Grid Computing and NorduGrid

Creating Grid infrastructure in Nordic countries

Mattias Ellert, Uppsala
Aleksandr Konstantinov, Oslo
Balázs Kónya, Lund
Oxana Smirnova, Lund
Anders Wäänänen, Copenhagen
The term **the Grid** appeared in the mid1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. Over the past years numerous protocols, services, and tools were produced to address the problem of

- intercommunication of resources,
- information discovery,
- resource management,
- secure remote access to resources.

Now **the Grid** refers to an infrastructure that enables the integrated, collaborative use of computing resources, networks, databases, data storage and scientific instruments owned and managed by multiple organizations. Grid applications often involve large amounts of data and/or computing and often require secure resource sharing across organizational boundaries.
Grid – first steps (cont.)

Time starts

Grid is here

Development of ideas, base tools

Predecessor implementations

Production implementations

First usage, feedback

Production usage

Widely used

Globus Toolkit™

cluster management software:
Condor, PBS, LSF, ...
remote access protocols
Internet protocols,
Specific implementations

EU Data Grid
EUROGRID
US ATLAS Grid
NorduGrid
...

ATLAS Data Challenge,
???
The objective is to build the next generation computing infrastructure providing intensive computation and analysis of shared large-scale databases, from hundreds of TeraBytes to Petabytes.

The Globus Project is developing fundamental technologies needed to build computational grids.

PPDG will develop, acquire and deliver vitally needed Grid-enabled tools for data-intensive requirements of particle and nuclear physics.
Globus Toolkit™ collection of protocols, libraries and services developed at Argonne National Laboratory and University of Southern California.

Why Globus?
- The only widely accepted *de-facto* standard for Grid computing.
- Mostly robust.
- Security built-in.
- Continuously evolving (sometimes too fast).

Why not Globus only?
- Missing **high-level** services:
  - grid-level scheduler
  - job data stage-in/stage-out
  - user-friendly grid entry points (*simple user-interface, web portals, etc.*)
  - grid-level authorization system
  - grid-level accounting and quotas
Future of Globus

Open Grid Services Architecture (Alpha scheduled for 2003)

- Protocol unification
- Added missing functionality
- Virtualization of end system
- ...

"The Open Grid Services Architecture (OGSA) is a proposed evolution of the current Globus Toolkit towards a Grid system architecture based on an integration of Grid and Web services concepts and technologies. Initial proposed technical specifications have been developed by the Globus Project and IBM, and are being put forward at the Global Grid Forum for discussion, refinement, and (we hope) eventual standardization."
**NorduGrid – Introduction**

- Launched in spring 2001, with the aim of creating a Grid infrastructure in the Nordic countries for interdisciplinary feasibility studies.
- Partners from **Denmark, Norway, Sweden, and Finland**.
- Powered mainly by ATLAS groups (Lund, Copenhagen, Stockholm, Uppsala, Oslo).
- Relies on very limited human resources (3 full-time researchers, few part-time ones) with funding from **NorduNet2**.
NorduGrid – History and Present

**Hardware resources available:**
- 4 dedicated small computer clusters – Copenhagen, Lund, Uppsala Oslo and Bergen
- 6 clusters are shared with other projects/users (number continuously increasing).

**Activities**

**04 2001** Gathering information and testing available software.

**06 2001** Evaluation of software being developed. Choosing base tools for future development. Testbed creation.

**02 2002** Definition of implementation requirements, development of architecture and definition of the tools for future toolkit.

**04 2002** First implementation and testing

**06 2002** Applying developed system to real tasks (ATLAS DataChallenge1, attracting miscellaneous users)

**10 2002** Refinement, extension or redefinition of architecture and main concepts based on experience obtained.
Grid Manager

*Uses* following parts of Globus Toolkit™

- **GridFTP** – fast and reliable data access for Grid
- **GASS Copy** interface – support for different data access protocols
- **Replica Catalog** – metadata storage
- **GRAM** – resource request (obsolete)
- **RSL** – *expandable* Resource Specification Language

*Adds*

- Data stage–in(stage–out + caching
- Job environment control
- Job and data submission through single interface
- Direct support for Globus Replica Catalog.
- E–mail notification of job status changes.
Information System

*Uses*
- MDS based, hierarchically distributed Information System (probably first fully implemented such system on real testbed).
- MDS hierarchy with dynamic site registrations

*Adds*
- new information model for clusters, queues, jobs, users, SE, RC with user-dependent view
- Efficient information providers (backends for collecting data)
- all the job monitoring, resource discovery, status monitoring and brokering are exclusively built on top of the MDS (unified source of information).
User Interface

*Uses*
- RSL for describing job.
- GridFTP and GRAM(obsolete) for job and data submission.

*Provides*
- xRSL – new attributes for RSL to allow more complex jobs.
- Multiple and conditional job submission with xRSL transformation.
- Decision making system (Resource Broker) to choose most appropriate computing resource for job.

Set of easy-to-use utilities to perform all necessary tasks for job submission, execution tracking, result retrieval, etc.

