VALIDATION OF THE CROSSGRID TESTBEDS

WP4

Document Filename:	**CG-4.4-D4.5-v1.0-LIP0XX-TestbedValidation.doc**
Work package:	**WP4**
Partner(s):	**LIP**
Lead Partner:	**LIP**
Config ID:	**CG-4.4-D4.5-v1.0-LIP0XX-TestbedValidation**
Document classification:	**PUBLIC**

Abstract:
This document describes the work developed in the context of the testbed verification and quality control.

Delivery Slip

	Name	Partner	Date	Signature
From				
Verified by				
Approved by				

Document Log

Version	Date	Summary of changes	Author
1-0-DRAFT-A	1/04/2003	Draft version	Jorge Gomes
1-0-DRAFT-B	24/04/2003	Second draft version	Jorge Gomes, Mario David

CONTENTS

1 INTRODUCTION

The stability of the CrossGrid production testbed is extremely important for development of the CrossGrid tools and applications. It is virtually impossible to develop complex applications and tools when the basis for these developments is unstable. In the case of CrossGrid the basis is provided by the Grid middleware and by the sites that comprise the production testbed. Problems in any of these components can render the testbed unstable and therefore useless for the users and application software developers.

CrossGrid also aims to extend the Grid technologies coverage to new countries where it was previously unavailable, therefore the deployment of a near production quality infrastructure is essential to attract users and foment new opportunities for further technological developments.

In order to achieve these goals the CrossGrid work package 4 (International testbed organisation) contains a task dedicated to testbed verification and quality control. The objective of this task is to ensure that the CrossGrid production testbed conforms to the quality levels that characterize a testbed production service. This is achieved through:

- Validation of the middleware and corresponding documentation.
- Validation of sites after each new middleware installation.
- Monitoring of the testbed.

Verification and quality control should not be confused with quality assurance. In fact validation and quality control can be considered as the last step of the quality assurance process where the final verification is performed to verify that the products are compliant with their specifications and perform according to the requirements. The verification and quality control does not interfere with the middleware development process that is handled inside the corresponding development work packages or external projects (such as Globus and EDG), however it provides feedback to the developers on the quality of the middleware and on the improvements that might be required.

The main goal of the testbed verification and quality control is to make sure that the services provided by the testbed are reliable, dependable and conformant therefore providing a stable base for the application and tools development.

The aim of this document is to provide an overview of the work that has taken place within the “Verification and quality control” activities.

2 ACRONYMS AND ABBREVIATIONS

ACL	Access Control List
API	Application programming interface
CA	Certification Authority
CASTOR	CERN Advanced Storage Manager
CE	Computing Element
CN	Common Name
CRL	Certificate Revocation List
CrossGrid	The EU CrossGrid Project IST-2001-32243
DataGrid	The EU DataGrid Project IST-2000-25182
DBMS	Database Management System
EDG	European DataGrid
FTP	File Transfer Protocol
HTTP	HyperText Transport Protocol
GDMP	Grid Data Mirroring Package
JDL	Job Description Language
JSS	Job Submission Service
LB	Logging and Bookkeeping
LDAP	Lightweight Directory Access Protocol
LCFG	Local ConFiGuration system
LFN	Logical File Name
MDS	Meta Directory Services
NFS	Network File System
OU	Organizational Unit
PKI	Public Key Infrastructure
PFN	Physical File Name
QoS	Quality of Service
GDMP	Grid Data Mirroring Package
GID	Unix Group ID
GIIS	Grid Information Index Service
GRAM	Grid Resource Allocation Manager
GRIS	Grid Resource Information Service
GSI	Grid Security Infrastructure
RC	Replica Catalogue
RM	Replica Manager
RDBMS	Relational Database Management System
RB	Resource Broker
RFIO	Remote File I/O
SE	Storage Element
UI	User Interface
UID	Unix User ID
VO	Virtual Organization
XML	Extensible Markup Language
WMS	Workload Management System
WN	Worker Node
WP	Work Package

3 STATUS AND HISTORY

The verification and quality control activities began in the end of 2001 before the official start of the project. The set of activities necessary for verifying and controlling the quality of the testbed were identified and a middleware test and validation procedure began to be developed. The actions initially identified were mostly concerned on testing the middleware. However the necessity of ensuring that the complex set of Grid middleware components was correctly deployed soon become extremely important, namely after the first tests with the EDG middleware released in the end of 2001. This fact has shift most of the activity to ensure that the systems involved in test and validation activities were properly installed and configured prior to perform more complex tests. At the same time the necessity of evaluating the behavior of the installed Grid systems also lead to some involvement in the Grid monitoring field. This effort was twofold verify the stability of the middleware and monitor the future production testbed hence allowing the extraction of metrics. At the project kickoff meeting at Krakow the verification and quality control task was presented, jointly with the first draft of a middleware test and validation procedure.

From the official start of the project until June of 2002 the efforts were centered on following the EDG developments in close contact with the EDG integration team and several of its developers. Many tests were performed starting with the installation of Grid services at the main test site (LIP in Lisbon). In this period important feedback was provide back to DataGrid namely on what concerns the LCFG installation software chosen by DataGrid as the installation and configuration method. The installation and test of the LCFG server was followed by the installation and test of a Computing Element, Worker Node, Storage Element and User Interface. The know-how obtained would be extremely valuable for later test activities.

The testing of the first central services started on June of 2002. These include the Resource Broker, VO server and MyProxy server. The deployment of these services allowed the creation of a small testbed where simple jobs could be submitted through a User Interface to a Resource Broker and then sent to a Computing Element for execution. At the same time other test sites were actively involved in deploying and testing Computing Elements, Storage Elements and Worker Nodes. When the central services were considered stable enough it was suggested their deployment for usage by the CrossGrid community, thus allowing the creation of the first CrossGrid testbed. Since the experience on the deployment and maintenance of the central services was restricted to the main test site and there wasn't yet the need for a dedicated production testbed, the services deployed in the context of the verification and quality control were converted to serve the whole project. As consequence the first CrossGrid testbed was born and named "initial" testbed. This testbed supported the development and user needs since July of 2002 until January of 2003. These were the first EDG based services to be setup outside DataGrid and also the first Resource Broker to be deployed outside CERN providing services to a real community. Other services were tested and provided soon after the initial deployment of the testbed namely a Replica Catalogue.

In September of 2002 a document explaining the architecture of the test and validation testbed used in the context of the “verification and quality control activities” was released. This document describes and discusses extensively the architecture of the validation testbed and its possible evolution taking into account the DataGrid and CrossGrid developments foreseen at the time.

During October and November of 2002 a mini infrastructure to test the EDG release of the LCFGng installation software was setup. During the tests many problems were identified and reported back to DataGrid. The objective of this effort was to contribute to the improvement of LCFGng and accelerate its deployment. LCFGng is a key piece that will enable de deployment of EDG middleware over RH 7.2 which is considered an important step for CrossGrid.

