DELIVERABLE D4.9

TESTBED FINAL REPORT

WORKPACKAGE 4

Document Filename:	**CG4-D4.9-v1.2-CSIC021-WP4-Testbed-Final-Report.doc**
Work package:	**WP4 International Testbed Organisation**
Partner(s):	**CYFRONET, ICM, INS, UvA, II SAS, FZK, PSNC, UCY, TCD, CSIC, UAB, USC, DEMO, AUTh, LIP**
Lead Partner:	**CSIC**
Config ID:	**CG4-D4.9-v1.2-CSIC021-WP4-Testbed-Final-Report**
Document classification:	**PUBLIC**

Abstract: The International Testbed of the CrossGrid project has been in operation for the last three years, including 16 sites in 9 countries across Europe, providing a framework for the development of middleware and tools in the project, and the execution of interactive and parallel applications. This document describes the status of the final testbed, and reports on the achievements in the different tasks: installation, deployment of software releases and other support activities, test and validation, coordination with EDG and EGEE projects. The integration effort is summarized, and the support for final demos is also discussed. Finally, collaboration experience and concluding remarks, including open questions, are presented .

Information Society Technologies

Delivery Slip

	Name	Partner	Date	Signature
From	Rafael Marco	CSIC	14-Jan-2005	
Verified by	David Rodríguez	CSIC	25-Jan-2005	
	Robert Pająk	CYFRONET	2-Feb-2005	
Approved by	Jesus Marco	CSIC	2-Feb-2005	

Document Log

Version	Date	Summary of changes	Author
0.1	14-Jan-05	Initial draft and template	Jesús Marco
0.2	18-Jan-05	Updated after final demo definition	Integration Team
0.3	20-Jan-05	Updated after reading D2.7, D3.7	Jesús Marco
0.4…0.9	21-Jan-05	Included revised validation	Integration Team
1.0	25-Jan-05	First draft release	Jesús Marco
1.1	31-Jan-05	Testbed statistics included	Jesús Marco, Jorge Gomes
1.2	2-Feb-05	Updated with reviewers comments	Jesús Marco
	2-Feb-05	Verified by the Quality Engineer	Robert Pajak

CONTENTS

1. INTRODUCTION 4

OVERVIEW 4
ON-LINE REFERENCES 4
DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 5

2. STATUS OF THE FINAL TESTBED INFRASTRUCTURE 6

3. COORDINATION WITH THE EDG, LCG AND EGEE PROJECTS 10

4. SUPPORT ACTIVITIES 11

5. QUALITY ASSURANCE AND VALIDATION SUMMARY 12

Testbed sites validation 12
Software test and validation. 16

6. INTEGRATION EFFORT 18

7. FINAL DEMOS: INTEGRATION AND TESTBED ISSUES 20

8. COLLABORATION EXPERIENCE 21

9. EXECUTIVE SUMMARY 22

10. CONCLUSION: PERSONAL REMARKS 24

Open Questions 25
What next? 25

1. INTRODUCTION

OVERVIEW

The CrossGrid project is approaching its exploitation phase. After a long, complex, and successful work, the project has reached its main objectives: a testbed with distributed computing resources in sixteen sites across nine different European countries has been setup, and four relevant applications in different areas with significant computing and storage demands and different interactive requirements have been integrated in this grid environment, using tools and middleware developed for this purpose.

The final report and demo description is the latest WP4 deliverable, although further work is being done regarding the exploitation plan and the technical implementation plan that will be only briefly introduced here.

Several points are covered in this document:

- a brief description of the final testbed infrastructure and its status (the reader is referred to the updated internal deliverable D4.8 for further details)
- the summary of coordination with other relevant initiatives like DataGrid and EGEE.
- the support offered to the different communities: users, developers, site responsibles
- the results of the validation effort
- a summary of the integration achievements (see updated Appendix in D4.7)
- the brief description of the demos from the testbed point of view, and an Appendix compiling the details
- the "collaboration" experience
- an executive summary, fully open, trying to transmit the lessons learned and pointing to the issues where further work is needed
- and finally, the possibilities for the short and mid term future

The document has been prepared by the end of January 2005, in time for the final project review in Amsterdam, but it will be updated with the very final details right after or along the Amsterdam meeting. Note that the length of each section has been limited on purpose, but further details can be obtained either in the previous deliverables (including the two previous internal deliverables, D4.7 and D4.8), or from the on-line documentation compiled below.

ON-LINE REFERENCES

- The WP4 main web page at IFCA
  http://grid.ifca.unican.es/crossgrid/wp4
  including the testbed tutorial and examples:
  http://grid.ifca.unican.es/crossgrid/wp4/Tutorial/CG4-Tutorial.pdf
- CrossGrid HelpDesk at IFIC
  http://cg1.ific.uv.es/hlpdesk/
- Testbed installation and support at LIP
  http://www.lip.pt/computing/projects/crossgrid/crossgrid-services/installing.htm
  and validation web pages:
  http://www.lip.pt/computing/projects/crossgrid/task4/middleware-validation.htm

- Integration Team Web Page, Forum, and Repository in GridPortal, at FZK
  http://gridportal.fzk.de/websites/crossgrid/iteam/
  including:
  - Developers' guide
    http://savannah.fzk.de/websites/crossgrid/iteam/devguide/
  - Code Review recommendations
    http://gridportal.fzk.de/websites/crossgrid/iteam/CG4-D4.7-Code-Review.doc
- Sites Installation Guide at UoA
  http://cgi.di.uoa.gr/~xgrid/cgfiles/LCFGng_v2.0.pdf

DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

CrossGrid/X#	The EU CrossGrid Project IST-2001-32243
DataGrid/EDG	The EU DataGrid Project IST-2000-25182
EGEE	The EU EGEE Project (6FP contract 508833)
Grid	Grid framework for sharing of distributed resources.
MPI	Message Passing Interface.
MPICH	A portable implementation of MPI (CH stands for Chamaleon)
WP	Work Package
VO	Virtual Organization
CE	Computing Element (EDG terminology).
SE	Storage Element (EDG terminology).
API	Application Programming Interface
RPM	Red Hat Package Manager
LCFG	Local Configuration Tool

2. STATUS OF THE FINAL TESTBED INFRASTRUCTURE

In its final configuration, the CrossGrid international distributed testbed shares resources across sixteen European sites ranging from relatively small computing facilities in universities to large computing centers, offering an ideal mixture to test the possibilities of the Grid framework. National research networks and the high-performance European network, GÉANT, assure the interconnectivity between all sites. The following figure shows the map with the different nodes, including the major network links.

