3rd Grid Plugtest Report

21/05/2007

Industrial Sponsors

In Collaboration With

Industrial Sponsors

With The Support Of

International Partners

European Partners

Contents

1. Introduction

Following the success of the first two Grid Plugtests, the 3rd Grid Plugtests has been held from November 27th to December 1st 2006. Co-organized by ETSI and INRIA OASIS team. OASIS is a joint team between INRIA, UNSA, I3S-CNRS which develops the ProActive Grid middleware. The objectives were: to test Grid interoperability and Grid middlewares in a large scale environment, and to discover, through user experiences and open discussions, what would be the features needed for Grid middlewares.

The 3rd Grid Plugtests consisted in several events: Conferences, Workshops, Tutorials and a Contest. Drawing over 240 participants from many different countries. The schedule was the following:

Table 1.1. Conference schedule

Mon 27 Nov.	GridCOMP meeting CoreGRID Institute on Resource Management and Scheduling Ibis Tutorial
Tue 28 Nov.	ProActive and GCM tutorial CoreGRID Institute on System Architecture CoreGRID Institute on Programming Model BIGG: Bridging Global Computing with Grid 3rd Grid Plugtests
Wed 29 Nov.	3rd ProActive and GCM User Group BIGG: Bridging Global Computing with Grid 3rd Grid Plugtests
Thu 30 Nov.	1st CoreGRID Industrial Conference 3rd Grid Plugtests
Fri 1 Dec.	1st CoreGRID Industrial Conference ETSI TC GRID#2 Standardization Meeting CoreGRID Institute on Grid Systems, Tools and Environments EGEE VO Management g-Eclipse Information meeting

Tuesday, Wednesday and Thursday two contests took place (Section 3) with 14 participating teams. For the contest, an intercontinental Grid has been setup (Section 2) by the OASIS Team using the ProActive middleware, which inter-operated with several other middlewares and protocols.

2. The Grid

2.1. Installation

As experiments testbed, a Grid has been setup for three days with the help of numerous partners. This Grid was deployed on 8 different countries, in more than 20 sites, gathering 4130 cores developing near than 2 Tflops (measured with the SciMark 2.0 benchmark).

Given its heterogeneity, each site had to be configured and finely tuned. This involved figuring out the operating system, installing an adequate Java Virtual Machine, figuring out the network/firewall configuration, job scheduler, etc. This work has been done by the OASIS Team, mainly Clement Mathieu, who prepared the Grid for the contest and Plugtests.

Application deployment was thus very simple and transparent for the Plugtests users. Indeed all the architectural details were hidden by the ProActive middleware which provides an uniform deployment model. This year, Plugtests users were not allowed to connect to Grid's machines.

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 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 arise 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 describe, 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 one hour to several days.

1. Invite partner to participate in the 3rd Grid Plugtests.
2. Request partner to open an account for the Grid Plugtests.
3. Analyze and configure the environment of the site: O.S., network, data storage, job scheduler, JVM, DNS configuration, etc.
4. If necessary, extends ProActive to support local technologies
5. Create a script for synchronizing ProActive libraries, JVMs and users' data with the site.
6. Create a cleaning script for the site.

7. Create an XML Deployment Descriptor for the site.
8. Test the site

Steps 1 and 2 were the longer as they require to contact teams and administrators.

Step 3 was simple and fast as gathering needed information is a repetitive task.

Step 4 consisted in working on Glite integration, 2.3.1, cluster-fork, 2.3.2, support and improving deployment, 2.3.3. Elton Mathias spent more than a month to have Glite support ready.

Steps 5 and 6 were fairly easy to build. Synchronization and cleaning were relying on a scripting framework. Adding a new site is an easy task, that does not require more than few minutes.

Steps 7 and 8 were the most time consuming. Testing each site was really hard and some sites changed their configuration multiple times during the test period.

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

Most of the sites were running various versions of Linux (Debian, Red Hat, Rocks, TurboLinux etc.). One cluster was running Mac OS X and no cluster was running Windows, HPUX, Solaris or AIX.

Sun's JVM in version 1.5.0_08 has been selected as default JVM and has been deployed everywhere it was possible. Depending on the architecture a 32 bits or a 64 bits version were deployed, 32 bits JVMs were used on Xeon IA32, Pentium III, Pentium IV and 64 bits JVMs were used on Opteron and Xeon IA64.

Exceptions to the uniq JVM rule had been made for Itanium 2 and G5 clusters. BEA's JRockit JVM has been used for Itanium 2 and Apple's JVM for G5.

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.

Plugtests users had not been allowed to log into Grid's machine. The only way to interact with the Grid resources was to perform a ProActive deployment.

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 inaccessible (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 ProActive's `rmi-ssh` tunneling features. *Restrictive* sites, were handled using the recently developed feature of hierarchical deployment and communication

No island sites were encountered.

No change to clusters configuration has been asked for the Plugtests. The Grid has been successfully deployed using the sites as they were.

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 required to synchronize each nodes.

Data storage was fully transparent for the Plugtests user. They had the ability to replicate a directory on the whole Grid simply by using a shell script. User files in this directory can be acceded in the same way from any computational resource.

