

## A GRID INFRASTRUCTURE FOR PARALLEL AND INTERACTIVE APPLICATIONS

J. Gomes<sup>a</sup>, G. Borges<sup>a</sup>, M. Montecelo<sup>a</sup>, M. David<sup>a</sup>, B. Silva<sup>a</sup>, N. Dias<sup>a</sup>, J.P. Martins<sup>a</sup>, C. Fernández<sup>b</sup>, L. García-Tarrés<sup>b</sup>, C. Veiga<sup>b</sup>, D. Cordero<sup>b</sup>, J. López<sup>b</sup>, J. Marco<sup>c</sup>, I. Campos<sup>c</sup>, D. Rodriguez<sup>c</sup>, R. Marco<sup>c</sup>, A. López<sup>c</sup>, P. Orviz<sup>c</sup>, A. Hammad<sup>d</sup>, M. Hardt<sup>d</sup>, E. Fernández<sup>e</sup>, E. Heymann<sup>e</sup>, M. A. Senar<sup>e</sup>, A. Padee<sup>f</sup>, K. Nawrocki<sup>f</sup>, W. Wislicki<sup>f</sup>, P. Heinzlreiter<sup>g</sup>, M. Baumgartner<sup>g</sup>, H. Rosmanith<sup>g</sup>, S. Kenny<sup>h</sup>, B. Coghlan<sup>h</sup>, P. Lason<sup>i</sup>, L. Skital<sup>i</sup>, J. Astalos<sup>j</sup>, M. Ciglan<sup>j</sup>, M. Pospieszny<sup>k</sup>, R. Valles<sup>l</sup>, K. Dichev<sup>m</sup>

- a) *Laboratório de Instrumentação e Física Experimental de Partículas, Lisboa, Portugal*
  - b) *Centro de Supercomputación de Galicia, Santiago de Compostela, Spain*
  - c) *Instituto de Física de Cantabria, IFCA, CSIC-UC, Santander, Spain*
  - d) *Steinbuch Centre for Computing, Forschungszentrum Karlsruhe, Karlsruhe, Germany*
  - e) *Universidad Autónoma de Barcelona, Barcelona, Spain*
  - f) *Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Warsaw, Poland*
  - g) *GUP Linz, Joh. Kepler University, Linz, Austria*
  - h) *Trinity College Dublin, Dublin, Ireland*
  - i) *Academic Computer Centre CYFRONET AGH, Krakow, Poland*
  - j) *Institute of informatics, Slovak Academy of Sciences, Bratislava, Slovakia*
  - k) *Poznan Supercomputing and Networking Center, Poznan, Poland*
  - l) *Instituto de Biocomputación y Física de Sistemas Complejos, Zaragoza, Spain*
  - m) *High-Performance Computing Center Stuttgart, Stuttgart, Germany*
- e-mail: {jorge, goncalo}@lip.pt*

**Abstract.** The int.eu.grid project aims at providing a production quality grid computing infrastructure for e-Science supporting parallel and interactive applications. The infrastructure capacity is presently about 750 cpu cores distributed over twelve sites in seven countries. These resources have to be tightly coordinated to match the requirements of parallel computing. Such an infrastructure implies high availability, performance and robustness resulting in a much larger management effort than in traditional grid environments which are usually targeted to run sequential non-interactive applications. To achieve these goals the int.eu.grid project offers advanced brokering mechanisms and user friendly graphical interfaces supporting application steering. The int.eu.grid environment is deployed on top of the

gLite middleware enabling full interoperability with existing gLite based infrastructures.

**Keywords:** Grid computing, parallel jobs, interactivity, interoperability, crossbroker, roaming access server, migrating desktop

## 1 INTRODUCTION

The Interactive European Grid project (int.eu.grid) [1] started in April 2006 and operates an advanced production quality grid infrastructure providing services for users with demanding interactive and parallel applications. The infrastructure capacity is presently about 750 CPU cores, distributed over twelve sites (clusters) in seven countries, and connected through the Geant Network. More than 200 users, registered in thirteen project virtual organizations, can run batch jobs, parallel MPI jobs and interactive tasks with immediate execution needs. The effort needed to support such advanced features is huge, representing a challenge from the infrastructure management point of view, both for local clusters and central services. The int.eu.grid project can be seen as a complementary extension of already existing European grid projects, such as EGEE [2], thus allowing multi-science applications to profit from the coordinated effort to support MPI and interactive environments over the grid. A strong effort has been put in establishing infrastructure interoperability scenarios and in creating technical solutions to enable virtual organizations to operate across different projects.