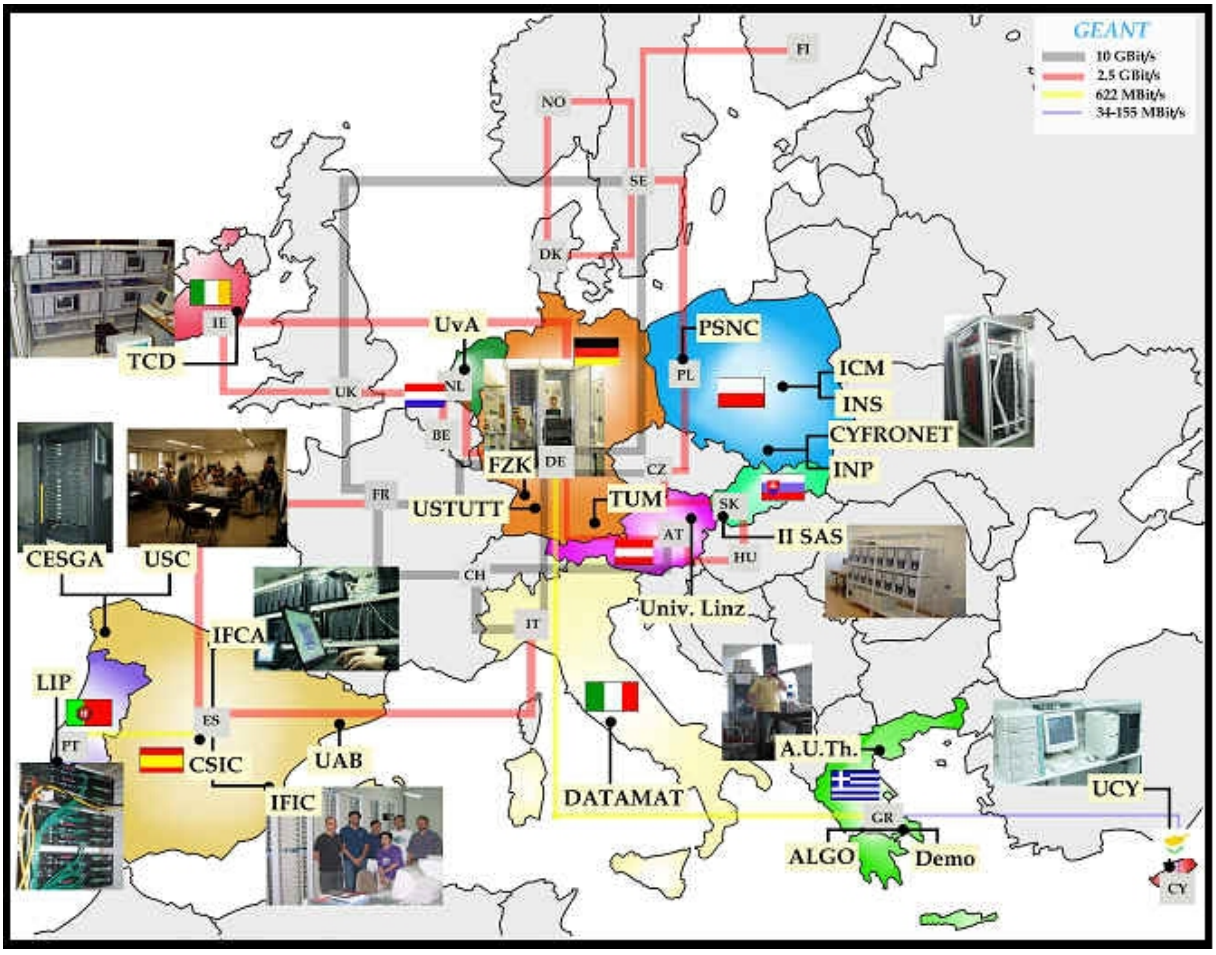


Figure 1 The CrossGrid testbed

The Grid framework provided on this infrastructure is based on basic job submission services, information systems, authentication, authorization and data transfer provided by the Globus Toolkit. These services are extended by a workload management system, replica location services, improved information system and Virtual Organization management system developed by the EDG project, incorporating also other middleware components from packages like Condor. This software is currently distributed in a version known as LCG-2, assembled and supported by the LHC Computing Project at CERN, and widely installed also in the EGEE project centers.

On top of these basic layers, CrossGrid has developed its own set of services:

- The user-friendly Migrating Desktop (MD) to access the Grid environment, complemented by a Roaming Access Server (RAS) and integrating support for Interactivity.
- An improved workload management system (WMS) with MPI local and distributed support, and prioritisation mechanism.
- Monitoring systems for infrastructure (JIMS), network traffic (SANTA-G), application execution (OCM-G), and resource usage prediction (GMDAT)
- Data access optimisation (UDAL).

All these services are used in the programming tools developed within the project to help application teams to tune their software for the grid environment:

- Grid benchmarking tools (GRIDBENCH)
- Performance Prediction for application kernels (PPC)
- Performance Monitoring tool for distributed applications (GPM)
- MPI code debugging and verification tool (MARMOT)

Finally, the testbed provides the Grid environment where four parallel and interactive applications on different areas (BioMedicine, Flooding Crisis Management, Particle Physics, and Meteorology and Air Pollution) have been developed making use of these services and tools. The testbed also provides the framework for the test and execution of these data and compute intensive applications.

For all these purposes, the final version of the CrossGrid testbed includes two distinct setups. The **"development"** testbed, with limited resources in five different sites, supports deployment and test of new software, that once validated is deployed in a larger and more stable **"production"** testbed, where the applications are executed. The following table summarizes these contributions.

Site	***Location***	***Testbed***
AUTH	Thessaloniki – Greece	Production + Development
CESGA	Santiago – Spain	Production
CYFRONET	Cracow – Poland	Production
FZK	Karlsruhe – Germany	Development
ICM	Warsaw – Poland	Production
IISAS	Bratislava – Slovakia	Production
IFCA	Santander – Spain	Production + Development
IFIC	Valencia – Spain	Production
INS	Warsaw – Poland	Production
LIP	Lisbon – Portugal	Production + Test
PSNC	Poznan – Poland	Production
TCD	Dublin – Ireland	Development
UAB	Barcelona – Spain	Development
UCY	Nicosia – Cyprus	Production
UoA	Athens – Greece	Production + Test
Demokritos	Athens – Greece	Production + Development
UvA	Amsterdam – Netherlands	Production

Table 1 Testbed sites

The integration of these resources is accomplished through the global collective services:

- **Resource Broker** (RB): implementing the workload management system, the CrossGrid RB supports single and parallel jobs (both inside a cluster via MPICH-P4 or distributed across several sites using MPICH-G2).
- **Information Index** (II): the root entry point for the MDS information tree that contains the resources information published by the CE and SE systems.
- **MyProxy**: repository of certificate proxies for long-lived jobs and portals.
- **Virtual Organization** server: repository of authorization information used by the testbed.
- **Replica Location** Services (RLS): storing information about the location of physical files. The RB uses the RLS to make scheduling decisions, and users can also access it through the Replica Manager (RM). The original EDG service has been extended regarding optimization.
- **Mapcenter**: an EDG tool enhanced by CrossGrid to monitor the connectivity.
- **Roaming Access Server** (RAS) and its **Job Submission Service** (RAS JSS): CrossGrid specific services supporting the Migrating Desktop (MD) and portal applications.
- **Grid Monitoring Data Analysis Tool** (GMDAT): a service to monitor and predict the usage of grid resources and the network bandwidth between sites.
- **Relational Grid Monitoring Architecture** (R-GMA): an EDG service to enable the publishing of monitoring information in distributed grid enabled relational database tables.

The global collective services are located mainly in four sites. The main production central systems are hosted by LIP, with the production RB installed in a Geant point of presence at FFCN improving the network connectivity and resilience of this critical service. The main global services for the development testbed are located at FZK in Germany. Duplication of the most critical central services is hosted at the IFCA site.

