2nd Grid Plugtests Report

October 10th-14th 2005

OFFICIAL SPONSORS

EUROPEAN PARTNERS

WITH SUPPORT OF

GRID@WORK CO-ORGANIZERS

Contents

Chapter 1

Introduction

Following the success of the 1st Grid Plugtests, during the 10th-14th of October 2005 the 2nd Grid Plugtests was held. Organized by ETSI and INRIA, the objectives were: to test Grid interoperability, and to learn, through the user experience and open discussion, about the future features needed for Grid middlewares.

The 2nd Grid Plugtests consisted of several events: Conferences, Workshops, Tutorials and a Contest. Drawing over 240 participants from many different countries. The events were organized as follows:

Monday During the Second ProActive User Group talks were given regarding the use of ProActive, from the introduction of the middleware, to descriptions of its use and it's current research status. Also, during the first day, EGEE "Introducing activities and their successes" took place.

Tuesday The ProActive Tutorial was held, along with the Unicore Summit.

Wednesday The second part of the Unicore Summit, the CoreGrid Workshop: "Grid Systems, Tools and Environments" and the Industrial Session took place.

Thursday The GridCoord Workshop: "The use of Open Middleware for the Grid", GAT Tutorial and Ibis Tutorial were held.

Friday The CoreGrid Workshop: "Programming models and components for the Grid", P-Grade Tutorial, NorduGrid Event: "ARC open day" took place.

Also, during the first three days (Monday-Wednesday) two contests took place (Section 3) with 8 participating teams. For the contest, a Grid was setup (Section 2) by the OASIS Team using the ProActive middleware, which inter-operated with several other middlewares and protocols.

These events were organized by ETSI Plugtests and the INRIA OASIS research team. OASIS is a joint team between INRIA, UNSA, I3S-CNRS which develops the ProActive Grid middleware, hosted by ObjectWeb. The event was officially sponsored by e-Europe, IBM, Microsoft, SUN Microsystems, and financially supported by Region PACA, INRIA, I3S. The Flowshop contest was sponsored by ROADREF.

Chapter 2

The Grid

2.1 Installation

To run experiments on Grid computing, a Grid was setup for three days with the help of numerous partners. This Grid was deployed on 13 different countries, in more than 40 sites, gathering 2700 processors for a grand total of more than 450GFlops (measured with the SciMark 2.0 benchmark).

Given the heterogeneity of the sites, each site had to be configured and fine tuned. This involved figuring out the operating system, installing an adequate Java Virtual Machine for the operating system (when not already installed), figuring out the network/firewall configuration, job scheduler, etc. This worked was handled by the OASIS Team, mainly Romain Quilici and Mario Leyton, who prepared the Grid for the contest and Plugtests.

The deployment was thus made very simple and transparent for the Plugtests users, who had all the architecture details hidden by the ProActive layer.

2.2 ProActive

ProActive is a LGPL Java library for parallel, distributed, and concurrent computing, also featuring mobility and security in a uniform framework. With a reduced set of simple primitives, ProActive provides a comprehensive API allowing to simplify the programming of applications that are distributed on Local Area Networks (LAN), on clusters of workstations, or on Internet Grids.

The deployment descriptors provide a mean to abstract from the source code of the application any reference to software or hardware configuration. It also provides an integrated mechanism to specify external process that must be launched, and the way to do it. The goal is to be able to deploy an application anywhere without having to change the source code, all the necessary information being stored in an XML Deployment Descriptor file.

Since programming the Grid cannot be achieved at a low-level of abstraction, ProActive is provided with a programming model. The complexity that arises from scale, heterogeneity, and dynamicity cannot be tackled with message-level primitives. As such, development of new Grid programming models have to rely on higher-level of abstraction that the current usage. These programming models are based on the component technology.

2.2.1 Methodology

The following steps describes, in a broadly manner, the methodology used to configure each site for the Grid. Average time of configuration varied depending on the complexity of the site from less than an hour to several weeks.

1. Invite partner to participate in the 2nd Grid Plugtests.
2. Request partner to open an account for the Plugtests.
3. Analyse and configure the environment of the site: operating system, network restrictions, data storage, scheduler, asymmetric access keys, Java virtual machine.
4. If necessary, develop ProActive support for this site.
5. Build script for synchronizing ProActive libraries and files with the site.
6. Build cleaning script for the site.
7. Build XML Deployment Descriptor for the site.
8. Test the site.

From all the steps described here, steps 1 and 2 usually took a long time, and depended on the response time of the site administrators.

Steps 3 and 4 were the most demanding for the OASIS Team, since they required careful inspection of the site, and sometimes protocol interoperability development (see Sections 2.3, 2.2.2).

Steps 5 and 6 were fairly easy to build, and proved to be most useful during the Plugtests. On the first place, to install (when it was requested by the contestant) the application libraries (jar). This was done to improve the deployment time by avoiding dynamic class loading. Secondly, for cleaning the Grid between each contestant's run.

Steps 7 and 8 were fairly simple when not dealing with complex configuration sites, but when facing problems usually required to go back and fine tune a previous step until eventually the site was correctly configured.

2.2.2 Environment Configuration

Figuring out the environment configuration of a site was a key process in building the Grid. Given the heterogeneousness of the Grid, the environment varied considerably from site to site. The most important aspects of the environment can be grouped into the following areas: Operating System & JVM, Schedulers and Site Access, Network and Firewalls and Data Storage.

2.2.2.1 Operating System and JVM

Since ProActive requires Java to operate, a very important aspect of site configuration is to determine whether a JVM is installed on the site, and therefore on each node of the Grid. On some cases, after searching in the remote sites, a proper JVM was found to be installed and was used.

When no JVM was found, the Operating System (`Linux, AIX, SGIIrix, MacOS, Solaris`) and hardware architecture had to be determined (`x86_32, x86_64, ia64, AIX, SGIIrix, PPC, Sparc`). Afterwards, a suitable JVM had to be installed, preferably the `Sun` version, but if not suitable, then an alternative was used (`IBM, Apple, Sgi`). To avoid requesting privileged permits on the site, the JVM installation took place on the home directory of the site account.

2.2.2.2 Schedulers and Site Access

We can classify the access into two categories depending on the scheduler/middleware installed on the site: remote or local access. Remote access is used with deployment protocols such as `Globus, Unicore, NorduGrid, GLite` where the job submission takes place directly from a client machine, usually with a certificate scheme provided by the protocol.

On the other hand local access protocols like: `LSF, PBS, OAR, PRUN` are used locally at the site, and therefore an `SSH` (or equivalent) connection must be combined with the job submission protocol. With ProActive, this can be easily done using the Deployment Descriptor. Nevertheless, to avoid password prompts an ssh passphrased key was installed on the remote sites to allow non interactive access using an ssh agent.

2.2.2.3 Network and Firewall Policies

The network can be classified into different levels of security policies.

Friendly Sites allowed all incoming/outgoing connections from/to machines on the ETSI Plugtests network.

Semi-Friendly Sites allowed only incoming ssh communications and all outgoing connections.

Restrictive Sites had an accessible public IP address frontend machine, and the internal nodes were either unreachable (firewalled) or unaccessible (private IPs with NAT) from the outside. The frontend can communicate with the inner nodes.

Island Like Restrictive, but outgoing connections are not allowed from the frontend or the inner nodes.

Friendly sites were the most easy configuration. *Semi-Friendly* sites were handled using ProActiveProActive'ss `rmi-ssh` tunneling features. *Restrictive* sites, were handled using the recently developed feature of hierarchical deployment and communication (see Section 2.3.1). Unfortunately, *Island* sites could not form part of the Grid because of their limited outgoing connection capabilities. The only way to solve this issue was to request the site administrators to be change their policy to comply at least with the *Restrictive* configuration. Note that for all the other cases, the administrators do not need to perform any type of network configuration since ProActive can handle this cases.

2.2.2.4 Data Storage

The data storage scheme varied from site to site. On many of them, the Network File System (NFS) was used, thus sharing the home user directory overall nodes on the site. These cases were the most simple to configure, since the software installation (ProActive and JVM if necessary), only had to take place once. On the other hand sites which did not share the user home directory proved to be very troublesome, specially for configuring the synchronization scripts.

One difference from last year with respect with data storage, was that some new protocols like `NorduGrid` or `Unicore` provide the concept of *Job Space*. When a job is submitted using any of this protocols, a specific space for the job is created on the cluster. This job space is temporal, and can be used by the process to store data. Nevertheless when the process finishes, the *Job Space* is destroyed. Thus making a persistence installation difficult. To solve this issue, our approach was to use *Deployment File Transfer* (see Section 2.3.2).

2.3 New Features Development

Support for several new deployment protocols were developed. This was necessary to include new partners into the Grid. Also, several new features were added to ProActive to cope with specific site configurations like Hierarchical Deployment and File Transfer.

Among the new deployment protocols that were developed to interface with other middlewares or schedulers we can find: `OarGrid, NorduGrid, Unicore` and `GLite`.

2.3.1 Hierarchical Deployment

Hierarchical Deployment was a key feature developed for this years Grid Plugtests. Following from last years experience, many sites had configurations that used internal IP networks, or were located behind a very restrictive firewall. During the 1st Grid Plugtests, it was up to the user to provide a forwarding mechanism for accessing the internal Grid nodes. Since this proves to be very complicated at the user application level, and taking last year's Plugtests experience into account, this year the OASIS Team, mainly Clement Mathieu, Romain Quilici and Matthieu Morel, worked on providing transparent support at the ProActive level for inner site nodes. As a result, sites could be added to the Grid requiring less configuration effort by the site's administrators.

Nevertheless this feature is still in a development status, with many improvements and bug fixes pending. For example, during the Plugtests one of the teams realized that the Group feature can not be combined, at this point, with Hierarchical Deployment. Thus, the Plugtests experience provided important feedback for ProActive improvements.

2.3.2 Deployment File Transfer

Another interesting feature that was developed corresponds to Deployment File Transfer support. This allows the user to specify files that need to be transfered at deployment time to the Grid nodes. The main result of this approach is that ProActive can be transfered *on-the-fly* along with the JVM on sites which do not allow persistant software installation (a job space is created for each submitted job, and later destroyed when the job is finished). Sites that used this mechanism were `NorduGrid`, `Unicore` and `GLite`.

2.4 Sites

For the 2nd Grid Plugtests, more than 40 sites located on 13 different countries were configured by the OASIS Team. The complexity of configuring each site varied, as described in Section 2.2.2.

2.4.1 Sites Description

Here we present the list of sites that formed part of the Grid. To easy the readability, we have sorted this sites alphabetically first, and secondly by site name. For this same reason we have also grouped them into four Tables: 2.1, 2.3, 2.4, 2.4. The columns of each table are described as:

Country The name of the country that the site belongs to: Australia, Brazil, Chile, China, France, Germany, Greece, Ireland, Italy, Netherland, Norway, Switzerland, USA.

Site_Name The name of the site.

Nodes/Proc The number of nodes (machines) provided by the site, and also the number of processors per machine.

Ghz The clock speed of the machine's CPU.

O.S. Operating System, and when relevant, certain architecture information: Linux (x86, ai64, x86_64), Macintosh (Motorola), AIX, Solaris (Sparc), SGIIrix.

Sched The scheduling (job submission) mechanism used to deploy on the site's machines: `LSF`, `PBS`, `SGE`, `OAR`, `Torque`, `Globus`, `Unicore`, `NorduGrid`, `GLite`, `SSH`.

JVM The type of Java Virtual Machine: Sun, IBM, Macintosh. Note that the versions are not specified, but were also heterogeneous. For example, for Sun JVM's the 1.4.x and 1.5.x versions were used.

