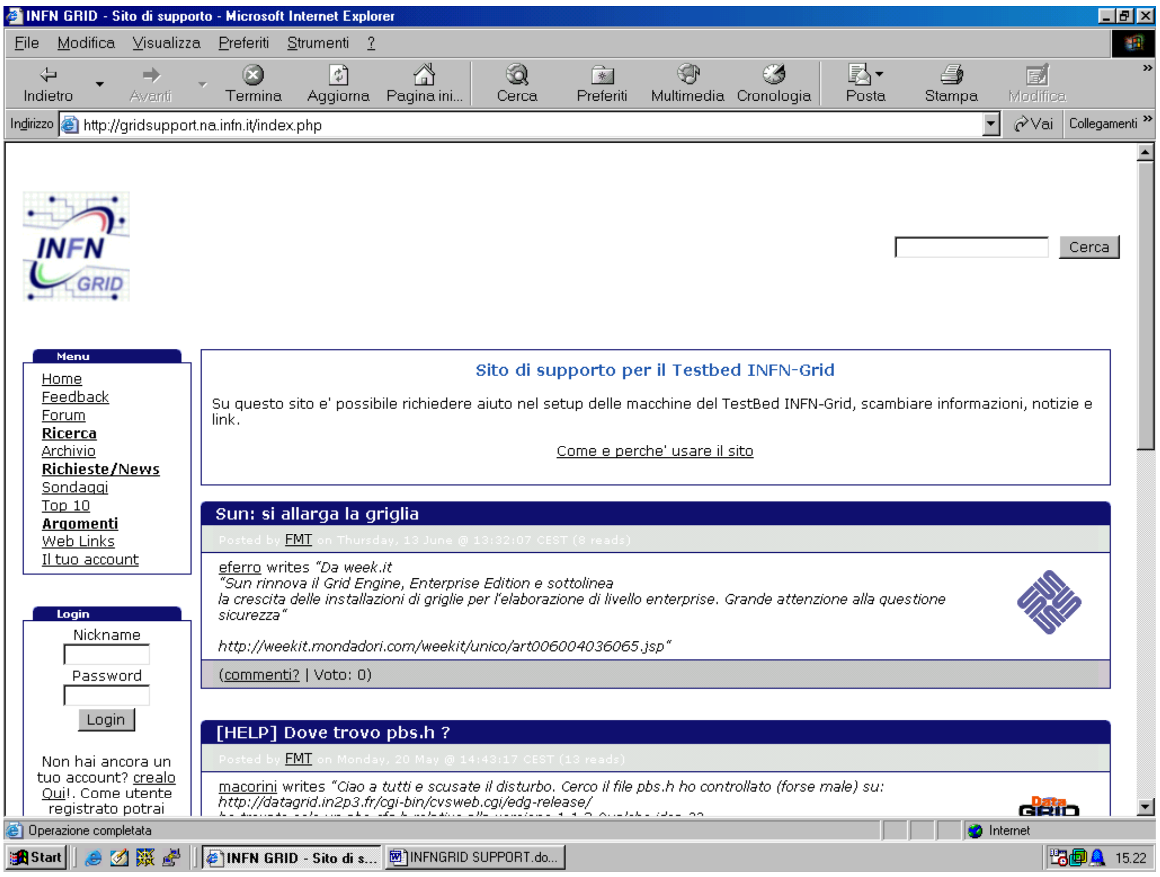
PHP-Nuke allows users to:

- insert requests, news, answers and comments via a form
- search in the knowledge base and documents
- access forums and surveys
- create some documents (e.g. mini how-to), directly online

At present the knowledge base is organized in several topics in a very flexible way (e.g. “LCFG”, “Examples”, “Software configuration”) according to the needs. In this way the users can restrict the field of their queries or search only in a specific topic. Once a request is posted to the support site, other people can read the open questions via a web browser and post a reply or a comment, extending the knowledge base. Users can search through documents, requests and answers posted on the site. There are also some specific forums, related to the “use cases” for example, and some surveys, used to verify the opinions of our users in fields like languages, ease of installation, or the most important problems.

