

Experience with the International Testbed in the CrossGrid Project

J. Gomes¹, M. David¹, J. Martins¹, L. Bernardo¹, A. García², M. Hardt², H. Kornmayer², J. Marco³, R. Marco³, D. Rodríguez³, I. Diaz³, D. Cano³, J. Salt⁴, S. Gonzalez⁴, J. Sánchez⁴, F. Fassi⁴, V. Lara⁴, P. Nyczyk⁵, P. Lason⁵, A. Ozieblo⁵, P. Wolniewicz⁶, M. Bluj⁷, K. Nawrocki⁷, A. Padee^{8,9}, W. Wislicki⁸, C. Fernández¹⁰, J. Fontán¹⁰, Y. Cotronis¹¹, E. Floros¹¹, G. Tsouloupas¹², W. Xing¹², M. Dikaiakos¹², J. Astalos¹³, B. Coghlan¹⁴, E. Heymann¹⁵, M. Senar¹⁵, C. Kanellopoulos¹⁶, A. Ramos¹⁷, and D. Groen¹⁷

¹ Laboratório de Instrumentação e Física de Partículas, Lisbon, Portugal

² Forschungszentrum Karlsruhe GMBH, Germany

³ Instituto de Física de Cantabria (CSIC-University of Cantabria), Santander, Spain

⁴ Instituto de Física Corpuscular (CSIC-University of Valencia), Valencia, Spain

⁵ Akademickie Centrum Komputerowe CYFRONET, Krakow, Poland

⁶ Poznan Supercomputing and Networking Center, Poznan, Poland

⁷ A. Soltan Institute for Nuclear Studies, Warsaw, Poland

⁸ Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Poland

⁹ Instytut Radioelektroniki PW, Warsaw, Poland

¹⁰ CESGA, Centro de Supercomputacion de Galicia, Santiago de Compostela, Spain

¹¹ National Center for Scientific Research "Demokritos", National and Kapodistrian University of Athens, Dep. of Informatics and Telecommunications, Greece

¹² University of Cyprus, Cyprus

¹³ Ustav Informatiky Slovenska Akademia Vied, Bratislava, Slovakia

¹⁴ Trinity College Dublin, Ireland

¹⁵ Universitat Autònoma de Barcelona, Spain

¹⁶ Aristotle University of Thessaloniki, Greece

¹⁷ Universiteit van Amsterdam, Netherlands

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. The main achievements in installation and operation are described, and also the substantial experience gained on providing support to application and middleware developers in the project. Results are presented showing the availability of a realistic Grid framework to execute distributed interactive and parallel jobs.

1 Introduction

The European CrossGrid project [1] has developed new components for interactive compute and data intensive applications in a Grid [2] framework.

The main objective of the CrossGrid testbed was to provide the framework where the new Grid services and programming tools could be developed and tested, and interactive user-friendly applications executed.

One of the initial objectives was to assure full interoperability with the European DataGrid (EDG) project [3] middleware to profit from their results and experience, and achieve an extension of this basic Grid framework across eleven European countries. This coordination has made possible also the successful contribution of many of the CrossGrid partners to the EGEE initiative [4] to setup a production-level Grid infrastructure in Europe.

Another important objective was to support the execution of parallel applications using the MPI [5] protocol, running either inside a cluster (using MPICH-P4 [7]) or across different sites (using MPICH-G2 [6]), and of the different services and tools oriented to support interactivity.

This article summarizes the experience gained not only on the deployment and operation of the distributed computing resources, but also on the corresponding support provided to developers and users.

Further technical details can be obtained from the project technical deliverables, and are also described in the CrossGrid book [8].

2 Implementation of the CrossGrid International Testbed

The CrossGrid international distributed testbed [9] shares resources across sixteen European sites and this is itself one of the challenging points of the project. The sites range 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, Geant, assure the inter-connectivity between all sites. The network includes usually three ranges: the local campus (typically inside a University or Research Center, via Fast or Gigabit Ethernet), the connection via the national network provider (at speeds that range from 34 Mbits/s to 622 Mbits/s or even Gigabit) to the national backbone, and finally the link to the Geant network (155 Mbits/s to 2.5 Gbits/s). The figure 1 shows a map with the different testbed sites, including the major network links.