2.3. Development of New Features

2.3.1. Glite integration

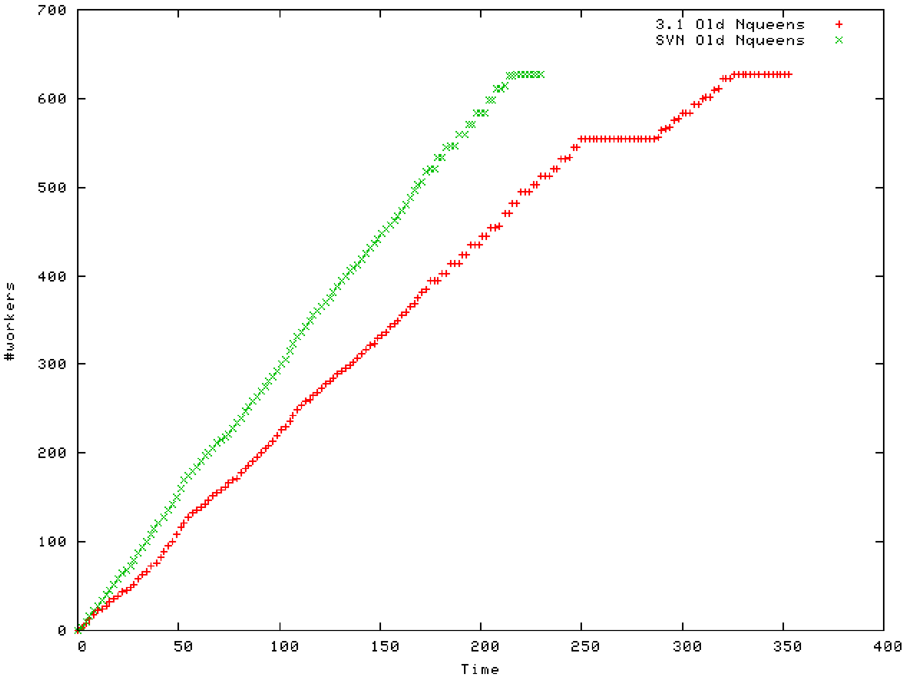
The ProActive gLite integration received two main improvments that have helped users deploying within EGEE environment:

- support of multiple deployment at once through gLite MPI jobs and
- compliance to hierarchical deployment model.

One of the main constraints of using EGEE/gLite in large computations is the delay caused by the platform to schedule a job. Single job deployment may last from half a minute to 10 minutes. Indeed, the submission time varies greatly, so that it is impossible to predict in advance deployment time. This constraint is even stronger for non-embarrassingly parallel applications that depends of the entire environment up before executing, as the deployment time may overlap the walltime, so making impossible the usage of the EGEE platform for such kind of application. The first improvement offers a solution to this limitation by supporting transparent multiple submission of ProActive runtimes at once in the EGEE grid through MPI jobs. By doing so, users may expect to have hundreds or thousands of ProActive nodes deployed by just paying the price of a single Job, so reducing the delay of deployment and processing/memory consumption, necessary to control jobs.

The second improvment makes possible the transparent local usage of EGEE resources through remote User Interfaces nodes (UI). Considering that most of the Compute Elements (CEs) offer only limited outbound connectivity, this tends to improve the usability of EGEE platform via ProActive. A further benefit of this improvement is the possibility of gathering into a single application/computation resources located on different Grid platforms, even if they don't offer a direct connection with the external world. These improvements included the necessary elements to the effective usage of EGEE/gLite in the Grid Crunching Day (7th Dec., Fribourg, Switzerland. http://www.gridinitiative.ch/gridcrunchday.html)

Figure 2.1. Improving deployment time before and after the Plugtests



and may also support next Plugtest editions, increasing even more the concept of interoperability in the ProActive middleware.

2.3.2. Cluster-fork integration

Rocks Linux distribution comes with a command named cluster-fork that allows users to run a given command on a set of CPUs sequentially or in parallel. Usually, when we encounter a Rocks Linux cluster we prefer to use Sun Grid Engine as it is also integrated and more powerful. But we had encountered some problems with the Benevento's SGE configuration so we decided to use to cluster-fork for this particular site. cluster-fork was also used at the INRIA Nef site.

2.3.3. Deployment improvements

Using a large scale Grid allowed us to speed up deployment time by two.

We analyzed ProActive with tools like, YourKit Java Profiler or JRockit Mission control to track thread and memory usage and to identify locks under high contention. The most significant improvement is located inside the deployment process: We have been able to divide the deployment time of our N-Queens application by two, see Fig. 2.1. Of course the application has not been modified.

2.4. Site Description

For the 3rd Grid Plugtests, more than 24 sites on 8 different countries were configured by the OASIS team. This year only reasonably powerful clusters were kept as Grid Plugtests users tend to not use the smaller ones. It has been decided to save the time required to configure theses clusters and spend it elsewhere like better automation or documentation. That explain why the 3rd Grid Plugtests used less sites than the 2nd one. It is a political choice and it is not due to technical limitations.

The table 2.1 presents the list of sites that formed part of the Grid. To increase the readability, the sites were sorted by country, and secondly by site name. The columns are described as:

Country The name of the country that the site belongs to
Site_Name The name of the site
Nodes The number of nodes provided by the site
CPUs The number of processors per node
Cores The number of core per CPU
CPU The CPU type
Freq The clock speed of the CPU
O.S. Operating System
Sched The scheduling (job submission) mechanism used to deploy on the site's machines
JVM The Java Virtual Machine
GFlops Represents a rough estimation of the site's computation capacity. Please not 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 2.5.