When the evaluation of the first release of LCFGng finished efforts were concentrated on EDG development releases 1.3.x and 1.4.x. During this period the developments of EDG were carefully followed and many contacts were performed to understand what could be the next production release of Datagrid. A dedicated infrastructure was setup to install and test the successive releases hence providing valuable feedback to the developers and contributing to the reduction of the integration and test time. The infrastructure created to test the new middleware gave origin to a dedicated “test and validation” testbed fully separated from the initial testbed. The initial testbed evolved then into a dedicated production infrastructure. The “test and validation” testbed was built around 4 sites in Lisbon, Demokritos, Karlsruhe and Cyfronet with all sites contributing actively to the verification and quality control effort. The creation of dedicated testbeds was already foreseen and recommended in several validation and quality control documents.

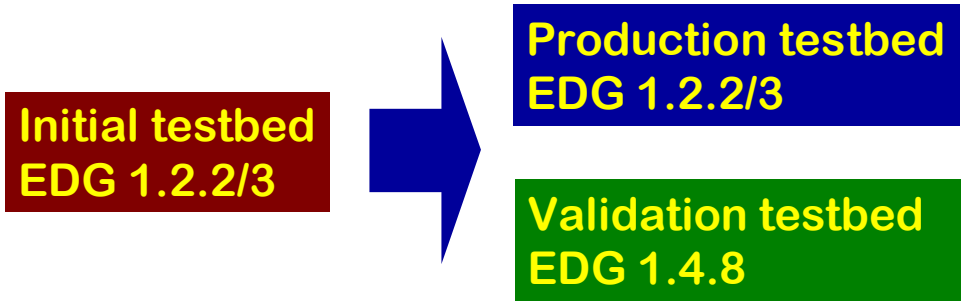

Figure 1. The initial testbed and its evolution.

The EDG release 1.4.8 is now considered stable and suitable for CrossGrid. Its validation phase has ended. In February a document describing the status of both production and development testbeds has been produced. This document describes many of the issues found during the tests and adds statistics that show clearly the benefits of upgrading to the EDG 1.4.8 middleware. This release is now being slowly deployed in the production testbed. The operation is delicate and is being carried out under the control of the verification and quality control task.

During February of 2002 the first CrossGrid components have been made available and integrated. Tests have also been performed on these components aiming to a first validation; reports have been produced on the integration and first tests with these components. A full integration with an installation kit should be available after EDG 1.4.8 is available at the integration sites, at that time deeper tests will be performed on the components aiming to their improvement and release into the production testbed. It is expected that versions 1.2.2/3 and 1.4.8 will coexist for some weeks until all sites perform the required upgrade and the certifications are performed.

Meanwhile the developments that will lead to the release of EDG 2.0 supporting RH 7.3.2 and LCFGng are now being followed closely. The first non-tagged releases have been deployed and are being tested in a new “test and validation testbed” that has been created for this purpose.

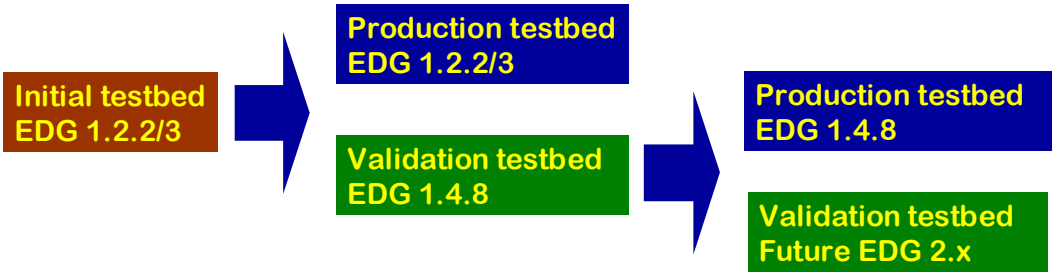

Figure 2. From the initial testbed until today.

CrossGrid is now on the cross road of moving from EDG 1.2.2/3 to 1.4.8 while it is already involved on the testing of the future EDG 2.x. Simultaneously it is preparing its own tools and middleware to run on EDG 1.4.8.

4 MIDDLEWARE TEST PROCEDURE

The test procedure followed by the verification and quality control task is described in the Middleware Test Procedure document released as annex of deliverable D4.1 in May of 2002 (CG-4-D4.1-004-TEST-1.1). In the document the validation cycle is defined as a set of interactions between the "verification and quality control team", the integration team and the developers. The developers can be from CrossGrid or from any other project providing middleware to CrossGrid. The integration team can also act as developer since as result of the integration process some "glue" software will be produced to integrated the components and also to install and configure them.

The objectives of the procedure as described in the reference document are:

- Reduce the middleware problems that would pass unnoticed to developers.
- Prevent the deployment of components with critical problems.
- Verify the conformance of the middleware with the specifications.
- Detect design errors.
- Identify potential bottlenecks.
- Detect middleware security issues.
- Detect documentation problems.
- Test the installation kits.
- Test the middleware components.
- Provide feedback to the developers.
- Produce recommendations to users and site administrators.

The cycle begins with a request from WP4 for testing a middleware release or component. After analysing the request the middleware is tested and validated in a dedicated testbed infrastructure. The dedicated infrastructure is required to ensure that nothing interferes with the validation tests and that the tests will not disturb the production testbed. Problems found during the validation are reported back to the developers directly. Small corrective actions can be accepted immediately and incorporated into the software and tested again. Once the tests are finished the

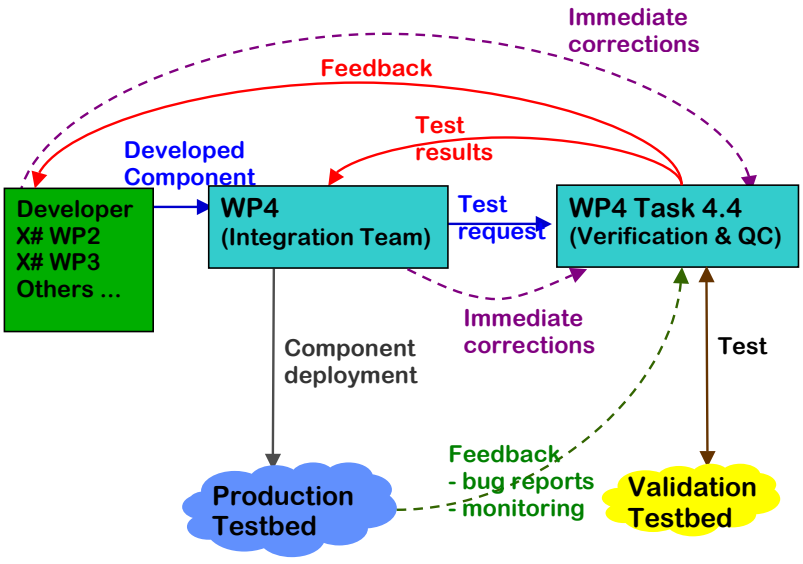

Figure 3. The middleware test procedure.

requester and the developers are informed of the results. If a release or component is found stable it can be deployed into the production testbed. Feedback about the middleware is collected from the production testbed in the form of bug reports and monitoring information to asses the behaviour of the testbed, improve the test procedure and methodology and finally spot components that may need improvement.