As indicated before, the basic Grid middleware was selected to guarantee interoperability with the large EDG project testbed.

At the most lower level, the basic job submission services, information systems, authentication, authorization and data transfer are provided by the Globus Toolkit [10]. 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 [11]. This software is currently distributed in a version known as LCG-2, assembled and supported by the LHC Computing Project at CERN [12].

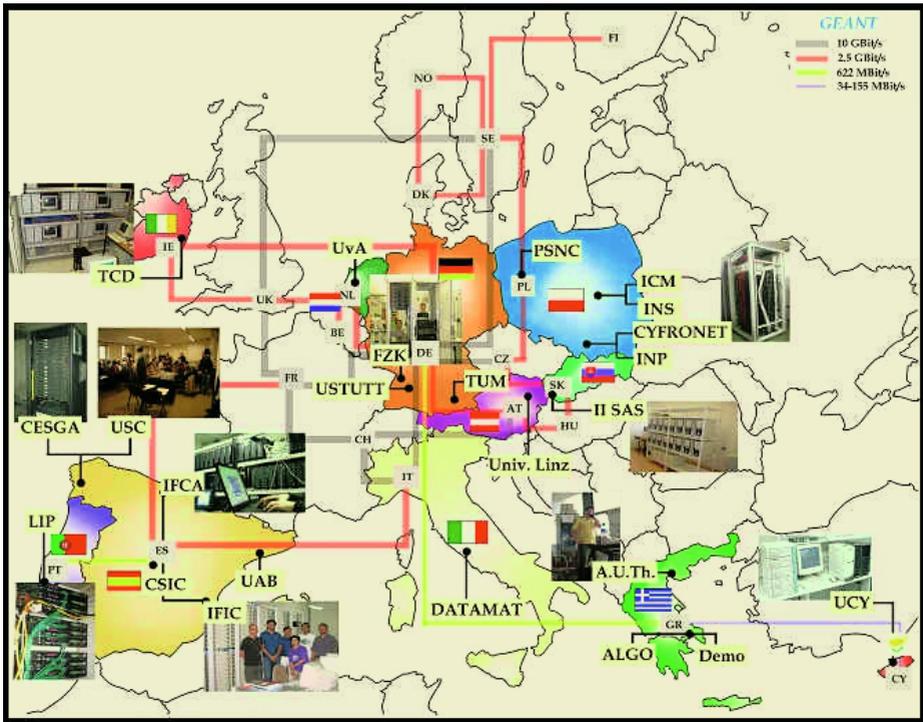


Fig. 1. The CrossGrid testbed sites

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

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

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.

Grid testbeds provide two different types of services: global collective services, and computing and data services using resources distributed across the sites.

In the CrossGrid testbed each site includes the following computing and data storage resources:

- A Computing Element (CE) machine providing the interface between the grid and the local processing farm.
- An Storage Element (SE) machine providing the interface between the grid and local storage resources.
- A User Interface (UI) machine enabling the local users to access the testbed.
- A farm with at least two Worker Nodes (WN) or a dual CPU system.
- An installation server using the LCFGng [14] software.

The CrossGrid testbed includes a total of about 200 CPUs and a distributed storage capacity above 4 Terabytes.

The integration of these resources is done by the global collective services:

- Resource Broker (RB): the heart of the testbed workload management system. The RB receives job requests sent by users and finds computing resources suitable to run the jobs. The CrossGrid RB supports single and parallel jobs (inside a cluster via MPICH-P4 or distributed 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 systems to generate the authorization databases. The CrossGrid VO server contains the list of all CrossGrid users authorized to use the testbed.
- Replica Location Services (RLS): a service that stores information about the location of physical files in the grid. The RB uses the RLS to make scheduling decisions based on the location of the files required by the jobs. The users can also use the RLS service through the Replica Manager (RM). Although this is an EDG service CrossGrid is developing components for its access optimization functionality.
- Mapcenter: an EDG tool enhanced by CrossGrid to monitor the connectivity of systems and services.
- Roaming Access Server (RAS): CrossGrid specific service that provides support for the CrossGrid Migrating Desktop (MD) and portal applications.