2.5. Grid Benchmarks

Benchmarks were done using Scimark that is a pure Java benchmark. As JVM types differ in vendor and version, comparing Mflops between site is pointless. Moreover, given the nature of a Grid of this nature (in size and location), for some sites it was not possible to gather all the resources at once, for these cases, the total capacity has been extrapolated. For all of these reasons, computed Grid benchmark is an approximation, and should be considered as a reference.

The Grid built for the 3rd Grid Plugtests shows a massive increase of its overall processing capacity power: 1700 Gflops. While the 2nd Plugtests Grid was about 450 Gflops and the 1st 100 GFlops. Details of this computation can be found in 2.1.

Figure 2.2 shows the distribution of number of cores per Grid site. Figure 2.3 shows the distribution of Grid processing capacity (in GFlops). Figure 2.4 shows both metrics.

2.6. Issues when building an internet Grid

As every year, many difficulties were encountered during the Plugtests. The next sections show some example of encountered problems.

2.6.1. Grid5000

Grid5000 is the main resources provider and the only site able to provide more than 2000 cores. To avoid issues that were encountered during the 2nd Grid Plugtests, the job scheduler (oar) has not been used. Instead a combination of reservation mechanism and the SSH deployment has been used.

During the event, LDAP and DNS servers repetitively crashed. After some investigations, a solution was found for LDAP servers. OpenLDAP can be recompiled to allow to open more file descriptors. Grid5000 administrators, deployed a recompiled version of OpenLDAP during the event.

DNS issue has not been solved. We cannot explain why so many requests were done and more investigations are needed. This problem is hard to reproduce and it is not clear if it is a Grid5000 issue or a ProActive one.

Table 2.1. Grid Sites

Country	Site Name	Nodes	CPUs	Cores	CPU	Freq	O.S.	Sched	JVM	GFlops
Australia	U. Melbourne	13	1	1	Pentium 4	2.4	Red Hat 7.3	PBS	Sun 1.5.0_08 i586	2
Chile	UDP	4	1	2	Xeon IA32	3.4	Centos 4.4	Torque	Sun 1.5.0_08 i586	2
China	BUPT	11	2	2	Xeon IA32	3.2	Red Hat AS 3	SSH	Sun 1.5.0_08 i586	15
China	SCC	80	4	1	Opteron 850	2.4	TurboLinux	LSF	Sun 1.5.0_08 amd64	163
China	SSCAS	25	2	2	Itanium 2	1.2	Red Hat AS 2	LSF	Sun 1.6.0 i586	23
China	Tsinghua	114	1	2	Xeon IA64				Sun 1.5.0_08 i586	48
France	Grid5000 Bordeaux	48	2	1	Opteron 248	2.2	Fedora Core 4	OAR	Sun 1.5.0_08 amd64	49
France	Grid5000 Lille	15	2	1	Opteron 252	2.6	Red Hat EL3	OAR	Sun 1.5.0_08 amd64	14
		53	2	1	Opteron 248	2.2	Red Hat EL3	OAR	Sun 1.5.0_08 amd64	55
France	Grid5000 Lyon	56	2	1	Opteron 246	2.0	Debian 3.1	OAR	Sun 1.5.0_08 amd64	51
		70	2	1	Opteron 250	2.4	Debian 3.1	OAR	Sun 1.5.0_08 amd64	76
France	Grid5000 Grenoble	103	2	1	Itanium 2	1.0	Red Hat EL3	OAR	Sun 1.5.0_08 i586	18
France	Grid5000 Orsay	216	2	1	Opteron 246	2.0	Debian testing	OAR	Sun 1.5.0_08 amd64	197
		126	2	1	Opteron 250	2.4	Debian testing	OAR	Sun 1.5.0_08 amd64	138
France	Grid5000 Nancy	47	2	1	Opteron 246	2.0	Debian testing	OAR	Sun 1.5.0_08 amd64	53
France	Grid5000 Rennes	64	2	1	Xeon IA32	2.4	Debian testing	OAR	Sun 1.5.0_08 i586	29
		99	2	1	Opteron 246	2.0	Debian testing	OAR	Sun 1.5.0_08 amd64	94
		64	2	1	Opteron 248	2.2	Debian 3.1	OAR	Sun 1.5.0_08 amd64	64
		64	2	1	G5	2.0	OS X	OAR	Sun 1.5.0_06 ppc	25
France	Grid5000 Sophia	105	2	1	Opteron 246	2.0	Rocks Linux	OAR	Sun 1.5.0_08 amd64	103
		56	2	2	Opteron 275	2.2	Rocks Linux	OAR	Sun 1.5.0_08 amd64	115
France	Grid5000 Toulouse	58	2	1	Opteron 248	2.2	Fedora Core 3	OAR	Sun 1.5.0_08 amd64	58
France	INRIA Nef	32	2	1	Opteron 246	2.0	Rocks Linux	Torque	Sun 1.5.0_08 amd64	21
France	OCA/U. Nice	48	2	2	Opteron 275	2.2	Red Had EL 4	LSF	Sun 1.5.0_08 amd64	110
France	Supelec	33	1	1	Pentium 4	3.0	Fedora Core 3	SSH	Sun 1.5.0_08 i586	14
Italy	Benevento	53	2	2	Xeon IA32	3.0	Rocks Linux	cluster-fork	Sun 1.5.0_08 i586	98
		15	1	1	Xeon IA32	2.8	Rocks Linux	SGE	Sun 1.5.0_08 i586	3
Italy	U. Pisa	32	1	1	Pentium 3	0.7	Fedora Core 1	SSH	Sun 1.5.0_08 i586	4
Japan	U. Tokyo	69	1	1	Pentium M	1.9	Debian 3.1	SSH	Sun 1.5.0_08 i586	25
Netherland	DAS-II	100	2	1	Pentium 3	1.0	Red Hat EL 3	prun	Sun 1.5.0_08 i586	36