This procedure has been applied to the validation tests performed until now with good results. Providing immediate feedback on detected problems results usually in prompt actions by the developers, the cycle helps improving the stability of the middleware. The monitoring information and problem reporting (performed mostly by Email) has proven to be a good indicator for the stability of the testbed and to identify problems in deployed middleware components. Most of the tests have been performed over middleware developed by external sources. This is due mostly to the CrossGrid middleware and tools being still under active development, and also due to some delays from DataGrid on supporting the platforms and features desired by the CrossGrid tools and middleware. However a first round of tests has already been performed on a first release made available by the CrossGrid integration team.

Most of the validations performed have been quite long specially the ones performed on EDG middleware. The main reason is the time that it takes for DataGrid from the first minor version up to the release of the production version. During this time a series of unstable minor versions with corrections is released and tested until a stable version is reached. During this process many interactions and bug reports are required. A similar approach will be followed while performing further tests on CrossGrid middleware and tools.

The proposed test procedure foresees two phases. In the first phase tests are performed by the task 4.4 members. These tests are centred on the validation of the documentation, installation and configuration procedures and on ensuring the correct behaviour of the middleware using test programs written for this purpose. In the second phase the validation testbed is opened to a reduced set of application users and developers that will validate the middleware by running their applications hence assuring that the applications that were running in previous testbed releases are also capable of running in the release being tested. Currently CrossGrid applications are still under active development and thus can't be used to accurately test the middleware. Therefore only the first test phase is being applied. Using applications to test the middleware in the current evolution state of the project could trigger the detection of problems either in the middleware or in the applications and it would be very difficult to distinguish both. The applications and the middleware are also evolving rapidly therefore many new features are now available for which there is no reference to compare the behaviour. Once the middleware and the applications achieve a more stable development state the second test phase will be applied.

5 MIDDLEWARE TEST METHODOLOGY

The test methodology used in CrossGrid comprises unit tests to verify components separately, system tests to exercise and validate the integrated software and finally stress tests to verify the software under high load and intensive usage.

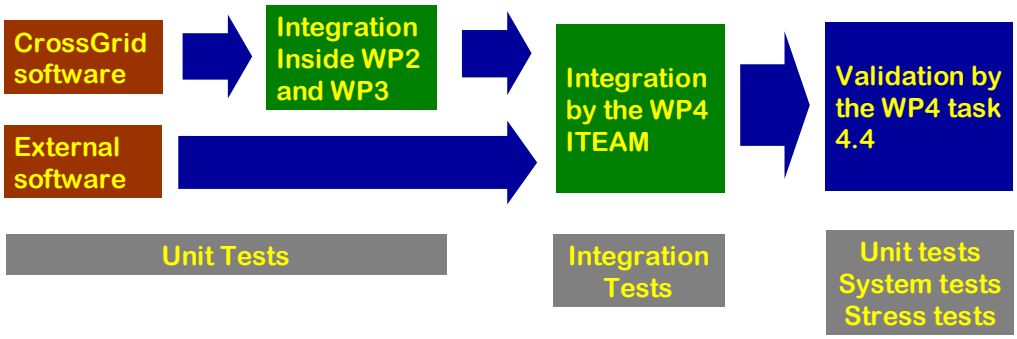

Figure 4. The test chain.

The software component developers should provide the unit tests. For CrossGrid software unit tests are developed and performed mostly by the corresponding development work packages namely through the developers and the integration task included in work packages 2 and 3.

However since the tests performed on the context of the task 4.4 are the last ones before the release of the software into the production testbed they can also be considered as acceptance tests. Therefore it is required that all software submitted for validation tests should be delivered with the corresponding unit tests. The tests can then be used and improved by the validation team. In the case of external components for which no unit tests exist they must either be developed or skipped. This is a difficult decision that must be carefully evaluated for each component since developing unit tests for external software can take too much time and impose considerable delays in the validation process, on the other hand if the component is recent and has not been sufficiently tested elsewhere unit tests might be required.

The integration team performs some integration tests during the software integration process. These are basic tests aiming to verify that the several components involved are capable of working together and result from the natural characteristics of the integration work.

The main goal of task 4.4 is the testbed validation and quality control hence the emphasis is put on the system and stress tests. These tests aim to verify the following software characteristics:

- Reliability
- Scalability
- Performance
- Security
- Installation and configuration
- Software documentation

The validation tests are performed and developed entirely by the test and validation team and try to exercise the integrated software has a whole instead of looking into each individual component as the unit tests do. The tests try to reproduce as possible real systems installation, configuration and usage situations.

The validation in task 4.4 starts with the analysis of the software documentation aiming to find possible problems such as errors and missing information. Then the software is installed and configured following the documentation. When correctly installed unit tests are performed followed by the system tests and the stress tests. The complexity and difficulty of the tests is incrementally increased. If possible the system tests and the stress tests are initially performed inside a single site and then extended to include other test sites.

Problems found are reported back to the developers according with the procedure. The results and experience gained during the validation tests also constitute valuable input for the deployment phase. The information is collected and used in the writing of the installation manual, support web pages and for providing support.

6 SITE VALIDATION PROCESS

CrossGrid has already a large testbed covering nine countries and sixteen sites. Since DataGrid spent its first year developing the first release of the EDG middleware it started to deploy and enlarge the testbed at the same time of CrossGrid, therefore both projects have similar experience in the deployment. Also some CrossGrid partners were involved and have followed closely the first DataGrid testbed trials hence all required deployment experience is available within the CrossGrid WP4.

CrossGrid has always aimed to make grid technologies as widely available as possible. One of the ways to achieve this goal is through a testbed covering as many countries and sites as possible. Therefore the first CrossGrid testbed had more countries and sites than DataGrid. In this sense CrossGrid faced the scalability issues of a wide testbed deployment before DataGrid.

Another reason for aiming to set a wide testbed managed apart from DataGrid is the nature of the applications developed and supported within the project. All applications require large amounts of CPU time and parallel processing. In this context parallel processing means a large number of CPUs communicating through MPI therefore a large testbed supporting MPI is required. DataGrid does not support MPI and never considered a large testbed as a priority, preferring to have smaller testbeds that could be easily managed. Although CrossGrid also uses EDG middleware distributions it is using it in different ways and also extending it to support MPI.

Supporting MPI across sites as in the case of CrossGrid demands a much more stable testbed than in classic single-processor computing. In fact in single-processor computing if a testbed site is badly installed or configured it could mean that jobs sent to it will be lost. This is not much of a problem since a Resource Broker can automatically reschedule the job to another site. Even if a Resource Broker is not used this means that a job will be lost but many others submitted to other sites will run without problems. This is not the case with MPI parallel jobs across sites. The experience gained with this kind of jobs shows that a single badly installed or configured site can cause much more problems. This is mainly due to the higher number of CPUs required. With MPI a single job may require a considerable amount of the testbed resources. If for some reason a job requiring a large number of CPUs across many sites is blocked due to a problem in a single site, this could mean that all resources allocated to the job can be locked and unavailable for some time. In this case jobs submitted later may not be able to find the appropriate number of CPUs. Even worse if there are still CPUs available at the site behaving badly then these CPUs can also block other large jobs. This kind of problems can render large portions of a testbed unavailable. Therefore it is essential to assure that all testbed sites are properly installed and configured. This is responsibility of the verification and quality control task and was considered from the beginning of the project as one of its most important activities to improve the testbed stability.