Service	Host	Site	Testbed
RB	rb01.lip.pt	LIP/FCCN	Production
II	li01.lip.pt	LIP/FCCN	Production
RLS	rc02.lip.pt	LIP	Production
VO	grid-vo.lip.pt	LIP	Production + Development
Mapcenter	mapcenter.lip.pt	LIP	Production
Myproxy	px01.lip.pt	LIP	Production
RAS-JSS	ui02.lip.pt	LIP	Production
RB	rb.fzk.de	FZK	Development
II	lc.fzk.de	FZK	Development
RLS	rls.fzk.de	FZK	Development
RAS	ras.fzk.de	FZK	Development
RAS	ras.man.poznan.pl	PSNC	Production
RAS-JSS	cedar.man.poznan.pl	PSNC	Production
RB	cgrb.ifca.org.es	IFCA	Production backup
II	cgii.ifca.org.es	IFCA	Production backup
Myproxy	cgpx.ifca.org.es	IFCA	Production backup
RB	loki09.ific.uv.es	IFIC	Development
RB	aorbgrid.uab.es	UAB	Development
R-GMA	cagraidsvr15.cs.tcd.ie	TCD	Production + Development
GMDAT	grid.fuw.edu.pl	FUW	Production + Development

Table 2 Testbed central services

As indicated also in the previous table, several other sites contributed to the collective services support effort by hosting services related with specific project developments for both testbeds.

Security services in the CrossGrid infrastructure rely on the Globus Security Infrastructure (GSI) using public key cryptography based on X.509 certificates. GSI provides the ability for secure communications over the Grid and for decentralized inter-organizational authentication enhanced by single sign-on features and delegation of credentials.

Ten national Certification Authorities are involved, six of them setup thanks to the CrossGrid project. All of them are now members of the European Grid Policy Management Authority (EUGridPMA), and serve many other current Grid projects across Europe.

3. COORDINATION WITH THE EDG, LCG AND EGEE PROJECTS

One of the main objectives of the CrossGrid testbed was to extend a Grid framework, like the one setup in the EDG project, into new European countries. The interoperability with EDG was one of the initial key decisions.

The first versions of the CrossGrid testbed were fully based on the DataGrid middleware distribution that already included a modified packaging of the Globus 2 toolkit. All the initial efforts were centered in providing a realistic Grid environment for the development of the CrossGrid services, tools and applications. The development of the DataGrid software was closely followed and the releases deployed at test sites. The first official CrossGrid testbed deployed in 2002 was fully based on the EDG 1.2 release, which was followed by EDG 1.4 in March of 2003. During the last months of 2003 CrossGrid faced the issue of the maintenance for the EDG releases as the DataGrid project was about to finish. Two possibilities existed, either install the last EDG 2.1 release incorporating all the latest DataGrid developments and maintain it, or deploy the LCG-1 release produced by the CERN LCG project that incorporated most of the stable EDG components and had a maintenance path in the future. After careful evaluation of both releases CrossGrid choose to deploy LCG-1 in December of 2003, which was upgraded to LCG-2.0 in the beginning of 2004. In June of 2004 the testbed was upgraded to LCG-2.2, which has remained the official basic middleware in CrossGrid until the end of the project.

Coordination with the EDG project was directed mainly through the corresponding testbed workpackages, WP6 in EDG and WP4 in CrossGrid. The experience was very successful and solid links between several of the CrossGrid partners and EDG ones were established and have been prolonged along participation in the EGEE project, started in April 2004, where several of the CrossGrid partners are participating in the SA1 package supporting the grid infrastructure operation.

Relevant examples of this coordination include:

- Collaboration on the deployment and cross-acceptance of Certification Authorities
- Test of releases and experience on configuration systems
- Improvements and tests on EDG packages, like the replica system, grid monitoring map, networking tools, RGMA tests.
- Discussions on common support tools, like the HelpDesk, auto-build tool, or the GridPortal repository
- Possible integration in EGEE of CrossGrid middleware: parallel (MPI) scheduler, interactivity mechanism.

CrossGrid WP4 representatives attended all EDG meetings, and regular presentations took place in the WP6 sessions. Conversely, EDG and EGEE representatives have participated in CrossGrid meetings in Cracow, Santiago, Cyprus, Lisbon, and will also share the Amsterdam event.

Since April 2004, participation in EGEE of several CrossGrid partners takes place through their correspondent “Federations”, and the partners participate in Regional Operating Centers (ROC). Direct input about new developments is received. EGEE testbed has been based on the LCG configuration, and, as indicated before, up to January 2005 the same basic version of middleware has been used also in CrossGrid testbed (LCG-2.2). However CrossGrid testbed will not follow the recent release LCG-2.3, as it includes significant changes in particular regarding installation tools (LCFG-ng is no longer supported), and none of the new main features is critical for our project.

4. SUPPORT ACTIVITIES

As stated in previous deliverables, the support within the project has been divided into three categories according to the target communities: “user support”, “developer support” and “site administrator support”. To satisfy the different needs of these communities several support channels have been created, from mailing lists to a dedicated helpdesk.

Relevant links for coordination and support are provided from the WP4 main web page (see http://grid.ifca.unican.es/crossgrid/wp4).

The main effort along the last months has been the consolidation of the different support lines. The level of experience gained on support for site administrators and developers is now quite adequate, while user support experience has been gained mainly through the different tutorial experiences (Cracow, Amsterdam, Madrid, Lisbon…) and also with the recent operation of a virtual organization for “external users”. Further work along this line is being prepared, as this will be a key component in the exploitation efforts.

Site administrators’ support

Support for site managers is available through the “Testbed Support” web site at LIP. It includes three sections dedicated to the production, development and validation activities, and provides information about the testbed infrastructure and also on installing and configuring sites.

The complexity of the CrossGrid software releases requires specialized tools to make the deployment process lightweight, fast and reliable. The release process relies in the software packing using the RPM format. The generated autobuild packages are made available through the GridPortal repository at FZK jointly with configuration profiles for automated installation using the LCFG installation and configuration software. These profiles contain configuration information that is common to all sites and are complemented by the site specific information provided by the systems administrators at each site. To further automate the deployment process, the cg-lcfg-go tool was developed. This approach assures that all sites have the same package versions configured properly and contributes to reduce the deployment time.

Additionally, a site installation manual “Cluster Installation in CrossGrid”, created and maintained by UoA and available from the “Testbed Support” web site, extensively explains all steps required to setup a new site, from the preparation of a LCFG-ng installation server to the deployment of a whole cluster. The manual has proven to be an excellent source of information for the sites installation, upgrade and maintenance.

Developers’ support

All CrossGrid partners contribute to the project development effort resulting in a complex environment where clear guidelines are needed. Using the EDG developer's manual as input FZK has created and maintained the official development reference manual for CrossGrid. Additionally FZK supports central development services including a CVS repository, bug tracker, webspace and autobuild, under the GridPortal umbrella. Along the last months, Installation and User Guides for all the software components have been prepared as described in the WP2 and WP3 deliverables.

User support

The main generic tool is the CrossGrid HelpDesk system, installed at IFIC, CSIC, Valencia, and accessible at http://cg1.ific.uv.es/hpdesk . Another important tool for user support is the CrossGrid Tutorial, that includes a section with exercises on the CrossGrid Testbed, intended for new users that would like to run their jobs or applications in the Grid framework. The tutorial explains all the steps that a user must follow to be able to execute jobs in the testbed: from the obtainment of the digital certificate to running MPI jobs and using the Replica services. All the examples, including source code, are available from http://grid.ifca.unican.es/crossgrid/wp4/Tutorial/Examples.