This paper describes the int.eu.grid infrastructure covering areas such as the production and development testbeds, accounting, monitoring, infrastructure performance, interoperability solutions and operation coordination.

## 2 INFRASTRUCTURE OVERVIEW

### 2.1 Production and Development Services

The int.eu.grid infrastructure is split in two independent grid computing services called *production* and *development*. The production service is composed of nine clusters and provides a reliable environment to run end-users applications. Each cluster deploys different CPU architectures (Xeon, Opteron, Pentium D, Itanium) and uses their preferred Local Resource Management Systems (LRMS) for job management control. The majority of applications running in int.eu.grid infrastructure require IA32, EMT64 or AMD64 architectures although the Itanium support was also available at the beginning of the project.

The development infrastructure is supposed to provide a realistic yet flexible environment for development activities. It is composed of four clusters and is used to support the testing and integration of the middleware and to allow early access to new middleware functionalities. For higher flexibility and to maximize the available resources, development sites make a heavy use of virtual machines.

Tables 1 and 2 provide a brief summary of the sites most important features, namely their capacity, cluster architecture, LRMS and bandwidth connection.

Prod Site	Country	Cores	Arch	Storage	LRMS	Bandwidth
BIFI	Spain	22	Xeon	0.1 TB	Torque	1 Gbts
CESGA	Spain	20	Xeon	4.8 TB	Torque	7.5 Gbts
CYFRONET	Poland	20	Xeon	1.0 TB	Torque	1 Gbts
ICM	Poland	32	Opteron	1.3 TB	Torque	1 Gbts
IFCA	Spain	314	Xeon	1.1 TB	Torque	$2 \times 622$ Mbts
IISAS	Slovakia	32	PD	0.5 TB	Torque	100 Mbts
LIP	Portugal	57	Opteron	1.6 TB	SGE	1 Gbts
PSNC	Poland	107	Itan/Xeon	4.4 TB	Torque	1 Gbts
FZK	Germany	100	Opteron	1.8 TB	Torque	1 Gbts

Table 1. Production sites properties

Dev Site	Country	Cores	Arch	Storage	LRMS	Bandwidth
FZK	Germany	4	Xeon	3.4 GB	Torque	1 Gbts
TCD	Ireland	26	PIII	3 TB	Torque	1 Gbts
UAB	Spain	30	PIV	380 GB	Torque	100 Mbts
GUP	Austria	10	Xeon/AMD	500 GB	Torque	100 Mbts

Table 2. Development sites properties

The production infrastructure has a total of  $\sim 700$  CPU cores and  $\sim 16$  TB of storage. Nevertheless, with the expected improvements coming in the near future, such as the deployment of higher capacity SRM DPM [3] storage elements at ICM and BIFI, the upgrade of the network bandwidth to 1 Gbts + 1 Gbts at IFCA, the deployment of an IBM General Parallel File System (GPFS) [4] parallel file system at IFCA, and the CPU and storage increase at CESGA and PSNC, the production infrastructure will soon become even more powerful and attractive to users.

## 2.2 Core and Local Services Architecture

The int.eu.grid infrastructure architecture is depicted in Figure 1 and relies on the same basic building blocks as EGEE. However, to enable interactivity and the possibility to run MPI parallel applications on grid environments, the int.eu.grid collaboration has enhanced pre-existing EGEE services, and deployed some additional ones.

The following set of specific int.eu.grid core services must be in place:

- The CrossBroker [5] represents the heart of the int.eu.grid workload management system providing efficient job management scheduling for parallel jobs inside clusters and across clusters. The CrossBroker was initially developed by the CrossGrid [6] project and is based on the LCG [7] Resource Broker and gLite [8] WMS. Its capacities will be described in detail in Section 3.