One of the most important steps towards assuring the correct site deployment is the site validation process. This is a procedure intended to validate testbed sites after

each major change, namely when a new middleware release is deployed after properly validated as explained in the previous sections.

The procedure starts with the installation and configuration of the site following the CrossGrid site deployment manual and the support web pages. The installation and configuration are performed with LCFG to reduce the number of possible installation and configuration problems. The software release is downloaded from the CrossGrid repository at FZK where the LCFG profiles of several sites are also available. The installation and configuration are responsibility of the site administrator.

Once the site is deployed the site administrator contacts the testbed administrators and provides the following information:

- Site administrator contact.
- The DNS domain name of the site.
- IP ranges for the grid nodes at the site.
- Name and IP address of the site Computing Element (CE).
- Name and IP address of the site Storage Element (SE).
- The certificate of the Storage Element.
- Name and IP address of the site User Interface (UI).

Some of the information is required for performing the site acceptance tests and later make the site available in the testbed while other information is required for security purposes, namely firewall configuration.

The site is added to the CrossGrid “mapcenter” grid monitoring service and the connectivity of the systems and services is tested. If all services are reachable the site CE and SE are added to the “Host Check” verification tool. This is a web enabled host verification tool that is capable of providing several installation, configuration and operation diagnostics for the CE and SE systems. Once the nodes are added to the “Host Check” web pages the site administrator can see by himself the list of problems detected and take appropriate measures to correct them. In this phase a strong interaction between the site administrators, the testbed administrators and the quality control is required to help site administrators quickly solve any detected problems and bring the site up to the required quality level. The interaction is also important to reduce the testbed deployment time and understand problems and situations that may have not been identified before.

Once all detected problems are corrected stress tests are performed on the site by submitting a large number of jobs and performing a large number of file transfers. These tests cover job submission through Globus and through the EDG Resource Broker.

When found stable the site is finally added to the list of official testbed sites and authorized to join the top MDS information index where all sites are registered.

The procedure has been successfully applied to the initial testbed running EDG 1.2.x and since then to all new sites. The procedure has been refined and improved and is now being applied to the deployment of EDG 1.4.x in the CrossGrid production testbed. The procedure has help to locate many problems that would pass unnoticed until the sites would be actually used by the users possibly creating major testbed disturbances.

7 TESTBED MONITORING

Sites must be continuously monitored both during the acceptance process and during operation. The objective of monitoring is to detect problems that can disturb the testbed and therefore contribute to instability. Monitoring also allows gathering statistics about the testbed that are useful to spot problems and evaluate its quality and usage.

The following tools are used to perform testbed and site monitoring:

- Mapcenter
- CE usage statistics
- RB usage statistics
- Host Check
- CRL verification tool
- MDS verification tool

These tools are more deeply explained in the next sections.

7.1 MAPCENTER

Mapcenter was developed by DataGrid and has been improved by CrossGrid with the support for usage statistics. Mapcenter has been a very important tool for testbed administrators, site administrators, general users and the verification and quality control. It provides a Global view of the testbed status in terms of network connectivity and network services availability. Mapcenter is very good at detecting network connectivity problems caused by the network layer, by firewalls or services being down. However it is not capable to verify whether a service that is answering is correctly configured or behaving well. Therefore other tools must complement its functionality. Mapcenter has also been enhanced with the visualization of CE and RB statistics provided by the CE and RB statistics modules developed within the CrossGrid WP4.

Mapcenter is used as the first diagnostics tool of the acceptance process and also as the first step while diagnosing site and testbed problems. The tool also stores historical information about alarms that is useful to understand past problems and study the testbed evolution in terms of connectivity problems.

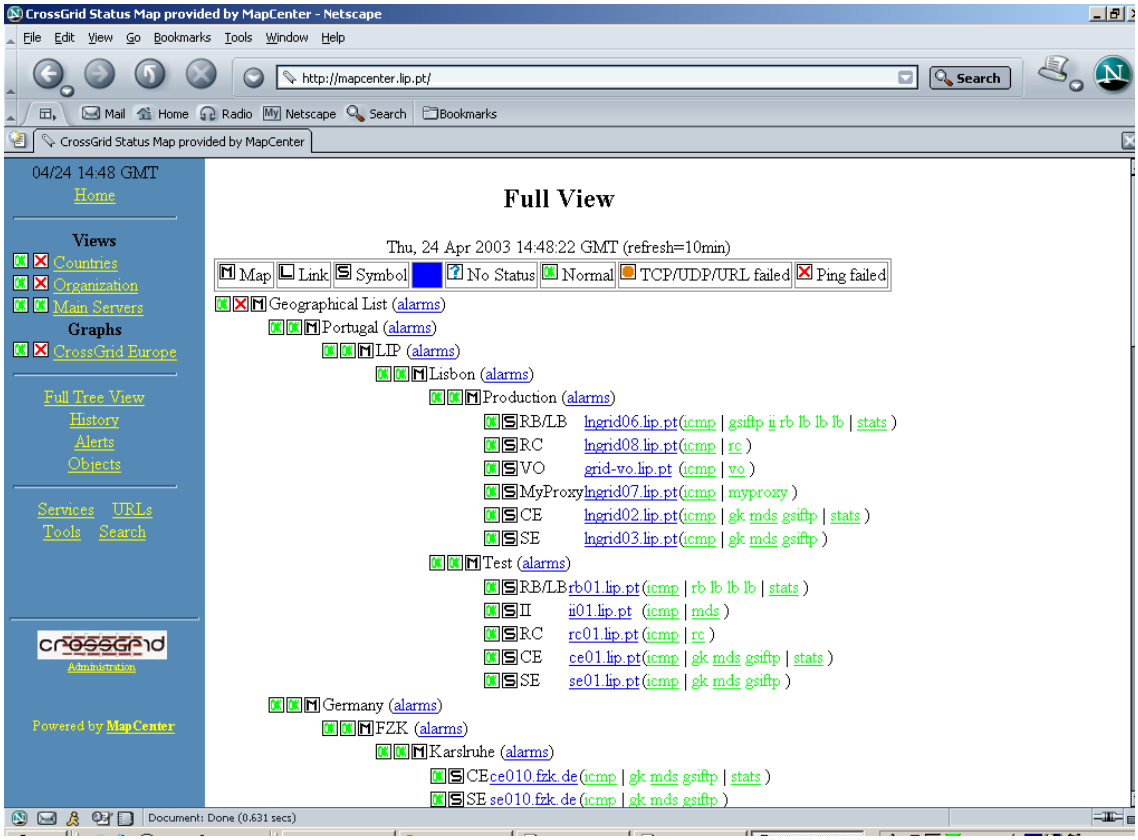

Figure 5. View of Mapcenter

7.2 CE STATISTICS

The CE statistics module developed in the context of CrossGrid WP4 is used to collect information about the jobs submitted to Gatekeepers. It obtains the Globus gatekeeper log file from each CE and processes it to produce statistics on jobs submitted and job submission errors. The information about the errors is extremely useful as an indicator of possible problems in the sites and elsewhere. It also provides a very good view on how the testbed is being used. Statistics collected at the level of the CE are very important for CrossGrid where the MPI job submissions have been performed so far directly through Globus, since the DataGrid RB does not support MPI and the CrossGrid RB is still under development. Thus accounting covering parallel job submissions can only be performed at the Globus gatekeeper level. A graphical web interface is currently being developed.

