



*NORDIC TESTBED FOR WIDE AREA
COMPUTING AND DATA HANDLING*

2003-01-15

**NORDUGRID – THE NORDIC TESTBED
FOR WIDE AREA COMPUTING AND DATA HANDLING**
FINAL REPORT

1. INTRODUCTION

The purpose of the NorduGrid project is to create the Grid computing infrastructure in the Nordic countries using the software tools provided by Grid R&D projects worldwide. During the evaluation of such tools by this project, it became clear that none satisfies the requirements set by the project participants. Therefore it was decided to develop an original and innovative set of tools in order to achieve the goal of enabling a production-level Grid services in the Nordic countries.

Project participants include universities and research centers in Denmark, Sweden, Finland and Norway. The active phase of the project started in May 2001, and involves the Niels Bohr Institute (Denmark), Lund and Uppsala Universities (Sweden), University of Oslo (Norway) and Helsinki Institute of Physics (Finland). National supercomputer centers in Norway (Parallab) and Sweden (NSC, HPC2N) entered the project by setting up their resources with the NorduGrid middleware.

From the very beginning, the NorduGrid testbed cooperated with the EU DataGrid project, initiated at CERN and aimed at creation of the computing infrastructure for the future High Energy Physics experiments. The two projects have to a large extent user base, and while relying on different architectural approaches, use the same basic tools.

Successful deployment of the NorduGrid Testbed, together with the robust and reliable performance of the NorduGrid Toolkit, allowed researchers in Nordic countries to use the Grid computing in their everyday work. One of the biggest success was participation in a real challenging tasks within the framework of the international High Energy Physics experiment ATLAS. Several other analyses and studies were done with the help of the NorduGrid facilities.

This document summarizes the achievements and gives the final status of the NorduGrid project. In the following sections, the description of the project organisation and deployed resources is given, followed by the summary of the developed tools and executed use cases.

2. ORGANIZATION

Due to the considerable geographical spread of the project member sites, the organization and management heavily relies on the Internet and teleconferences. The project Web site (<http://www.nordugrid.org>) opened in December 2000, and is regularly updated with the most detailed information. This Web site serves not only as a project reference tool, but also for dynamic monitoring of the testbed status and as a portal to the software and middleware repositories, databases and other services.

2.1. RESPONSIBILITIES

a) Steering Panel

Prior to the project's start, the steering panel was appointed to provide adequate management. It consisted of the project coordinator, John Renner Hansen, representing Niels Bohr Institute, Denmark; Tord Ekelöf, Paula Eerola and Sten Hellman, representing Uppsala, Lund and Stockholm Universities of Sweden; Farid Ould-Saada from the University of Oslo, Norway, and Jorma Tuominiemi, representing Helsinki Institute of Physics, Finland. The group holded monthly phone

meetings.

b) New positions

Three new postdoctoral positions, funded by the project, were created. Each employed researcher carried responsibility for one of the NorduGrid development nodes: in Lund, Uppsala and Oslo, and provided support and assistance to other sites in the area. Positions were announced in December 2000, and chosen by the steering panel were:

Name	Location	Start
Dr. Balázs Kónya	Lund University	2001-05-01
Dr. Mattias Ellert	Uppsala University	2001-05-15
Dr. Aleksandr Konstantinov	University of Oslo	2001-06-20

Table 1. New positions

These researchers carried out most of the project's activities, including set up and maintenance of the testbed sites, evaluation, middleware development, and user support.

c) Technical Working Group

For the fast evaluation of existing demands and problems, a technical working group was set up. It consisted of the three abovementioned postdocs and the researchers, responsible for the cooperation with the EU DataGrid (see Section 5): Anders Wäänänen (Testbed and Demonstrators Work Package), and Oxana Smirnova (HEP Applications Work Package).

Date	Location
June 21-23, 2001	Copenhagen
August 21, 2001	Lund
October 31 - November 1, 2001	Oslo
January 15-16, 2002	Uppsala
February 5, 2002	<i>Phone call</i>
March 9-11, 2002	Geneva
May 7-9, 2002	Copenhagen
July 4-5, 2002	Lund
October 1-2, 2002	Oslo
November 11-13, 2002	Uppsala

Table 2. Technical working group meetings

The group was convening bi-monthly at rotating locations across the Nordic countries

to assess the immediate directions of the project technical development, discuss the status of the project, evaluate other available middleware tools, discuss various related problems, and provide an immediate short-term planning.

2.2. GENERAL WORKSHOPS

Apart of the steering panel and the technical working group meetings, four general workshops have been arranged.

Date	Location
February 5-6, 2001	Lund
November 1-2, 2001	Oslo
May 23-24, 2002	Helsinki
November 11-13, 2002	Uppsala

Table 3. NorduGrid Workshops

The inaugural 1st NorduGrid Workshop took place in Lund, February 5-6, 2001. Present were coordinators of Grid-related activities in participating institutions, and researchers, otherwise involved in the project. The purpose of the workshop was the general presentation of the project and its planned activities, review of the situation in the Nordic countries, and discussion of organizational issues.