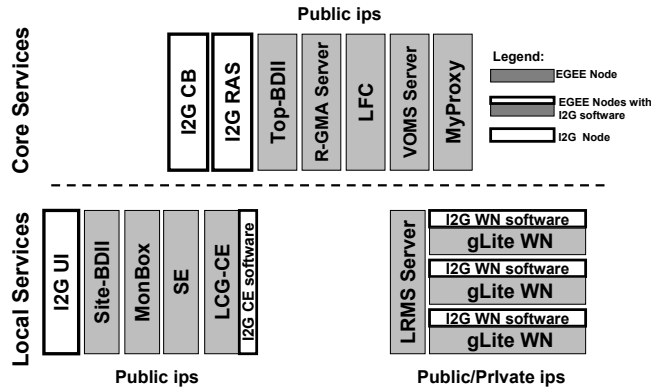


Fig. 1. Int.eu.grid core and local services

- Roaming Access Server (RAS) is a service that acts as a relay between the grid and the Migrating Desktop (MD) [9], a thin Java [10] client enabling end-users access to grid resources through a graphical interface. Together, the MD and the RAS are two key components of the int.eu.grid approach to an integrated user-friendly grid environment. Both were initially developed within the CrossGrid project and have been extensively improved in the int.eu.grid context. An extensive description will be presented in Section 3.

The remaining core services (Top Berkeley Database Information Index, MyProxy-Server, Logical File Catalogue, Virtual Organization Membership Service and R-GMA server) are used and deployed as in Egee infrastructures without any additional change (apart from configuration aspects needed to fit int.eu.grid specific needs).

The production core services are provided mainly by LIP and IFCA (see Figure 2). Both sites offer the same set of services that are thus duplicated, a feature which is important for load balancing and for providing better resilience to failures. Nevertheless, the replication mechanism implemented regarding VOMS and LFC servers at both sites is worth being explained. The VOMS server is the core service for managing authorization data in the int.eu.grid project, which provides a database of users, groups, roles and capabilities that are grouped in Virtual Organizations. When needed, users query the VOMS Server to get their VO grid credentials. Furthermore, VOMS-enabled services (CE, SE and others) use VOMS aware plug-ins to validate user access requests. Since the VOMS Server is a very critical service which must be available all the time, the int.eu.grid project developed a simple service replication schema to keep databases synchronized and to provide a distributed and transparent access for users and grid services. The architecture of the replication process in int.eu.grid project relies on a simple master-slave schema where the master host receives all administrations tasks and new membership requests for each VO, and the slave one is automatically updated according to master host. Moreover, the slave host keeps a read-only database which allows proxy renewal. Finally, load balance was implemented providing scalability, fault tolerance and high availability in the grid infrastructure. The LFC server, providing the namespaces and physical location of data

files over the grid, follows the same replication schema as the VOMS server. Finally, CESGA supports the core R-GMA [11] registry server used for accounting and monitoring. Int.eu.grid developed very important new features aimed to the proper monitoring and accounting of parallel and interactive computing tasks, will be described in detail in Section 2.4.

The core services for the development infrastructure are maintained by FZK. The FZK site relies on a very flexible setup based on virtual machines ensuring fast recovery of the site systems.