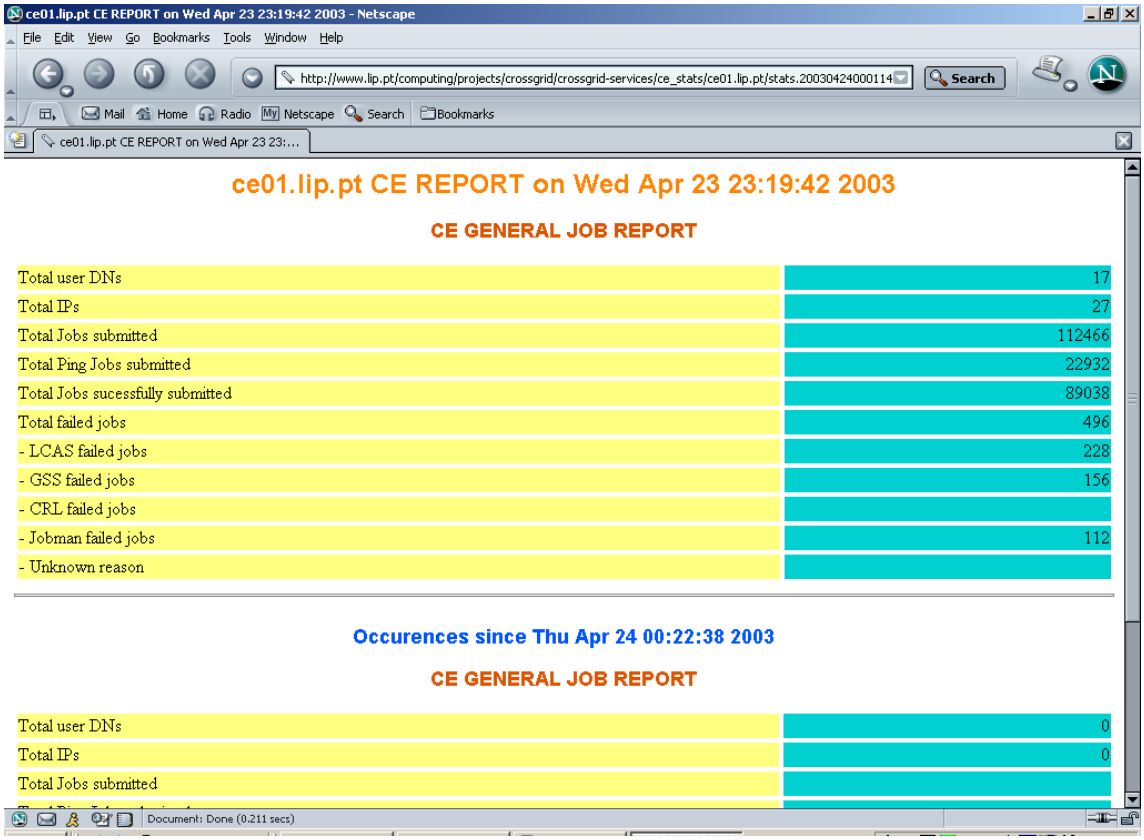

Figure 6. Computing Element statistics

7.3 RB STATISTICS

The RB statistics module developed in the context of CrossGrid WP4 is used to collect information about the jobs submitted through a Resource Broker. Although the EDG Resource Broker does not support MPI jobs it is foreseen that support for MPI jobs inside a cluster can be added. On the other hand CrossGrid wants to run MPI applications across clusters and is already doing so directly with Globus. CrossGrid is also improving the EDG Resource Broker so that it will allow submission of MPI jobs across clusters. Since MPI submission will be moved from using Globus to using a RB, software has been developed to gather statistics about RB job submissions and job submission errors. Meanwhile the software is used to monitor the production and validation RB's which currently do not support MPI. The statistics software runs inside each RB and produces statistics from the Logging and Bookkeeping relational database. There are also statistics components that produce web pages with tables containing the total usage and a second component that produces web pages with graphical output showing the usage and the errors in function of the time.

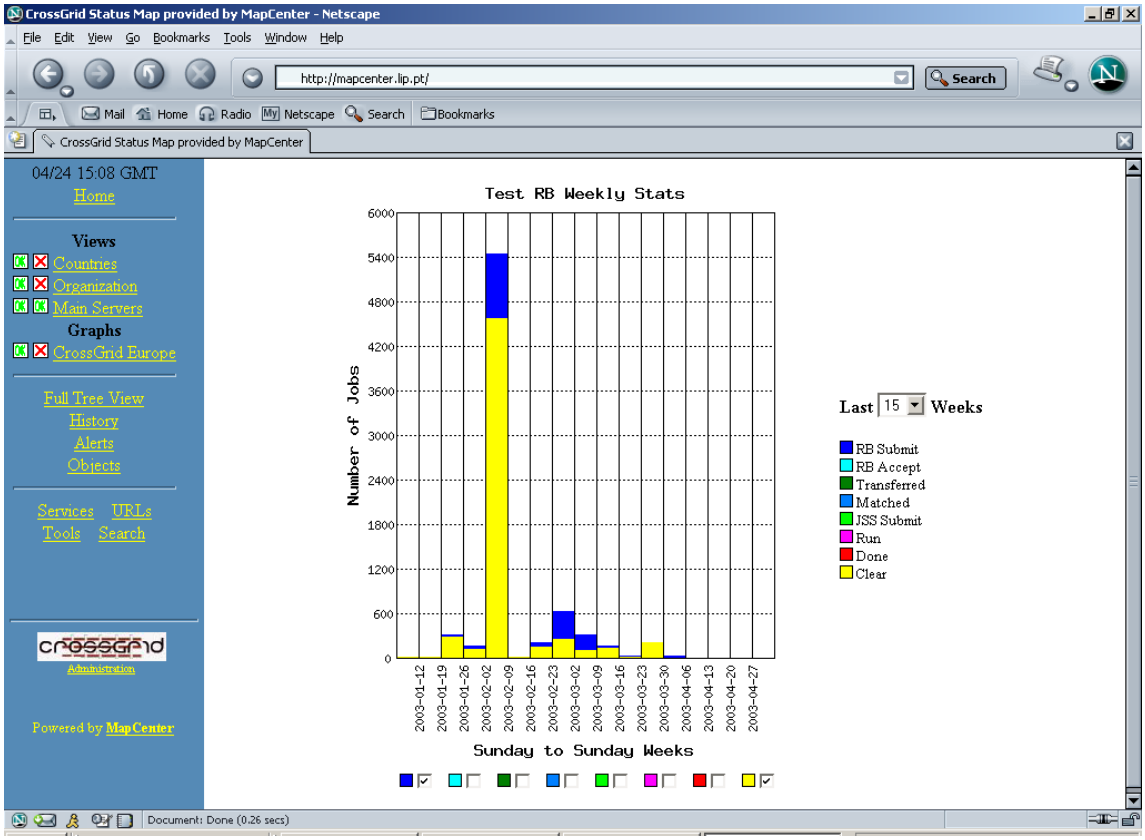

Figure 7. Resource Broker statistics: Graphical view

7.4 HOST CHECK

Host check is the main tool used in the site validation process. It is designed to verify the installation and configuration of Computing Elements and Storage Elements. Host check runs from a central system at LIP and performs its verifications by submitting jobs, performing file transfers and looking at the MDS published information. Host Check is capable of detecting most of the known configuration and installation problems. Its web interface allows a site administrator to schedule tests when desired and see the results. Site administrators can also compare the results of the tests performed on their sites with the results of other tests performed over other testbed sites. Unscheduled tests are performed for all testbed CE and SE systems twice per day. The current Host Check tool is an interim solution while more powerful and flexible tool is being developed.