The 2nd Workshop took place in Oslo, November 1 to 2, 2001. Major issues were first hands-on experience with installation and usage of the hardware and existing Grid middleware, demonstrations of the Globus Toolkit™ installation. Also discussed were the ways of cooperation with the national computing centers and plans towards using the NorduGrid Testbed for research applications, such as ATLAS Data Challenges (see Section 7).

The 3rd Workshop in Helsinki was dedicated to the first test release of the NorduGrid toolkit. Apart of the detailed presentations of all the aspects of the NorduGrid architecture and technical solutions, a live demonstration of the operational NorduGrid testbed was shown, using few generic test applications, as well as the real example of the research task.

The 4th Workshop in Uppsala summarized the achievements of the project, demonstrating the maturity of the NorduGrid middleware and stability of the testbed, which reached a production level and included not only test clusters, but big production farms of the national computing centers. Issues of cooperation with other international Grid projects and plans for the future development were addressed as well.

All the Workshops gathered not only immediate NorduGrid project participants, but also local researchers and several invited speakers from other related projects.

3. RESOURCES

The NorduGrid environment consists of five core sites, mostly dedicated to development and testing of the middleware, and several production clusters, scattered across the Nordic countries. The whole project infrastructure relies on the existing networks in the Nordic countries. The connectivity chart for the participating sites is shown in Figure 1

From the project start, part of existing computing resources was assigned for the test installations. The LSCF cluster at NBI (ca. 50 heterogeneous processors) was used for initial middleware and resource management tests. In Lund, a mini-farm (2 PC's) was set up at the Particle Physics department. In Uppsala University and Helsinki Institute of Physics, a stand-alone PCs were originally used for initial tests.

In addition to existing resources, five new dedicated development clusters were set up: in Lund, Uppsala, Oslo, Bergen and Copenhagen. Lund and Uppsala clusters upgraded and replaced previously used local test facilities.



Figure 1. Connectivity map of the NorduGrid participating sites

Computing resources are fairly heterogeneous, hardware- and software-wise. Most of the processors are various types of Intel Pentium. Operating systems are various flavors of Linux: different distributions of Red Hat, Mandrake, Slackware and Debian.

The core sites run small test clusters (a few CPUs) and comprised the initial testbed, put up in May 2002. Since this testbed was shown to be stable, reliable and capable of executing production-level tasks, more sites have joined with large production clusters. sites have joined with large production clusters. At the time of writing, they are: LSCF in Copenhagen, *Grendel* in Uppsala, *Ingvar* in Linköping, Para//ab in Bergen and *Seth* in Umeå. Several other clusters in Sweden and Denmark considering joining the Grid are evaluating the NorduGrid middleware, as well as sites in other

countries, such as Canada and Japan.

	<i>Location</i>	<i>Type</i>
<i>Test clusters</i>	Uppsala University	Linux PC cluster
	Lund University	Linux PC cluster
	University of Oslo	Linux PC cluster
	University of Bergen	Linux PC cluster
	NBI Copenhagen	Linux PC cluster
<i>Production farms</i>	LSCF, NBI Copenhagen	Linux PC farm
	Para//ab, Bergen	Linux PC farm
	NSC <i>Grendel</i> , Uppsala University	Linux PC farm
	NSC <i>Ingvar</i> , Linköping	Beowulf cluster
	SCFAB, Stockholm	Linux PC farm
	<i>Seth</i> , Umeå	Linux PC super cluster
<i>Other sites</i>	<i>Sleipner</i> , Lund University	Linux PC cluster
	HIP, Helsinki	Linux PC node
	NORDITA, Copenhagen	Linux PC cluster

Table 4. NorduGrid Testbed sites

Table 4 summarises resources at the disposal of the NorduGrid project (test cluster and production farms), as well as lists some other sites which participate in the project activities.

The 5 core clusters were acquired and set up by the NorduGrid project itself. Detailed specifications for each cluster are shown below.

3.1. UPPSALA CLUSTER

Name	grid.tsl.uu.se	Contact person	Mattias Ellert	grid.support@tsl.uu.se	
Address	Uppsala University, Department of Radiation Sciences, Uppsala, Sweden				
HARDWARE					
Nodes	<i>Quantity</i>	<i>CPU</i>	<i>RAM</i>	<i>Disks</i>	<i>Other notes</i>
	1	P-III 866 Mhz	512 MB	40 GB + 60 GB	Gatekeeper
	2	2 × P-III 866 Mhz	512 MB	40 GB	Computing nodes
Network	100 Mbit/s				
Mass storage	None				
SOFTWARE					
OS	RedHat Linux 7.1				
	Kernel 2.4.9-13 on the gatekeeper, kernel 2.4.9-13smp on the computing nodes glibc 2.2.4-19				