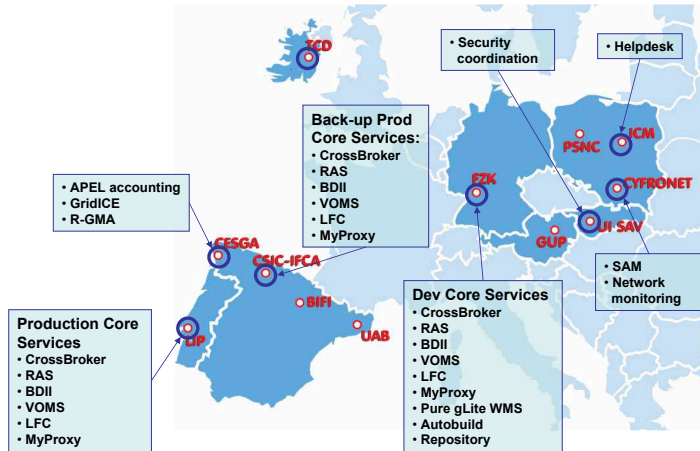


Fig. 2. Int.eu.grid services geographical distribution

Regarding local services, both production and development infrastructure follows the same design as EGEE project meaning that each site must deploy a User Interface (UI), a LCG Computing Element (lcg-CE), a Storage Element (SE), a Monitoring Box (Mon-Box) and several Worker Nodes (WNs). The most strong difference comes in the UI which provides all the int.eu.grid workload management client command tools to properly interact with the CrossBroker. Other services as the lcg-CE and WNs just have to support a thin int.eu.grid software layer on top of the ordinary gLite services. This software layer deploys and installs the MPI implementations supported within the project (OpenMPI [12] and PACX-MPI [13]) and the MPI-START mechanism, a solution developed inside the int.eu.grid project to encapsulate the MPI, batch system and cluster specific details (see Section 3).

### 2.3 General Tools

The FZK site is also responsible for maintaining a wide set of very important project tools, such as the *Autobuild system*, the *APT-repository* and the project *Wiki*.

The test-autobuild [14] is an open source tool that was installed on two dedicated build servers where it continuously checks for modifications in the int.eu.grid SVN repositories. In case changes have been made to a package, it downloads and compiles it. The status of

the latest build is available through a web page showing an overview about all packages and more detailed information per package in individual pages. The successful output of a build generates RPM packages which are then copied to an APT repository.

APT stands for Advanced Package Tool. It indexes its repository so that APT clients can obtain a detailed description of available packages. APT clients are installed on every int.eu.grid infrastructure machine. The clients frequently check for the availability of packages newer than the ones installed on the client machine. If newer packages are found, they can be downloaded and installed. This process can be done automatically or by a system administrator. In int.eu.grid we have defined three different repositories: one for the packages that are autobuilt; this repository is used only by experimental sites in order to test and validate new packages. If tested and validated successfully the package can be moved into the development repository to which all development sites subscribe. If the package performs well in development it will be released in a production release.

The project wiki server [15] provides an easy tool for documenting those parts of the project that are changing frequently, e.g. test and validation reports and central services hostnames. This is also the place where some admin documentation is available, as for example the *Developers Guide* describing how to use the development infrastructure and the policies of the project in more detail.

## 2.4 Monitoring and Accounting

Monitoring is an important proactive service that contributes to the detection and isolation of infrastructure problems. To monitor the int.eu.grid infrastructure, two main tools have been deployed, GridIce and Service Availability Monitoring (SAM).

GridIce is a distributed monitoring tool designed for grid systems by INFN in Italy [16]. Different aggregations and partitions of monitoring data are provided based on the specific needs of different users categories (VO, Site, etc.). Two servers have been deployed and configured to monitor production [17] and development [18] sites.

The Service Availability Monitoring (SAM) [19] developed at CERN aims to provide a site independent, centralized and uniform monitoring tool for all grid services (called sensors). It is the main source of monitoring information for high-level grid operations based on the execution of different VO dependent periodic tests such as job-submission and file replication. To increase the monitoring features of the int.eu.grid project, a set of new tests has been developed to monitor the parallel and interactive capabilities of the infrastructure. These tests were included inside the CE sensor, are started on an hourly basis, and check if:

- OpenMPI and PACX-MPI parallel jobs using two and three CPUs (respectively) can be properly executed at all sites
- Interactive jobs can be executed in all CEs.

OpenMPI and PACX-MPI tests requesting a greater number of nodes (eight CPUs per job) are also submitted once a day. New sensors are presently being developed to check the Migrating Desktop and the CrossBroker capabilities. Based on the results from SAM tests, a notification service has been set-up to communicate to site admins a change in their site availability. The different SAM components (server, client and portal) can be assessed at [20].