7.5 MDS VERIFICATION TOOL

The MDS verification tool belongs to the second generation of tools being developed by the CrossGrid test and quality verification task. The tool is a replacement for the MDS tests performed by the Host Check tool. A beta version is already being used. It is developed to be more scalable and flexible, namely is capable of performing more sophisticate tests and can be more easily expanded. Since the MDS reflects most of

the host and site configuration the tool can be used to spot many configuration problems.

7.6 CRL VERIFICATION TOOL

The CE statistics allowed the detection of authentication failures caused by problems in the release of CRL's by the CA's. The CRL verification tool is another recently developed test tool. It aims to verify the issuance of Certificate Revocation Lists (CRL's) by the Certification Authorities (CA's).

The CA's must issue periodically a list containing the indexes of the revoked certificates this is called a CRL. The list must be signed and published into the corresponding CA web server. Each CRL has a maximum lifetime usually set to 30 days. The Grid nodes (CE's, SE's and RB's) must download these lists at least once a day. The lists must be installed in the system and used to verify whether the certificates used in authentication operations are still valid. The software that implements the authentication refuses all authentication attempts when the corresponding CRL corresponding to the certificates being authenticated has expired. The rationale behind this behaviour is that it is unsafe to perform authentication operations with information that is not updated since certificates could have been revoked and a CRL issued but not installed locally. Therefore it is essential to verify that the CA's are issuing the CRL's according with the rules agreed within the Certification Authorities task force.

The CRL verification tool performs CRL expiration checks by downloading the latest CRL's from the CA's web sites and verifying their expiration time. The tool is also capable of sending notification alarms to the CA's administrators when the safety time intervals for CRL issuance (issue the CRL at least 7 days before expiration) are not being respected. Currently the tool is monitoring CrossGrid and DataGrid recognized CA's.

8 MAKING THE INFORMATION AVAILABLE

The information gathered by the verification and quality control task is made available to the CrossGrid community in several ways. Detected bugs, software and documentation problems are reported directly to the developers. Site problems are reported to the site administrators and to the testbed administrators.

Problems that require immediate actions and are of relevance for the whole testbed community and reported through the CrossGrid mailing lists maintained at IFIC.

crossgrid-wp4	General testbed mailing list.
crossgrid-wp4-sites	Site administrators mailing list.

Advices to users and site administrators are published in the testbed deployment web pages maintained at http://www.lip.pt/computing/crossgrid-services. Web pages produced by verification tools have their own URLs but can be reached from the testbed deployment web pages. The URLs for the verification and quality control related web pages are the following:

Mapcenter	http://mapcenter.lip.pt
CE statistics	http://www.lip.pt/computing/crossgrid-services/ce_stats
RB statistics	http://www.lip.pt/computing/crossgrid-services/rb_stats http://www.lip.pt/computing/crossgrid-services/rb01_stats
RB statistics (graph)	http://mapcenter.lip.pt/php/plotstats.php?db=rblbserver http://mapcenter.lip.pt/php/plotstats.php?db=rb01lbserver
Host Check	http://www.lip.pt/computing/crossgrid-services/site_check
MDS verification tool	NOT AVAILABLE TO THE INTERNET
CRL verification tool	NOT AVAILABLE TO THE INTERNET

9 METRICS

As stated before one of the objectives of the development of statistics software and of the monitoring activities is to provide metrics for the testbed stability. Metrics are essential to evaluate correctly the behaviour of the testbed and help decide whether actions to improve the testbed are needed and where those actions should concentrate to have the maximum benefits.

In the first testbed releases the activities have been centred on verifying as extensively as possible the testbed components before deployment and during operation. This has been a difficult and time consuming effort since the middleware used as the basis for the old initial and production testbeds has based on EDG 1.2.x which had many problems and insufficiencies. With the deployment of the new production release CrossGrid is entering in a new phase where the quality of the services provided by the testbed will be very important to test more complex and demanding software prototypes. The first testbed was mainly intended to demonstrate the possibilities and the concepts, now it is expected that the testbed will provide a near production environment. Therefore it is necessary to define metrics and deploy the measurements mechanisms necessary to implement those metrics. Four areas have been identified which can be decomposed into more specific metrics. They the following:

- Anomalies evolution.
- Service reliability.
- Service efficiency.
- Support efficiency.

Obtaining metrics from a Grid testbed is a difficult task. Most middleware components are still under development and don't have yet all the necessary functionalities to produce the information required for the metrics. Therefore in many cases only the end user will know whether a specific action has succeeded or failed. Even when information is available it is frequently impossible to know whether an operation has failed due to a middleware problem or due to an erroneous user input or action.

9.1 ANOMALIES EVOLUTION

This is the evolution of problems reported by the testbed users. It can be obtained from the problem reporting tools, submitted either by Email or through the helpdesk application. Anomalies can be categorized by problem type as in the next list.

- Globus job submission problems.
- RB job submission problems.
- MPI Globus job submission problems.
- MPI RB job submission problems.
- Replica management problems.
- Site deployment/maintenance problems (sent by site administrators).

- VO problems.
- Information system problems (MDS).
- Portal and user interface problems.
- Monitoring tools problems.
- Application development and performance analysis tools problems.

In some cases the problems can be further decomposed by the specific middleware component. The categorization of the problems must be performed after a positive diagnostic is achieved. The support experts will perform the diagnostics.

The metrics will show the number of problems submitted per month for each of the categories.

9.2 SERVICE RELIABILITY

Reliability reflects the number of problems that occur while performing actions on the testbed. The information to calculate reliability metrics is hard to obtain since many errors aren't even logged by the corresponding components. However for some components this information is produced and can be treated to produce some metrics. The following are reliability metrics that will be calculated.

- Globus job submission reliability: obtained from the Globus gatekeeper log files. This metric can be further decomposed by the type of job submission error.
- RB job submission reliability: obtained from the RB logging and bookkeeping database.

Even with information available these metrics are difficult to calculate. For instance its not possible to obtain the number of total job submission attempts for a Gatekeeper since this number contains also the TCP port scans which sometimes are even bigger than the actual number of real job submissions. The logging and bookkeeping database can also contain incomplete information since it is not only originated from inside the RB but also from the CE's and the UI's. Even so these can be very good metrics to evaluate the testbed reliability and overall behaviour.

9.3 SERVICE EFFICIENCY

The efficiency of the testbed services is also difficult to measure. For a heavy loaded testbed the number of jobs processed per unit of time can be a very good metric. However it is not expected that the CrossGrid production testbed would be heavy loaded, therefore the efficiency measured this way would be very high and would not mean much. The emphasis for the CrossGrid production testbed is on providing reliable and dependable services for application and tool developers, and not on providing a true high performance production service. Therefore the only efficiency metric that can be of any value is the services down time.