5. QUALITY ASSURANCE AND VALIDATION SUMMARY

The previous sections have covered the workpackage tasks 4.1 (infrastructure), 4.2 (coordination with EDG), and 4.3 (support), and here the last and critical task 4.4 on verification and quality control is summarized.

It is a common error to underestimate the importance of quality assurance in a computing project, and even more in a distributed framework. The testbed task 4.4, oriented to verification and quality control, lead by LIP, has proved to be of critical importance both to improve the testbed infrastructure operation and to assure a correct deployment of middleware on it.

Two main areas emerged from the task objectives:

- **Testbed test and validation.**
  - Validation of sites after each new installation or upgrade.
  - Continuous monitoring of the testbed.
- **Software test and validation.**
  - Validation of the software to be deployed in the testbed.
  - Validation of software documentation.

Testbed sites validation

The testbed test and validation aims to make sure that the services provided by the testbed are reliable, dependable and conformant therefore providing a stable base for the development of the CrossGrid application and tools.

Since the beginning of the project it became necessary to ensure that the middleware was correctly deployed at all testbed sites. This need become even more evident after the first tests with the EDG middleware in the end of 2001. These tests have shown the complexity of the testbed deployment and coordination. A considerable amount of resources was then allocated to ensure that the systems involved in these initial activities were properly installed and configured before more complex tests could be performed. Simultaneously the need to evaluate the behaviour of the installed grid systems also led to some involvement in the grid monitoring field. This effort was twofold: verify the stability of the middleware and monitor the future production testbed.

Due to the nature of the software being developed within the project the correct configuration and behaviour of the testbed sites is also extremely important. Besides being CPU and data intensive the CrossGrid software is characterized by its interactive and parallel nature. To satisfy the parallel computing needs MPI was chosen due to its capacity of supporting parallel applications running inside local clusters and also running across clusters using the Globus toolkit. However for an MPI application to run with success all the intervening nodes and sites must be correctly configured. One single badly configured site or node can make the application fail. The same is true for the interactive applications, since a badly configured site can make the interactive jobs being queued instead of entering immediately into execution. Due to the high number of CPUs required by the parallel applications the probability of encountering a site with a problem is also considerably higher. Therefore, it is essential to assure that all testbed sites are properly installed, configured and behaving well

First of all, a site validation procedure is applied after each new release installation at a given site. The procedure starts with the site manager sending technical information required for performing the site acceptance tests and later make the site available in the testbed. 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 team is required to help site administrators quickly solve any problems that are detected 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 the problems that were detected have been 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 Resource Broker.

When the site is found to be stable, it is 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 helping to locate many problems that would pass unnoticed until the sites would be actually used by the users possibly creating major testbed disturbances.

Several testbed metrics are considered to evaluate the performance, like the number of testbed sites, number of users, job submission success and testbed sites uptime. Several monitoring tools have been developed or adapted to implement these testbed quality indicators.

Sites must be continuously monitored both during the acceptance process and during operation. The aim 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
- Site uptime

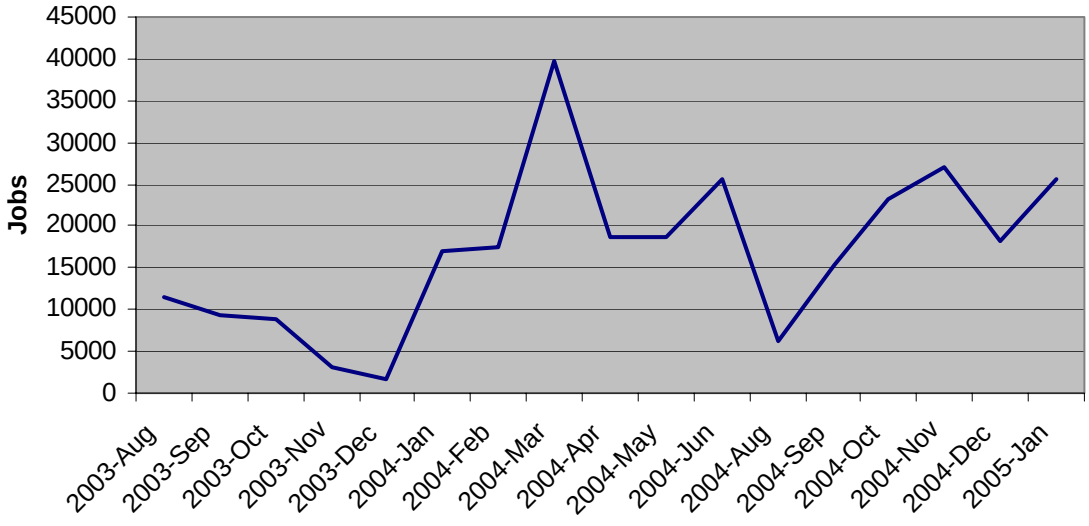
Details and updated statistics can be obtained from the deliverable D4.8. A summary with updated figures corresponding to these statistics is included below.

The first chart shows the testbed quality indicators collected since August of 2003 in the context of the testbed quality assurance activities. The table shows the number of testbed sites, the number of users registered in the CrossGrid VO, the number of active users (users that have been using the testbed during the month), the ration between the active and the total users, the total number of submitted jobs, the jobs that were successfully submitted, the ratio between the total jobs and the active jobs and finally the testbed uptime in percentage. The number of sites reflects the total number of testbed sites. The users refer to the users registered in the main CrossGrid Virtual Organization. The number of jobs was obtained from the production testbed sites.

Month	Sites	Users			Jobs			Uptime
		Total	Active	Ratio	Total	Ok	Ratio	
2003-Aug	16	79	28	35%	11424	11059	97%	90%
2003-Sep	16	79	39	49%	9278	8781	95%	85%
2003-Oct	16	81	32	40%	8919	8772	98%	82%
2003-Nov	16	83	32	39%	3118	1950	63%	56%
2003-Dec	16	87	29	33%	1627	1565	96%	77%
2004-Jan	16	92	46	50%	16882	16526	98%	76%
2004-Feb	15	96	40	42%	17471	17394	100%	92%
2004-Mar	15	100	55	55%	39674	39357	99%	-
2004-Apr	15	101	40	40%	18620	18501	99%	94%
2004-May	15	104	43	41%	18648	18307	98%	87%
2004-Jun	15	105	42	40%	25687	24403	95%	94%
2004-Aug	15	111	31	28%	6148	6125	100%	96%
2004-Sep	15	114	44	39%	15381	14610	95%	95%
2004-Oct	16	119	57	48%	23279	23235	100%	97%
2004-Nov	16	119	43	36%	27151	26899	99%	99%
2004-Dec	16	120	44	37%	18106	18056	100%	100%
2005-Jan	16	121	39	32%	25546	25521	100%	100%