Resource manager	OpenPBS 2.3.2
File system	/home and /usr/local shared through NFS
Databases	None
MIDDLEWARE	
Gatekeeper	/O=Grid/O=NorduGrid/CN=grid.tsl.uu.se/jobmanager, /O=Grid/O=NorduGrid/CN=grid.tsl.uu.se/jobmanager-pbs
Grid	Globus Toolkit, NorduGrid
NETWORK	
Configuration	Private network

Table 5. Uppsala cluster specifications

3.2. LUND CLUSTER

Name	Lund Grid Cluster	Contact person	Balázs Kónya	grid.support@grid.quark.lu.se	
Address	Elementary Particle Physics Department of Lund University				
HARDWARE					
Nodes	<i>Quantity</i>	<i>CPU</i>	<i>RAM</i>	<i>Disks</i>	<i>Other notes</i>
	1	PIII (Coppermine) 1GHz 256KB cache	512 MB	40 & 60 GB	Front-end machine, Powerware 5115 UPS
	2	Dual PIII (Coppermine) 1GHz 256KB cache	512 MB	30 GB	Dual processor computing nodes
	1	PIII (Coppermine) 1GHz 256KB cache	256 MB	16 GB	Single processor computing node
Network	100Mb/s private network with a 3Com OfficeConnect 16 100MB Switch				
Mass storage	None				
SOFTWARE					
OS	Linux Mandrake 8.0 distribution, kernel-2.4.3				
Resource manager	OpenPBS 2.3.12				
File system	All the nodes have local disks and share the NFS mounted /scratch and /home area of the front-end machine				
Databases	None				
MIDDLEWARE					
Gatekeeper	grid.quark.lu.se 2119 port				
Grid	Globus Toolkit, NorduGrid				
NETWORK					
Configuration	All the nodes are on a closed private network behind the front-end machine.				

Table 6. Lund cluster specifications

3.3. OSLO CLUSTER

Name	Oslo University NordGrid gatekeeper	Contact person	Aleksandr Konstantinov	grid.sysadmin@fys.uio.no grid.support@fys.uio.no	
Address	University of Oslo, Department of Physics				
HARDWARE					
Nodes	<i>Quantity</i>	<i>CPU</i>	<i>RAM</i>	<i>Disks</i>	<i>Other notes</i>
	1	2 x Intel PIII 1 GHz	256MB	39266MB	computing node
	1	Intel PIII 1 GHz	256MB	39266MB	computing node
	1	Intel PIII 870 MHz	128MB	2 x 41174MB	gatekeeper
Network	100Mbps ethernet cards EtherExpress Pro100				
Mass storage	2 x 1 TB disk arrays				
SOFTWARE					
OS	Gatekeeper - distribution: RedHat 7.1 , kernel: 2.4.2-2 (i686) , libc: 2.2.2				
	Nodes - distribution: Slackware 7.1 , kernel: 2.2.19 (i686) (SMP & UP) , libc: 2.2.3				
Resource manager	OpenPBS 2.3.12				
File system	ext2fs				
Databases	None				
MIDDLEWARE					
Gatekeeper	grid.uio.no 2119				
Grid	Globus Toolkit, NordGrid				
NETWORK					
Configuration	Nodes are situated on private physically isolated network with 100Mbps connection through Allied Telesyn FS708 network switch.				

Table 7. Oslo cluster specifications

3.4. BERGEN CLUSTER

Name	Bergen University NordGrid gatekeeper	Contact person	Aleksandr Konstantinov	grid.sysadmin@fys.uio.no grid.support@fys.uio.no	
Address	University of Bergen, Department of Physics				
HARDWARE					
Nodes	<i>Quantity</i>	<i>CPU</i>	<i>RAM</i>	<i>Disks</i>	<i>Other notes</i>
	2	2 x Intel PIII 1 GHz	1024MB	41174MB	computing node
	1	Intel Celeron 800 MHz	512MB	2 x 41174MB	gatekeeper
Network	100Mbps ethernet cards EtherExpress Pro100				
Mass storage	None				
SOFTWARE					
OS	Gatekeeper - distribution: RedHat 7.1 , kernel: 2.4.7-10 (i686) , libc: 2.2.4				
	Nodes - distribution: Slackware 7.1 , kernel: 2.4.14 (i686) (SMP) , libc: 2.2.3				
Resource manager	OpenPBS 2.3.12				
File system	Nodes: ext2fs, gatekeeper: ext3fs				

Databases	NA
MIDDLEWARE	
Gatekeeper	grid.uio.no 2119
Grid	Globus Toolkit, NorduGrid
NETWORK	
Configuration	Nodes are situated on private physically isolated network with 100Mbps connection through network switch.