Mflops Represents a rough estimation of the site's computation capacity. Please note that this benchmarks correspond to a *rough* estimation with several approximations, and should therefore not be regarded as a scientific reference or certification. The main goal of providing this information, is to have a rough reference metric of the Grid, and not to make comparisons between sites. For information on how this estimation was computed, and why comparing this metric between sites is pointless see Section 2.5.

Country	Site Name	Nodes/Proc	Ghz	O.S.	Sched	JVM	Mflops
Australia	U. Melbourne	13/1	2.0	Linux	PBS	Sun	1658
Brazil	LNCC	8/2	3.2	Linux	SSH	Sun	2464
Chile	DCC U. Chile	15/2	1.6	Linux ia64	SSH	Sun	2917
Chile	CMM U. Chile	11/1	1.5	Linux	SSH	Sun	
Chile	UTFSM	30/1	2.0	Linux x86_64	SSH	Sun	5103
China	BUPT	11/1	3.2	Linux	SSH	Sun	3576
China	CNGrid HKU	32/1	2.4	Linux	PBS	Sun	3487
China	CNGrid ICT	6/2	1.8	Linux	PBS	Sun	1119
China	CNGrid SCCAS	25/4	1.3	Linux	LSF	Sun	4593
China	CNGrid SCCNET	100/4	2.4	Linux x86_64	LSF	IBM	90209
China	CNGrid USTC	16/2	1.5	Linux ia64	LSF	Sun	N/A
China	CNGrid XJTU	2/2	2.0	Linux ia64	PBS	Sun	N/A

Table 2.1: Grid Sites: Australia-China

Country	Site Name	Nodes/Proc	Ghz	O.S.	Sched	JVM	Mflops
France	G5K Bordeaux	48/2	2.2	Linux	OAR	Sun	N/A
France	G5K Grenoble icluster2	103/2	0.9	Linux ai64	OAR	Sun	10606
France	G5K Grenoble idpot	32/2	2.4	Linux	OAR	Sun	
France	G5K Lyon	126/2	2/2.4	Linux x86_64	OAR	Sun	23703
France	G5K Orsay	216/2	2.0	Linux	OAR	Sun	93524
France	G5K Rennes Paraci	64/2	2.4	Linux	OAR	Sun	26091
France	G5K Rennes Parasol	64/2	2.2	Linux x86_64	OAR	Sun	30618
France	G5K Rennes Tartopom	32/2	2.0	MacOS PPC	OAR	Mac	9739
France	G5K Sophia	105/2	2.0	Linux x86_64	OAR	Sun	47939
France	G5K Toulouse	32/2	2.2	Linux	OAR	Sun	14571

Table 2.2: Grid Sites: France - France G5K

Country	Site Name	Nodes/Proc	Ghz	O.S.	Sched	JVM	Mflops
France	IDRIS-Deisa	2/32	1.3	AIX	SSH	IBM	7970
France	INRIA Cluster	32/2	2|0.9	Linux	LSF	Sun	10606
France	INRIA Nef	32/2	2	Linux x86_64	Torque	Sun	
France	LIFL	53/2	2.4	Linux x86_64	SSH	Sun	18374
France	LORIA	2/30	0.7	SGIIrix	SSH	SGI	4655
France	LORIA	8/2	1.4	Linux	SSH	Sun	
France	Supelec	33/2	3.0	Linux	SSH	Sun	6693
Germany	Unicore	4/1	2.6	Linux|AIX	Unicore	IBM	446
Greece	FORTH ICS	16/2	1.6	Linux x86_64	SSH	Sun	4125
Ireland	Queen's U. Belfast	14/1	1.5	Linux x86_64	SSH	Sun	2147

Table 2.3: Grid Sites: France IDRIS - Ireland

Country	Site Name	Nodes/Proc	Ghz	O.S.	Sched	JVM	Mflops
Italy	Benevento A	7/1	1.6	Linux	SGE	Sun	3465
Italy	Benevento B	18/1	2.8	Linux	SGE	Sun	
Italy	ISTI	4/1	2.2	Linux	SSH	Sun	457
Italy	Pisa University	33/1	0.7	Linux	SSH	Sun	2385
Netherland	Vriej University	20/2	1.0	Linux	PRUN	Sun	1346
Norway	NorduGrid	22/1	0.45	Linux	NorduGrid	IBM	2328
Switzerland	CERN/GILDA	N/A	N/A	N/A	GLite	N/A	N/A
Switzerland	EIF	25/1	0.5	Solaris9	SSH	Sun	508
Switzerland	ETHZ	21/1	3|2.4|1.2	Linux	SSH	Sun	3410
USA	UC Irvine	14/1	1.6	Linux	LSF	Sun	1213
USA	USC - CGT	8/2	2.4	Linux	Globus4	Sun	1966

Table 2.4: Grid Sites: Italy - USA

2.4.2 Sites Map

Figure 2.1 shows a graphical representation of the Grid. The location of sites are pointed out with flags in the map. This maps shows how we reached a worldwide dissemination, with sites in Asia, Australia, Europe, North and South America. The details for each site can be found in Tables: 2.1, 2.3, 2.4, 2.4.

Figure 2.1: Grid Sites Map

2.5 Grid Benchmarks

To benchmark the Grid we used the SciMark agent for computing. This measure was taken using a pure Java benchmark. Since the types of JVMs used were heterogeneous in vendor and version, comparing *Mflops* between sites is pointless. More over, given the instability of a Grid of this nature (in size and location), for some sites we were unable to obtain all the provided resources at the moment of benchmark. In this cases, we extrapolated to estimate the total site capacity. Because all of this reasons, the specified Grid benchmark is a very rough one, and should be considered as a reference, and not a certification or a rigorous scientific study.

Considering the 1st Grid Plugtests benchmark (`100GFlops`), this years corresponds to a significant improvement: `450GFlops` (approx). The details of this computation can be found in Tables: 2.1, 2.3, 2.4, 2.4.

2.5.1 Benchmark Grid Graphs

Figure 2.2 shows the distribution of number of CPUs per Grid site. Figure 2.3 shows the Distribution of Grid Processing capacity (in Mflops). The graph in Figure 2.4 holds both results in a pie like graph.