Important documents are published or linked on the site, and several of them can be composed and modified online.

This support site will grow not only thanks to the support group and its documents or answers, but mainly thanks to the "mutual help" between the users involved in the INFN-GRID project. Using this tool in addiction to the already existing testbed mailing list the support team offers a pervasive presence in order to guarantee fast response to the problems.
The INFN-GRID support web portal is shown in fig. 16.



INFN GRID - Sito di supporto - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Avanti Termina Aggiorna Pagina ini... Cerca Preferiti Multimedia Cronologia Posta Stampa Modifica
Indirizzo http://gridsupport.na.infn.it/index.php
Vai Collegamenti
INFN GRID
Cerca
Menu
Home
Feedback
Forum
Ricerca
Archivio
Richieste/News
Sondaggi
Top 10
Argomenti
Web Links
Il tuo account
Login
Nickname
Password
Login
Non hai ancora un tuo account? crealo Qui!. Come utente registrato potrai
Operazione completata
Internet
Start
INFN GRID - Sito di s...
INFNGRID SUPPORT.do...
15.22

Sito di supporto per il Testbed INFN-Grid

Su questo sito e' possibile richiedere aiuto nel setup delle macchine del TestBed INFN-Grid, scambiare informazioni, notizie e link.

Come e perche' usare il sito

Sun: si allarga la griglia

Posted by FMT on Thursday, 13 June @ 13:32:07 CEST (8 reads)

eferro writes *"Da week.it*
"Sun rinnova il Grid Engine, Enterprise Edition e sottolinea
la crescita delle installazioni di griglie per l'elaborazione di livello enterprise. Grande attenzione alla questione sicurezza"

http://weekit.mondadori.com/weekit/unico/art006004036065.jsp"

(commenti? | Voto: 0)

[HELP] Dove trovo pbs.h ?

Posted by FMT on Monday, 20 May @ 14:43:17 CEST (13 reads)

macorini writes *"Ciao a tutti e scusate il disturbo. Cerco il file pbs.h ho controllato (forse male) su:*
http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/edg-release/

Fig 16 - A screenshot of *gridsupport*

5 The evolution of the grid layout: from Testbed0 up to Production testbed

As it has been described in the previous chapters the Testbed needs have been evolving, according to the different phases of the INFN-GRID and DataGrid plans. The Testbed0 was setup to test the Grid core services based on the Globus Toolkit, and it was characterized by small Computing Elements widely distributed. Then the Testbed1 was mainly focused on the DataGrid release validation, for which not only CE's, represented by more powerful farms, but also new grid resources like Storage Elements and new services have been deployed in a limited number of sites. Now the middleware is ready to allow a production environment setup.

The most important roles in the setting up of a *production testbed* are represented by the CERN LCG project driven by the LHC experiments and by the new national FIRB-GRID project [46] where several research communities are involved.

LHC experiments, other INFN experiments like Virgo, CDF and BaBar and the FIRB communities are grid users with different computing needs and the main goal of the INFN-GRID production testbed planning, coordination and management is to deploy an integrated grid computing system satisfying the needs of all the users involved. Each research community is a Virtual Organization whose 'virtual' computing grid is dynamically configured.

5.1 LCG Grid Deployment

The main goal of the LHC Computing Grid project is to prepare the computing infrastructure for the simulation, processing and analysis of LHC data for all four LHC experiments (Alice, Atlas, CMS and LHCb). This computing infrastructure will be implemented as a geographically distributed Computational Data Grid based on resources located at CERN and in a large number of regional centres in many different countries.

The grid deployment workplan foresees to have an LCG Global Grid Service (LCG-1) during the first half of year 2003.

INFN is committed to this plan providing the so-called T1-like and T2-like farms (according to the "Tier" schema developed by the MONARC project [47]) located at Bologna (CNAF), Milano, Legnaro and Torino.