Table 8. Bergen cluster specifications

3.5. COPENHAGEN CLUSTER

Name	Niels Bohr Institute Nordugrid gatekeeper	Contact person	Anders Waananen	grid.sysadmin@nbi.dk grid.support@nbi.dk	
Address	Niels Bohr Institute, Copenhagen				
HARDWARE					
Nodes	<i>Quantity</i>	<i>CPU</i>	<i>RAM</i>	<i>Disks</i>	<i>Other notes</i>
	3	2 x AMD 1.2 GHz	512MB	41174MB	Computing node
	1	Intel 1GHz	512MB	2 x 41174MB	gatekeeper
Network	100Mbps ethernet cards EtherExpress Pro100				
Mass storage	None				
SOFTWARE					
OS	Gatekeeper - distribution: RedHat 7.2 , kernel: 2.4.9-13 (i686) , libc: 2.2.4 Nodes - distribution: RedHat 7.2, kernel: 2.4.9 -13 (i686) (SMP) , libc: 2.2.4				
Resource manager	OpenPBS 2.3.12				
File system	Ext3, nfs				
Databases	None				
MIDDLEWARE					
Gatekeeper	grid.nbi.dk 2119				
Grid	Globus Toolkit, NorduGrid				
NETWORK					
Configuration	Nodes are situated on private physically isolated network with 100Mbps connection through network switch.				

Table 9. NBI cluster specifications

4. THE NORDUGRID MIDDLEWARE

Evaluation of the functionality of existing Grid solutions, such as the Globus Toolkit™ and the EU DataGrid middleware by the NorduGrid researchers, showed that those tools are either not sufficient or not robust enough for production-level tasks. In February 2002, the NorduGrid steering committee, together with the technical working group, decided to abandon part of the EU DataGrid commitments, and to develop a new, lightweight and scalable, Grid solution for the Nordic countries.

The NorduGrid middleware was consequently designed to be a lightweight, non-invasive and portable solution, suitable for any kind of available resources and requiring minimal intervention on the part of system administrators. This solution allows NorduGrid to embrace all kinds of available resources, providing authorized

users with a widely distributed continuously operational production environment. The NorduGrid is a dynamic, heterogeneous Grid with fluctuating and inflating number of available resources of different kinds.

4.1. ARCHITECTURE

The NorduGrid architecture was carefully planned and designed to satisfy the needs of a generic user, as well as those of a system administrator. The chosen philosophy can be outlined as follows:

- ◆ Avoid architectural single points of failure
- ◆ Resource owners retain full control over their resources
- ◆ Installation details (method, operation system, configuration etc.) should not be dictated
- ◆ As little restriction on the site configuration as possible
 - Computing nodes should not be required to be on the public network
 - Clusters need not be dedicated
 - NorduGrid software should be installed only on a front-end machine
- ◆ NorduGrid software should be able to use the existing system and eventual pre-installed Globus Toolkit™ versions
- ◆ Start with something simple that works and proceed from there

The NorduGrid tools are designed to handle job submission and management, user area management, a minimal necessary data management, and monitoring. Figure 2 depicts the basic components of the NorduGrid architecture and schematic relations between them.

Job submissions in the NorduGrid architecture are performed via User Interfaces, which use the Information System and Replica Catalog information about Storage Elements contents to discover resources (clusters) and select the best ones. More detailed description of these components is given in what follows.