The new int.eu.grid accounting portal [21] is the graphical front-end to the int.eu.grid accounting data. Its underlying design is based on the one deployed in EGEE [22] but extended to fulfill the requirements of the int.eu.grid community. The accounting web user interface was enhanced and includes new job properties as the *Waiting time* defined as the time difference between submission and execution times. All data can also be displayed according to new categories:

- Job type: Job distribution as a function of the job nature (*Sequential Batch, Parallel Batch or Interactive*).
- Parallel type: Jobs distribution as a function of the parallelism type (MPICH, Open-MPI, PACX-MPI, plain, other).
- Interactive: Interactive agents distribution for interactive jobs (*Batch or I2GLogin*).
- Node Number: Requested CPUs distribution for parallel jobs.

As in EGEE, most of the data is obtained from the project R-GMA server. The new information is gathered from the production CrossBrokers mySQL databases containing the job requirements for each individual submission. This data is collected, parsed and transformed into a suitable database field format and then stored in a local mySQL database.

### 3 INFRASTRUCTURE CAPABILITIES

Int.eu.grid's main focus is to enable interactive and parallel user-friendly computing over the grid but maintaining the same ordinary features as other gLite based infrastructures. A user can choose to access to the infrastructure capabilities both from a gLite user interface enhanced with the int.eu.grid middleware or from the Migrating Desktop (MD) GUI developed within the project. While the gLite classic user interface is a good solution for developers working in porting applications, since it combines the Linux development environment with the gLite and Int.eu.grid capabilities; the MD GUI is a good solution for the end-user who wants to focus on achieving his or her scientific goals, without bothering with the underlying grid details.

The MD GUI and RAS work together in a client/server fashion. The RAS is a core service installed at a central location, with direct network connectivity as any regular server; while the MD is a client which connects to the RAS and can be run from the private workstations of the grid users (as long as it can connect to the appropriate RAS ports). This architecture provides a higher decoupling between the MD and the underlying grid middleware while at the same time enables the end-user to access the infrastructure from a wider range of platforms. The MD and RAS combination addresses issues still considered open in grid environments: it is a powerful GUI that can be started using Java from any desktop, platform or portable computer independently from its location, thus making the grid easily available to non-experts. The MD/RAS server version in production supports Java 1.6 and enables the submission of batch, interactive and parallel jobs using both OpenMPI and PACX-MPI (including data management and job management grid operations).

The CrossBroker matchmaking capabilities for parallel applications are a key functionality for the int.eu.grid infrastructure since they enable the correct scheduling of the jobs to the most appropriate sites without dropping the standard LCG Resource Broker

features available in gLite based infrastructures. If resources inside single clusters are not enough, the CrossBroker can join resources from multiple sites. This feature exploits PACX-MPI and enables to run parallel jobs larger than the sites capacity optimizing the resource usage. The CrossBroker uses a highly flexible start-up approach for parallel jobs through the invocation of the MPI-START mechanism developed in int.eu.grid to encapsulate MPI, batch system and cluster specific details. This approach decouples the details of the clusters from the Broker implementation which is still a serious limitation of the LCG and WMS resource brokers making extremely difficult to support multiple MPI implementations and heterogeneous sites. Finally, a new major functionality has been introduced with the Condor glidein [23] mechanism. When this functionality is active, glidein instances are attached to the jobs submitted by the CrossBroker. Once started in the WNs these glidein instances declare themselves to the CrossBroker as usable computing resources. The CrossBroker can then submit job instances directly to these resources bypassing the gatekeeper and enabling much faster start-up time and higher reliability for interactive job submissions.

The interactivity functionalities are based on the compression of the OpenGL [24] graphics output generated by the applications running in the clusters which is sent to the RAS server through a Glogin [26] communication channel. This stream is then forwarded to the MD where it is rendered by a client. GVID [25] performs the encoding and decoding of the graphics on both sides. For steering purposes the input events generated by the user such as mouse clicks are sent back from the MD to the application running in the remote cluster. An application specific plug-in for the MD has to be developed implementing the dialogs and buttons showed through the MD; and their translation to events that are sent to the running application. The application itself has to be adapted to react to these input events.

The added value of these tools is clear and these int.eu.grid developments are already being used by other grid projects. EGEE and EELA projects are using MPI-START and are actively collaborating in providing feedback and enhancements for this component. The MD tool is also being used in several projects and regional infrastructures such as Baltic Grid. The IBERGRID initiative joining Portuguese and Spanish grid resources is planning to deploy the full int.eu.grid framework to support MPI and interactivity and in this way to complete the gLite capabilities.