5.2 Experiments' testbed deployment

The INFN components of the LHC experiments will use the LCG resources (T1 and T2 centers) and their own resources (T2 or T3/UI). Other experiments like Virgo and CDF, in addiction to their national and international resources, can count on the INFN-GRID national testbed to increase their available computing power. In fact the real benefit of having most grid resources (belonging to INFN and to other international research institutions) shared in a common grid is the optimization of their availability. In this way it is possible to satisfy the peak needs through the usage of such a shared facility.

5.3 DataGrid application testbed

The DataGrid project foresees two other major software releases (release 2 at the end of 2002 and release 3 at fall 2003), that will improve the existing grid middleware services and will also add new services. As for the previous Release this new software has to be validated by the applications provided by the HEP, Biology and EO communities on the European testbed set up by the DataGrid partners. INFN is participating with resources located at all the INFN sites involved in the INFN-GRID project.

5.4 FIRB resources

One of the main goals of the FIRB project is the development and deployment of a national hardware and software grid infrastructure enabling an easy and reliable access to the grid resources. According to this goal, the FIRB Work Package 3 is focused on the implementation of some “core” services in order to build not only a “demo” testbed but also a “production” one for the Biology, Astrophysics, Earth Observation and Volcanology research communities. These services can be provided by adapting the existing grid toolkits or by developing brand new software where needed.
A deployment plan describing the foreseen software functionality needs is going to be produced.

The INFN experiments and the Biology and Geophysics communities will provide dedicated resources in appropriate sites, mainly where the presence of multiple different research groups can permit an aggregation of the resources. For example, Catania, Bologna, Genova and Padova are the candidate sites for INFN.

5.5 The INFN-GRID testbed

According to the mentioned requirements (some of which already correspond to existing testbed infrastructures) the INFN-GRID project should be able to design and implement a pre-production testbed. This infrastructure should integrate all the available resources in a grid topology in order to satisfy the needs of the involved user-communities in agreement with each VO grid guidelines.

5.6 The VO Scenario

The research communities sharing their resources in a common computational grid environment have to define the policies that govern the sharing. Since the sharing relationships are dynamic (according to the scientific activities of each VO), mechanisms for defining the sharing degree of each resource at each moment are required. In other words a new participant joining a VO must be able to determine which resources (s)he is able to access, the "quality" of these resources, and the policies that govern access. Therefore there is the need of specific VO services to manage the authentication and authorization steps, to define and publish the grid access policies, and other related services. Tools like the Virtual Organization Membership Service (VOMS) focused on the user authorization provide a prototype of a VO-based management system.

One of the most important grid services implemented by DataGrid is the grid scheduler made up of several components already described (User Interface, Resource Broker, Logging & Bookkeeping and Job Submission Service). So far this service has been configured in a central grid topology but in a production environment shared among multiple virtual organizations a VO-based scheduling service seems to be more suitable. Such a topology has to be carefully investigated.

The Information Service has to provide information about the resource status to the grid schedulers and must also be used to have a snapshot of the grid resources of the collaborating communities.
Data management is one of the main tasks of the grid. In order to keep track of the physical location of the data and of their replicas and to manage the data replication some special services (the Replica Catalogues and the Replica Managers) have been developed. These services have been implemented in a VO perspective.

Therefore each VO manages its own Replica Catalog and Replica Manager.

In fig. 17 the grid domain is represented from the VO viewpoint, showing the VO dedicated services and both private and shared resources.

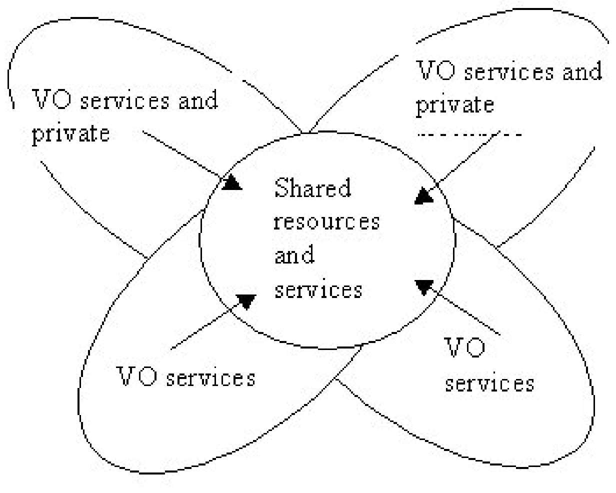


Fig. 17 - VO's shared resources and services distribution

The grid services must be configured in such a way that each VO has the *view* of the resources (private and shared) it can use, can manage the user access to them, organize the job scheduling and the data distribution in a convenient way, in agreement with the policies agreed upon by the resource owners. For example a VO has to determine and install the appropriate amount of UI's and RB's according to the distribution of its users; it has also to manage the VOMS server, the replica catalogues and the monitoring tools.