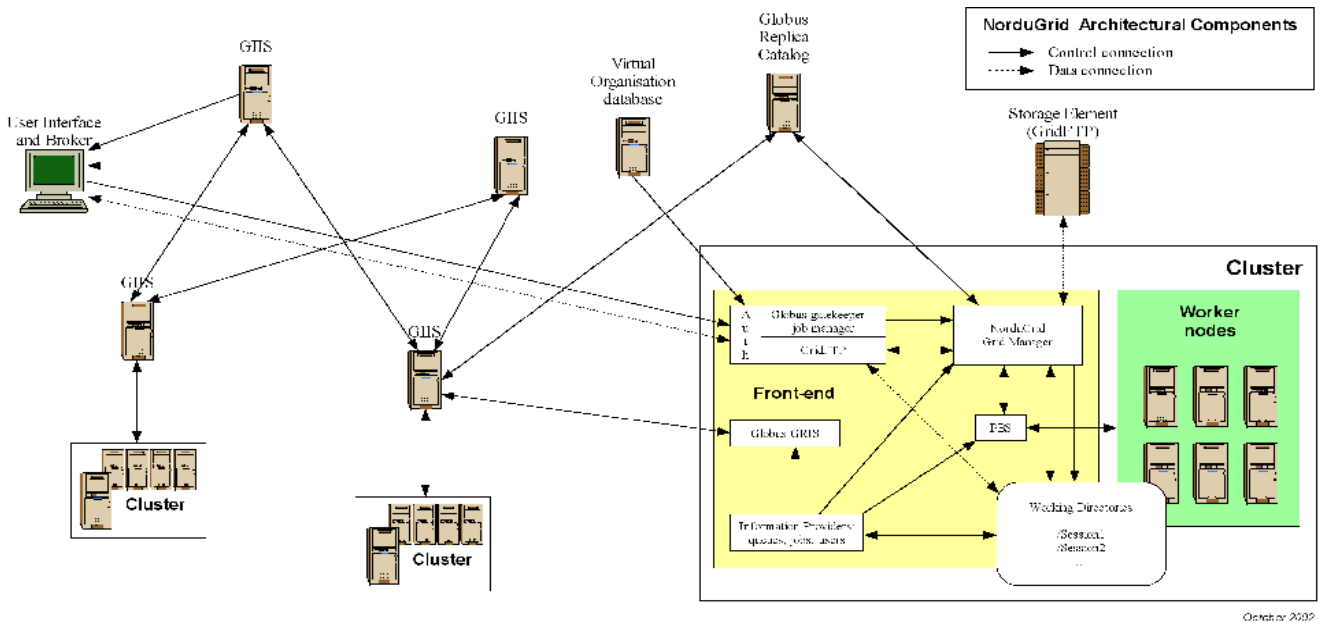


Figure 2. NorduGrid Architectural Components

A) USER INTERFACE

User Interfaces (UI) are among the key components, as they perform the task of the workload management, distributing job requests to resources. The UI comes simply as a client package, which can be installed on any machine - typically, an end-user desktop computer. There can be as many User Interfaces as users find convenient.

B) INFORMATION SYSTEM

Information System in the NorduGrid architecture does not require a dedicated machine, although it is not excluded as an option. Normally, cluster front-ends are capable of providing the necessary Information System services.

C) COMPUTING CLUSTER

A cluster is the natural computing unit in the NorduGrid architecture. It consists of a front-end (master) node which manages several back-end (worker) nodes, normally via a private closed network. The software component currently consists of a standard batch system setup plus extra components to interface with the Grid. The operation system of choice is Linux in its many flavors, however, other Unix-like systems (e.g. HP-UX, Solaris) can be used as well.

The configuration of the batch system is very liberal, respecting local setup and configuration policies. Much effort has been put into ensuring that the NorduGrid tools should not change current installations or impose any restrictions, but rather try to be an add-on component which hooks the local resources onto the Grid and allows for Grid jobs to run along with the conventional jobs.

Front-end, the gatekeeper:

The responsibility of the front-end machine is to handle job requests from the Grid and translate them into requests to the local resource management system (LRMS). This task is shared such that the authentication and authorization of requests is

handled by the gatekeeper, while the Grid Manager takes authorized requests and does the job management.

The gatekeeper is expandable through plugins. It uses the job submission plugin to submit jobs and acts as a normal GridFTP file server.

Once a request from the Grid has been accepted, the control is passed to the Grid Manager. The Grid Manager from now on manages the job, from actually retrieving input files for the job and submitting the job to the LRMS, to cleaning up after the job execution and uploading output files to the user-specified locations.

The LRMS which is normally used in NorduGrid is either OpenPBS or a PBSPro, eventually with e.g. the Maui scheduler. However, there are no specific design limitations as to which batch system to use.

Worker nodes:

The worker node (WN) is the backend of the system, where the execution of an application is performed. WNs are managed entirely through the LRMS, and no NorduGrid middleware has to be installed on them.

An important part of the NorduGrid architecture is that the WNs are not required to have a direct access to the Internet. Thus applications can not expect to be able to retrieve files via e.g. WWW or from global file systems like AFS. This concept is a part of the philosophy that as few restrictions as possible should be imposed on a site.

The only requirement for the cluster setup is that there should be a shared filesystem (e.g. NFS) between the frontend and the back-end nodes. This so-called session directory is then available to the job during the execution and to the Grid Manager for input and output file staging.

D) STORAGE ELEMENT

A concept of a Storage Element (SE) is not fully developed by the NorduGrid at this stage. So far, SEs are implemented as plain GridFTP servers. A software used for this is either the GridFTP server provided as a part of the Globus Toolkit™, or the one delivered as a part of the NorduGrid Grid Manager. The latter is preferred at the moment, since it allows configuring the access based on the user Grid identity (Distinguished Name of a certificate) instead of the local identities to which users are mapped. At present, NorduGrid operates two high-capacity Storage Elements situated at Oslo, Norway, and several test set-ups.

E) REPLICA CATALOG

The NorduGrid project uses the Globus Replica Catalog (RC) developed by the Globus project for locating and registering data sources. The RC implementation in Nordugrid follows closely the one from Globus.

The RC server is based on an OpenLDAP server with an LDBM back-end. There are no significant changes introduced into the original objects schema. Since the applications are not expected to perform heavy concurrent writing to the RC, some advanced features, like rollbacks, are not used.

Unfortunately, the OpenLDAP has appeared to be unstable under heavy load during production runs, and is overall somewhat problematic. One of the problems appeared

as the size of the catalogue grew up. It appeared that the OpenLDAP has problems with transferring relatively big amounts of data over an authenticated/encrypted connection. Some problems were fixed by applying appropriate patches, and some by automatic restart of the server. Together with a fault tolerant behavior of the client part, these fixes made the system usable.

4.2. MIDDLEWARE

The NorduGrid middleware is almost entirely based on the Globus Toolkit™ API, libraries and services. In order to support the NorduGrid architecture, several innovative approaches were used, such as the Grid Manager and the User Interface. Other components were extended and developed further, such as the Information Model and the Extended Resource Specification Language.