Figure 2.2. Number of cores

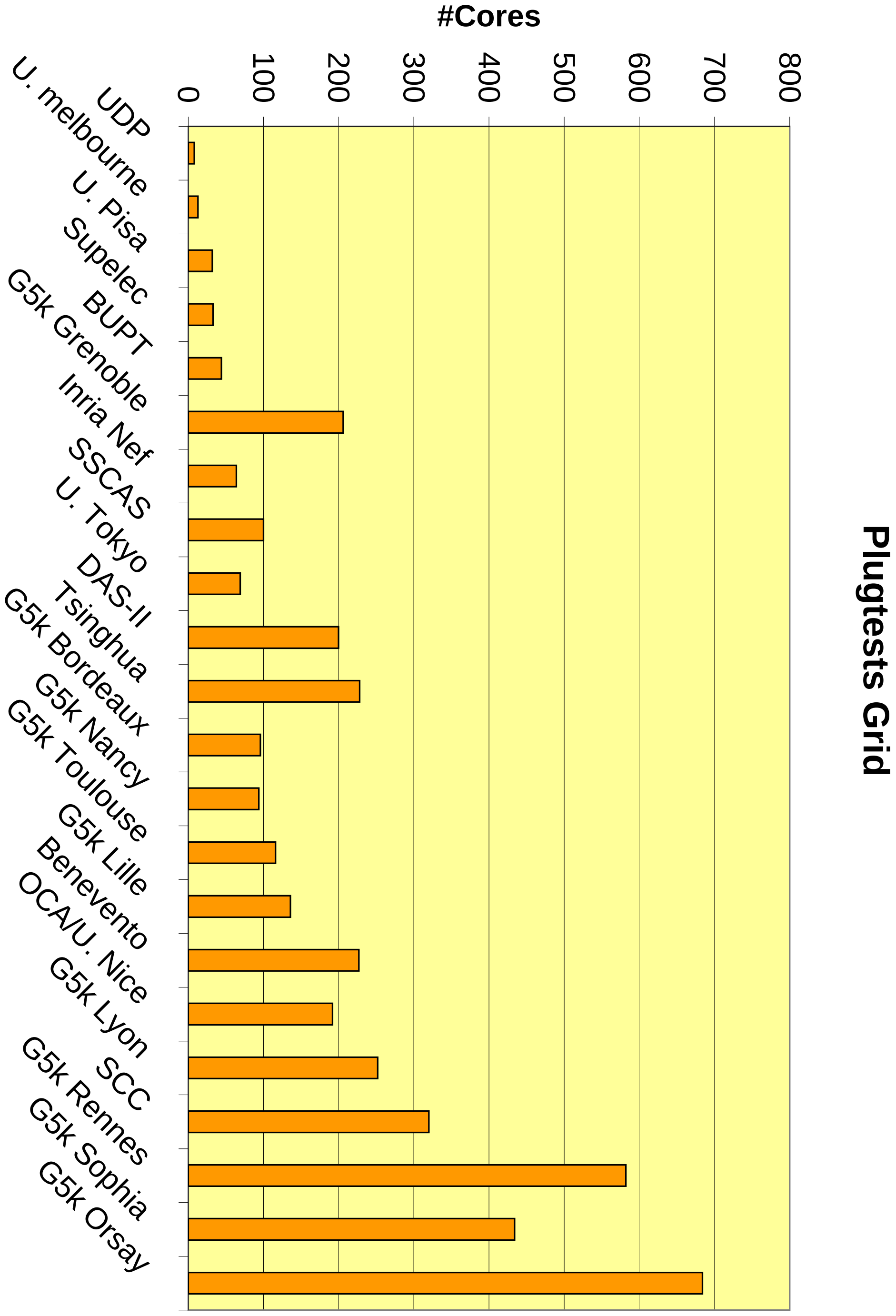


Figure 2.3. Processing Capacity