- Roaming Access Server Job Submission Service (RAS JSS): a component of the CrossGrid RAS server that performs the actual job submission.
- 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): a 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. Resulting from the collaboration between the Laboratório de Instrumentação e Física de Partículas (LIP) in Lisbon and the Portuguese academic network (FCCN) on Grid technologies the main production central systems at the LIP/FCCN site are physically hosted at the FCCN Network Operations Center in Lisbon. The systems share a 100Mbits/s network connection to the multi-Gigabit Pan-European network backbone Geant. The FCCN collaboration allowed the the production RB to be installed in a Geant point of presence improving the network connectivity and resilience of this critical service. The less demanding central production services have been hosted at the LIP computer centre facilities also in Lisbon. Redundancy has been introduced into the production central services with the duplication of the most critical central services at the Instituto de Física de Cantabria (IFCA) site in Santander. This duplication allows the production testbed to continue to operate in case of a major failure at the Lisbon sites.

The main global services for the development testbed are located at the Forschungszentrum (FZK) in Karlsruhe.

Several other sites contributed to the collective services support effort by hosting services related with specific project developments for both the production and development 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) [13], and serve many other current Grid projects across Europe.

3 Support Tools in the CrossGrid Testbed

The support within the project can be divided in three categories according to the target communities: “user support”, “developer support” and “site administrator support”. Site administrators are responsible for the installation, maintenance and daily operation of the testbed sites. They need mostly good documentation covering the software installation and configuration, and tools to help them to verify their sites configuration and behavior. Developers are special users that are deeply involved in writing software and as such need support not

only in the testbed usage but also in the software testing, packaging, integration and deployment. Developers also need access to specialized tools such as a central CVS [20] repository, bug tracking tools and packaging tools. End-users are interested in running the applications accessing to the testbed in a user friendly way and being isolated as much as possible from the software and infrastructure complexity.

To satisfy the needs of these different communities several support channels have been created, from mailing lists to a dedicated helpdesk. Relevant links for coordination and support are provided from the testbed workpackage (WP4) main web page (<http://grid.ifca.unican.es/crossgrid/wp4>).

3.1 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 end users can also found relevant information about the setup of the testbed and examples on how to use it.

Additionally the site also hosts the testbed verification and quality control web pages. These pages contain information about the test and validation procedures for testbed sites and software packages, a validation request web form and a database with information about the validation requests and corresponding reports is also available.

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 RedHat Package Manager (RPM) format. The generate auto-built packages are made available through the Grid-Portal repository at FZK jointly with configuration profiles for automated installation using the LCFG [14] 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` [23] 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" [27], created and maintained by UoA and available from this "Testbed Support" web site, extensively explains all steps required to setup a new site, from the preparation of a LCFGng installation server to the deployment of a whole cluster. The manual has proved to be an excellent source of information for the sites installation, upgrade and maintenance.

3.2 Developers Support

All CrossGrid partners contribute to the project development effort. In such a complex development environment clear guidelines are needed to coordinate the development process. Using the EDG [3] developer's manual as input FZK

has created and maintained the official development reference manual [17] for CrossGrid.

Additionally FZK supports central development services including a CVS [20] repository, bug tracker, webspace and autobuild.

The autobuild [21] tool was developed by EDG and adapted for CrossGrid use. The tool takes the source code from the repository and produces RPM packages suitable from deployment. This fully automated process is performed every four hours.

3.3 User Support

The CrossGrid HelpDesk system is the main generic tool of the User Support Team. The system is installed at the Instituto de Física Corpuscular (IFIC) in Valencia, and can be accessed in <http://cg1.ific.uv.es/hlpdesk>.