#### 4 INFRASTRUCTURE USAGE AND RELIABILITY

The int.eu.grid project is supporting 13 virtual organizations (VOs) and more than 200 users. Table 3 shows the complete list of project specific virtual organizations.

Since the beginning of the int.grid project (April 2006) until end of 2007, more than 270 000 jobs were submitted to the infrastructure, 68% of which were of the batch type, 27% of the parallel type and 5% of the interactive type. One should note that int.eu.grid only entered a real production phase after the first year of operation meaning that we expect a huge increase on the number of parallel and interactive jobs from now on.

The distribution of job submissions per VO shows that the main VO has the higher number of job submissions (41.5%) followed by the imon VO (26.3%). These can be explained by the extensive monitoring tasks scheduled each hour to all infrastructure sites by the central monitoring services. Nevertheless, looking to the CPU time usage, we



VO Name	User VO	Description
imain	YES	Open to all project members
imon		Restricted to infrastructure monitoring tools
itut		Aimed for tutorials and dissemination events
itest		Aimed for supporting software test and user induction
ifusion	YES	Nuclear fusion applications
iusct	YES	Ultra sound computer tomography applications
ienvmod	YES	Environment-modelling applications
ihep	YES	High Energy Physics applications
iplanck	YES	Planck space mission application
ibrain	YES	Brain images processing applications
icompchem	YES	Computational chemistry
icesga	YES	CESGA computing centre users
iwien2k	YES	Wien2k application

Table 3. List of VO's supported in int.eu.grid project

conclude that the number of job submissions does not always match a high usage of the CPU resources. Indeed, the applications VOs are the more CPU time consuming VOS. The ifusion VO leads this rank with 55.6 % of all consumed CPU time followed by the ienvmod VO with 20.1 %.

It is possible to assess the infrastructure reliability based on the information stored in the logging and bookkeeping database from the production CrossBrokers. This database keeps records on the history of each job allowing to determine the success rates within the infrastructure and the cause/reason of the most common failures. Table 4 reports the most relevant job states, as seen from LIP production CrossBrokers from March 2007 to November 2007, and the number of jobs going through each of those states.

Jobs States	Total Jobs	Fraction (%)
Submitted	226444	100.0
Accepted	223269	98.5
Matched	159097	70.2
Run	147411	65.0
Done OK	144135	63.6
Cleared	134239	59.2

Table 4. Number of jobs processed by LIP production brokers from March to November 2007

During the time period considered in this analysis, 226 444 jobs have been successfully registered, meaning that they were successfully authenticated and authorized, and gave origin to new job registrations. Only 1.5 % was not accepted for crossbrokering probably due to bad use of submission protocols/syntax or to problems in the input sandbox file transfers. After job acceptance, the requirements of each job are matched against the available infrastructure resources, giving origin to a list of CEs where the job can be

executed. The highest number of job submission failures happens during this step (28.3%). Looking to the reasons of such failures, one concludes that most failures are caused by incoherent user specified requirements such as requesting for features that do not exist, or asking for resource quantities not available to the VO at a given time. Due to the parallel computing specificities of int.eu.grid infrastructure, a user may ask for a large amount of processors strongly decreasing the matchmaking probability for other users with the same kind of requests since the total number of CPUs is limited.

The jobs are then submitted to a given lcg-CE, namely to its globus GRAM gatekeeper service, chosen in agreement with the default or user defined ranking expression. The gatekeeper authenticates and authorizes the job at the cluster level and submits it to local resource management system (LRMS), which in int.eu.grid case can be the PBS/Torque or SGE batch systems, via the proper job-manager interface. The file which is actually executed is a *job wrapper* produced by the CrossBroker. This script is responsible for fetching the input sandbox files from the broker, starting abstraction layers such as MPI-START and interactivity agents, or the executable itself, and finally transferring the output sandbox files back to the broker. In average 65.0% of the original submissions entered *Running* state meaning that the job wrapper started to execute. The state *Done OK* corresponds to successful jobs. The number of jobs that entered in *Running* state but did not succeed was 1.4%. This value also includes cancelled jobs by the user and jobs failing due to file transfer problems. Finally, the *Cleared* job status stands for jobs where the output sandbox files have been retrieved by the user, an action which is clearly dependent on the user side.