- `ngsub` – submission
- `ngstat` – status of jobs and clusters
- `ngcat` – track execution
- `ngget` – retrieve the result
- `ngkill` – kill running job
- `ngclean` – delete a job
- `ngsync` – to recreate local information about jobs
NorduGrid – Implementation

Virtual Organization and Certification Authority

- NorduGrid **CA** was put into operation in May 2001 in Copenhagen to provide Trusted Public Key Infrastructure
  - a user possesses a private key and a certificate signed by CA
  - sites has copies of public key and use them to authenticate users
  - NorduGrid CA is responsible for providing certificates for scientists from Denmark, Sweden, Norway and Finland.
- User access control is based on Distinguished Name of the user.
- List of the users is managed at **VO** database.
  - users are added by VO administrators
  - list of VO users is used to establish access control at sites
- Sites can apply **local policies and fine–grain access control** by filtering VO users.
NorduGrid – Implementation

Configuration and Packaging

- NorduGrid Toolkit come packaged in RPMs
  - easy to install
  - quick to start
  - direct access to development source code is available too
- Server part has centralized common configuration (2 configuration files)
  - no need to know all the configuration files for the individual components
    (Globus MDS has plenty of them)
  - easy to manage
  - quick to start
  - same syntax everywhere
NorduGrid – Implementation

Load Monitor

Provides

- Nice web interface to NorduGrid Information System and VO.
- Gives both statistical and detailed information.
It is possible to run any application with predefined set of input and output data

- From as simple as "Hello World"

```sh
gensub '(& (executable=/bin/echo) (arguments="Hello World") (stdout=out.txt))
```
to as difficult as Atlas Data Challenge

```bash
&(executable="ds2003.sh") (arguments="1" "10000"
"0") (stdout="dc1.002003.simul.00001.hlt.pythia_jet_11.log")
(join="yes") (maxCPUPtime=1200) (maxDisk=1200) (ftpThreads=10)
(rsl_substitution=("RCINP" "rc://@grid.uio.no/1c=Dataset_2003,rc=Nordugrid,dc=nordugrid,dc=org"))
(rsl_substitution=("RCOUT" "rc://@grid.uio.no/1c=Output_2003,rc=Nordugrid,dc=nordugrid,dc=org/2003"))
(inputfiles=
  (dc1.002003.evgen.0001.hlt.pythia_jet_11.root $(RCINP)/dc1.002003.evgen.0001.hlt.pythia_jet_11.root)
)
(outputFiles=
  (dc1.002003.simul.00001.hlt.pythia_jet_11.log $(RCOUT)/log/dc1.002003.simul.00001.hlt.pythia_jet_11.log)
  (dc1.002003.simul.00001.hlt.pythia_jet_11.zebra
   $(RCOUT)/zebra/dc1.002003.simul.00001.hlt.pythia_jet_11.zebra)
  (dc1.002003.simul.00001.hlt.pythia_jet_11.his $(RCOUT)/his/dc1.002003.simul.00001.hlt.pythia_jet_11.his)
  (dc1.002003.simul.00001.hlt.pythia_jet_11.AMI $(RCOUT)/ami/dc1.002003.simul.00001.hlt.pythia_jet_11.AMI)
  (dc1.002003.simul.00001.hlt.pythia_jet_11.MAG $(RCOUT)/mag/dc1.002003.simul.00001.hlt.pythia_jet_11.MAG)
)
(jobname="dc1.002003.simul.00001.hlt.pythia_jet_11")
(runTimeEnvironment="ATLAS-3.2.1") (runTimeEnvironment="DS2003")
```
NorduGrid – Atlas DC1

Input:
- dataset 2000 (15 partitions)
- dataset 2003 (whole)
- total of ca 200 GB were distributed around the clusters.

Output:
- dataset 2000 – 161497 events
- dataset 2003 – 125799 events
- total of ca 765 GB are kept at the dedicated storage area at dc1.uio.no

Processing time:
- ca 530 CPU–days in total (CPU speeds 700 Mhz – 1500 MHz)

Problems:
- Atlas software problems (mostly "ZEBRA banks screwed up")
- NorduGrid software problems (DC1 was very useful for fixing bugs)
- Globus software (still not 100% stable, as expected)
## NorduGrid – Atlas DC1

### Processed Events per Site

<table>
<thead>
<tr>
<th>Dataset 2000</th>
<th>Dataset 2003</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bergen</strong></td>
<td>24359 49121</td>
</tr>
<tr>
<td><strong>Grendel, Uppsala</strong></td>
<td>1261 6845</td>
</tr>
<tr>
<td><strong>Ingvar</strong></td>
<td>– 18411</td>
</tr>
<tr>
<td><strong>ISV</strong></td>
<td>6261 3329</td>
</tr>
<tr>
<td><strong>Lund</strong></td>
<td>9657 15660</td>
</tr>
<tr>
<td><strong>LSCF, Copenhagen</strong></td>
<td>72558 103688</td>
</tr>
<tr>
<td><strong>NBI, Copenhagen</strong></td>
<td>– 1823</td>
</tr>
<tr>
<td><strong>Oslo</strong></td>
<td>9670 4193</td>
</tr>
</tbody>
</table>

No major problems in Grid part were detected during DC1.
The *minimal* functioning environment for Grid computing is established.

Additional layer of tools/services were developed to provide required infrastructure.

A lot of things to do:
- Closer to **DataGrid** and runtime data handling.
- Accounting.
- Better support for different LRMS.
- Enhanced Information System – more stability, access control, better and richer information providers etc.
- ...
- Integration with other projects (EDG, participation in GGF).