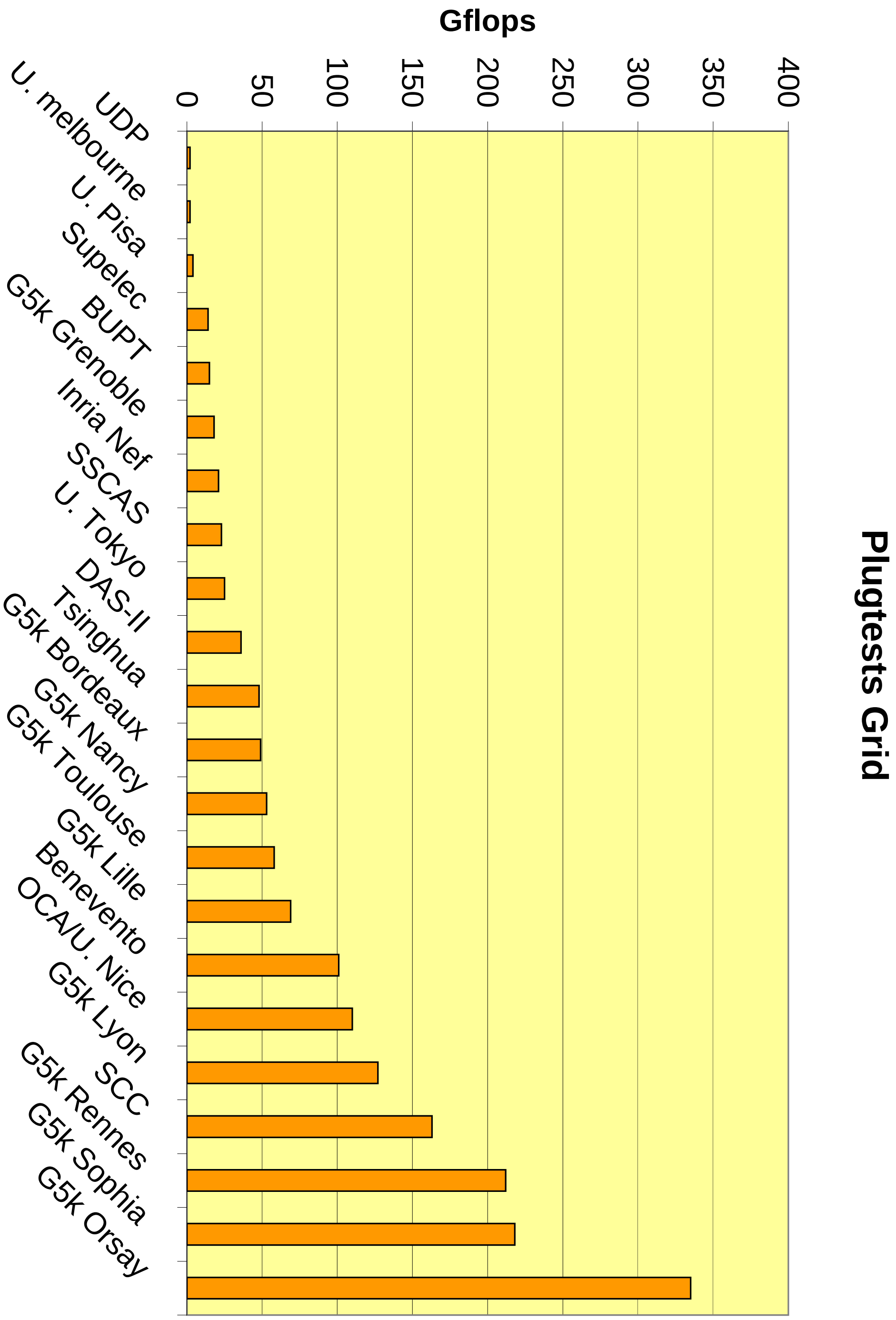
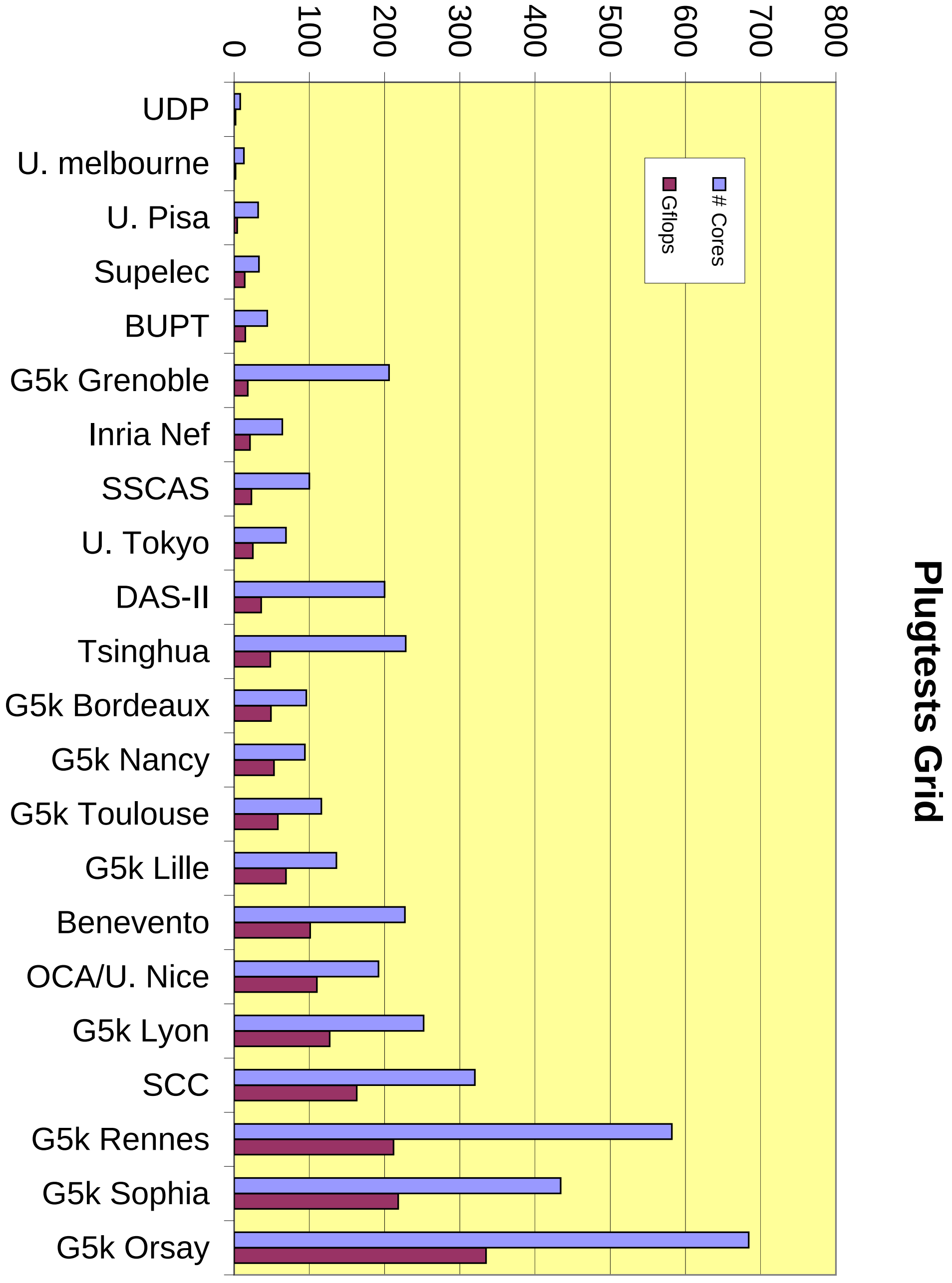