Based on the previous analysis one can conclude that from the 159 097 matched jobs, 144 135 run successfully corresponding to a CE job submission success rate of 90.6%. This value is in accordance with other gLite based infrastructures.

## 5 INTEROPERABILITY

The int.eu.grid project is committed to provide interoperability with other gLite based infrastructures. Several scenarios have been studied aiming at detailing how an already existing VO using other gLite based infrastructures could take advantage of the int.eu.grid capabilities. Several levels of interoperability are proposed, from simple job submission interoperability to site integration in the int.eu.grid infrastructure [27].

VOs wishing to use the int.eu.grid infrastructure should start by establishing a service level agreement with the project. This agreement is then implemented through the addition of the VO to the int.eu.grid infrastructure. This allows the VO users to access and use both infrastructures and provides authorization interoperability.

A first level of interoperability consists in the job submission to different infrastructures from the same UI server. This step requires the deployment of a small set of int.eu.grid packages without changing the existing gLite setup. Users are then allowed to submit jobs to gLite infrastructures using gLite commands and to the int.eu.grid infrastructure using int.eu.grid commands. Both types of commands are slightly different while the arguments and language syntax is fully compatible.

When a given VO wishes to share their gLite based resources with int.eu.grid project, profiting from both infrastructures advantages, this requires the deployment of a small number of additional int.eu.grid packages in the WNs. Moreover, it also implies the

deployment of a dedicated int.eu.grid CE (recommended) or the deployment of additional packages on an existing CE together with some additional configurations. The batch system has also to be configured to support MPI applications.

Finally, VOs can achieve full interoperability at the infrastructure services level including publishing of accounting information in the int.eu.grid portal and site monitoring by int.eu.grid central monitoring services. This involves the deployment of a separated MonBox or the configuration of the existing MonBox to publish in the R-GMA servers of both infrastructures and the inclusion of the site-BDII in int.eu.grid top-BDII.

Using any of the previous approaches, the VOs will benefit from the MPI and interactivity capabilities as well as the access to int.eu.grid resources without losing their current functionalities and at the cost of a reduced deployment effort. Most of infrastructure complexity and operational costs remains hidden in the core services maintained by int.eu.grid.

## 6 CONCLUSIONS

The int.eu.grid infrastructure and its generic capabilities have been presented. More than 750 CPUs and 16 TB of storage capacity, distributed over twelve sites and seven countries, are available for end-users to run their applications. The quality assurance metrics, the monitoring, the coordinated MPI support and the middleware releases aimed at enabling interactive and parallel job execution over grid environments represent a huge effort never developed in previous projects.

Along the first year, int.eu.grid project implemented a considerable set of mechanisms which allow batch, parallel and interactivity multi-science applications to successfully execute over a reliable, production quality grid infrastructure. The success of int.eu.grid approach is shown by the fact that other grid projects, such as EGEE, EELA and Baltic-Grid, are adopting int.eu.grid middleware to enable local MPI support and a user friendly access to their resources. The second year of the project is aiming at pushing for a higher usage of int.eu.grid resources through the establishment of a strong business model and through implementing service level agreements with exterior VOs, always ensuring an interoperability framework with other gLite based infrastructures.

## REFERENCES

- [1] Int.eu.Grid web site. Available on: <http://www.interactive-grid.eu/>.
- [2] EGEE web site. Available on: <http://www.eu-egee.org>.
- [3] Available on : <https://twiki.cern.ch/twiki/bin/view/LCG/DataManagementTop>.
- [4] Available on : <http://www-03.ibm.com/systems/clusters/software/gpfs>.
- [5] FERNÁNDEZ, E.—HEYMANN, E.—SENAR, M. A.: Supporting Efficient Execution of MPI Applications Across Multiple Sites. In: 12<sup>th</sup> International Euro-Par Conference, Dresden, Germany, Lecture Notes in Computer Science, Vol. 4128, 2006, pp. 383–392.
- [6] GOMES, J. et al: Experience with the Internation Testbed in the CrossGrid Project. In: Advances in Grid Computing – EGC 2005, LNCS 3470, Amsterdam, The Netherlands, February 2005, pp. 98–110.