It is a web based system based on the OneOrZero software [26], version v1.2 RC2 Red Lava incorporating PHP [15], JavaScript and MySQL [16]. This product is designed to be fully customizable.

All kind of questions related to the Testbed (i.e. CrossGrid V.O., Testbed installation and upgrades, network, security, resource broker, etc.) can be posed by users creating classified tickets, that are answered by a team of specialized supporters.

A Knowledge Base contains the stored problems reported by users with their corresponding solutions.

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.

This tutorial explains all the steps that a user must follow to be able to execute jobs in the testbed: from obtaining 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>.

4 Test and Validation

The CrossGrid testbed workpackage includes a task dedicated to the testbed verification and quality control. The task aims to ensure that the CrossGrid testbeds conforms to the quality requirements of a near production service. Two main areas of activity emerged from the task objectives:

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

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 testbed validation is responsible for the verification of the CrossGrid sites and services.

4.1 Software Validation

The software test and validation procedure followed by CrossGrid is defined as a set of interactions between the "verification and quality control team", the "integration team" and the developers. The procedure covers the documentation, the installation and the software functionalities. The documentation provided by the developers is followed to install the software in systems dedicated or allocated for the validation. The software is then tested using unit tests to verify the correct behavior of the software. System tests with the software integrated with the remaining testbed components are then performed. These tests are performed first at local level and then between sites if applicable. Finally stress tests are performed exercising the components.

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.

The problems detected during the validation are classified regarding their severity and priority. 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.

While applying the test and validation procedure to the CrossGrid software packages produced until November of 2004 a total of 142 software issues and 34 documentation issues have been detected these issues are distributed as follows:

4.2 Testbed Sites Validation

One of the most important steps toward the assurance of the correct site deployment is the site validation. This is a procedure intended to validate testbed sites after each major change, namely when a new middleware release is deployed.

Table 1. Issues found during validation

Severity		Priority	
Medium	72	Medium	57
High	70	High	64
Critical	21	Immediate	38
Low	13	Low	17

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 were initially performed with LCFG and later with LCFGng to reduce the number of possible installation and configuration problems. The software release is downloaded from the CrossGrid repository at FZK where the common LCFG profiles 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 site details.

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, a web enabled host verification tool that is capable of providing several installation, configuration and operation diagnostics. Once the nodes are added 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.

4.3 Testbed Monitoring

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 have been developed at LIP and are used to perform testbed and site monitoring:

- Mapcenter: developed by DataGrid, it has been improved by CrossGrid with the support for usage statistics and testbed services uptime.
- CE usage statistics: Collects and presents information about the jobs submitted to Gatekeepers. Data is obtained from the gatekeepers log file and processed to produce statistics about the job submission errors.

- RB usage statistics: Collects information about the jobs submitted through a Resource Broker. The statistics are collected from the RB logging and bookkeeping database and are processed to produce usage graphics.
- 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 is capable of detecting most of the known configuration and installation problems.
- CRL verification tool: 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.
- Site uptime: Collects and processes the Mapcenter alarms data to produce an historical view of the systems and services uptime.

5 Results

The experience with the CrossGrid testbed has shown that is feasible to use grid technologies to support interactive and parallel applications. However this class of applications requires careful testbed planning, deployment and operation thus infrastructure monitoring is a key factor for success.

Deploying and maintaining a grid site is a complex task that requires a considerable degree of knowledge covering many aspects of the operating system, grid middleware, security and networking. Experience has shown that manual configuration although possible can be difficult and highly susceptible to errors. Furthermore human resources with the necessary knowledge and practice to perform these operations correctly are frequently not available at all sites. For grid technologies to become widely available it is necessary to simplify the site management tasks.