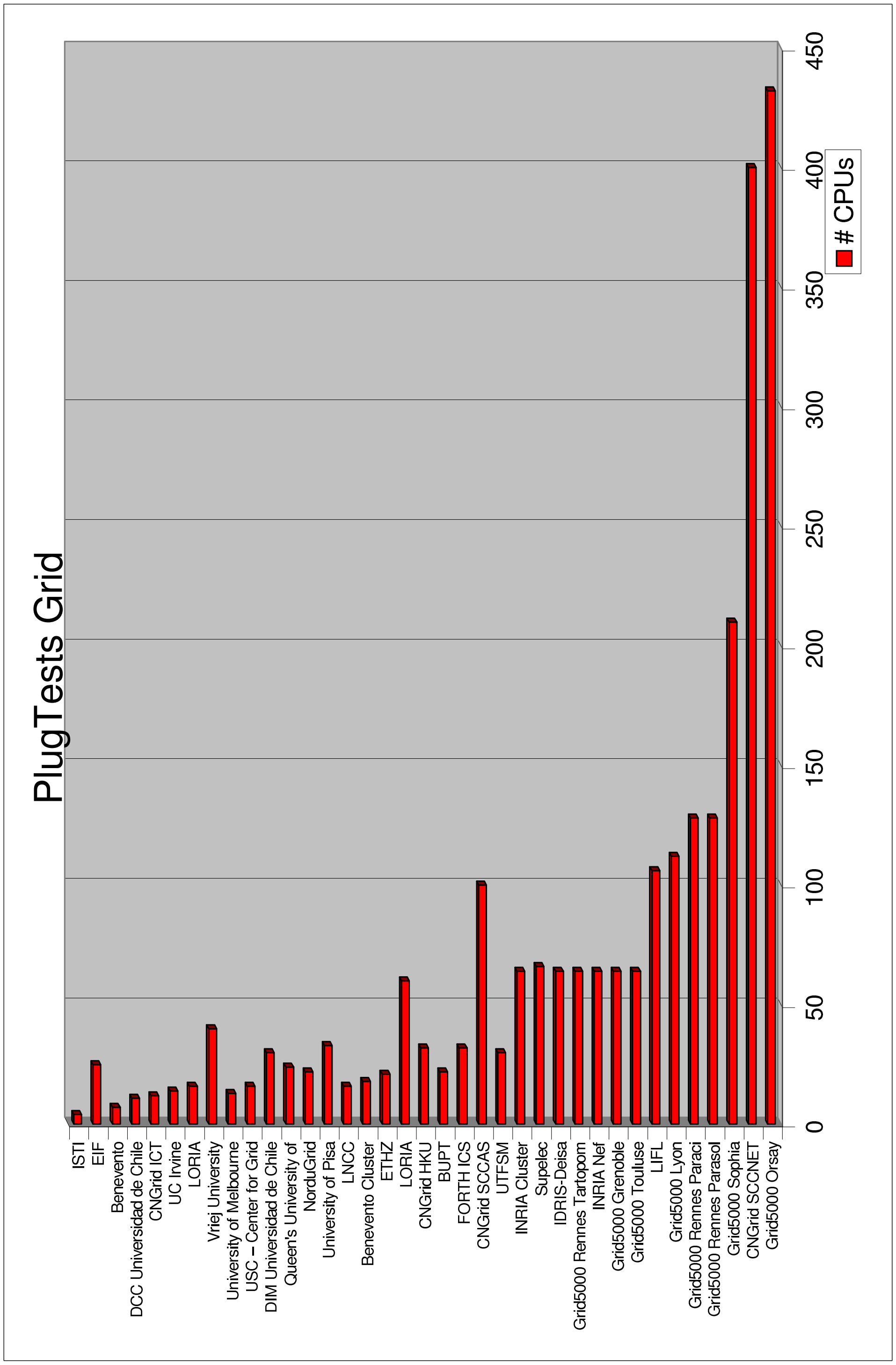

Figure 2.2: Number of CPUs

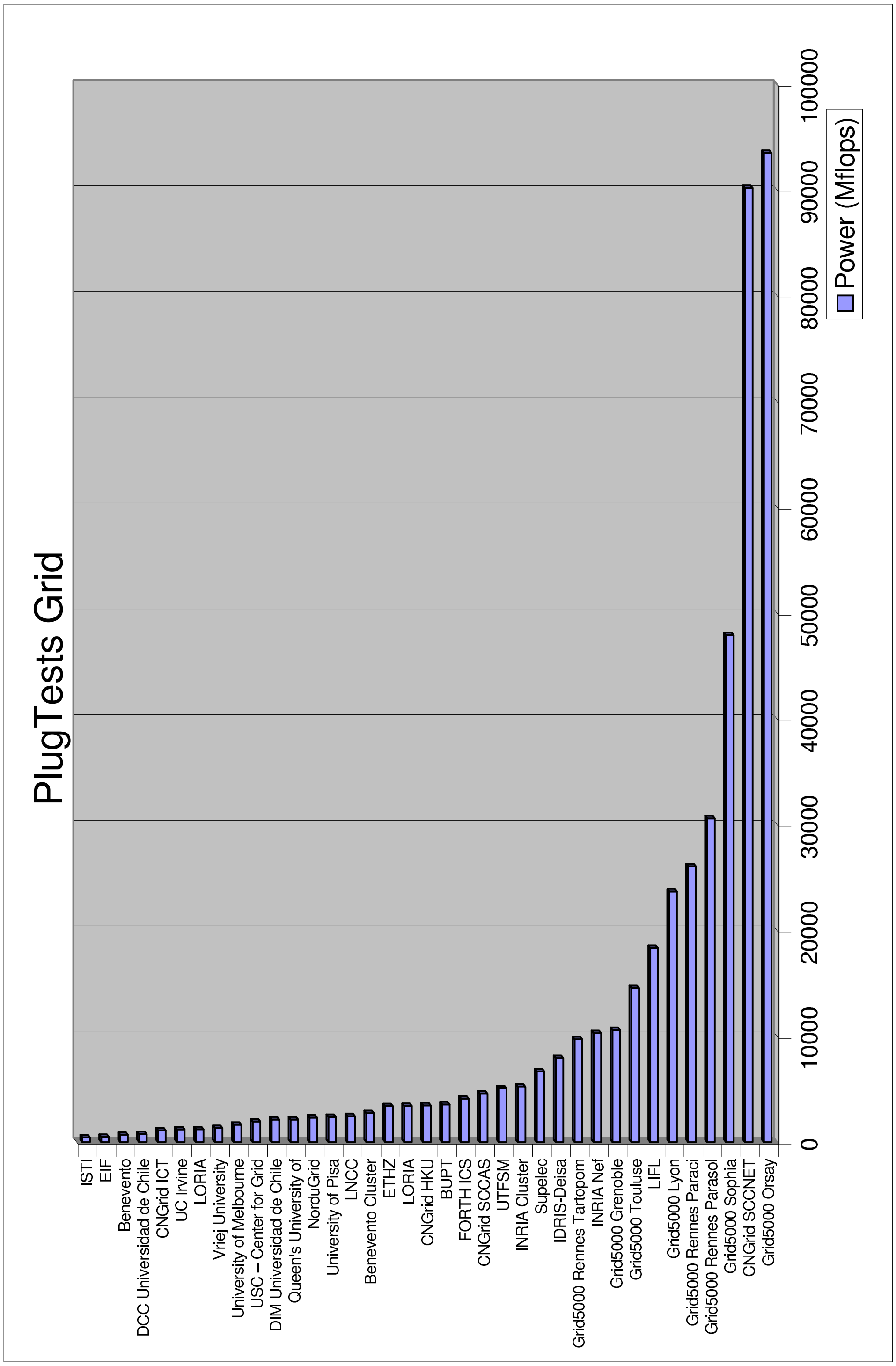

Figure 2.3: Processing Capacity

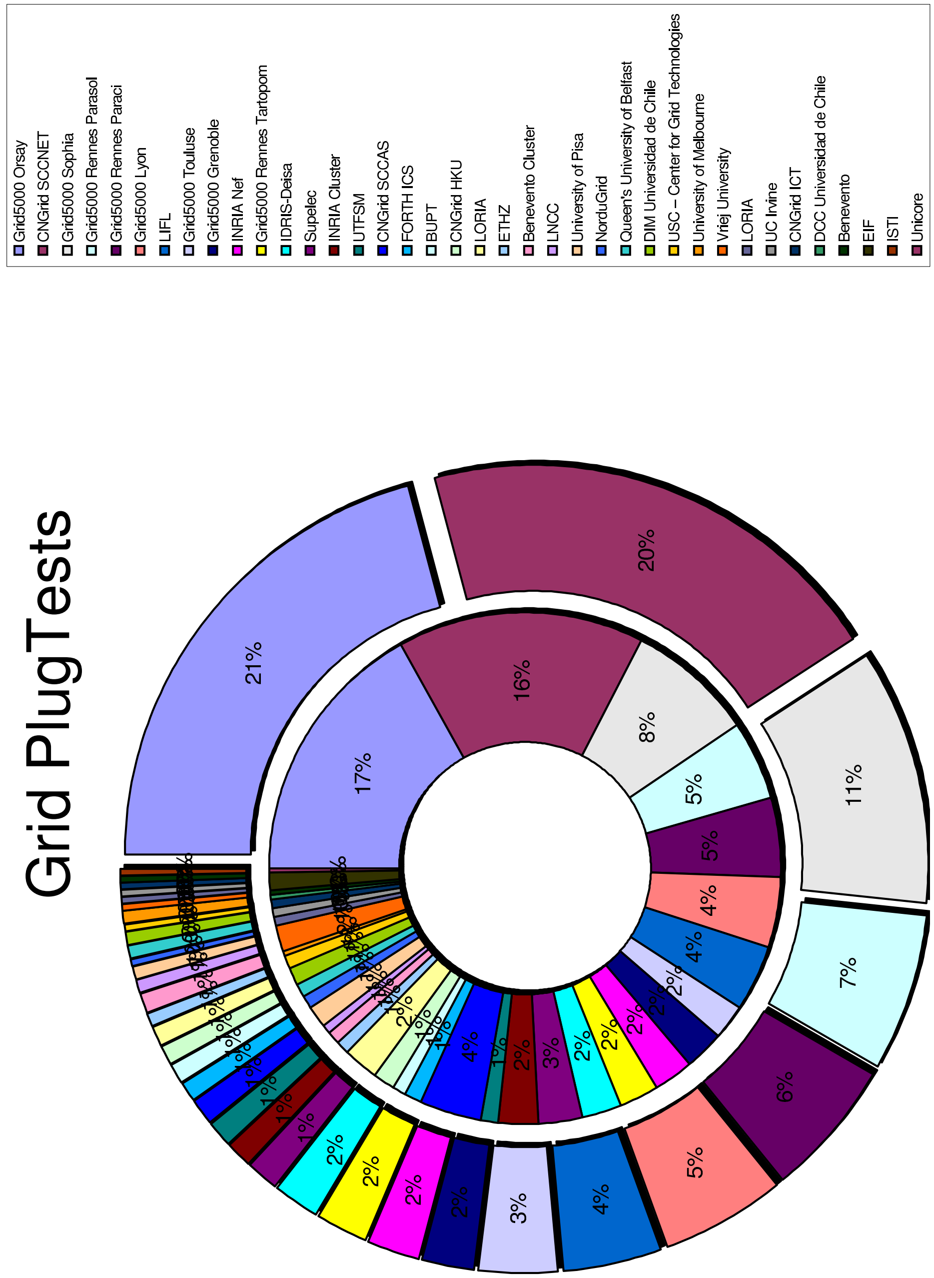

Figure 2.4: Processing Capacity and Number of Processors

2.6 Difficulties and Problems

As last year, many difficulties were encountered when setting up the Grid.

For example, when dealing with Grid5000 we faced some problems with the `oardel` command on some sites. The command did not execute correctly a job deletion, and we had to write a script to do this. Other problems we faced were `oarsub` not working correctly with parameter "all" on the Grid500 Sophia site, and other small details. Nevertheless, the monitoring webpage provided by Grid5000 proved to be very useful to diagnose and solve the problems.

Also on Grid5000, we developed the support for the `oargrid` submission protocol, which was finally not used (we had to fall back on the `oar` protocol), because the `oargridsub` command provided a very rigid behaviour: exactly all the requested nodes were returned for all sites, or no nodes were returned at all. When dealing with requests for hundreds of nodes, it is very likely that some might fail. For us, it would have been much more useful if the `oargridsub` command provided more flexibility by allowing the specification of "at most" or "at least".

On other Grids we also faced some problems. For CNGrid we had to implement a custom `LSF` protocol for one of the sites. Also, the provided resource for CNGrid were very busy (not exclusive access for the Plugtests), and most of the time we were unsuccessful at submitting a large job reservation.

For Unicore we developed our own interface using the testing server. Unfortunately, we were unable to test our interface when dealing with big sites, since Unicore only provided virtual sites with one processor.

With the GLite interface we faced some problems when trying to deploy from a Fedora Core 3 (FC3) machine. We discovered at this point that the GLite client library is not supported for FC3 and newer. We managed to solve this problem by using ProActive's remote job submission features. To achieve this, we deployed the job submission first into a GLite client compatible machine using ssh, and from there submitted the job to the GLite machines. The transfering of the GLite JDL file was handled using ProActive's File Transfer mechanism.

Even though hierarchical deployment was a key feature for this Plugtests, it still lacks maturity and development for more complex scenarios. We would like to continue development this feature, since we believe is fundamental for building the Grid.

Finally, we had some problems with the Unix/Linux open file (connection) limitation. For a Grid of this size the default limitation on current distributions is 1024 . This is too small when we take into account that this years Grid involved over 2700 processors. ProActive provides a mean to reduce the number of open connection by specifying that this should be closed in the deployment descriptor files. None the less, this optimization was not enough for a Grid of this size. We therefore incremented the default value to (16K) for the contests machines. Nevertheless, we only realized during the Plugtests that hierarchical deployment provided an even harder stress on the open file limits. For example, we had to contact the administrator to increment this limitation for the Grid500 Sophia site.

Overall these difficulties proved to be a valuable part of the Grid interoperability, and will help us to continue improving and developing the Grid.

Chapter 3

The Contests

This year, two contest were organized during the 2nd Grid Plugtests. Like the last year, the *N-Queens Counting Problem* was present: How many ways can N queens be placed on a NxN chess-board. Also, a new problem was added this year, the *Flowshop Problem*: What is the optimum way of using *M* machines for *J* jobs were a job *j* in a machine *m* takes P_{jm} time.

These events were strictly an engineering event, not a conference, nor a workshop. As such, an active participation was requested from the companies/organizations which had to write their own implementation of the problem. There was no compulsory programming language, all teams used Java, and when possible, some used native code inside a Java wrapper.

3.1 N-Queens Counting Problem

Four teams competed this year in the N-Queens contests. The criterion for deciding the winners were based on:

- Greatest number of solutions found.
- Biggest number of processors used.
- Fastest algorithm.

Each team was allocated one hour of exclusive access to the Grid, for computing the N-Queens challenges.

3.1.1 Teams

3.1.1.1 AlgoBAR (France)

Organization INRIA, CNMA (University of Nice) + NTU and MCU.

Members Philippe HENRI, Yuh-Pyng (Arping) Shieh, Philippe SIGAUD and Sylvain Bellino.

3.1.1.2 BEATRIX (Netherlands)

Organization Vrije Universiteit

Members Thilo Kielmann, Ana Maria Oprescu, Andrei Agapi Rob van Nieuwpoort

3.1.1.3 DCC Universidad de Chile (remote participation)

Organization Departamento de Ciencias de la Computacion (DCC), Universidad de Chile.

Members Jaime Hernandez, Alfredo Leyton, Nicolas Dujovne, Luis Mateu (coach).

Remote Interlocutor Florian Martin (INRIA OASIS).

3.1.1.4 LSC/UFSM (Brazil)

Organization Laboratorio de Sistemas de Computacao (LSC), Universidade Federal de Santa Maria (UFSM)

Members Elton Nicoletti Mathias (coordinator), Tiago Scheid, Benhur Stein.

3.1.1.5 BUPT

Organization Beijing University of Posts and Telecommunications (BUPT)

Members Han Yunan, Gong Zheng, Xu Ming, Huang Xiaohong, Zhang Bin, Wu Yongjuan, Su Yujie, Wang Zhenhua, Pu Mingsong, Liu Wen, Cui Yinhua.

3.2 Flowshop Problem

The FlowShop contest was sponsored by ROADREF providing a prize of 500 Euros. Each team had 1 hour to run their application on the Grid. During this period, they were expected to solve Taillard's instances of the FlowShop problem[1]. The instances were required to be solved exactly with proof of optimality. This means that the program must find the exact solution, and prove that it is the optimal. If more than one team solved the problem correctly, the winner was the one that solved the problem in less elapsed time. If more than one team solved the same problem in the same amount of time, the final criteria for deciding the winner was the number of workers (number of CPUs) used.

3.2.1 Teams

3.2.1.1 PutAT3AM

Organization Poznan University of Technology (PUT)

Members Filip Gorski, Pawel Marciniak, Maciej Plaza, Stanislaw Stempin.