- [7] LCG web page. Available on : <http://lcg.web.cern.ch/LCG/>.
- [8] gLite web page. Available on : <http://cern.ch/glite>.
- [9] KUPCZYK, M.—LICHWAŁA, R.—MEYER, N.—PALAK, B.—PŁÓCIENNIK, M.—WOLNIEWICZ, P.: “Applications on demand” as the Exploitation of the Migrating Desktop. *Future Generation Computer Systems*, Vol. 21, 2005, No. 1, pp. 37–44.
- [10] java web page. Available on : <http://java.sun.com/>.
- [11] R-GMA web page. Available on : <http://www.R-GMA.org/>.
- [12] GABRIEL, E. et al.: Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. In: D. Kranzlmüller, P. Kacsuk and J.J. Dongarra (Ed.), *Proceedings of the 11<sup>th</sup> European PVM/MPI, Users Group Meeting, Lecture Notes in Computer Science (LNCS)*, Budapest, Hungary, Vol. 3241, 2004, pp. 97–104.
- [13] KELLER, R.—GABRIEL, E.—KRAMMER, B.—MULLER, M. S.—RESCH, M. M.: Towards Efficient Execution of MPI applications on the Grid: Porting and Optimization issues. In: *Journal of Grid Computing*, Vol. 1, 2003, No. 2, pp. 133–149.
- [14] Autobuild web page. Available on : <http://autobuild.org>.
- [15] Int.eu.grid wiki web page. Available on : <https://wiki.fzk.de/i2g>.
- [16] GridIce web page. Available on : <http://grid.infn.it/gridice/>.
- [17] Available on : <http://gridice.i2g.cesga.es/gridice/site/site.php>.
- [18] Available on : <http://gridice2.i2g.cesga.es/gridice/site/site.php>.
- [19] SAM web page. Available on : <http://sam-docs.web.cern.ch/sam-docs>.
- [20] Available on : <https://sam.cyfronet.pl/i2g-sam/sam.py?>
- [21] Int.eu.grid accounting web page. Available on : <http://acct.i2g.cesga.es>.
- [22] REY, P.—LÓPEZ, J.—FERNÁNDEZ, C.—KANT, D.—GORDON, J.: The Accounting Infrastructure in EGEE. In: *Ibergrid 1<sup>st</sup> Iberian Grid Infrastructure Conference proceedings*, May 2007, pp. 103–112.
- [23] Condor glidein web page. Available on : <http://www.bo.infn.it/condor-mirror/manual/v6.6/condor.glidein.html>.
- [24] OpenGL web page. Available on : <http://www.opengl.org/>.
- [25] POLAK, M.—KRANZLMÜLLER, D.: Interactive Videostreaming Visualization on Grids. In: J. Dongarra, B. Tourancheau (Eds.), *Cluster and Computational Grids for Scientific Computing, Future Generation Computer Systems, Special Section*, Elsevier Science, Amsterdam, The Netherlands, Vol. 24, 2008, No. 1, pp. 39–45.
- [26] ROSMANITH, H.—KRANZLMÜLLER, D.: glogin – A Multifunctional Interactive Tunnel into the Grid. In: R. Buyya (Ed.), *Proc. Grid 2004, 5<sup>th</sup> IEEE/ACM Intl. Workshop on Grid Computing*, IEEE Computer Society, ISBN 0-7695-2256-4, Pittsburgh, PA, USA, November 2004, pp. 266–272.
- [27] Available on : <http://www.lip.pt/grid/I2G-EGEE-Interoperability-v0.2.doc>.



**Jorge GOMES** is a senior computing research scientist at LIP where he coordinates the Computer Centre and the grid computing team. He is currently participating in the EGEE, i2g, LCG and IBERGRID projects. He is the i2g infrastructure operations manager.