- Services down time: This measure can be partially obtained from Mapcenter. With Mapcenter it is possible to verify whether a specific TCP port is answering but not to verify whether the service is actually behaving well.

9.4 SUPPORT EFFICIENCY

This information can be only provided when the problems are submitted through a helpdesk interface that keeps track of the problem submissions and corresponding answers. With such system two metrics can be obtained:

- Time that it takes to provide answers to the submitted problems.
- Number of pending requests for which no solution has been found.

10 PROCEDURES

The verification and quality control task has contributed to many WP4 activities besides the ones already mentioned. The objective is always to contribute to increase the quality of the testbed services provided within CrossGrid. In this context task 4.4 is developing and proposing procedures covering:

- Security.
- Site acceptance.
- Testbed operation.
- Middleware validation.
- Testbed extension.

10.1 SECURITY

A security group has been recently established within CrossGrid. Site administrators compose the group with experience in system and network security. Members of the verification and quality control are also present in this body that aims to take actions and define guidelines, procedures to improve the testbed security.

10.2 SITE ACCEPTANCE

The site acceptance process has been defined and conducted by the verification and quality control (task 4.4) in close contact with the testbed setup and incremental evolution (task 4.1).

10.3 TESTBED OPERATION

Guidelines for the management of VO's, monitoring of testbed sites and services and maintenance of central services are being applied.

10.4 MIDDLEWARE VALIDATION

A procedure to test the middleware produced both within the project and also from external sources as defined. The procedure is discussed in the first sections of this document and as been released as annex of deliverable D4.1 in May of 2002 (CG-4-D4.1-004-TEST-1.1).

10.5 TESTBED EXTENSION

A proposal on how to deal with the testbed extension to cover new sites not initially foreseen and also users external to the project has been produced. The proposal is being evaluated by the steering group and has result already in the creation of a new VO for external users.

11 EXPERIENCE AND TEST OF EDG RELEASES

Prior to the official start of the Crossgrid project, first tests were performed with the setup of a small testbed. This phase was considered a “learning phase” which allowed not only to test the Globus toolkit and EDG middleware, but also to learn its usage. It also allowed gaining experience with the automatic installation of testbed machines through LCFG. This first phase was quite hard due partly to the instability of the released software. Close collaboration with the Datagrid Integration Team helped in the solution of bugs and towards more stable versions of the EDG middleware.

An installation server with the LCFG software was deployed with also the required supporting services such as DHCP and PXE for IP address distribution and network boot. Several tests were performed on how to establish the best possible set-up for the installation server since this would be a fundamental tool in the months to come.

At the start of the Crossgrid project there was already some experience with the setup and operation of a testbed with a minimal number of elements namely: a Computing Element, a Storage Element, a Worker Node and a User Interface. This experience eased the first testbed deployments across the Crossgrid sites.

The evolution of LCFG has been towards more automatic installation and configuration of an increasing number of different testbed elements, with the time frame for installation/upgrade/configuration always decreasing, and easing the testbed management task. This allowed having more time for the development of test and validation tools and towards an automatic validation procedure for the deployment of a new site, or the upgrade and test of new middleware.

11.1 TESTS PRIOR TO EDG RELEASE 1.2.3

The LCFG server was the first machine to be deployed. This testbed element is of utter importance since it allows automatic installation, upgrade and configuration of the other testbed elements. It also eases the task of maintenance and management of those other elements.

The LCFG server publishes XML profiles containing the type and configuration of a given element. It is also a software repository through which the other elements are installed and upgraded. The software to be installed and/or upgraded is given in file containing a list of packages for a given element. Given that the supported platform is RedHat, the software are packaged with RPM.

Although the first deployment of the LCFG server was not to dificult, when it came to the point of installing the first LCFG client (a Computing Element) many problems have arisen. Those problems can be divided into the following categories: incorrect or missing steps in the documentation, bugs in the LCFG software and software package lists with wrong/missing packages.

In the first installation of a Computing Element we encountered several severe bugs. Since in this early release the platform was RedHat 6.2, the kernel used for initial installation did not recognized new Intel processors like the Pentium IV and the module for the ethernet card was missing in the kernel. Workarounds were found to overcome such bugs which included the substitution of the ethernet card with one which was supported by the kernel and modification of the installation script in the LCFG server which was distributed by EDG. Those bugs were reported back and solved in later releases. In the next step the RPM lists had to be modified according to the packages and versions actually in the repository with additional packages to satisfy dependencies. It should be noted that problem is almost allways present whenever there is a new release.

Another issue which has to be carefully followed are the configuration profiles for each element. In principle, only two profile files have to be modified to reflect the configuration for a given site. This is not allways true since the other profiles (which should be static for a given release) had the configuration variables hardwired. This leads to the verification of every profile whenever there is a new release increasing the amount of time for an upgrade. This has been more than one time reported back to Datagrid and presently there is almost no need to touch in those other profiles. Presently, the configuration for a given site is made in only two profile files specific for that purpose.

After sucessful instalation of the Computing Element, it remains some things which have to be done manually, namelly: a host certificate had to be requested and installed, the file containing the authorized users had to be generated, the pooled accounts had to be activated in a special way not mentioned in the documentation. Some of the deamons had to be started manually and configured to start after the machine rebooting. Finally the job scheduler must also be configured and started manually, although this was performed only after the installation of a Worker Node. The job scheduler which has been used is Open PBS.

The time lost in the installation and configuration of the first LCFG client was time gained in the installation and configuration of the next elements, namelly : a User Interface, a Worker Node and a Storage Element. New problems arised specially in understanding the configuration and operation of the SE in what concerns the GDMP. There is another configuration file in which authorized SE's are inserted, users of a given VO are allowed to copy/replicate files between our SE and those SE's.

The PBS job scheduler was configured in the CE (the server) and in the WN (the client). At this time simple tests were perfomed through the Globus Toolkit. Submission of jobs through the '*globus-job-run*' command to the CE and SE using the '*fork*' jobmanager and to the CE using PBS. File transfer through the '*globus-url-copy*'. Queries to the GRIS's of the CE and SE through '*ldapsearch*'. File replication/mirroring using GDMP commands between our SE and the SE's at other sites in Europe.

Those tests allowed us not only to learn how to operate this small testbed but also to detect the kind of problems which could arise. It should be noted that at this time we were not registering in any Resource Broker, so the EDG midleware was not being tested yet.

In the next step we asked to Integration Team of Datagrid for our CE to be included in their RB. At this point some problems were detected mainly in the configuration of the PBS. This was possible due to a close collaboration between both projects. Tests of job submission through a RB to our CE using the EDG midleware proved sucessful.

Meanwhile we started the deployment of a RB and VO server for the Crossgrid project. A large fraction of the RB configuration has to be made manually after installation from the LCFG server. This includes seting up and configuring a MySQL database, a PostgreSQL database and Condor, since they are used by several services running in the RB. The Information Index also runs on this machine, this service gathers the information from all Gatekeepers which are allowed to publish it in the RB.