3.2.1.2 outPUT

Organization Poznan University of Technology (PUT)

Members Mariusz Mamonski, Szymon Wasik, Pawel Lichocki

3.2.1.3 LIFL1

Organization LIFL - OPAC

Members Tantar Alexandru-Adrian

3.2.1.4 LIFL 2

Organization University of Bejaia (in Algeria)

Members Ahcene Bendjoudi, represented by Tantar Alexandru-Adrian.

3.2.1.5 INRIA

Organization INRIA Project OASIS

Members Cedric Dalmasso, Clement Mathieu, Didier Dalmasso, Alexandre di Costanzo.

3.3 Local ETSI Contest Machines Configuration

For the contests and tutorial, 25 machines were installed and configured by the OASIS Team. The main software installed on the machines were: Fedora Core 3, Sun JDK1.4.9, Eclipse, ProActive and other contest environment configuration. One of the machines was configured as a central server for the user accounts using NFS. In order of arrival to the ETSI Plugtests room, each team was assigned an account on the machines, from `team1` to `teamN`. Contestants spent the first day (and part of the second) testing and fine tuning their code for the Grid.

3.4 Online Remote Contest Participation: Santiago de Chile

Florian Martin, from the OASIS Team, worked on preparing the remote contest participation at Santiago. Using the time zone difference, the remote contest took place mainly during the night, which corresponded to the afternoon in Santiago, allocating exclusive access to the Grid during this period.

Basically Florian Martin's job was to contact Grid actors in south America, negotiate access and configure them into the Grid. He also had to organize the Plugtest in Santiago, to allow the local teams to participate to the event. For this, a special room was reserved for the event. Each participant used an individual local computer, and each machine was connected to one of the ETSI contest machines, thus allowing them access to the Grid.

Chapter 4

Results

4.1 N-Queens Contests Results

These results are taken from the ETSI *2nd Grid Plugtests N-Queens Challenge Results* report[2]. The contests results are as follows:

- The first place was awarded to LSC/UFSM (Brazil) for computing the maximum number of solutions.
- The second place was awarded to BEATRIX (Vrije Universiteit) for using the maximum number of nodes.
- The third place was awarded to DCC Universidad de Chile for being the most efficient.

Note that:

- No team could be awarded more than one place.
- Team LSC/UFSM managed to compute almost 3 more times the number of solutions than the winners of the 1st Grid Plugtests (~800 vs ~2 202 billions), deployed on almost twice the number of nodes than last years maximum (560 vs 1106), and found the first N=21 challenge in almost half the time (24 vs 13 minutes).
- Some teams managed to combine ProActive with other Grid tools. For example, BEATRIX team used ProActive for deployment, and Ibis[11] for communication.

Challenge	Elapsed	Nodes	Solutions
21	>1h	237	N/A

Table 4.1: N-Queens Results. Team: Algobar

Challenge	Elapsed	Nodes	Solutions
N=20 x 1	N/A	200	39 029 188 884
N=21 x 3	14mn 49s (N=21x1)	200	943 998 668 136
			983 027 857 020

Table 4.2: N-Queens Results. Team: BEATRIX

Challenge	Elapsed	Nodes	Solutions
N=18 x 5	64s	114	3 330 453 120
N=19 x 4	140s	114	19 872 231 392
N=20 x 2	650s	114	780 583 777 768
			101 261 062 280

Table 4.3: N-Queens Results. Team: DCC Universidad de Chile

Challenge	Elapsed	Nodes	Solutions
N=21 x 7	13 min	1106	2 202 663 558 984

Table 4.4: N-Queens Results. Team: LSC/UFSM

4.2 FlowShop Contests Results

These results are taken from the ETSI *2nd Grid Plugtests Flowshop Challenge Results* report[3]. The results are detailed as follows:

- The first place was awarded to Team PUTaT3AM. They computed all exact cases for Flow-Shop case #21 to #30 (20x20). Only this team was able to do this in less than one hour, and using 370 Nodes.
- The second place was awarded to Team outPUT. They managed to find the optimal solution for the FlowShop case #28.

Challenge	Elapsed	Nodes	Solutions
<>	60 min	568	N/A

Table 4.5: FlowShop Results. Team: LIFL 1

Challenge	Elapsed	Nodes	Solutions
#28	33 min	1007	OK

Table 4.6: FlowShop Results Team: outPUT

Challenge	Elapsed	Nodes	Solutions
#21	>1h	1596	Not OK

Table 4.7: FlowShop Results. Team: INRIA

Challenge	Elapsed [s]	Nodes	Solutions
#21	435	370	OK
#28	148	370	OK
#29	187	370	OK
#30	139	361	OK
#22	219	361	OK
#24	234	361	OK
#25	351	361	OK
#26	515	361	OK
#27	607	352	OK
#23	1746	352	OK

Table 4.8: FlowShop Results Team: PUTaT3AM

Challenge	Elapsed	Nodes	Solutions
#21	>1h	360	Not OK

Table 4.9: FlowShop Results Team: LIFL2

4.3 Offline Remote N-Queens Challenge

After the Plugtests the N-Queens Challenge was extended for one month. This gave an opportunity for the motivated teams to continue testing Grid operability. The remote challenge results are shown in Figures 4.12, 4.12, and.

4.12.

Challenge	Elapsed	Nodes	Solutions
21x4	5min 14s	1086	1 258 664 890 848
22x3	32min 43s	1086	8 073 026 104 932
			9 331 690 995 780

Table 4.10: Offline Remote N-Queens Results. Team: Algobar

Challenge	Elapsed	Nodes	Solutions
22	25 min	961	2 691 008 701 644
22	26 min	841	2 691 008 701 644
22	28 min	761	2 691 008 701 644
			5 382 017 403 288

Table 4.11: Offline Remote N-Queens Results. Team: BEATRIX

Challenge	Elapsed	Nodes	Solutions
16	3s	28	14 772 512
17	6s	28	95 815 014
18	11s	28	666 090 624
19	1min 3s	28	4 968 057 848
20	7min 59s	28	39 029 188 884
21	1hr 7min 7s	28	314 666 222 712
22	9h 45min 55s	28	2 691 008 701 644
			3 050 448 849 238

Table 4.12: Offline Remote N-Queens Results. Team: BUPT

Using the same criteria as for the Plugtests N-Queens Challenge, the results were as follows:

- The first place was awarded to Algobar for computing the maximum number of solutions.
- The second place was awarded to BEATRIX (Vrije Universiteit) for using the maximum number of nodes.

- The third place was awarded to BUPT for being the most efficient.

The full report of the Offline Remote N-Queens Challenge can be found at[12].

4.4 1st and 2nd Grid Plugtests Comparisons

Table 4.13 shows a comparison chart of the 1st and 2nd Grid Plugtests. This differences have been mentioned through this report, but are summarized here. From the table, it is easy to see that the 2nd Grid Plugtests embraced an even wider range of the Grid community.

	2004	2005
Plugtests number of participants	80	240
Plugtests number of events	3	13
Grid: number of involved countries	12	13
Grid: number of sites	20	40
Grid: number of CPUs	800	2700
Grid: GFlops	~100	~450
Hierarchical Deployment support	No	Yes
File Transfer support	No	Yes
Number of contests	1	2
Number of teams	6	8
Contestant max CPU used for successful computation	560	1106
Contestant max CPU deployed	800	2111
Contestant Max N-Queens # solutions found	~800 billion	~2 202 Billion

Table 4.13: 1st and 2nd Grid Plugtests Comparison Summary Chart

4.5 Grid Heterogeneousness and Interoperability

The Grid gathered for the 2nd Grid Plugtests proved to be heterogeneous in many levels: Computer Arquitecture, Operating Systems, Java Virual Machines, Deployment Protocols and Network Configurations. The diversity of resources is detailed as follows:

- *Computer Architectures:* x86, ia64, x86_64, PPC, AIX, SGIIrix, Sparc.
- *Operating Systems*: Linux, MacOS, AIX, SGIIrix, Solaris.
- *Java Virtual Machines*: Sun, IBM, Apple, AIX.
- *Deployment Protocols*: `GLite`, `Globus`, `LSF`, `NorduGrid (ARC)`, `OAR`, `PBS`, `PRUN`, `SGE`, `SSH`, `Unicore`.

- *Network Configurations*: Friendly, Semi-Friendly, Restrictive (see Section 2.2.2.3).

The technical challenge was to virtually merge all the heterogeneous gathered computing resources into a single world-scale computing Grid. Using the ProActive middleware, the interoperability was thus achieved and tested by successfully deploying on the Grid the N-Queens and Flowshop contestant's applications.

Chapter 5

Conclusions

The 2nd Grid Plugtests, co-organized by INRIA and ETSI, pleased all the participants. It was an event useful for the Grid community: users and developers. The Conferences and Workshops helped the community to exchange their views, objectives, difficulties and user experience for developing the Grid. Also, with the Tutorials, the gap between the users and the Grid was narrowed by presenting the different Grid tools and middlewares.

In the specific case of ProActive, the Plugtests gave us the opportunity to develop new and interesting features, while testing the middleware at a new level of complexity. The results shown during the N-Queens and Flowshop contests left us very happy, since they showed that the applications could take advantage of the heterogeneous Grid in a simple way.

As usual, setting up the Grid proved to be a lot of hard work with problems and difficulties. The OASIS Team had to implement new deployment protocols, and new ways to adapt to network configurations. This new tools were an important advancement from last year, since they enabled more restrictive sites to join the Grid with less effort from the sites administrators. Nevertheless, after the Plugtests experience we believe these tools still require further development before they can become an integral feature of ProActive. The Plug & Play Grid is still not a reality, but after the Plugtests we can happily say that it lies one step closer.

Given the positive experience of the event, we would like to organize a 3rd version. In this occasion, we would like to encourage a wider usage palette of tools for accessing and programing the Grid. We would also like to have a wider community involvement, including new organizations, for example, GGF and EGA.

Bibliography

[1] Talbi Flowshop Problem Challenges. Talbi http://www.ifl.fr/~talbi/challenge

[2] ETSI 2nd Grid Plugtests N-Queens Challenge Results. http://www.etsi.org/plugtests/History/DOC/N-QueensChallengeRealTimeJURYRecords2005final.pdf

[3] ETSI 2nd Grid Plugtests Flowshop Challenge Results. http://www.etsi.org/plugtests/History/DOC/FlowShopChallengeRealTimeJURYRecords2005final.pdf

[4] Involved Sites Technical Contacts. http://www-sop.inria.fr/oasis/plugtest2005/technical_List.html

[5] How To Prepare for the 2nd Grid Plugtests Contest Guide. http://www-sop.inria.fr/oasis/plugtest2005/HowTo_Prepare_Contest.html

[6] Grid Architecture Summary Chart. http://www-sop.inria.fr/oasis/plugtest2005/Machines_Clusters.html

[7] Sites Providers CPU Ranking. http://www-sop.inria.fr/oasis/plugtest2005/Providers.html

[8] ETSI 2nd Grid Plugtests webpage. http://www.etsi.org/plugtests/History/2005GRID.htm

[9] ETSI 1st Grid Plugtests webpage. http://www.etsi.org/plugtests/History/2004GRID.htm

[10] 1st Grid Plugtests Report. http://www.etsi.org/plugtests/History/DOC/1stGRIDplugtest_report.pdf

[11] Ibis Grid Software. http://www.cs.vu.nl/ibis/

[12] ETSI 2nd Grid Plugtests Offline Remote N-Queens Challenge Results http://www-sop.inria.fr/oasis/plugtest2005/RemoteRecordsN-QueensFinalReport.pdf