A) GRID MANAGER

The NorduGrid Grid Manager (GM) software acts as a smart front-end for job submission to a cluster. It runs on the cluster's front-end and provides job management and data pre- and post-staging functionality, with a support for metadata catalogues.

Since data operations are performed by an additional layer, the data are handled only at the beginning and end of a job. Hence the user is expected to provide complete information about the input and output data. This is the most significant limitation of the used approach.

The GM is based on the Globus Toolkit™ libraries and services. It was written because the services available as parts of the Globus Toolkit™ did not have the required functionality to implement the architecture developed by the NorduGrid project. The following parts of Globus are used:

- ◆ GridFTP: fast and reliable data access for Grid with integrated security,
- ◆ GASS Transfer: support for HTTP(S)-like data access protocol,
- ◆ Replica Catalog: meta-data storage,
- ◆ Replica Location Service, designed by a collaboration between Globus and EDG
- ◆ RSL: extendable Resource Specification Language .

To provide a GridFTP interface for submission of specialized jobs, a more Grid-oriented GridFTP server (GFS) was developed. It has the following features:

- ◆ Virtual directory tree, configured per user.
- ◆ Access control, based on the Distinguished Name stored in the user certificate.
- ◆ Local file system access, implemented through loadable plugins (shared libraries). There are two plugins provided with GFS:
 - The local file access plugin implements an ordinary FTP server-like access,
 - The job submission plugin provides an interface for submission and control of jobs handled by the GM.

The GFS is also used by NorduGrid to create relatively easily configurable GridFTP based storage servers (often called Storage Elements).

The GM accepts job-submission scripts described in Globus Resource Specification Language (RSL). Since RSL only defines a syntax and does not put any restrictions on the used attributes, some additional attributes were introduced to reflect the features of the GM.

For every job, the GM creates a separate directory (the *session directory*) and stores the input files specified in the job submission script into it. There is no single point (machine) that all the jobs have to pass, since the gathering of input data is done directly on a cluster front-end.

Then the GM creates a job execution script and launches it using the LRMS. Such a script can perform additional actions, such as data staging to a computing node, and setting the environment for third-party software packages requested by a job.

After a job has finished, all the specified output files are transferred to their destinations, or are temporarily stored in the session directory to allow users to retrieve them later.

Additionally, the GM can cache input files requested by one job and make them available to another job. If a file is taken from the cache, it is checked (if the protocol allows that) against the remote server containing that file, whether a user is eligible to access it. To save disk space, cached files can be provided to jobs as soft links.

The GM therefore provides the following functionality:

- ◆ Unified interface for job submission and data staging from a user machine and retrieval of results (GridFTP is used).
- ◆ Support for GridFTP, FTP, HTTP or HTTPS-like servers for data staging with a possibility to lookup/register information in meta-data catalogues.
- ◆ Fault tolerant transfers.
- ◆ Caching of downloaded data.
- ◆ Email notification of job status. The job execution log is sent to a user in case a problem was encountered, and can also be provided to a user upon request.
- ◆ Support for software packages runtime environment configuration.
- ◆ Handling of session directories (creation, cleaning, staging to/from a computing node).

The GM is written mostly in C/C++. It uses a bash-script Globus-like implementation of an interface to an LRMS. This makes it easy to adapt the GM to inter-operate with any new LRMS. Currently, the GM comes with the interface to the PBS only.

The GM uses the file system to store the state of handled jobs. This allows it to recover safely from most system faults, after a restart.

The GM also includes user utilities to handle data transfer and registration in meta-data catalogues. These are `ngmove` for transferring/registering data and `ngremove` for removing/unregistering.

B) INFORMATION SYSTEM

The NorduGrid information system is a distributed dynamic system which provides information on the status of the Grid resources. The information is generated on a

resource upon request from a user or an agent when a status query is performed (pull model). The generated information can then be cached at different levels in the distributed database. The implementation consists of local information providers, local databases (first level cache), information indices with caching mechanism (higher level caches), a soft registration mechanism and an information model (schema) for representing the Grid resources. The dynamic soft registration mechanism makes it possible to construct different topologies of information indices and resources. The most typical Grid topology is a tree-like hierarchy. The User Interface with the built-in broker and the monitoring system are the main consumers of the dynamic information produced and stored in the information system.

The design of a Grid information system is always a delicate question of how to represent the Grid resources, what kind of information should be stored, what is the best structure of presenting this information to the Grid users and to the Grid agents (i.e. brokers). In case of an LDAP-based system, these questions have their technical answers in the so-called LDAP schema files. NorduGrid designed its own information model (schema) in order to properly represent and serve its environment. A working information system, as a part of the NorduGrid architecture, has been built and put into operation already in May 2002. While some other information models have been proposed recently (e.g. CIM, Glue), the NorduGrid schema is up to now the only one which has been extensively tested and used in an actual production. The schema develops continuously in order to satisfy the user requirements.