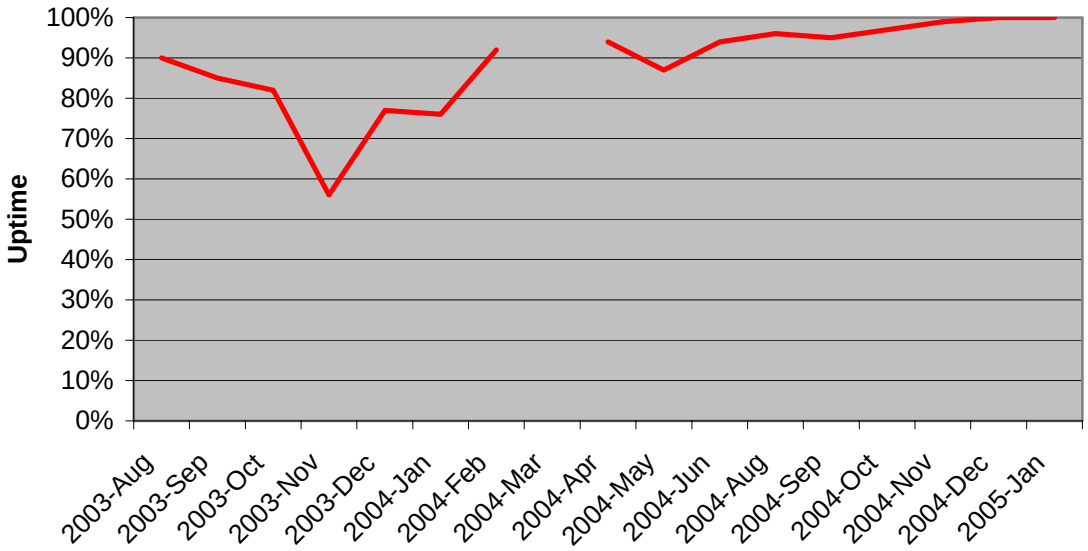
Testbed evolution in numbers

The next figure shows the evolution of the testbed usage measured as the number of jobs submitted to computing elements using the Globus GRAM service. The data was obtained from the Globus gatekeeper log files all computing elements from the production and development testbeds.



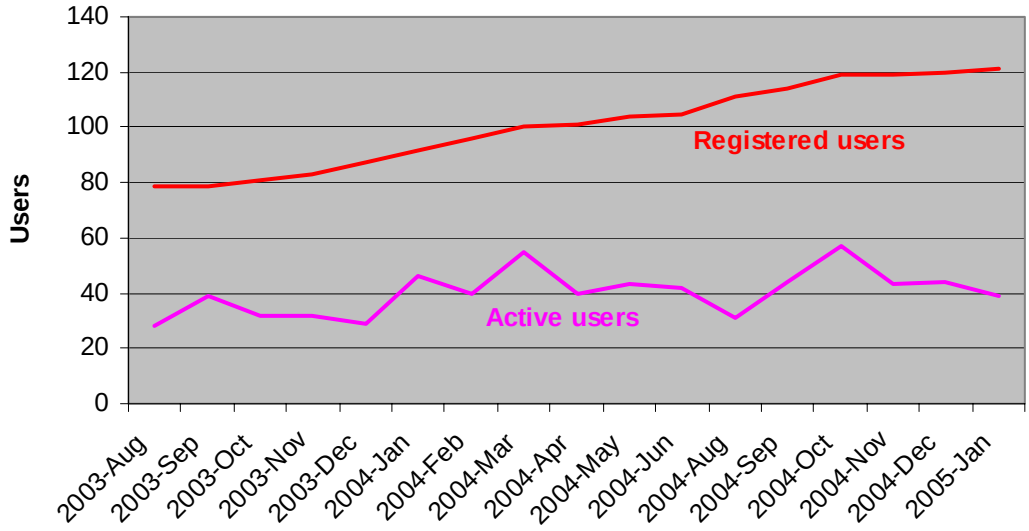
Number of jobs submitted per month

The evolution of the testbed uptime can be observed in the next figure. The missing part corresponds to March of 2004 for which information was not collected due to the upgrade of the Mapcenter software from which the uptime data is obtained. A clear improvement of the testbed uptime can be observed showing an increased stability.



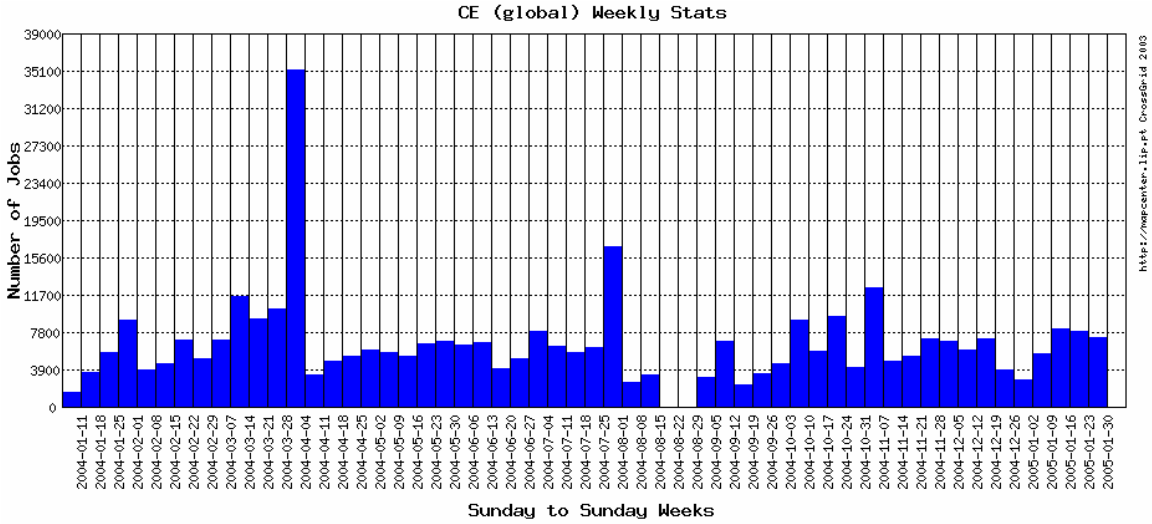
Testbed uptime per month

Next figure shows the evolution of the number of users registered in the CrossGrid VO and the number of users actually submitting jobs.



Evolution of the CrossGrid VO in terms of users

And the last graph shows the evolution of the job submissions using the Globus GRAM service that includes jobs submitted by all resource brokers and jobs submitted directly using Globus.



Number of jobs submitted using Globus GRAM

Software test and validation.

The software test and validation is the last step of the CrossGrid quality assurance process and is responsible for the execution of the software acceptance tests. The verification and quality control does not interfere with the software development process that occurs inside the project work packages or inside external projects such as Globus and DataGrid, however it provides feedback to the developers on issues that affect deployment, security, usability and functionality of the software.

Initially the software validation efforts were centred in the DataGrid middleware distribution as the CrossGrid software was still under development and the EDG software was being used as the basis for the CrossGrid developments. These activities allowed the detection of issues that could affect the CrossGrid developments and the testbed stability, simultaneously they contributed to increase the know how on the DataGrid software internals and characteristics that become important for the support, development and deployment activities. In the end of 2003 the basic middleware provided by the EDG distribution was replaced in CrossGrid by the LCG distribution also fully compatible and based on the EDG middleware but with long-term support. Again this decision was supported by the results of the tests performed on both distributions. These activities contributed actively to the improvement of the DataGrid and LCG software through the reporting of the detected issues back to the developers. All basic middleware releases used by CrossGrid from EDG 1.2 to LCG 2.0 were previously tested and validated prior to deployment.