Appendix A

Plugtests Agenda

Mon 10 Oct.	**2nd ProActive User Group** Amphi ATHENA – ETSI Main Building **EGEE: Introducing activities and their successes** Amphi IRIS – ETSI Main Building
	7.00 pm: **ProActive User Group Reception** ETSI Hall – Main Building

Tue 11 Oct.	**ProActive Tutorial Hands-On Grid Programming** Amphi ATHENA – ETSI Main Building **ProActive Tutorial with PCs** Mozart Room – ETSI Einstein Building **Unicore Summit** Amphi IRIS – ETSI Main Building **Unicore Summit with Pcs** IRIS 4 Room – ETSI Main Building
	2nd Grid Plugtests: **N-Queens Contest and FlowShop Contest** Interop Room - ETSI Einstein Build

Wed 12 Oct.	**CoreGRID Workshop: "Grid Systems, tools and environments"** IRIS 2&3 Room – ETSI Main Build. **Unicore Summit** Amphi IRIS – ETSI Main Building **Unicore Summit with Pcs** IRIS 4 Room – ETSI Main Building **Industrial session** Hermes Room – ETSI Main Building **NextGrid Plenary Session** (for private members only) Amphi Athena – ETSI Main Building **NextGrid Breakout Session** Debussy Room - ETSI Einstein Building
	2nd Grid Plugtests: **N-Queens Contest and FlowShop Contest** Interop Room - ETSI Einstein Build

Thu 13 Oct.	**GridCoord Workshop:"The use of open middleware for the Grids"** Amphi Athena - ETSI Main Build **GRID Application Toolkit (GAT) Tutorial** (morning) IRIS 2&3 Room – ETSI Main Building **Ibis Tutorial (afternoon)** IRIS 2&3 Room – ETSI Main Building **NextGrid Plenary Session (for private members only)** Amphi Iris – ETSI Main Building **NextGrid Breakout Session** Debussy Room - ETSI Einstein Building
	7.00 pm: **Grids@Work Banquet**

Fri 14 Oct.	CoreGRID Workshop: **"Programming models and components for the Grid"** Amphi Athena – ETSI Main Building **P-GRADE Tutorial (morning)** IRIS 2&3 Room – ETSI Main Building **P-GRADE Tutorial Hand on (afternoon)** IRIS 2&3 Room – ETSI Main Building **NorduGrid : ARC Tutorial and discussions** IRIS 4 Room – ETSI Main Building **NextGrid Plenary Session (for private members only)** Amphi Iris – ETSI Main Building **NextGrid Breakout Session** Debussy Room - ETSI Einstein Building

DAY 1 : MONDAY OCTOBER 10TH, 2005

Second Proactive User Group

8.30-9:15	**Registration**
9:15-9:30	**Welcome address** ***speaker TBC***
9.30-11:00	**Session 1 : Basic Programming features**
	ProActive Overview : Asynchronous Method Calls, Futures, Groups, Openness ***Denis Caromel*** UNSA
	Components and Legacy Code ***Matthieu Morel*** INRIA
	Deployment and Firewall ***Romain Quilici*** UNSA
11:30-13:00	**Session 2 : "Recent and advanced features"**
	Exceptions Management, Fault Tolerance and Checkpointing ***Christian Delbé*** INRIA
	Automatic Load Balancing of ProActive Applications ***José Piquer*** Universidad de Chile, Santiago
	Peer-To-Peer and Branch-and-Bound ***Alexandre di Costanzo*** INRIA
13:00-14:00	**Lunch break on site**
14:00-18:00	**Session 3 : User presentations**
	Grid Computing with an Extension of ProActive Groups ***Eugenio Zimeo, Nadia Ranaldo*** RCOST University of Sannio, Benevento, Italy
	Improving Peer-to-Peer Resource Usage Through Idle Cycle Prediction ***Elton Mathias, Marcelo Pasin*** Univ. Federal de Santa Maria, Departamento de Eletrônica e Computação, Brazil
	Reflections on Using ProActive for Skeleton Programming ***Marco Danelutto, Marco Aldinucci*** University of Pisa, Italy
	Distributed Financial Computations with ProActive ***Stephane Vialle*** Supelec, Metz, France
15:45-16:15	**Coffee break**

	Using ProActive for Dynamic Discovery of Components and Services ***Diego Puppin, Domenico Laforenza*** ISTI-CNR, Italy
	Grid Computing and Satellite Communications with ProActive ***Vincent Cavé, Jean-Christophe Honnorat*** Alcatel Space
	Functional and Dataflow Parallelism with ProActive: the JavaF project ***Paul N. Martinaitis, Andrew L. Wendelborn*** University of Adelaide, Australia
	ProActive Dynamic Security; ProActive and the EU Provenance Project ***Arnaud Contes*** Cardiff University, UK
18:00-19:00	**Session 4: "Perspectives"**
	Perspectives, Discussions, Open Mic. Q&A: - Teaching the Grid with ProActive - On-going work File Transfer ***Mario Leyton*** OSGi ***Virginie Legrand*** Understanding and Optimizing Communication Patterns ***Fabrice Huet*** NAS Benchmarks ***Christian Delbé*** Componentization of an Electromagnetic Parallel Code ***Nikos Parlavantzas*** Code Coupling of MPI Legacy ***Stéphane Mariani - EADS*** - Release plans - User Needs and Priorities

EGEE "Introducing activities and their successes"

09:00	**Managing EGEE and dissemination** (1h00')	**David Fergusson** *(NeSC)*
10:00	**Operating the World's largest grid infrastructure** (1h00')	**Roberto Barbera** *(INFN, University of Catania)*
11:00	**Developing the next generation of middleware** (1h00')	**Roberto Barbera** *(INFN, University of Catania)*
12:00	LUNCH	
14:00	**Securing the production grid** (1h00')	**Roberto Barbera** *(INFN, University of Catania)*
15:00	**Bringing Applications to the grid** (1h00')	**Johan Montagnat**
16:00	Coffee	
16:30	**Training and induction to EGEE** (1h00')	**David Fergusson** *(NeSC, Edinbrugh)*
17:30	**EGEE relating to other grids and policy definition** (1h00')	**Roberto Barbera** *(INFN, University of Catania)*

DAY 2 : TUESDAY OCTOBER 11TH, 2005

Proactive Tutorial and Hands-On Programming

09:00	**Welcome**	
9.15-12:00	Tutorial Basic ProActive functionalities through an example: computing decimals of PI.	
12:00-13:00	Install party (ProActive, Java, Cygwin/ssh)	
13:00-14:00	**Lunch break on site**	
14:00-18:00	Tutorial tracks proposed :	
	Groups/OOSPMD	Peer-to-Peer / Branch and Bound
	Fault-tolerance	Web Services
	Migration	Components
	Any questions on ProActive are welcome.	
18:00	**End of tutorial**	

Unicore Summit

Agenda Tuesday October 11th, 2005:

General Track	
9:30 - 9:45	Opening ***Geerd-Rüdiger Hoffmann*** *Director of the UNICORE Forum Board, Deutscher Wetterdienst, Germany*
9:45 - 10:45	The Status of Grid Computing in Research and Industry ***Wolfgang Gentzsch*** *D-Grid, Germany*
10:45 - 11:15	**Coffee Break**
11:15 - 12:15	NaReGI and UNICORE ***Satoshi Matsouko*** *Tokyo Institute for Technology, Japan*
12:15 - 13:00	The UNICORE Grid Software ***Achim Streit*** *Research Center Juelich, Germany*
13:00 - 14:00	**Lunch Break**
User Track	
14:00 - 14:30	OpenMolGRID - A UNICORE-based System for Molecular Science and Engineering ***Uko Maran*** *Tartu University, Estonia*
14:30 - 15:00	Computational Steering with VISIT in UNICORE ***Thomas Eickermann*** *Research Centre Juelich, Germany*
15:00 - 15:30	Service-Grids based on UNICORE ***Alfred Geiger*** *T-Systems Solutions for Research, Germany*
15:30 - 16:00	**Coffee Break**
16:00 - 16:30	Lightweight Client Access to UNICORE ***Roger Menday*** *Research Center Juelich, Germany*
16:30 - 17:00	UNICORE as a frontend for the IBM Infrastructure Solution ***Thomas Rueter*** *IBM, Germany*
UNICORE Forum	
17:00 - 19:00	UNICORE Forum Meeting *Restricted to members of the UNICORE Forum*

DAY 3 : WEDNESDAY OCTOBER 12TH, 2005

Unicore Summit

Agenda Wednesday October 12th, 2005:

Developer Track	
9:30 - 10:15	Future Developments in UniGrids and NextGRID ***David Snelling*** *Fujitsu Laboratories of Europe, United Kingdom*
10:15 - 11:00	Brokering in UNICORE ***John M. Brooke*** *University of Manchester, United Kingdom*
11:00 - 11:30	**Coffee Break**
11:30 - 12:00	UniGrids and GPE - A Client Framework for Interoperability ***Thomas Kentemich*** *Intel Software & Solutions Group, Germany*
12:00 - 12:30	Plugin-Development - Extensibility of UNICORE ***Piotr Bala*** *ICM - Warsaw University, Poland*
12:30 - 13:00	Integration of GridFTP in UNICORE ***Simone Lanzarini*** *CINECA Bologna, Italy*
13:00 - 14:00	**Lunch Break**
Administrator Track	
14:00 - 14:45	UNICORE and the fastest Supercomputer in Europe ***Rosa M. Badia*** *Barcelona Supercomputing Center, Spain*
14:45 - 15:30	DEISA - Easy Integration of HPC-Systems with UNICORE ***Andrea Righi*** *CINECA Bologna, Italy*
15:30 - 16:00	**Coffee Break**
16:00 - 16:30	UNICORE in production is easy to do ***Gert Ohme*** *T-Systems Solutions for Research, Germany*
16:30 - 17:00	GPFS and LL-MC as a basis for Grids with UNICORE ***Jean-Yves Girard*** *IBM, France*

CoreGrid Workshop : Grid Systems, Tools and Environments

09:00-09:15	**Welcome Message, Vladimir Getov, WP7 CoreGrid Leader**
09:15-10:50	**Session I - Components for Grids**
09:15-10:00	**Keynote Talk** Enterprise Grid Alliance Component Reference Model Dave Pearson, ORACLE and EGA EMEA Steering Committee
10:00-10:25	Redesigning the SEGL Problem Solving Environment: A Case Study of Using Mediator Components

	Natalia Currle-Linde, Michael Resch, Gosia Wrzesinska and Thilo Kielmann , HLRS , VUA
10:25-10:50	Mixing Aspects and components for Grid Computing Jean-Marc Menaud, Jacques Noye and Piere Cointe, INRIA
10:50-11:15	**Coffee-Break**
11:15-12:30	**Session II - Tools**
11:15-11:40	Deployment and File Transfer Tools for the Grid Mario Leyton and Romain Quilici , UChile, INRIA
11:40-12:05	Automatic Deployment of Interoperable Legacy Code Services G. Kecskemeti, Y.Zetuny, T.Kiss, G.Sipos , P.Kacsuk , G. Terstyanszky and S. Winter Westminster, SZTAKI
12:05-12:30	Towards Automatic Creation of Web Services for Grid Component Composition Jan Dunnweber, Francoise Baude, Virginie Legrand, Nikos Parlavantzas and Sergei Gorlatch, MUENSTER, INRIA
12:30-14:00	**Lunch Break**
14:00-15:35	**Session III: Grid Platforms**
14:00-14:45	**Keynote Talk** GridKit: Deep Middleware for the Divergent Grid Paul Grace, Computing Department, Lancaster University, UK
14:45-15:10	User Profiling for Lightweight Grids Lazar Kirchev, Minko Blyantov, Vasil Georgiev, Kiril Boyanov, Maciej Malawski, Marian Bubac, Stavros Isaiadis and Vladimir Getov IPP-BAS, CYFRONET, Westminster
15:10-15:35	Lightweight Grid Platform: Design Methodology Rosa M. Badia, Olav Beckmann, Marian Bubak, Denis Caromel, Vladimir Getov, Stavros Isaiadis, Vladimir Lazarov, Maceik Malawski, Sofia Panagiotidi, J. Thiyagalingam UPC, IPP-BAS, Imperial, CYFRONET, Westminster, INRIA
15:35-16:00	**Coffee-Break**
16:00-17:30	**Session IV - Integrated Environments**
16:00-16:25	Integration of GEMLCA and the P-GRADE Portal T.Kiss , G.Sipos, P.Kacsuk, K. Karoczkai, G. Terstyanszky and T. Delaitre Westminster, SZTAKI
16:25-16:50	GRID superscalar enabled P-GRADE portal Robert Lovas, Raul Sirvent, Gergely Sipos, Josep M. Perez, Rosa M. Badia, Peter Kacsuk, SZTAKI, UPC
16:50-17:15	Integrating of the OCM-G Monitoring System into the GRID superscalar Wlodzimierz Funika, Marcin Smetek, Marian Bubak CYFRONET
17:15	**Panel Discussion**