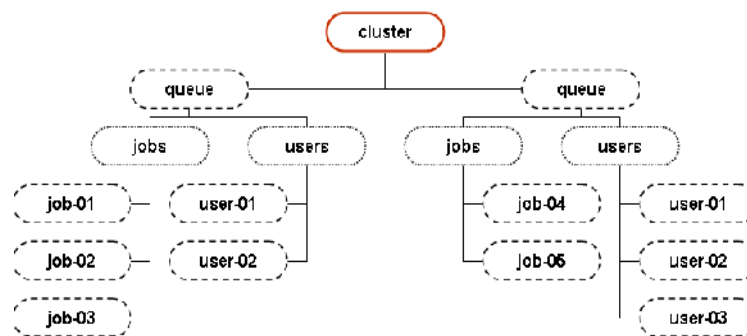


Figure 3. NorduGrid Information System LDAP subtree corresponding to a cluster resource

The information model:

The NorduGrid production environment consists of different resources located at the different sites. The list of implemented Grid services involves the computing resources, Storage Elements and Replica Catalogs. The NorduGrid information model naturally maps these resources onto an LDAP-based tree, where each resource is represented by a subtree. The LDAP-tree, or DIT, of a cluster is shown in Figure 3.

The clusters are described by the *nordugrid-cluster* LDAP objectclass. The objectclass has several attributes to characterize the hardware, software and middleware properties of a cluster. The model supports both dedicated and non-dedicated Grid clusters and queues, as it contains attributes which make it possible to distinguish between the Grid-shared and non-Grid-shared resources.

A cluster provides access to Grid-enabled queues which are described by the

nordugrid-queue objectclass. Under a queue entry, the *nordugrid-authuser* and *nordugrid-job* entries can be found grouped in two distinct subtrees (the branching is accomplished by the *nordugrid-info-group* objectclass). Every Grid user who is authorized to submit jobs to a queue should have an entry under the queue. Similarly, every Grid job submitted to the queue is represented by a job entry.

The *nordugrid-authuser* entry contains all the userdependent information of an authorized Grid user. Within the user entries the Grid users can find out, among other things, how many CPUs are available for them in that specific queue, what is the available disk space their job can consume, etc. The NorduGrid philosophy is that the user-dependent information is crucial on the Grid, since the Grid users are more interested in their personal resources rather than the general resources of a cluster.

The *nordugrid-job* entries describe the Grid jobs submitted to a cluster. The detailed job information includes the job's unique Grid identifier, the certificate subject of the job's owner, the job status, the job's submission machine, etc. The job monitoring is done solely by querying the information system. The job entries are kept on the execution cluster, thus implementing a distributed status monitoring system.

The schema contains the *nordugrid-se* and the *nordugridrc* objectclasses for describing Storage Elements and Replica Catalogues respectively, although these resources are not fully implemented yet.

The software:

The information system makes use of the MDS (Monitoring and Discovery Services) of Globus. The standard MDS package is an extensible toolkit for creating a Grid information system. It is built upon and heavily relies on the OpenLDAP software. Such an information system is implemented as a pseudo-distributed LDAP database, with the Grid information being stored in the attribute-value pairs of the entries of the LDAP trees.

The LDAP entries are generated on the fly by the information providers of a Grid resource upon a search request from a user or agent. The information providers are small efficient programs which interface to and gather information from the Grid resources, like clusters, Storage Elements, etc. NorduGrid has created its own set of information providers for populating the entries of the NorduGrid schema.

The local database of a resource is responsible for implementing the first-level caching of the output of the providers and presenting the formulated LDAP entries through the LDAP protocol. For the local database, the GRIS (Grid Resource Information Service) LDAP back-end of the Globus MDS is used.

The local databases are linked together via the GIIS (Grid Information Index Services) information indices. These indices are implemented as a special GIIS LDAP back-end of the Globus MDS. Local resources register dynamically to the indices, which themselves can further register to other indices. Through the usage of the dynamic registration it is possible to create different topologies of indexed resources. The NorduGrid operate a multi-level tree hierarchy, based upon the geographical location of resources: resources within a city register to the index of the city, city index servers register to the country indices, which in turn register to toplevel NorduGrid indices. In order to avoid any single points of failure, NorduGrid operate a multi-rooted tree with several top-level indices. Figure 4 illustrates the NorduGrid Information System