Later the EDG and LCG software become more mature and the CrossGrid software reached a development state where it could be integrated and deployed. At this stage the test and validation activities become focused in the CrossGrid software. An iterative sequence of validations started then, with the developers producing new releases that were integrated, validated and deployed. The procedure was refined and improved along the project. Two major changes were introduced in 2004, the possibility of bypassing the validation procedure for minor fixes and the use of the bugtracker to keep a record of the detected issues.

Guidelines have been defined for the actual validation that must cover the following points:

- Documentation.
- Usability.
- Installation.
- Functionality.
- Compatibility.
- Security.
- Previously reported issues.

The problems detected during the validation are classified regarding their severity and priority as Citical/Inmediate, High, Medium, or Low. These two attributes define the urgency and impact of the problem and help to determine which issues must be addressed first. For each severity level described a guideline action is recommended.

The validation procedure has been applied 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 excellent cooperation between the developers, integration team members and validation team contributed highly to the success of the validation process. Since the validation process was seen as a method to improve the quality of the released packages a close contact between developers and testers was established. The developers provide support and additional information and receive feedback from the test results and difficulties experienced during the tests. Frequently during the validations issues were detected and immediately corrected by the developers before the end of the validation, hence improving the quality of the releases and reducing the time between development and deployment.

The web tools used to submit validation requests and obtain information on the status and results of the validations have contributed considerably to keep track of the validation process making it easier, more transparent and well controlled.

More than 50 validations have been performed, including all CrossGrid components, as can be seen in http://www.lip.pt/computing/projects/crossgrid/wp4task4/test-results.htm.

The final relevant versions are given below:

- MARMOT 1_1_9
- GridBench 0_9_9
- PPC, version 1_3_1
- **G-PM, version 0_8_2**
- **RAS/MD, version 4_0_15**
- **Resource Broker, version 2.1.15 (release 2.6)**
- GMDAT 1_10
- **OCM-G, version 1.9.6**
- JIMS, version 1_5_02
- SANTA-G version 1.4.1
- UDAL, version 1.3.4

Further details can be obtained from the updated version of the previous deliverable D4.8.

6. INTEGRATION EFFORT

Integration in a software project is the process by means of which different software pieces, developed by programmers or team of programmers, are "assembled" together in a common framework and single product. For a complex and distributed project like CrossGrid, planning of the architecture and a strong effort on integration are key issues for success.

The Integration Team effort was started inside WP4 in the first project year with the objective of achieving an adequate level of integration for the basic middleware before deployment in the testbed sites. Right after the first annual meeting, in Santiago, it was recognized that the effort should be extended, and involve all workpackages. Integration meetings have taken place in Poznan, Cyprus, Dagstuhl and Lisbon, and following the recommendation of the project reviewers, an integration manager was appointed, and a detailed integration plan for the last year was defined, with a clear schedule and list of actions. Weekly meetings via videoconference have helped to track the progress, discuss issues and exchange information along the last year, with the final meeting taking place in Amsterdam from 10th of February for the rehearsal of the final demos.

The detailed plan is described (and updated) in the deliverable D4.7, appendix 1, and includes a list of actions and corresponding detailed plans inside each workpackage:

1. ***Test and validation of all individual components***
2. ***Code review***
3. ***Integration of key components***
4. ***Business model and testbed components***
5. ***Synchronization of schedules of all components and definition of releases***
6. ***Integration of applications***
7. ***Preparation of demos***

The six main integration milestones (M1-M6) for this last year are:

- Milestone 1 - Release of CrossGrid Tag 1.0 (**M1** 20-May, *reached on 3rd June*)
  Tag 1.0 is based on LCG-1 and includes components tested by first deadline (RB, JSS)
- Milestone 2- Release of CrossGrid Tag 2.0 (**M2** 15-July, *reached on 23rd July*)
  Tag 2.0 based on LCG-2, and includes all revised (and code reviewed) key components: RB, RAS/MD, OCM-G/G-PM
- Milestone 3- Applications running on Tag 2.0 (**M3** 30-July, *reached on 29th July, Dagstuhl*)
  The demo of applications will be oriented to show the use of Tag 2.0 and integration issues
- Milestone 4- Full Integration of Key components (IS, CONTROL, MPI) and applications
  (**M4** 15-October, *reached on 20th October, Lisbon*)
- Milestone 5- Consideration of web services and testbed integrated components
  (**M5** 15-December, *delayed to mid February*)
- Milestone 6- Final Application Demos
  (**M6** 15-February, *to be reached on 16th February, Amsterdam*)

The planning has been followed not without difficulties, and in particular the schedule of releases after validation has put a lot of pressure on developers. Regarding the two last milestones, it has to be said that the final milestone will be reached on time, as can be inferred from current progress and the discussions through the latest videoconference integration meetings. However, the milestone M5 is delayed to mid February at least by two related reasons: the two main basic middleware providers for our project, the Globus team and the LCG/EGEE team are both late with the release of their web service grid packages. For Globus, the Globus World meeting by the end of January will give a hint on the release of WSRF based GT4; for EGEE the advance in the ARDA package will be known by the time of the first review. A careful consideration of the best approach inside CrossGrid by the

architecture team and the integration team will be important for the technology implementation plan that has been postponed to M38. Similarly the action 4 on business model and testbed components will be also finished in this framework, profiting also of the participation of the integration team components in the exploitation activities.

Another very important action along the last months has centered on the integration of the key components. After the definition of these three components, the RAS/MD system, the CrossGrid scheduler for parallel and interactive applications, and the OCM-G/GPM application monitoring system, priority has been given to their integration. This action has also meant a strong pressure on the developers and on the supporting members of the integration team, for example on the effort on validation or in the consideration for its use in applications and demos.

The next section describes in more detail the last integration action, preparation of demos, that was already well advanced by the Lisbon meeting but has centered most of the work in these last months.

7. FINAL DEMOS: INTEGRATION AND TESTBED ISSUES

The final demos are based on CrossGrid full applications, but in order to be effective they should:

- Be based on a clear, effective and well motivated use scenario
- Benefit of an appealing and user friendly interface
- Show the possibilities of a Grid framework, and the relevance of the CrossGrid components
- Execute perfectly in the actual testbed infrastructure

The demos have been initially prepared by the application teams (workpackage 1, see deliverable D1.7), but they have been discussed in detail in the integration team meetings regarding the two last points. In particular, explicating which tools and middleware in a given application demo has been discussed to provide a set of demos as complete as possible. As an example along this line, the role of key components has been emphasized.

This deliverable, D4.9, includes an appendix that will be updated until all final demos' details regarding the testbed use and the integration issues are included.

Three demo packages have been prepared and will be shown at the EGC'05 CrossGrid demo session in Amsterdam (16th February 2005, 14-18h):

- **Biomedical application**
- **Environmental package**
- **Interactivity demo**

The biomedical application will demonstrate also the role of the Migrating Desktop (one of the key components), visualization possibilities with GVK, and the job submission mechanism (to several MPI enabled sites in the testbed).

The environmental package comprises the Flooding application and the combined Meteo Prediction-Air Pollution demo. Again the integration through the Migration Desktop including user friendly and graphics-rich plugins and parallel MPI jobs submission will be demonstrated.