Industrial Session

14.00-14.15	**Opening**
14.15-15.00	Scheduling the Future: The changing world of workload distribution Tony Kay, Sun Microsystem's.
15.00-15.45	Role of ETSI in Grid standardisation Patrick Guillemin ETSI
15.45-16.00	**Coffee break**
16.00-16.45	IBM Grid Technical Strategy Jean-Pierre Prost IBM
16.45-17.30	

DAY 4 : THURSDAY OCTOBER 13TH, 2005

GridCoord Workshop : The use of Open Middleware for the Grids

9.00-9.30	Opening: ***Denis Caromel*** University of Nice Sophia-Antipolis, INRIA, CNRS-I3S, IUF, and Rep. GridCoord
	ProActive and the ObjectWeb Consortium Strategy for Open Source Grids ***Christophe Ney*** Executive Director of ObjectWeb
	W3C's Approach to Grids ***Daniel Dardailler*** W3C Associate Chair for Europe
10.30-11.00	**Coffee break**
	The EU EGEE Grid Infrastructure project ***Bob Jones*** CERN
	Unicore-UniGrids Activities and Strategies for Open Source Grids ***John Brooke*** University of Manchester
	The CoreGrid NoE Activities for Open Technologies for the Grid ***Marco Danelutto*** University of Pisa, CoreGrid
12.30-13.30	**Lunch break**
	Provenance Activity and Expected Architecture, and Envisioned Tools. ***Luc Moreau*** University of Southampton, Provenance Project Architect
	Strategy and Vision of the NextGRID Architecture ***Mark Parsons*** EPCC, and NextGRID Project Director

	Middleware Strategy for the DEISA Infrastructure ***Victor Alessandrini*** IDRIS, and DEISA Project Director
15.00-15.30	**Coffee break**
	GGF Activity in Programming Models and Environments ***Craig Lee, Thilo Kielmann*** The Aerospace Corporation, GGF Steering Group, Applications & Programming Models Vrije Universiteit Amsterdam
	The Open Middleware Infrastructure Institute ***Alistair Dunlop*** OMII, UK e-Science Core Program
	Open Grids in Japan and Asia Pacific ***Satoshi Matsuoka*** TITECH, NII NAREGI
17.00-18:30	**Panel**: Industrial Views on Existing and Future Grid Middlewares Chair: ***Denis Caromel*** University of Nice Sophia-Antipolis, INRIA, CNRS-I3S, IUF, and GridCoord *After hearing the academic views on middleware trends, the industrial actors will present their wish list of features, and how they would like, ideally, to be using Grid Middleware.*

Grid Application Toolkit (GAT) (Morning) Tutorial and Hands-On

9:00-10:15	**GAT and its programming models**
10:30-12:00	**Hands-on Session**

Ibis Tutorial and Hands On

14:00-15:30	**Ibis and its programming models**
15:30-18:00	**Hands-on Session**

DAY 5 : FRIDAY OCTOBER 14TH, 2005

CoreGrid : Programming models and components for the Grid

9.00 *Opening* (D. Caromel and M. Danelutto)	
Session Basic Programming Models	
9h30-9h50	OO SPMD for the Grid: an alternative to MPI and the road to Component F. Baude, L. Baduel, D. Caromel
9h50-10h10	The SAGA Task Model: Asynchronous Operations for Grid Applications A. Merzky, T. Kielmann, Vrije Universiteit
10h10-10h30	From (Data-)Flow Based Programs to Web Service Composition using XSLT Andrew Wendelborn, University of Adelaide, Australia
10h30-11h00	*Break*
11h00-11h20	TBA
11H20	Basic Programming Models: Discussions, Direction, Recommendation Chair: Pierre Kuonen
12.30 *Lunch*	
Session Grid Component Models (CoreGRID GCM)	
14h00-14h30	Report on required features to be included in GCM as well as the XML schema Massimo Coppola
14h30-15h00	Multicast and Gathercast interfaces for the GCM D. Caromel, M. Morel, Nikos Parlavantzas
15h00-15h30	Behavioral Specification and Verification of Compositional Grid Components" Tomas Barros, Univ. Chile, Eric madelaine, INRIA OASIS
15h30-16h00	TBA
16h00	*Break*
16h30-18h00	Session GCM: Discussions, Direction, Recommendation. Chair M. Danelutto
18h00	End of the workshop

P-Grade Tutorial and Hands on Grid Programming

9.15-10:45	**Session 1 :**
	Introduction to P-GRADE Parallel program development by a high-level, graphical language Creating PVM, MPICH and MPICH-G2 code Parallel debugging Parallel check-pointing and load-balancing Application migration in the Grid Workflow concept Accessing supercomputers, clusters and grids
11.15-13:00	**Session 1 :**

	Introduction to P-GRADE portal Workflow editor Workflow execution Workflow monitoring and visualization File management, file staging Fault-tolerance Certificate management Grid resource administration Grid resource monitoring Legacy code execution (GEMLCA) Demonstration
13:00-14:00	**Lunch break on site**
14:00-15:30	**Session 2 :**
	Hands-on experience and real-size demos Reaction-diffusion-advection system on SEE-Grid Urban traffic simulation on the UK NGS
15:30-18:00	**Session 3 :**
	Hands-on experience Workflow creation and execution

NorduGrid Event: ARC open day

9:00-9:15	**Welcome** ***B.Kónya*** , NorduGrid technical coordinator
9:15-11:00	ARC Tutorial (please register)
	ARC introduction ***A.Wäänänen*****,** Niels Bohr Institute
	Strengths ans weaknesses of ARC ***A.Konstantinov*****,** Vilnius University/University of Oslo
	ARC usage ***O.Smirnova*****,** Lund University
	Hands-on exercises ***B.Kónya*****,** Lund University
11:15-13:00	**Discussion panel "Grid diversity"**
	Open discussion on positive and negative sides of Grid diversification and how to deal with it ***Panelists from NorduGrid and other Grid projects***
13:00-14:00	**Lunch break on site**
14:00-19:00	**Open discussions, contributed presentations**
	New Grid features brainstorming ***S.Haug*** Oslo University

Appendix B

Technical Information

B.1 Involved Sites Technical Contacts

This document is taken from the on-line version [4].

```
Australia   UNIVERSITY OF MELBOURNE
            Rajkumar Buyya <raj@cs.mu.OZ.AU>
            Srikumar Venugopal <srikumar@cs.mu.OZ.AU>
Brazil      LNCC
            Bruno Schulze <bruno.schulze@gmail.com>
Chile       DCC Universidad de Chile
            Jose Piquer <jpiquer@nic.cl>,
            Florian Martin <Florian.Martin@sophia.inria.fr>,
            Luis Mateu <lmateu@dcc.uchile.cl>
Chile       UTFSM
            Xavier Bonnaire <xavier.bonnaire@inf.utfsm.cl>
China       BUPT
            MA Yan <mayan@bupt.edu.cn>
            Xiaohong Huang <huangxh@buptnet.edu.cn>
China       CNGRID
            Zhang Xiaoming <xmzhang@sccas.cn>
China       CNGRID-ICT
            Zhang Xiaoming <xmzhang@sccas.cn>
China       CNGRID-NHPCC
            Zheng Fang <zhengfang510@sohu.com>,
            <nhpccxa@mail.xjtu.edu.cn>
China       CNGRID-HKU
            Lin Chen <lchen2@cs.hku.hk>
China       CNGRID-SCCAS
            Zhang Xiaoming <xmzhang@sccas.cn>,
            Sungen Den <dsg@sccas.cn>
```

China	CNGRID-SCCNET Jiang Kai <kjiang@ssc.net.cn>
China	CNGRID-USTC PengZhan Liu <pzliu@mail.ustc.edu.cn>
France	IDRIS-DEISA Victor Alessandrini <va@idris.fr>, Philippe Collinet <collinet@idris.fr>, Gilles Gallot <Gilles.Gallot@idris.fr>
France	INRIA SOPHIA-ANTIPOLIS Nicolas Niclausse <Nicolas.Niclausse@sophia.inria.fr>, Francis Montagnac <Francis.Montagnac@sophia.inria.fr>, Janet Bertot <Janet.Bertot@sophia.inria.fr>, Jean-Luc Szpyrka <Jean-Luc.Szpyrka@sophia.inria.fr>, Antoine Zogia <Antoine.Zogia@sophia.inria.fr>, Regis Daubin <Regis.Daubin@sophia.inria.fr>
France	GRID5000-BORDEAUX Aurelien Dumez <aurelien.dumez@labri.fr>
France	GRID5000-GRENOBLE Nicolas Capit <<nicolas.capit@imag.fr>
France	GRID5000-LYON Frederic Desprez <frederic.desprez@ens-lyon.fr>, Stephane D'Alu <sdalu@ens-lyon.fr>
France	GRID5000-ORSAY Philippe Marty <philippe.marty@lri.fr>, Gilles Gallot
France	GRID5000-RENNES Guillaume Mornet <gmornet@irisa.fr>, David Margery <David.Margery@irisa.fr>
France	GRID5000-SOPHIA Sebastien Georget <Sebastien.Georget@sophia.inria.fr>, Nicolas Niclausse <Nicolas.Niclausse@sophia.inria.fr>
France	GRID5000-TOULOUSE Celine Juan <cjuan@cict.fr>, Pierrette Barbaresco <pb@cict.fr>
France	LIFL Melab Nouredine <Nouredine.Melab@lifl.fr>, El-ghazali Talbi <El-ghazali.Talbi@lifl.fr>, Sebastien Cahon <Sebastien.Cahon@lifl.fr>
France	LORIA Xavier Cavin <Xavier.Cavin@loria.fr>, Bertrand Wallrich <Bertrand.Wallrich@loria.fr>, Alain Filbois <Alain.Filbois@loria.fr>, Olivier Demengeon <olivier.demengeon@loria.fr>, Benjamin Dexheimer <Benjamin.Dexheimer@loria.fr>
France	SUPELEC