Figure 2.4. Processing Capacity and Number of Processors



To conclude, 3rd Grid Plugtests showed up that deployments involving hundred of nodes scattered on several clusters put Grids under very high pressure. An open topic is for Grid administrators and Grid middleware developers to work together to identify and to eliminate bottlenecks appearing during large deployments.

2.6.2. Large Scale Deployment

Using a large grid allowed all the constants to experiment large scale deployments and spot weakness points of their applications, middlewares or infrastructure. For ProActive the main problem we encountered were memory consumption and lack of speed during the deployment.

Resources required by a large scale deployment can't be fulfilled by a single host. Memory and system resources (thread, file descriptors etc.) consumption evolve linearly with the number of hosts. Computing resources must be cut up in smaller subsets (hierarchical deployment) or runtimes should be persistants (using a P2P infrastructure [p2p] for example).

Taktuk[Taktuk] perfomed much better than ProActive in term of deployment speed and memory usage. It spreads itself using an adaptive algorithm (work-stealing). In fact it was a bit too fast and LDAP, NFS or DNS servers were under very high pressure (some crashed).

The ProActive SSH deployment must also be optimized in term of memory space and speed.

2.6.3. Others

In addition to the previous problems, we had to deal with the following ones:

Some sites had quotas that was set too low to fulfill Plugtests requirements. This issue has been discovered when users tried to synchronize their datas.

Some clusters changed their configuration or had a strange behavior during the event. Torque [Torque] deployment has been dropped for the INRIA cluster.; It has been replaced by a cluster-fork one for example.

Some descriptors has been rewrited to fulfill Plugtests users needs. Some teams required a scattering of Grid5000 into three pools of resources.

A lot of hard drives died during the event leading from lost of several hosts (local hard drive failure) to lost of cluster (NFS server hard disk failure).

And to finish we had to deal with a global power outage that lasted more that 45 minutes. Fortunately no data was lost, and the run in progress had been able to complete successfully.

3. The Contests

In addition, two contests were organized during the 3rd Grid Plugtests. Topics were the N-Queens Counting problem and the FlowShop problem. These events were strictly engineering events, 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 but one used Java, and when possible, some used native code inside a Java wrapper. The IMAG team choose to use Kaapi as middleware and to write their application in C++.

3.1. N-Queens Counting Problem

This problem can be summarized as follows:

How many ways can N queens be placed on a NxN chessboard ?
Ten teams competed this year in the N-Queens contest. Teams competed in 3 categories:

1. Maximum number of solutions computed in 1 hour
2. Maximum number of nodes deployed in one successful computation
3. Fastest computation

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

3.1.1. Teams

LSC/UFSM (Brazil)

Organization Laboratorio de Sistemas de Computacao (LSC), Universidate Federal de Santa Maria (UFSM)
Members Tiago Scheid (coordinator), Benhur de Oliveira Stein, Marcelo Pasin, Rodolfo Leffa de Oliveira, Marcio Parise Boufleur, Guilherme Piegas Koslovskir

ChinaGrid (China)

Organization Computer School of Huazhong University of Sci. & Tech.
Members Yuhua Huang, Chao Ma, Jumhe Xiao, Zhao Chen, Dong Pan, De Jiang, Wei Zhu
Contact Yongwei Wu

BUPT (China)

Organization Beijing University of Posts and Telecommunications (BUPT)
Members Huang Xiaohong (corrdinator), Ma Yan, Su Yujie, Wu Yongjuan, Du Nan, Han Yunan, Sun Zheng, Yang Fan, Ding Toa, Jiang Lili, Xiao Qiangju, Liu Benjin, Chen Zhihui, Li Feiyun
Contact Yongwei Wu

FIT (China)

Organization
Members Weiyuan Huang (coordinator), Yulai Yuan, Zhongqiang Zhang, Le Du
Contact Yongwei Wu

UDP (Chile)

Organization University Diego Portales (UDP)
Members Tomas Barros (coordinator), Jorge Lamzarotti, Fernando del Campo, Juan Pablo Illanes, Nicolas Bersano Juan Pablo Serra
Contact Mario Leyton

MOAIS/Kaapi (France)

Organization Institut d'Informatique et de Mathematique Appliquees de Grenoble (IMAG)
Members Thierry Gautier (coordinator), Serge Guelton, Xavier Besseron, Vincent Danjean

Eight Samourai (Japan)