Finally, the Interactivity demo is based on a High Energy Physics application: the interactive ANN (Artificial Neural Network) training, executed from the MD and displaying in real time the progress of this distributed data mining tool.

A nice and encouraging fact is that the demos have put no extra requirement on the testbed: no special ad-hoc modification of the usual setup will be done to support the final demos.

8. COLLABORATION EXPERIENCE

Before summarizing the whole workpackage experience, a section is required to discuss the task 4.0 (testbed coordination and management).

If already "Grid is everything about collaboration", we can only insist on the importance of collaboration for the successful completion of a project like CrossGrid, and even more on the areas of testbed and integration.

Several project management mechanisms have been used, starting with a balanced and agreed sharing of tasks with assignment of work and roles to centers and responsible people. The initial planning, included already in the project Technical Annex, has made this task easier. Examining it in perspective, it can be said that all tasks have been completed, deliverables prepared, and milestones reached, including the last one, M4.4, final testbed setup.

The use of mailing lists, the very frequent contact through video-audio conference (thanks to the VRVS system), and an adequate number of presential meetings (about 2-3 each year) have been used to keep a regular contact at all times. Also the flow of information from the project management and with WP1, WP2 and WP3 workpackage managers has been kept as fluent and open as possible.

However the main reason for the good collaboration experience has been the interest and collaborative attitude of all workpackage and integration team members. Probably the fact that the project was addressing a new and promising framework has motivated in particular the young people (that form a large majority of the participants).

Trust relations have been also established, what is critical for a Grid environment to be operative. The success of the cross-acceptance of certificates, and the mapping of the virtual organization users to local accounts are not minor issues.

It must be said that all the coordination activities did take a substantial time, much more than initially expected, and the workpackage responsible has been able to cope with them only thanks to the help and support of the different task responsibles[1].

Real and hard work on testbed sites and integration was the common effort of **ALL** partners[2]

Special thanks should be given also to the people involved in the organization of the presential meetings in Krakow, Linz, Santiago, Poznan, Cyprus, Dagstuhl, Lisbon and Amsterdam, and to the VRVS team for providing a very good tool that has regularly supported weekly meetings with 20-30 participants.

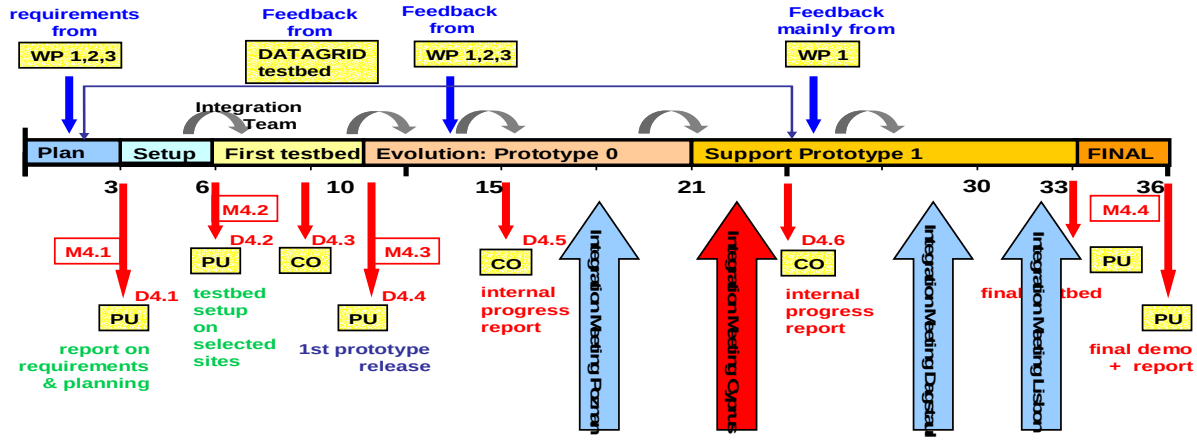
---

[1] *Jorge Gomes, LIP, responsible for task 4.4, and also deputy of the whole workpackage; Ariel Garcia, Marcus Hardt and Harald Kornmayer, FZK, lead task 4.2, and also much of the integration work in particular in the last months, Jose Salt responsible for task 4.3 together with Santiago Gonzalez, IFIC, who helped to coordinate the integration work in Dagstuhl, Cyprus and Poznan meetings. Last, but not least, Rafael Marco, IFCA, was responsible for task 4.1 and close advisor for all the infrastructure issues.*

[2] *Mario David, Joao Martins and Luis Bernardo from LIP; David Rodriguez, Celso Martínez-Rivero, Daniel Cano and Irma Diaz from IFCA, Alvaro Fernández, Javier Sanchez, Farida Fassi and Vicente Lara from IFIC, Piotr Nyczyk, Patryk Lason, Andrzej Ozieblo, Lukasz Dutka, Bartosz Balis, Kaziemirz Balos and Marcin Radecki from CYFRONET, Pawel Wolniewicz, Marcin Plociennik and Miroslaw Kupczyk from PSNC, Michal Bluj, Krzysztoff Nawrocki and Adam Padee from ICM, Carlos Fernández and Javier Fontán from CESGA, Vangelis Floros from UoA, George Tsouloupas and Wei Xing from UCy, Jan Astalos and Branislav Simo from II-SAS, Brian Coghlan, Stuart Kenny and David O'Callaghan from TCD, Enol Fernandez, Carlos Moreno, Antonio Hervas, Elisa Heymann and Miquel Senar and from UAB, Christos Kanellopoulos and Dimitris Zilaskos from AuTH , Alfredo Tirado-Ramos and Derek Groen from UvA, Bettina Kramer from HLRS, Roland Wismüller from TUM, Francisco Fernandez Rivera, Marcos Boullon and Carlos Mouriño from USC, Marco Sottilaro and Stefano Becco from DATAMAT.*

9. EXECUTIVE SUMMARY

As indicated in planning scheme included below, the final phase of the project is reached, marked by the deployment of the final testbed supporting the application demos that will be shown in the final project review, in the framework of the CrossGrid Day of the European Grid Conference EGC'05 in Amsterdam on February 16th.



After the successful integration meeting in Lisbon, the final plans for integration and preparation of demos were fixed. A final deadline for validation requests for all the software to be included in the final release was set: 20th of December. This deadline has allowed the installation in the development testbed of the final components that included mostly minor improvements with respect to previous versions, and its test for the final demos. In parallel, the validation procedure has been performed with the objective of the deployment of the "final" **CrossGrid release, 3.0**, in the production testbed by the end of January.

The most technically challenging point in this last phase has been the integration and validation of the interactive setup, including the new "glide-in" mechanism for prioritization of interactive jobs. By the time of writing this report, the initial interactive example developed by a joint effort of the teams working on task 3.1, 3.2 and 1.3, and shown in the Lisbon meeting, has evolved to a realistic setup demo (the HEP ANN training from task 1.3) currently running in the development testbed.

This final report describing the testbed has been released by the end of January, as deliverable D4.9: it summarizes the main achievements, pointing also to the different sections of the two previous internal deliverables D4.7 (oriented to integration) and D4.8 (describing the final testbed setup).