A MyProxy server machine was also deployed. This Service is needed for long lived jobs, when the initial user proxy does not have a suficiently long time interval for a given job to run.

The RB, VO server and MyProxy server are elements providing central services for a complete testbed.

11.2 EDG 1.2.2/3 WIDE DEPLOYMENT AND TEST

Crossgrid sites were asked to deploy the EDG midleware in their testbeds. After deployment, the site administrators performed basic tests over their testbeds, namelly : basic job submission with globus, file transfer with grid-ftp and ldap queries to the gatekeepers. Meanwhile the administrators of the central services also performed those basic tests and also some check on the configuration parameters. After passing this tests the site was added to the RB and Information Index, the respective SE certificate was added to the Crossgrid VO in order to be acessible throughout the testbed for file replication and mirroring.

Although initially a given site could be OK, this does not mean stability along time. One of the most serious problems was that if a given site failed for some reason to publish all its information to the Information Index, or a timeout ocurred, it was suficient for the whole testbed to crash, i.e., the RB would give an error when submiting jobs to it. On the other hand, even when the job submission was sucessful it could take a quite long response time since it is required a ldap query to all sites.

Moreover, a given site goes in or out of the RB through the addition of a few lines in a configuration file, so that the verification if all sites were responding was very time consuming.One of the most early scripts to be developed was an automatic ldap query to all sites with verification if all the information was present, if so that site was

added to the configuration file, at the end the script restarted the deamon of the Information Index. Sites for which the test failed an automatic mail was sent to the respective administrator to take proper measures. A list of common problems is given below.

Another essential element was deployed in the testbed, namelly a Replica Catalog machine. This element is necessary to keep track of the files which are kept in the different SE's. It runs a ldap server containing information about the files and storage location.

The only documentation available for the installation instructions of such machine was in the form of a 'README' file contained in the ReplicaCatalog server package. There was no LCFG profile for that element and the RPM list was incomplete. It was our job to make a minimal LCFG profile and a RPM list containing the necessary software packages, assuring at the same time that no conflits would arise in the installation. Both the profile and RPM list was sent to the Datagrid Integration Team to be in a later release.

Direct contact with Datagrid members allowed us to sucessfully install and configure the Replica Catalog by 'filling' the blank lines of the installation, configuration and catalog initialization procedure. This element is presently running the Crossgrid Replica Catalog for the whole testbed. Catalogs for the ATLAS and CMS VO's were also created for testing puroposes, since there are official catalogs for those VO's running in the Datagrid testbed.

Tests were sucessfuly performed by copying files between SE's and registering them in the Replica Catalog.

11.2.1 Common problems with Gatekeepers

Several configuration problems may prevent a job submission to a Gatekeeper to succeed. Below a list of some of these issues can found.

- The TCP port 2119 used by the Globus GRAM service might be blocked for inbound access.
- The TCP port 2135 used by the Globus MDS service might be blocked for inbound access, this only affects jobs submitted through the RB.
- The gatekeeper hostname and its DNS reverse do not match.
- The user certificate distinguish name may not exist in the **gridmapfile**.
    - The user submitting the job is not registered into any of the VO's recognized by the gatekeeper.
    - If the daemon **edg-gridmapfile-upgraded** is running the **gridmapfile** is automatically rebuilt from the VO servers specified in mkgridmap.conf. This means that any modifications made by hand to the **gridmapfile** will be lost each time the daemon does a rebuild.

    - If the daemon **edg-gridmapfile-upgraded** is not running then any changes performed to the VO servers content or in the **mkgridmap.conf** will not be reflected into the gridmapfile.
- The user certificate distinguish name may be mapped to an invalid Unix account.
- The CRL of the certification authority that signed the user certificate may have expired in the gatekeeper.
- The polled accounts the directory **/etc/grid-security/gridmapdir** may not exist. When using pooled accounts this directory must contain one empty file for each username used by the pooled accounts. Each file must be named with one username, it must be empty and be owned by the root user.
- The directory **/etc/grid-security/gridmapdir** must be shared between the worker nodes, gatekeeper and storage elements (this applies to the SE only if the SE shares directories via NFS or AFS with the WN's).
- The home directories are not being shared by NFS. All grid user accounts must be shared between the worker nodes and gatekeeper through NFS.
- The **flatfiles** is not being shared. This directory must be shared through NFS between the SE, the worker nodes and the gatekeeper.
- The user corresponding CA certificate is missing in the gatekeeper. The "root" certificate corresponding to the certification authority that has signed the user certificate is not in the gatekeeper directory **/etc/grid-security/certificates**. This can be due to:
    - The CA not being administratively accepted/recognized at the gatekeeper site.
    - Site administrator failure to install the CA certificate.
- Problems with the local scheduler. The batch scheduler used by the gatekeeper to distribute the processing load by the worker nodes may not be running or configured properly.

11.2.2 Problems that may affect the transfer of the input and output files (sandbox):

- The worker node may not have outbound TCP connectivity. Outbound connectivity is required to transfer both the input and output files of jobs submitted through the RB.
- The information in the worker node directory **/etc/grid-security/certificates** may be outdated. The directory contains all the certification authority certificates and most recent CRL's. The gatekeeper should share this directory through NFS with all worker nodes. This is required because the worker nodes do not run **edg-crl-upgraded** daemon, therefore the daemon must be executed by the gatekeeper and the CRL's shared with the WN's.

11.2.3 Problems configuring the MDS

- When PBS is down information regarding the CE object class **"ComputingElement"** will not be published.

- If PBS is configured but the PBS queue does not point to any worker node then the information regarding the CE object class "**ComputingElement"** also will not be published. Therefore all sites must have at least one worker node.

11.2.4 Known limitations of this release

The RB is known to lose long-lived jobs, which outdates the time of the proxy certificate.

When a very high rate of jobs is submitted to the RB, a large fraction of those jobs are lost, this is known as stress testing the RB. The RB statistics are one way to demonstrate this.

If one site is down or not giving the Resource Information to the Information Index it can render a total breakdown of the whole testbed.

Overall this EDG release is not yet very scalable.

Despite these limitations, the EDG 1.2.2/3 is presently the production release in the Crossgrid testbed.

11.3 EDG 1.4.8 TESTING

After the EDG1.2.3 release, it became available the 1.3.x series. These series of releases never became the production one since they were quite unstable. More stable EDG midleware became available with the 1.4.x series. Four Crossgrid sites (LIP Lisbon, Demokritos Athens, FZK Karslruhe and Cyfronet Krakow) are part of the ‘test and validation’ testbed.

At all sites the following elements were deployed: a CE, a SE, one or more WN’s and a UI. Additionally at LIP a RB and a RC were also deployed. From the initial 1.4.0 release, these sites have continually tested the several releases up to the present one EDG 1.4.8. These series are still based in RH 6.2 and LCFG so the same LCFG server was used for this testbed at least in some sites.

The middleware released in these series is incompatible with EDG 1.2.2/3.

11.4 INTEGRATION/TESTING OF CROSSGRID MIDLEWARE AND APPLICATIONS

The first Crossgrid Integration Meeting allowed the first tests of some tools being developed by WP1, WP2, WP3 and WP4 in the testbed. A large number of issues were

11.5 TOWARDS EDG 2

A new testbed