Following the approach introduced by DataGrid the site management in CrossGrid was initially performed with the LCFG installation and management software. At a later stage LCFG was replaced by its successor LCFGng that become the official installation method until the end of the project. In spite of the problems encountered, the CrossGrid testbed has been successfully managed with LCFG. To help the site administrators in the deployment and daily maintenance of their systems a LCFG cluster installation and management manual [27] was written by UoA and common configuration profiles for each new release have been developed by FZK greatly contributing to a reduction in the deployment and upgrade times and enabling the maintenance of a homogeneous infrastructure.

The CrossGrid testbed supports parallel applications using both MPI restricted to clusters and MPI across clusters. Supporting MPI applications requires careful testbed configuration. Parallel applications that use MPI across multiple machines have an increased probability of failure in comparison with non-parallel applications.

The approach selected was to extend the DataGrid workload management system to support the submission of MPI jobs using two methods:

MPI inside clusters: Job submission from the resource broker to the best possible cluster according to the job specified requirements and using globus GRAM. The job runs inside a cluster using MPICH-P4.

MPI across clusters: Job submission from the resource broker to the best possible clusters using globus GRAM. Each application instance is started as a separate globus sub job, and can be run across different clusters. The communication mechanism between the sub jobs is provided by MPICH-G2, an MPICH implementation that uses the globus communication mechanisms (globus IO).

The last versions of the modified resource broker have shown remarkable MPICH-P4 job submission reliability to which the careful testbed monitoring and configuration has also greatly contributed.

The support for MPICH-G2 in the resource broker become available in 2004. The experience showed several new problems. The most common problem is caused by sub jobs that stay queued at a site while the remaining sub jobs enter into execution at other sites. Interactivity also introduced similar problems, as interactive applications need to be started immediately. This basic requirement collides with the nature of the load balancing systems used today in most clusters, such as PBS. These systems are designed to support the requirements of batch applications, where waiting in a queue does not constitute a problem. The interactivity issues are extremely similar to the ones from which MPICH-G2 suffers. A patch for the PBS job manager as also been developed within CrossGrid to specify that a job must enter immediately into execution, if the job remains queued it will be killed giving a change for the resource broker to restart it at some other location, or returning the control to the user.

Table 2. Testbed metrics

Month	Sites	Users			Jobs			Uptime
		Tot	Act	Ratio	Tot	Ok	Ratio	
2003 Aug	16	79	28	0.35	11424	11059	0.97	90%
2003 Sep	16	79	39	0.49	9278	8781	0.95	85%
2003 Oct	16	81	32	0.40	8919	8772	0.98	82%
2003 Nov	16	83	32	0.39	3118	1950	0.63	56%
2003 Dec	16	87	29	0.33	1627	1565	0.96	77%
2004 Jan	16	92	46	0.50	16882	16526	0.98	76%
2004 Feb	15	96	40	0.42	17471	17394	0.99	92%
2004 Mar	15	100	55	0.55	39674	39357	0.99	
2004 Apr	15	101	40	0.40	18620	18501	0.99	94%
2004 May	15	104	43	0.41	18648	18307	0.98	87%
2004 Jun	15	105	42	0.40	25687	24403	0.95	94%
2004 Jul	15	109	56	0.51	27326	27248	0.99	98%
2004 Aug	15	111	31	0.28	6148	6125	0.99	96%
2004 Sep	15	114	44	0.39	15381	14610	0.95	95%
2004 Oct	16	119	57	0.48	23279	23235	0.99	97%

A more sophisticated method to support interactive applications based on the Condor glide-in feature is also being introduced in the resource broker. This method aims to bypass the local load balancing system when submitting interactive applications. This method ensures that interactive jobs are started immediately and faster.

All these situations require careful resource monitoring. The host check tool was improved with tests that include job submissions using MPICH thus detecting the sites with problems. Tools developed at FZK also allowed the detection of SSH issues inside the clusters and the monitoring of relevant queue parameters.

The number of jobs submitted and jobs successfully executed since last year shows a clear improvement during the last months corresponding to the deployment of more stable middleware releases, including better support for MPI jobs in the workload management software.