	Stephane Vialle <vialle@metz.supelec.fr>,
	Patrick Mercier <Patrick.Mercier@supelec.fr>
Germany	UNICORE
	Daniel Mallmann <d.mallmann@fz-juelich.de>
Greece	FORTH ICS
	Manolis Marazakis <maraz@ics.forth.gr>
Ireland	QUEEN’S UNIVERSITY OF BELFAST
	Ron Perrott <r.perrott@qub.ac.uk>,
	Andrew Carson <a.carson@Queens-Belfast.AC.UK>
Italy	BENEVENTO
	Eugenio Zimeo <zimeo@unisannio.it>,
	Nadia Ranaldo <ranaldo@unisannio.it>
Italy	ISTI
	Domenico Laforenza <domenico.laforenza@isti.cnr.it>,
	Ranieri Baraglia <Ranieri.baraglia@isti.cnr.it>,
	Giancarlo Bartoli <giancarlo.bartoli@isti.cnr.it>
Italy	UNIVERSITY OF PISA
	Marco Danelutto <marcod@di.unipi.it>,
	Pietro Vitale <vitale@di.unipi.it>
Netherland	VRIEJ UNIVERISTY
	Kees Verstoep <versto@cs.vu.nl>,
	Henri Bal <bal@cs.vu.nl>,
Norway	NORDUGRID
	Oxana Smirnova,
	Aleksandr Konstantinov,
	Balazs Konya,
	Alexander Lincoln Read,
	<nordugrid-discuss@nordugrid.org>
Switzerland	CERN/GILDA TESBED (Italy)
	Bob Jones <Robert.Jones@cern.ch>,
	Marc Ozonne <Marc.Ozonne@sophia.inria.fr>,
	Roberto Barbera <roberto.barbera@ct.infn.it>,
	Giuseppe Platania <giuseppe.platania@ct.infn.it>
Switzerland	EIF
	Pierre Kuonen <pierre.kuonen@eif.ch>,
	Jean-Francois Roche <jfrancois.roche@eif.ch>
Switzerland	ETHZ
	Luc Girardin <girardin@icr.gess.ethz.ch>
USA	UC IRVINE
	Stephen Jenks <sjenks@uci.edu>
USA	USC - CENTER FOR GRID TECHNOLOGIES
	Mats Rynge <rynge@isi.edu>

B.2 Technical Sites Information and Instruction Form

Informations that we need to know about machines you are going to provide for the Grid Plugtests:

- Number of machines available:
- For each machine (use range with the form [1 .. N] or [a ..z] if possible) Name: IP Address: OS (linux, windows, solaris): CPU (Mhz or Ghz) Nb of processors (monopro, bipro,...):
- Disc quota for a single user (Quota - Unlimited):
- Specific questions about dedicated machines (cluster) Name and IP address of the frontend: What is the access protocol? (ssh, globus, other):
- What is the Job Scheduler? (LSF, PBS, Sun Grid Engine, other):
- Specific questions about desk machines and cluster without Job Scheduler How can we access machines from outside? (ssh, rsh, rlogin, other):
- General Questions Are those machines located behind? (Firewall - NAT - Other - Nothing):
- What is the file System (NFS - Other):

Actions you should perform on the machines:

- Open 1 accounts for Romain Quilici: rquilici
- On this account we will put ProActive (50M) and Java (90M) in order for all teams to access java and ProActive in a common location
- Unfilter machines with full access IP-bidirectionnal between available machines and following addresses: `138.96.251.74` (Test machines), `212.234.160.0/24` `192.80.24.96/27` (Contest Machines).

Those actions should be performed as soon as possible and no later than 30 of September

B.3 How To Prepare for the 2nd Grid Plugtests Contest Guide

This document has been taken from the online version[5].

B.3.1 Introduction

Two contests will take place during the Plugtests: The N-Queens Counting Problem and The Flowshop Problem. To solve these problems, a world wide Grid will be configured, composed of a rich diversity of systems (architecture, operating system and Java virtual machines).

The following document is intended to help contestants fine tune their applications to compete at the Plugtests event. Grid Architecture

The Grid will be composed of more than a 1000 CPUs. These CPUs will be distributed all around the world, and grouped into sites. The size of all sites will be heterogeneous, ranging from a handful to hundreds.

To deploy on each site, contestants will use already configured Deployment Descriptors. There will be one deployment descriptor per site, configured with a virtualnode named "plugtest". The name of this virtualnode is the one that should be hard-coded into the contestant's application code. The length of the node array (number of nodes) returned from the virtualnode will vary depending on the size of the site.

B.3.2 ProActive nodes and CPUs

The machines on a site may have one or more CPUs. For each site with more than one CPU the configuration can be one of the following:

- A node will be created for each processor on the machine. This means that if a site has X machines, and each machine has Y processors. Then the expected number of nodes is X*Y.
- Only one node will be created on the machine. This means that if a site has X machines, and each machine has Y processors. Then the expected number of nodes is X. In this cases, it is advisable to deploy Y active objects per node to take advantage of all the processors on the machine. Nevertheless, there is one case when this is not recommend, and that is when the active object uses static variables (see Warnings section).

B.3.3 Warnings

Warning on Static Variables

It is highly discouraged to use static variables in the deployed active objects. Sometimes, more than one active object will be deployed on the same same java virtual machine, which may produce a race conflict between the active objects.

Last year experience shows this is a latent risk, and must be avoided.

Warning on Native Code

Native code, is highly discouraged. The first reason is the heterogeneousness of the Grid, since code will require specific compilation for each site. The second reason is that size of the Grid, which makes it unfeasible to compile and copy the native code to the remote sites during the plugtest event. The third and last reason not to use native code is that by using it your team will limit the amount of machines to which it can deploy, reducing the Grid capacity.

The ProActive / Plugtests staff will not provide support for native code during the plugtest event.

B.3.4 Note on Private IP Sites

The machines of a site can have: private or public IPs. For sites with private IPs, ProActive has provided a new feature that will allow deployment between the site's frontend and the innermachines.

Nevertheless, the current status of this feature does not support inner node communication between two different sites. That is to say, if site A and site B have inner nodes: A1...AN, B1...BM, then Ax will not be able to communicate with By.

For security reasons, solutions which require communication between tasks will be limited to a subset of the sites known as Grid5000 (composed of more than a thousand CPUs).

B.3.5 Note on Parallel Deployment

Due to the large number of descriptor files, the deployment time is significant. Therefore it is recommended to contestants to deploy each descriptor file in parallel thread.

Moreover, the process of placing active objects on the nodes can also be done in parallel, using a thread pool.

B.3.6 Expected Capabilities

Teams are expected to:

- Deploy using several Deployment Descriptor Files. During the plugtest, the files will be located in a read-access directory.
- Deploy more than one active object per node when necessary.
- Handle communication of a large number of nodes.
- Provide a nqueen.jar file with the application at the contest.

B.3.7 Useful Links Important Links

- Architecture summary chart [6] will be updated constantly with information for all sites.
- Providers Ranking [7], will be update constantly with information for all sites.

B.3.8 Reference Links

- 2nd Grid Plugtests website (2005)[8]
- 1st Grid Plugtests website (2004) [9]
- 1st Grid Plugtests Technical Report (2004) [10]

B.3.9 Quick References

General Concepts

- One descriptor file per site, with more than 20 sites.
- VirtualNode name: plugtest
- Contestants accounts: team1,...,teamN

Configuration Files

- Deployment descriptors location: /0/plugtest/xmlfiles
- Plugtest environment configuration: ~/.profile_plugtest
- Team custom environment configuration: ~/.profile

Software

- Java Home: /usr/java/j2sdk1.4.2_09/
- Eclipse Home: /usr/local/eclipse/
- ProActive Location: /0/plugtest/ProActive/

Others

- Required java/proactive option: -Dproactive.useIPaddress=true
- Further questions: Mario.Leyton@sophia.inria.fr

B.4 Pictures

Figure B.1: Contestants in Plugtests room

Figure B.2: Contestants in Plugtests room

Figure B.3: Contestants in Plugtests room

Figure B.4: Contestants in Plugtests room

Figure B.5: Remote Plugtests Contestants: Santiago de Chile

Figure B.6: N-Queens contest winning team

Figure B.7: Flowshop contest winning team

Figure B.8: Panel of experts

Figure B.9: Panel of experts

Figure B.10: Auditorium audience

B.5 Descriptor Examples

Figure B.11: Descriptor Example: Hierarchical ssh

```xml
<?xml version="1.0" encoding="utf-8"?>
<ProActiveDescriptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DescriptorSchema.xsd">
  <componentDefinition>
    <virtualNodesDefinition>
      <virtualNode name="plugtest" property="multiple" />
    </virtualNodesDefinition>
  </componentDefinition>
  <deployment>
    <mapping>
      <map virtualNode="plugtest">
        <jvmSet> <vmName value="newJvm1" /> </jvmSet>
      </map>
    </mapping>
    <jvms>
      <jvm name="newJvm1">
        <creation> <processReference refid="sshHierarchical1" /> </creation>
      </jvm>
    </jvms>
  </deployment>
  <infrastructure>
    <processes>
      <processDefinition id="internalJVM">
        <jvmProcess class="org.objectweb.proactive.core.process.JVMNodeProcess">
          <classpath>
            <absolutePath value="/home/rquilici/ProActive/lib/log4j.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/ProActive.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/asm.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/xercesImpl.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/fractal.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/bouncycastle.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/jsch.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/javassist.jar" />
            <absolutePath value="/home/rquilici/ProActive/lib/nqueen.jar" />
          </classpath>
          <javaPath>
            <absolutePath value="/usr/java/j2sdk1.4.2_04/bin/java" />
          </javaPath>
          <policyFile>
            <absolutePath value="/home/rquilici/ProActive/scripts/proactive.java.policy" />
          </policyFile>
          <log4jpropertiesFile>
            <absolutePath value="/home/rquilici/ProActive/scripts/proactive-log4j" />
          </log4jpropertiesFile>
        </jvmProcess>
      </processDefinition>
      <processDefinition id="frontendJVM">
        <jvmProcess class="org.objectweb.proactive.core.process.JVMNodeProcess">
          <extendedJvm refid="internalJVM" overwriteParameters="yes" />
          <jvmParameters>
            <parameter value="-Dproactive.communication.protocol=rmissh" />
            <parameter value="-Djava.rmi.server.hostname=syntagma.local" />
          </jvmParameters>
        </jvmProcess>
      </processDefinition>
      <processDefinition id="ssh_list">
        <processList class="org.objectweb.proactive.core.process.ssh.SSHProcessList"
        fixedName="compute-0-" list="[0-14]^[1,6,13]" domain="">
          <processReference refid="internalJVM"></processReference>
        </processList>
      </processDefinition>
      <processDefinition id="sshHierarchical1">
        <hierarchicalProcess class="org.objectweb.proactive.core.process.ssh.SSHHierarchicalProcess"
        hostname="undisclosed" internal_ip="undisclosed.local" username="rquilici">
          <processReference refid="frontendJVM" />
          <hierarchicalReference refid="ssh_list" />
        </hierarchicalProcess>
      </processDefinition>
    </processes>
  </infrastructure>
</ProActiveDescriptor>
```