Organization Departement of Information and Communication Engineering University of Tokyo
Members Hideo Saito (coordinator), Ken Hironaka, Shogo Sawai, Yu Watanabe, Takshi Sekiya, Kei Takahashi, Tatsuya Shirai, Kenjiro Taura

VU (Netherlands)

Organization Vrije University
Members Jason Maassen (coordinator), Thilo Kielmann, Rob van Nieuwpoort, Niels Drost

PUTat3am POZNAN (Poland)

Organization Poznan University of Technology
Members Mariusz Mamonski, Grzegorz Pawlak, Filip Gorski, Pawel Marciniak, Maciej Plaza, Stanislaw Stempin

OUTPUT POZNAN (Poland)

Organization Poznan University of Technology
Members Pawl Lichocki, Mariusz Mamonski, Szymon Wasik, Krysztof Witkowski, Grzegorz Pawlak

3.2. FlowShop Problem

Each team had one hour to run their application on the Grid. During this period, they were expected to solve Taillard's instances of the FlowShop problem. 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 one. If more than one team solves the same problem in the same amount of time, the final criteria for deciding the winner is the number of workers (numbers of cores) used.

3.2.1. Teams

BUPT (China)

Organization Beijing University of Posts and Telecommunications (BUPT)
Members Huang Xiaohong (corrdinator), Ma Yan, Su Yujie, Wu Yongjuan, Du Nan, Han Yunan, Sun Zheng, Yang Fan, Ding Toa, Jiang Lili, Xiao Qiangju, Liu Benjin, Chen Zhihui, Li Feiyun
Contact Yongwei Wu

Kanban System (Japan)

Organization Departement of Information and Communication Engineering University of Tokyo

Members Hideo Saito (coordinator), Ken Hironaka, Shogo Sawai, Yu Watanabe, Takshi Sekiya, Kei Takahashi, Tatsuya Shirai, Kenjiro Taura

UTat3am POZNAN (Poland)

Organization Poznan University of Technology
Members Mariusz Mamonski, Grzegorz Pawlak, Filip Gorski, Pawel Marciniak, Maciej Plaza, Stanislaw Stempin

OUTPUT POZNAN (Poland)

Organization Poznan University of Technology
Members Pawl Lichocki, Mariusz Mamonski, Szymon Wasik, Krysztof Witkowski, Grzegorz Pawlak

3.3. Local ETSI Contest Machines Configuration

For the contests 25 machines were installed and configured by the OASIS Team. 24 of them were workstation used by the teams and the other one was acting as NFS server. Two machines were allocated to each team. They had to be used to start and develop their application on the grid. Users accounts, deployment descriptors, documentation and shell scripts were shared by the NFS server.

Following software have been installed on the workstation:

- Fedora Core 6
- Eclipse / Netbeans
- Sun's JDK 1.5.0_08

4. Results

4.1. N-Queens Contests Results

These results are taken from the ETSI 3rd Grid Plugtests N-Queens Challenge Results report[NQueens Report]. The contests results are as follows:

- The 3rd ProActive Prize winner is Vrije University. They calculated N=22 Queens in 27 minutes.
- The 2nd ProActive Prize winner is ex-aequo BUPT and FIT with ~5 000 Billions solutions found on ~680 workers.
- The 1st ProActive Prize winner is Eight Samourai with ~6 467 Billions solutions found deployed on 2193 workers.
- The "Prix special du Jury" is MOAIS/Kaapi. They calculated 8 times N=22 Queens (~21 528 Billions solutions in 4600s on 1348 workers). They also computed N=22 Queens in 488s, and N=23 Queens in 4415s. N=23 Queens represent about 24233 Billions solutions.

Note that:

- No team could be awarded more than one place.
- MOAIS/Kaapi has not won the contest because they asked to be able to use SSH to start their job.

4.2. FlowShop Contests Results

These results are taken from the ETSI 3rd Grid Plugtests FlowShop Challenge Results report[Flowshop Report]. The results are detailed as follows:

- The winner is Kanban System: 553s, 207 workers

4.3. 1st, 2nd and 3rd Grid Plugtests Comparisons

	2004	2005	2006
Plugtests number of participants	80	240	240
Plugtests number of events	3	13	15
Grid: number of involved countries	12	13	8
Grid: number of sites	20	40	22
Grid: number of CPUs	800	2700	4130
Grid: GFlops	~100	~450	~1700
Hierarchical Deployment support	No	Yes	Yes
File Transfer support	No	Yes	Yes
Number of contests	1	2	2
Number of teams	6	8	14
Contestant max CPU used for successful computation	560	1106	2193
Contestant max CPU deployed	800	2111	2193
Contestant Max N-Queens # solutions found	~800 Billions	~2 202 Billions	~24 233 Billions

Table 4.1. 1st and 2nd Grid Plugtests Comparison Summary Chart

4.3.1. Grid Heterogeneousness and Interoperability.

The Grid gathered for the 3rd Grid Plugtests was heterogeneous in many levels: Computer Architecture, Operating Systems, JVM, Deployment Protocols and Network Configuration. The diversity of resources is detailed as follows:

- Computer Architectures: x86, ia64, AMD64, EMT64, PPC, Sparc
- Operating Systems: Linux, OS X
- Java Virtual Machines: Sun, BEA, Apple
- Deployment Protocols: cluster-fork, Glite, LSF, OAR, PBS, Prun, SGE, SSH, Torque
- 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.