The table 2 shows the evolution of the testbed quality indicators for the production testbed since August of 2003. The Table shows for each month the number of testbed sites, registered users, active users, jobs submitted, jobs successful, and the average testbed uptime obtained from the Mapcenter monitoring tool. It can be observed the growth of the number of users, the improvement of the job submission ratio and of the average testbed uptime. The uptime values for March of 2004 were not collected due to the upgrade of the monitoring software.

6 Conclusions

The installation and operation of the CrossGrid International Testbed has been described. More than 200 CPUs and 4 Terabytes of storage are available for application execution.

Despite the intrinsic difficulty of organizing a real distributed Grid framework across 16 different sites, in 9 different countries, and thanks to the implementation of strong quality assurance mechanisms and support tools, the statistics show that the testbed is functional and allows the development and execution of interactive jobs and parallel applications.

Further effort along the next months will be devoted to improve on those areas more critical for the execution of interactive jobs, like prioritisation mechanisms, role of the load balancing systems, and optimized data transfer.

Acknowledgements

This work has been supported by the European project CrossGrid (IST-2001-32243) and by the Polish Committee for Scientific Research through the grants KBN/115/SPB/5.PRUE/DZ206 and KBN/115/SPB/5.PRUE/DZ208.

The authors would like to thank their colleagues in EDG and EGEE projects for their nice collaboration, and also to the CrossGrid reviewers that have pushed for the improvement of quality assurance mechanisms and support tools.

References

1. The European Project CrossGrid: <http://www.eu-crossgrid.org>
2. I. Foster and C. Kesselman (eds.), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.
3. The European Project DataGrid: <http://www.edg.org>
4. The European Project EGEE (Enabling Grids for e-Science): <http://www.eu-egee.org>
5. Message Passing Interface Forum, MPI: A Message Passing Interface Standard, June 1995. <http://www.mpi-forum.org>
6. A grid enabled implementation of the MPI standard: <http://www3.niu.edu/mpi/>
7. An implementation of the MPI standard for applications running inside clusters: <http://www-unix.mcs.anl.gov/mpi/mpich>
8. The CrossGrid Project Book, in press, Springer LNCS Series.
9. J. Gomes et al., "First Prototype of the CrossGrid Testbed", Proc. AcrossGrids Conf., Santiago, February 2003; ISBN 3-540-21048-2 Lecture Notes in Computer Science 2970: 67-77, 2004.
10. The Globus Project: <http://www.globus.org>
11. Condor Project Homepage: <http://www.cs.wisc.edu/condor>
12. The LHC Computing Grid Project: <http://lcg.web.cern.ch>
13. J.Astalos et al., "International Grid CA Interworking". Submitted to EGC'05, <http://www.eugridpma.org>
14. The Local Configuration software: <http://www.lcfg.org>
15. The Hypertext Preprocessing scripting language: <http://www.php.net>
16. The MySQL relational database: <http://www.mysql.org>
17. The CrossGrid Developers' Guide: <http://gridportal.fzk.de/websites/crossgrid/iteam/devguide/devguide-html>
18. The CrossGrid central software repository and development server, GridPortal: <http://gridportal.fzk.de>
19. The Savannah software: <http://gna.org/projects/savane>
20. CVS Documentation: <http://www.cvshome.org>
21. Autobuild entry page: <http://savannah.fzk.de/autobuild/i386-rh7.3-gcc3.2.2>
22. Autobuilt RPMs: <http://savannah.fzk.de/distribution/crossgrid/autobuilt>
23. cg-lcfg-go: <http://cvs.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp4/config/cg-wp4-lcfg-server-files/cg-lcfg-go>
24. Jorge Gomes et al., "CrossGrid Deliverable D4.1 Appendix D" (<http://www.lip.pt/computing/projects/crossgrid/doc/deliverables/TestProcedure.pdf>)
25. CVS Documentation: <http://www.loria.fr/~molli/cvs/doc/cvstoc.html>
26. OneOrZero software: <http://helpdesk.oneorzero.com>
27. The CrossGrid cluster installation manual: <http://cgi.di.uoa.gr/xgrid/archive.htm>