Moreover a proper information system has to be deployed in order to collect the resource information for each RB and at the same time to provide the whole picture of the grid. An example is shown in fig. 18.

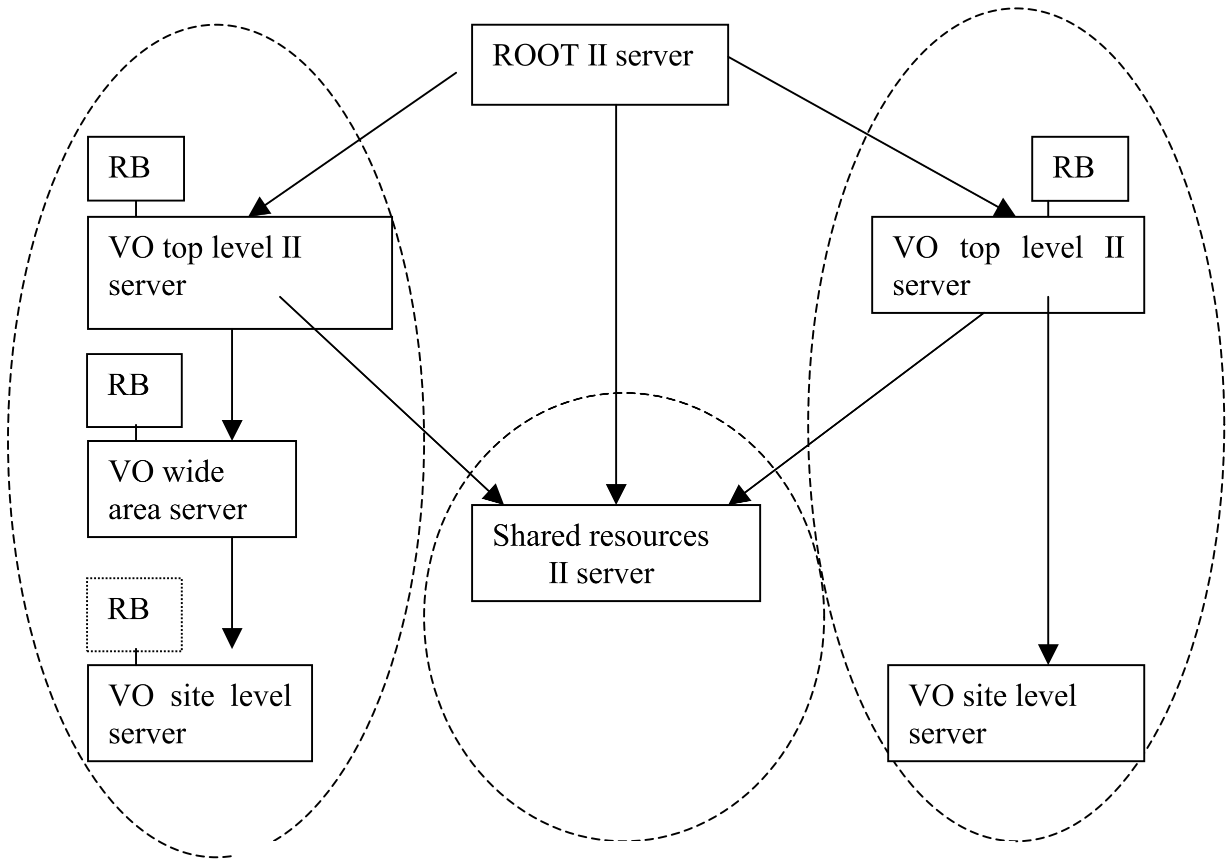


Fig 18 - Example of an Information System and RB topology in a VO environment

The *root II server* is optional; its only function is to collect the information about all the existing GIIS's.