5. Conclusions

The 3rd Grid Plugtests, co-organized by INRIA and ETSI, pleased all the participants. It was an event useful for the Grid community: users, developers and system administrators. The Conferences and Workshops helped the community to exchange their views, objectives, difficulties and user experiences 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 features, while improving the scalability of the middleware. 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 optimize ProActive to allow big deployments. The feedback provided by the Grid Plugtests, will help the OASIS Team to improve the ProActive middleware. From the contestants point of view, it has been clear that the Grid can help them to run their computation faster.

Given the positive experience of the event, we will organize a 4th version that will be co-located in France and China. This 4th edition promise to raise new interesting challenges; a bigger grid, two basements, high latency to reach the biggest clusters etc. Don't forget to come to the 4th edition of the Grid Plugtests !

Bibliography

[sites ranking] Sites Providers Core Ranking. Providers http://www-sop.inria.fr/oasis/plugtest2006/Providers.html

[grid arch] Grid Architecture Summary Chart. Machines clusters http://www-sop.inria.fr/oasis/plugtest2006/Machines_Clusters.html

[1st Plug report] 1st Grid Plugtests Report. 2004 http://www.etsi.org/plugtests/History/

[2nd Plug report] 2nd Grid Plugtests Report. 2005 http://www.etsi.org/plugtests/History/

[Kaapi] Kaapi homepage. Kaapi https://gforge.inria.fr/projects/kaapi/

[Taktuk] Taktuk homepage. Taktuk http://taktuk.gforge.inria.fr/

[ProActive] ProActive homepage. ProActive http://www-sop.inria.fr/oasis/ProActive/

[Torque] Torque homepage. Torque http://www.clusterresources.com/pages/products/torque-resource-manager.php

[Flowshop] Talbi FlowShop Problem Challenges. Talbi Challenge http://www.ifl.fr/~talbi/

[p2p] Clement Mathieu, Denis Caromel, Alexandre di Costanzo. Peer-to-peer for computational grids: Mixing clusters and desktop machines. *Parallel Computing Journal on Large Scale Grid*, 2007. To appear.

[Contacts] Involved Sites Technical Contacts:http://www-sop.inria.fr/oasis/plugtest2006/technical_List.html

[NQueens Report] N-Queens Contests Jury Report:http://portal.etsi.org/docbox/GRID/Open/GRID%20Plugtests%202006/2006%20N-Queens%20Contests%20Jury%20Report%20v11.doc

[Flowshop Report] Flowshop Contests Jury Report:http://www-id.imag.fr/Laboratoire/Membres/Gautier_Thierry/Perso/TG/KAAPI%20winner%20of%20plugtest%202006_files/3rd%20ETSI%20GRID%20Plugtests%20N-Queens%20FlowShop%20results%20v2-5.pdf

A. Involved Sites Technical Contacts

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

Australia	UNIVERSITY OF MELBOURNE Rajkumar Buyya <raj@cs.mu.OZ.AU> Srikumar Venugopal <srikumar@cs.mu.OZ.AU> Chee Shin YEO <csyeo@csse.unimelb.edu.au>
Chile	UDP Tomas Barros <tomas.barros@inf.udp.cl>
China	BUPT MA Yan <mayan@bupt.edu.cn> Xiaohong Huang <huangxh@buptnet.edu.cn>
China	CNGRID-SCCNET Jiang Kai <kjiang@ssc.net.cn> <ydgui@ssc.net.cn>
China	CNGRID-SCCAS niutsccas <niut@sccas.cn> <haili@sccas.cn> <zhupeng@sccas.cn>
China	Tsinghua Yongwei Wu <wuyw@tsinghua.edu.cn> <zws@mail.tsinghua.edu.cn>
France	GRID5000-BORDEAUX Aurelien Dumez <aurelien.dumez@labri.fr>
France	GRID5000-GRENOBLE Nicolas Capit <<nicolas.capit@imag.fr>
France	GRID5000-LYON Stephane D'Alu <sdalu@ens-lyon.fr>
France	GRID5000-ORSAY Philippe Marty <philippe.marty@lri.fr>,
France	GRID5000-RENNES Guillaume Mornet <gmornet@irisa.fr>, David Margery <David.Margery@irisa.fr>
France	GRID5000-SOPHIA Nicolas Niclausse <Nicolas.Niclausse@sophia.inria.fr>
France	GRID5000-TOULOUSE Celine Juan <cjuan@cict.fr>,
France	IDRIS-DEISA

	Vincent Ribaillier <vincent.ribaillier@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	OCA/U. Nice Sophia Alain Miniussi <alain.miniussi@oca.eu> Marie-Laure Miniussi <Marie-Laure.Miniussi@oca.eu>
France	SUPELEC Stephane Vialle <vialle@metz.supelec.fr>, Patrick Mercier <Patrick.Mercier@supelec.fr>
Italy	BENEVENTO Eugenio Zimeo <zimeo@unisannio.it>, Nadia Ranaldo <ranaldo@unisannio.it>
Italy	UNIVERSITY OF PISA Marco Danelutto <marcod@di.unipi.it>, Pietro Vitale <vitale@di.unipi.it> Massimo Coppola <coppola@di.unipi.it>
Japan	UNIVERSITY OF TOKYO Ken Hironaka <kenny@logos.ic.i.u-tokyo.ac.jp>
Netherland	VRIEJ UNIVERISTY Kees Verstoep <versto@cs.vu.nl>, Henri Bal <bal@cs.vu.nl>,
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>