Regarding Grid infrastructure setup, covered under ***task 4.1***, central services (see http://www.lip.pt/computing/projects/crossgrid/crossgrid-services/central-services.htm) have been maintained by LIP in Lisbon, supporting the production environment, with the Replica Location Server installed and strongly supported at FZK. Backups for the Resource Broker and the Proxy server machines have been kept running at IFCA. The development testbed central services have been supported at FZK. A new webpage setup at FZK shows now the status of installation of testbed releases (see http://savannah.fzk.de/DocumentRoots/cvs/monitoring/prod-stat.html) .

Certification authorities are in coordination with the EUGRIDPMA initiative, and they have met in Marseille along the last days of January.

For ***task 4.2***, the tracking of EDG/LCG/EGEE middleware evolution has come to its final point. The new LCG version (2.3.0) being deployed in all EGEE sites by end of January, does not include support for LCFG-ng installation, but rather returns to a more simple script approach, and the default Linux version is now Scientific Linux. All this changes will not be followed in our testbed. However, an excellent relation is kept with EGEE and LCG projects, and discussions regarding the possible collaboration to support an "interoperable interactive testbed" in the short term future have started.

Under the infrastructure support, ***task 4.3***, the CrossGrid software releases have continued taking place under the coordination and support of FZK. The final release, CG3.0, is installed in all sites by the end of January, and only minor patches will be allowed after this installation is completed. The new Virtual Organization for external users has been deployed, and explicit instructions are available from the renewed WP4 web pages.

Under verification and quality control, ***task 4.4***, as indicated, the large work to test and validate the final versions of CrossGrid software has been also finished. The "express" procedure for validation of new versions with only minor changes has proved to be quite useful in the last months of the project.

10. CONCLUSION: PERSONAL REMARKS

Before closing this last WP4 deliverable, two extra sections are included providing an open point of view on the achievements from the workpackage manager perspective[3].

From a privileged testbed/integrator/developer/user view, the question "What have we learned in these three years of project" is a very interesting one.

The first lesson is that the Grid "idea" is as interesting as difficult to put in practice: using the network, and open standards, join resources from different sites (16 organizations, about 200 CPUs in our case) and make them available to the user in a transparent way: sit the user in front of a 300 GHz PC (the "common" PC in 2020 if Moore law holds by then, or much sooner if the "cell processor" promise becomes real, who knows…). Theoretically it makes sense: the network is ready now to support it (and evolving faster than CPU power), and the basic middleware for secure communication also exists. But the key point is on "sharing and collaboration": this is extremely difficult, and the question is why just bother and not use always resources at a single site (like powerful clusters or supercomputing centers). The answer is two-fold: the e-Science needs can only be handled with a distributed framework when the "basic" information is also distributed (like for example in a network of hospitals), and the second reason are metacomputing possibilities that can provide an umbrella for ambitious projects providing access to larger resources.

Three years ago we were considering the first wave of Grids starting: based on UNICORE and Globus middleware, mainly oriented to cover large streams of long batches (a High Throughput Computing problem). In this framework, CrossGrid proposed a new path for a useful Grid environment: interactive, bringing enough power to the tips of the finger of a user (a physician, an engineer, or a physicist, *a human in the loop*) waiting in front of an screen for an answer to a complex problem that she/he can not handle in a PC or small cluster; and built on top of existing middleware to keep compatibility with the existing initiatives like EDG (*extending the Grid across Europe*).

The bet included selected real life problems (from medicine, environment and physics areas), started from existing middleware (Globus 2 and the EDG layer), and tried, by running in parallel (MPI support and adapted scheduler), to bring enough power to the tips of the finger of the user in front of an screen in a friendly way through the Migrating Desktop or the Portal. This effort was complemented with monitoring and performance tools, and distributed data access techniques.

Along these three years we understood and went through all steps to setup a testbed interoperable with EDG: installed local resources at each site (a configuration machine, a computing element and worker nodes, and a storage element), defined collective resources (resource broker, replica location service, virtual organization) and established compatible certification authorities. And all this effort was complemented with the deployment of test, validation and check mechanisms. A good support for the communities (users, developers, administrators) was established through the GridPortal, Webpages and HelpDesk, and coordinated through mailing lists and videoconference meetings.

New services required new middleware and supporting machines that were deployed when needed (for MPI the CrossGrid RB, for user friendly access the MD and RAS server), and the infrastructure was operated with regular releases and upgrades, security updates and patches, and solving all issues. At the same time a large effort on integration was done.

So the final question is "is there a Grid available after these three years?"

The answer is yes: we now have a clear idea of how to setup a grid infrastructure and how to use it. A clear comparison with the way a local cluster is used can be done as shown in the table below.

[3] Most of these ideas were already expressed in previous presentations/slides at different fora.

LOCAL CLUSTER	GRID FRAMEWORK
Get an account/password from your local administrator, get access to a connected terminal	Get a digital certificate, register in the VO, get access to the Migrating Desktop from your laptop
Develop your own code using CVS	Use the CVS repository at GridPortal for coding
Compile using MPI, get an executable, save it in your private or group account disk space	Compile with MPI, make your executable and save it in your Virtual Directory in an SE
Set links to your input data	Assign logical file name to data, that will reside in an SE
Submit a script to the PBS queue system from command line or from a web interface	Submit your job from the MD, indicate the resources needed (from a single site, MPICH-P4, or many sites, MPICH-G2)
Monitor job status	Check the job status from the MD
Retrieve output & logs from the queue	Get output stored in your Virtual Directory
Display the output locally in the screen	Display the output via a MD plugin, including real time refresh in interactive mode

So, applications can be ported easily to this framework, and executed with some advantages, but also some restrictions, that we have experienced in our testbed.

On the advantages, the recognition of Grid possibilities, like launching in parallel to many cluster sites several MPI applications to cover different scenarios (like for the medical application) or execute a cascade of simulations (like for the environmental applications). Even a not so large testbed has been able to cope with the usual requests (around 20 CPUs as maximum). And there are always sites available ('on') to execute the jobs, despite own resources may be unavailable. New applications can be ported quite easily, and new users can work in a few days in the new environment.

On the drawbacks, the "loss" of total control over the application, that is executed remotely, and with reduced debugging and tracing capabilities. Data management is also an important issue, as the data management is not yet completely optimal nor automatic, and problems with large data replication have been encountered.

Open Questions

These last remarks bring us to the point of "open questions"; this is a very brief list:

- Real possibilities of the prioritization scheme
- Comparison of interactivity channels (JSS vs. glogin)
- Data management
  - Lack of flexibility of worker nodes vs. storage elements
  - Scalability of replication
- Global "Universal" Monitoring (everything into R-GMA?)
- Support on platforms different than IA32 and installation solution

And of course the definition of a possible Grid economy model (before defining a Grid business model!)

What next?

The final question…

Answering briefly this question related to real exploitation possibilities (so see the exploitation deliverable D5.3.14), e-Science is one of the natural frameworks: CrossGrid testbed already had an impact on national programs, and also in EGEE. A devoted testbed supporting interactivity is being considered inside EGEE, and could well be the future of our current testbed resources and expertise. Next months, M36-M38, will be devoted to establish an agreement on how to proceed.

Application oriented testbeds to support specialized communities like medical or environmental ones, are also being considered.

Regarding possibilities in a more general commercial framework, the main technology transfer expected is on the know-how, and in particular on consultancy.

However the opportunity to build some commercial/industrial components from the best features of our testbed and integration experience is yet open!