The *VO top-level Information Index server* collects the information about all the grid resources (private and shared) of a VO and the *Shared resource top level II server* contains the information about all the shared grid resources only.

5.7 Multiple VO Testbed Management

The capability to provide a ‘production grid facility’ implies the solution of the complex problem of the operational management of a grid.

On this subject some important issues have already been identified:

a. Grid services functionality: appropriate test-suites must be developed with each new release to test the configuration and the functionality of each grid service and of the grid as a whole.
b. Grid monitoring: specific grid monitoring tools must be integrated in order to monitor the jobs and the resources and services status.
c. User support: a world-wide computational system used by world-wide user communities requires an appropriate user support management and tools.
d. Deployment of VO specific services and resources: the Virtual Organizations require the definition of virtual grid domains that can frequently be changed. VO-based grid layouts have to be designed and tested according to the VO needs.

It is very likely that other important issues will emerge when a production grid will be used by more and more users. However the capability of managing a production grid will take advantage by the experience of a production environment.

6 Conclusions

INFN-GRID has the ambitious aim of demonstrating the feasibility of a production grid, based on the services so far developed by DataGrid, Globus, and Condor, to satisfy the computing needs of the different INFN experiments (the LHC experiments, Virgo, CDF and Babar), and in general of other academic and research communities.

To achieve this goal a multiple VO oriented production testbed is being deployed and managed. So far a huge amount of work has been done and many new problems have been faced.

The challenge is to demonstrate the capability to manage a wide distributed computing grid, constituted by private and shared resources, where each VO can benefit of the usage of whatever available resources according to its needs (planned or unpredicted) in a transparent way.

The Testbed WP has to provide an effective solution to manage the access policies to the grid resources according to the agreements between the VO's and the resource owners.

The project's success will continue to increase as the simplicity of its infrastructure and the efficiency of its support allow it to be easily used by ever increasing user communities.

Acknowledgments

The topics and results presented in this document are the outcome of two years of intense and profitable collaboration among many people.
The authors are very grateful to all the colleagues involved in INFN-GRID project, first of all the INFN site managers and the Testbed Work Package members, for the effort lavished during this period: Antonio Amoroso, Enrica Antonioli, Stefano Barale, Massimo Biasotto, Alessandro Brunengo, Ernesto Cangiano, Maria D'Amato, Alessandro De Salvo, Carlo De Vecchi, Domenico Diacono, Federica Fanzago, Enrico Fasanelli, Luigi Fonti, Michele Gambetti, Roberto Gomezel, Tullio Macorini, Michele Michelotto, David Rebatto, Silvia Resconi, Carlo Rocca, Franco Semeria, Antonio Silvestri, Sandro Spanu, Paolo Veronesi and Domenico Vicinanza.
Special thanks are due to Massimo Sgaravatto, Elisabetta Ronchieri, Massimo Mezzadri and Francesco Prelz for their help in managing the WP1 services.
Thanks also to the DataGrid Project colleagues, expecially to the Integration Team group, for the helpful cooperation.