Figure B.12: Descriptor Example: Hierarchical Grid5000

```
<?xml version="1.0" encoding="utf-8"?>
<ProActiveDescriptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DescriptorSchema.xsd">
  <componentDefinition>
    <virtualNodesDefinition>
      <virtualNode name="plugtest" />
    </virtualNodesDefinition>
  </componentDefinition>
  <deployment>
    <mapping>
      <map virtualNode="plugtest">
        <jvmSet>
          <vmName value="JvmSophia" />
        </jvmSet>
      </map>
    </mapping>
    <jvms>
      <jvm name="JvmSophia">
        <creation>
          <processReference refid="sshSophia" />
        </creation>
      </jvm>
    </jvms>
  </deployment>
  <infrastructure>
    <processes>
      <processDefinition id="JVM_Forwarder">
        <jvmProcess class="org.objectweb.proactive.core.process.JVMNodeProcess">
          <classpath>
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/asm.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/bouncycastle.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/fractal.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/log4j.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/ProActive.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/xercesImpl.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/javassist.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/nqueen.jar" />
            <absolutePath value="/home/sophia/plugtest/ProActive/lib/jsch.jar" />
            <absolutePath value="/home/sophia/plugtest/flowshop.jar" />
          </classpath>
          <javaPath>
            <absolutePath value="/home/sophia/plugtest/jdk/bin/java" />
          </javaPath>
          <policyFile>
            <absolutePath value="/home/sophia/plugtest/ProActive/scripts/proactive.java.policy" />
          </policyFile>
          <log4jpropertiesFile>
            <absolutePath value="/home/sophia/plugtest/ProActive/scripts/proactive-log4j" />
          </log4jpropertiesFile>
          <jvmParameters>
            <parameter value="-Djava.system.class.loader=org.objectweb.proactive.core.classloader.ProActiveClassLoader" />
            <parameter value="-Djava.rmi.server.RMIClassLoaderSpi=org.objectweb.proactive.core.classloader.ProActiveRMIClassLoaderSpi" />
            <parameter value="-Dlog4j.defaultInitOverride=true" />
            <parameter value="-Dproactive.communication.protocol=rmissh" />
            <parameter value="-Dproactive.rmi.port=2030" />
          </jvmParameters>
        </jvmProcess>
      </processDefinition>
      <processDefinition id="JVM_Worker">
        <jvmProcess class="org.objectweb.proactive.core.process.JVMNodeProcess">
          <extendedJvm refid="internalJVM" overwriteParameters="yes" />
        </jvmProcess>
      </processDefinition>
      <processDefinition id="sshSophia">
        <hierarchicalProcess class="org.objectweb.proactive.core.process.ssh.SSHHierarchicalProcess"
        hostname="undisclosed" username="plugtest" internal_ip="undisclosed">
          <processReference refid="JVM_Forwarder" />
          <hierarchicalReference refid="oarSophia" />
        </hierarchicalProcess>
      </processDefinition>
      <processDefinition id="oarSophia">
        <oarProcess class="org.objectweb.proactive.core.process.oar.OARSubProcess"
        bookedNodesAccess="ssh">
          <processReference refid="JVM_Worker" />
          <commandPath value="/usr/bin/oarsub" />
          <oarOption>
            <resources>nodes=all,weight=2</resources>
            <scriptPath>
              <absolutePath value="/home/sophia/plugtest/ProActive/scripts/unix/cluster/oarStartRuntime.sh" />
            </scriptPath>
          </oarOption>
        </oarProcess>
      </processDefinition>
    </processes>
  </infrastructure>
</ProActiveDescriptor>
```

Figure B.13: Descriptor Example: GLite

```xml
<?xml version="1.0" encoding="utf-8"?>
<ProActiveDescriptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DescriptorSchema.xsd">
  <componentDefinition>
    <virtualNodesDefinition>
      <virtualNode name="plugtest" property="multiple"
      timeout="900000" waitForTimeout="false" />
    </virtualNodesDefinition>
  </componentDefinition>
  <deployment>
    <mapping>
      <map virtualNode="plugtest">
        <jvmSet>
          <vmName value="Jvm1" />
        </jvmSet>
      </map>
    </mapping>
    <jvms>
      <jvm name="Jvm1">
        <creation>
          <processReference refid="sshProcess" />
        </creation>
      </jvm>
    </jvms>
  </deployment>
  <infrastructure>
    <processes>
      <processDefinition id="localJVM">
        <jvmProcess class="org.objectweb.proactive.core.process.JVMNodeProcess">

          <classpath>
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/classes" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/ProActive_examples.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/ProActive.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/bcel.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/asm.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/log4j.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/xercesImpl.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/components/fractal.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/bouncycastle.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/ws/servlet-api.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/glite/glite-wms-jdlj.jar" />
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/lib/glite/classad.jar" />
          </classpath>
          <policyFile>
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/scripts/proactive.java.policy" />
          </policyFile>
          <log4jpropertiesFile>
            <absolutePath value="/afs/cern.ch/user/m/mozonne/public/ProActive/scripts/proactive-log4j" />
          </log4jpropertiesFile>
        </jvmProcess>
      </processDefinition>
      <processDefinition id="gLiteProcess">
        <gLiteProcess class="org.objectweb.proactive.core.process.glite.GLiteProcess"
        virtualOrganisation="gilda"
        executable="/usr/java/j2sdk1.4.2_08/bin/java"
        JDLFileName="job.jdl" Type="Job" stdError="error.log"
        stdOutput="stdout.log" retryCount="3">
          <processReference refid="localJVM" />
          <requirements>
          RegExp(".*infn.it:*",other.CEID)</requirements>
          <gLiteOptions>
            <JDLFilePath>
              <relativePath origin="user.home" value="/jdl" />
            </JDLFilePath>
            <JDLRemoteFilePath>
              <absolutePath value="/home/mozonne/JDL" />
            </JDLRemoteFilePath>
            <outputSandbox>error.log stdout.log</outputSandbox>
          </gLiteOptions>
        </gLiteProcess>
      </processDefinition>
      <processDefinition id="sshProcess">
        <sshProcess class="org.objectweb.proactive.core.process.ssh.SSHProcess"
        hostname="undisclosed" username="mozonne">
          <processReference refid="gLiteProcess" />
        </sshProcess>
      </processDefinition>
    </processes>
  </infrastructure>
</ProActiveDescriptor>
```

Figure B.14: Descriptor Example: NorduGrid

```
<?xml version="1.0" encoding="utf-8"?>
<ProActiveDescriptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DescriptorSchema.xsd">
  <componentDefinition>
    <virtualNodesDefinition>
      <virtualNode name="plugtest" timeout="1200000" />
    </virtualNodesDefinition>
  </componentDefinition>
  <deployment>
    <mapping>
      <map virtualNode="plugtest">
        <jvmSet>
          <vmName value="Jvm1" />
        </jvmSet>
      </map>
    </mapping>
    <jvms>
      <jvm name="Jvm1">
        <creation>
          <processReference refid="ngProcess" />
        </creation>
      </jvm>
    </jvms>
  </deployment>
  <FileTransferDefinitions>
    <FileTransfer id="ng_transfer">
      <file src="http://grid.uio.no/runtime/j2re1.4.2_08.tar.gz"
      dest="j2re1.4.2_08.tar.gz" />
      <file src="dist/ProActive/ProActive.jar"
      dest="ProActive.jar" />
      <file src="lib/asm.jar" dest="asm.jar" />
      <file src="lib/components/fractal.jar" dest="fractal.jar" />
      <file src="lib/bouncycastle.jar" dest="bouncycastle.jar" />
      <file src="lib/log4j.jar" dest="log4j.jar" />
      <file src="lib/xercesImpl.jar" dest="xercesImpl.jar" />
      <file src="scripts/proactive-log4j" dest="proactive-log4j" />
      <file src="scripts/proactive.java.policy"
      dest="proactive.java.policy" />
    </FileTransfer>
  </FileTransferDefinitions>
  <infrastructure>
    <processes>
      <processDefinition id="localJVM1">
        <jvmProcess class="org.objectweb.proactive.core.process.JVMNodeProcess">

          <classpath>
            <absolutePath value="ProActive.jar" />
            <absolutePath value="asm.jar" />
            <absolutePath value="fractal.jar" />
            <absolutePath value="bouncycastle.jar" />
            <absolutePath value="log4j.jar" />
            <absolutePath value="xercesImpl.jar" />
          </classpath>
          <javaPath>
            <absolutePath value="$JAVA_HOME/bin/java" />
          </javaPath>
          <policyFile>
            <absolutePath value="proactive.java.policy" />
          </policyFile>
          <log4jpropertiesFile>
            <absolutePath value="proactive-log4j" />
          </log4jpropertiesFile>
        </jvmProcess>
      </processDefinition>
      <processDefinition id="ngProcess">
        <ngProcess class="org.objectweb.proactive.core.process.nordugrid.NGProcess"
        hostname="undisclosed">
          <processReference refid="localJVM1" />
          <FileTransferDeploy refid="ng_transfer">
            <copyProtocol>processDefault</copyProtocol>
            <sourceInfo prefix="file:///0/plugtest/ProActive" />
          </FileTransferDeploy>
          <ngOption>
            <executable>
              <absolutePath value="/0/plugtest/ProActive/scripts/unix/cluster/ngStartRuntime.sh" />
            </executable>
            <count>28</count>
            <outputFile>output.txt</outputFile>
            <errorFile>error.txt</errorFile>
          </ngOption>
        </ngProcess>
      </processDefinition>
    </processes>
  </infrastructure>
</ProActiveDescriptor>
```

Figure B.15: Descriptor Example: Unicore

```
<?xml version="1.0" encoding="utf-8"?>
<ProActiveDescriptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DescriptorSchema.xsd">
  <componentDefinition>
    <virtualNodesDefinition>
      <virtualNode name="plugtest" property="multiple" />
    </virtualNodesDefinition>
  </componentDefinition>
  <deployment>
    <mapping>
      <map virtualNode="plugtest">
        <jvmSet>
          <vmName value="JvmTestSite" />
        </jvmSet>
      </map>
    </mapping>
    <jvms>
      <jvm name="JvmTestSite">
        <creation>
          <processReference refid="unicoreProcessTestSite" />
        </creation>
      </jvm>
    </jvms>
  </deployment>
  <FileTransferDefinitions>
    <FileTransfer id="ProActiveLite">
      <dir src="ProActive" />
    </FileTransfer>
  </FileTransferDefinitions>
  <infrastructure>
    <processes>
      <processDefinition id="jvmTestSite">
        <jvmProcess class="org.objectweb.proactive.core.process.JVMNodeProcess">
          <classpath>
            <absolutePath value="ProActive/lib/ProActive.jar" />
            <absolutePath value="ProActive/lib/asm.jar" />
            <absolutePath value="ProActive/lib/bouncycastle.jar" />
            <absolutePath value="ProActive/lib/fractal.jar" />
            <absolutePath value="ProActive/lib/jsch.jar" />
            <absolutePath value="ProActive/lib/log4j.jar" />
            <absolutePath value="ProActive/lib/xercesImpl.jar" />
            <absolutePath value="ProActive/lib/javassist.jar" />
            <absolutePath value="ProActive/lib/nqueen.jar" />
          </classpath>
          <javaPath>
            <absolutePath value="/opt/j2sdk1.4/bin/java" />
          </javaPath>
          <policyFile>
            <absolutePath value="ProActive/scripts/proactive.java.policy" />
          </policyFile>
          <log4jpropertiesFile>
            <absolutePath value="ProActive/scripts/proactive-log4j" />
          </log4jpropertiesFile>
        </jvmProcess>
      </processDefinition>
      <processDefinition id="unicoreProcessTestSite">
        <unicoreProcess class="org.objectweb.proactive.core.process.unicore.UnicoreProcess"
        jobname="ProActivePlugtestJob" submitjob="true">
          <processReference refid="jvmTestSite" />
          <unicoreDirPath>
            <absolutePath value="/0/plugtest/certificates/unicoretestsite" />
          </unicoreDirPath>
          <keyFilePath>
            <absolutePath value="/0/plugtest/certificates/unicoretestsite/keystore_testsite" />
          </keyFilePath>
          <unicoreOption>
            <usite name="Gate Europe" type="CLASSIC" url="http://testgrid.unicorepro.com:4000" />
            <vsite name="SUPRENUM" nodes="1" processors="1"
            memory="256" runtime="3600" priority="normal" />
          </unicoreOption>
          <FileTransferDeploy refid="ProActiveLite">
            <copyProtocol>processDefault</copyProtocol>
            <sourceInfo prefix="/0/plugtest/ProActiveLite" />
            <destinationInfo />
          </FileTransferDeploy>
        </unicoreProcess>
      </processDefinition>
    </processes>
  </infrastructure>
</ProActiveDescriptor>
```