7 References

[1] The INFN-GRID Project - http://www.infn.it/grid
[2] GARR, the italian academic and research network – http://www.garr.it/
[3] The DataGrid Project - http://www.eu-datagrid.org
[4] High Energy Physics - http://datagrid-wp8.web.cern.ch/DataGrid-WP8/
[5] Earth Observations (EO) - http://styx.esrin.esa.it/grid/
[6] Bio-Informatics - http://marianne.in2p3.fr/datagrid/wp10/index.html
[7] http://public.web.cern.ch/Public/SCIENCE/lhccolexp.html
[8] The Virgo experiment - http://www.virgo.infn.it/
[9] The CDF experiment - http://www-cdf.fnal.gov/
[10] The BaBar experiment - http://www.slac.stanford.edu/BFROOT/
[11] The DataTAG project – http://www.datatag.org/
[12] The LHC Computing Grid Project - http://lhcgrid.web.cern.ch/LHCgrid/
[13] The GridPP Project - http://www.gridpp.ac.uk/
[14] The Globus toolkit - http://www.globus.org/toolkit/
[15] R. Alfieri, C. Anglano, et al., Report on the INFN-GRID Globus evaluation, March, 26 2001INFN-TC-01-09 and http://grid.infn.it/globus/Docs/infn-globus-evaluation.pdf
[16] http://server11.infn.it/grid/dist.html
[17] http://www.pi.infn.it/GRID/dist.html
[18] EDG Software Release Plan - http://edms.cern.ch/ document/333297
[19] LCFG (Local ConFiGuration system) – http://www.lcfg.org/
[20] DataGrid Deliverable D8.2 - http://edms.cern.ch/document/334920
[21] DataGrid Deliverable D6.4 - http://edms.cern.ch/document/336389
[22] The HEPCAL activity – http://lcg.web.cern.ch/LCG/peb/web/rtags.htm
[23] http://hep-proj-grid-fabric.web.cern.ch/hep-proj-grid-fabric/
[24] http://cern.ch/hep-proj-grid-fabric-config/documents/cfglan.pdf
[25] http://www.rpm.org
[26] http://www.lnl.infn.it/datagrid/wp4-install
[27] http://cvs.infn.it/cgi-bin/cvsweb.cgi/testgrid/
[28] http://marianne.in2p3.fr/datagrid/TestPlan/
[29] http://alice-grid.ca.infn.it
[30] http://www.globus.org
[31] Security Requirements and Testbed1 Security Implementation, DataGrid-07-D7.5-0111-4-0 - http://edms.cern.ch/document/340234
[32] http://cvs.infn.it/cgi-bin/cvsweb.cgi/Auth/
[33] https: //security.fi.infn.it/CA/
[34] http://marianne.in2p3.fr/datagrid/documentation/quick-start.html
[35] http://www.gridpp.ac.uk/gridmapdir/
[36] http://www.ietf.org/html.charters/pkix-charter.html

[37] http://www.globus.org/security/CAS/
[38] http://cvs.infn.it/cgi-bin/cvsweb.cgi/Auth/voms/
[39] I. Foster, C. Kesselman, *The Grid: Blueprint for a new computing infrastructure***,** Morgan Kaufmann, 1999
[40] The Nagios system – http://www.nagios.org
[41] K. DeBisschop, E. Galstad, H. Gayosso**,** *Nagios plug-in development guidelines***,** 2000-2001
[42] E. Galstad, *Nagios documentation*, 1999-2001
[43] See http://infn-tb:guest@infngrid.ct.infn.it
[44] The EDG bug reporting system – http://marianne.in2p3.fr/datagrid/bugzilla/
[45] PHP-NUKE – http://www.phpnuke.org
[46] The FIRB project - http://grid.infn.it/firb-grid
[47] The MONARC project - http://monarc.web.cern.ch/MONARC

8 Glossary

AFS	Andrew File System
API	Application Programming Interface
ATF	DataGrid Architecture Task Force
CA	Certification Authority
CAS	Community Authorisation Service
CE	Computing Element; a Grid-enabled computing resource.
CERN	Centre Europeen pour la Recherche Nucléaire
CERT	X.509 Certificate
CNRS	Centre National de la Recherche Scientifique
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CVS	Concurrent Versioning System
DAQ	Data Acquisition
DBMS	Data Base Management System
EDG	European Data Grid; (official project name is "DataGrid").
EO	Earth Observation
ES	Earth Science
ESD	Event Summary Data; used in HEP: information required for detailed analysis and high-level reconstruction.
ESRD	Earth Science Requirement document
EU DataGrid	European Data Grid Project
FTP	File Transfer Protocol
GDMP	Grid Data Mirroring Package; a WP2 application.
GGF	Global Grid Forum
GIS	Grid Information Service; (e.g. IMS or Globus MDS).
Globus	Project that aims at developing fundamental technologies needed to build computational grids.
GMA	Grid Monitoring Architecture; monitoring architecture defined by GGF.
GRAM	Grid Resource Allocation Management
GriFIS	Grid Fabric Information Service
GS	Grid Scheduler; service responsible for selecting which Grid resources to use for a given job.
GSI	Grid Security Infrastructure (Globus Security mechanism).
GUI	Graphical User Interface
HEP	High Energy Physics
HSM	Hierarchical Storage Manager. HSM software allows infrequently accessed data to be migrated to less expensive offline storage automatically.
HW	Hardware
I/O	Input/Output
IDL	Interactive Data Language
IMS	Information and Monitoring System; catalogue and distribute static and dynamic data about the Grid.

IST	Information Society Technologies
JDL	Job Description Language; to describe Grid jobs.
JSS	Job Submission Service
LB	Logging and Bookkeeping
LCAS	Local Centre Authorisation Service
LCG	LHC Computing Grid
LCMAPS	Local Credential MAPping Service
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
LFN	Logical File Name; A globally unique name to identify a specific file which is mapped by the RC onto one or more PFNs.
LHC	Large Hadron Collider
LRMS	Local Resource Management System; controls resources within a CE e.g. PBS or LSF.
LSF	Load Sharing Facility
MDS	Globus Meta-computing Directory Service
MPI	Message Passing Interface
MPI	Standard API for exchanging messages between networked computers. Several implementations of MPI are available on different platforms. MPICH is a freely available, portable implementation of the MPI standard. A Globus-aware MPICH implementation is proposed on Globus home page
MR	Monitoring Repository
MSA	Monitoring Sensor Agent
MSS	Mass Storage System
MUI	Monitoring User Interface
MySQL	Widely distributed SQL open source implementation.
NFS	Network File System
NIS	Network Information System
NMA	Node Management Agent
OS	Operating System
PB	PetaByte
PDS	Payload Data Segment
PFN	Physical File Name; URL of actual physical instance of an LFN.
PKI	Public Key Infrastructure
PM	Project Manager
PMB	Project Management Board
PTB	DataGrid Project Technical Board
QoS	Quality of Service
RB	Resource Broker
RC	Replica Catalog; associates an LFN to one or more PFNs.
RCS	Revision Control System
RD	Research and Development
RDBMS	Relational Database Management System
Replica	A copy of a file that is managed by the Grid middleware.
Replica Service	The combination of Replica Catalogue, Replica Manager, and all associated components.
RM	Replica Manager; provides several services related to replicas.

RMS	Resource Management Subsystem
RPC	Remote Procedure Call
RPM	Red Hat Package Manager
RSL	Globus Resource Specification Language
RUP	Rational Unified Process
SAN	Storage Area Network.
SE	Storage Element; a Grid-enabled storage system.
SI2000	SpecInt 2000 Benchmark
SP	Software Package
SQL	Standard Query Language for relational databases.
SR	Software Repository
SSL	Secure Sockets Layer
SW	Software
TAG	Event Selection Tag; used in HEP: used for fast event selection.
TeV	Tera electron Volt
Testbed 0	DataGrid initial Glogus toolkit deployment
Testbed 1	First release of DataGrid Software - Year 1
Testbed 2	Second release of DataGrid Software - Year 2
Testbed 3	Third release of DataGrid Software - Year 3
TFN	Transport File Name; URL to access a given file on a SE.
TFTP	Trivial File Transfer Protocol
UI	User Interface
URL	Uniform Resource Locator
VO	Virtual Organization; A set of individuals defined by certain sharing rules - e.g. members of a collaboration.
VPN	Virtual Private Network
WM	Workload Management
WMS	Workload Management System
WP	Work Package
WP1	Grid Work Scheduling
WP2	Grid Data Management
WP3	Grid Monitoring Services
WP4	Fabric Management
WP5	Mass Storage Management
WP6	Integration Testbed and Demonstrators
WP7	Network Services
WP8	HEP Applications
WP9	Earth Observation Applications
WP10	Biology Applications
WP11	Dissemination
WP12	Project Management
WWW	World Wide Web
XML	eXtensible Markup Language

9 Index