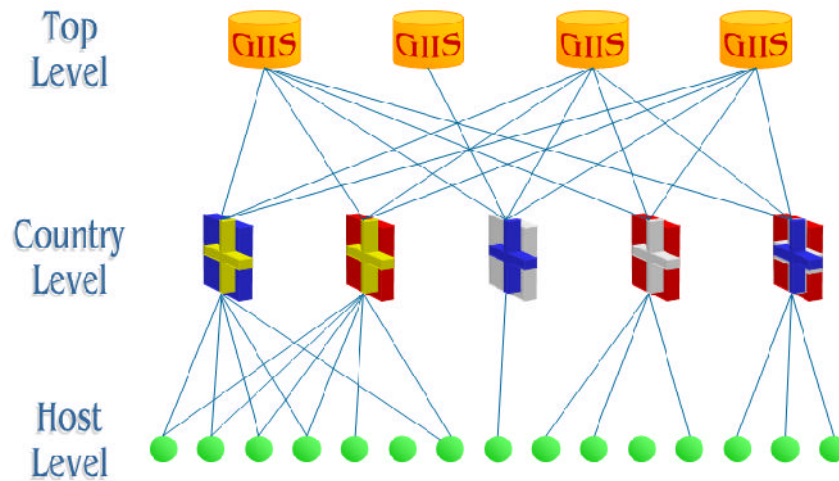


Figure 4. The NorduGrid MDS Hierarchy

hierarchy.

Besides the indexing function, the GIIS indices are capable of performing queries by following the registrations and caching the results of these search operations. However, the present implementation of the caching capability of the GIIS back-ends were found to be unreliable and unscalable in the everyday usage, therefore the NorduGrid operates with disabled GIIS caching. In such a layout, the GIISes are merely used as resource indices, while the query results are obtained directly from the resources (local databases).

c) USER INTERFACE AND RESOURCE BROKERING

The user interacts with the NorduGrid through a set of command line tools. There are commands for submitting a job, for querying the status of jobs and clusters, for retrieving the data from finished jobs, for killing jobs etc. There are also tools that can be used for copying files to, from and between the Storage Elements and replica catalogues and for removing files from storage elements and replica catalogs. The full list of commands is given in Table 10.

Command	Action
ngsub	Job submission
ngstat	Show status of jobs and clusters
ngcat	Display stdout or stderr of a running job
ngget	Retrieve the output from a finished job
ngkill	Kill a running job
ngclean	Delete a the output from the cluster
ngsync	Recreate the user interface s local information about running jobs
ngmove	Copy files to, from and between Storage Elements and Replica Catalogs
ngremove	Delete files from Storage Elements and Replica Catalogs

Table 10. The NorduGrid User Interface Commands

When submitting a Grid job using *ngsub*, the user should describe the job using extended RSL (xRSL) syntax. This piece of xRSL should contain all the information needed to run the job (the name of the executable, the arguments to be used, etc.) It should also contain a set of requirements that a cluster must satisfy in order to be able to run the job. The cluster can e.g. be required to have a certain amount of free disk space available or have a particular set of software installed.

When a user submits a job using *ngsub*, the User Interface contacts the Information System: first to find available resources, and then to query each available cluster in order to do requirement matching. If the xRSL specification states that some input files should be downloaded from a Replica Catalog, the Replica Catalog is contacted as well, in order to obtain information about these files. The User Interface then matches the requirements specified in the xRSL with the information obtained from the clusters in order to select a suitable queue at a suitable cluster. If a suitable cluster is found, the job is submitted to that cluster. Thus, the resource brokering functionality is an integral part of the User Interface, and does not require an additional service.

D) RESOURCE SPECIFICATION LANGUAGE

The Globus project develops a Grid middleware toolkit, which makes use of the Resource Specification Language to parse job options and definitions to resource management systems. RSL is particularly convenient as a job description language, since it is extendable by design, thus allowing a lot of flexibility and adaptability. The NorduGrid architectural solution was designed primarily to suit specific High Energy Physics tasks. While using extensively the Globus Toolkit™, this solution requires certain extensions to the core Globus RSL. This concerns not only the introduction of new attributes, but also the differentiation between the two levels of the job options specifications:

- ◆ User-side RSL - the set of attributes specified by a user in a job description file. This file is interpreted by the User Interface, and after the necessary modifications is parsed to the Grid Manager (GM)
- ◆ GM-side RSL - the set of attributes pre-processed by the UI, and ready to be interpreted by the GM

This differentiation reflects the dual purpose of the RSL in the NorduGrid architecture: it is used not only by users to describe job requirements, but also by the UI and GM as a communication language. Such a functional separation, as well as re-defined and newly introduced attributes, prompted NorduGrid to refer to the used resource specification language as xRSL, in order to avoid possible confusion. xRSL uses the same syntax conventions as the core Globus RSL, although changes the meaning and interpretation of some attributes.

Most notable changes are those related to the file movement. The implementation of the Grid Manager implies that while most of the GRAM RSL attributes can be re-used, they have to be interpreted differently. For example, the standard *executable* attribute becomes obsolete, although it is preserved in the xRSL to preserve compatibility with Globus. By using specific syntax conventions, the main executable will be either uploaded by the UI from the submission node, or will be located by the GM on the execution node.

The major challenge for NorduGrid applications is pre- and post-staging of considerable amount of files, often of a large size. To reflect this, two new attributes were introduced in xRSL: *inputFiles* and *outputFiles*. Each of them is a list of local-remote file name or URL pairs. Local to the submission node input files are uploaded to the execution node by the UI; the rest is handled by the GM. The output files are moved upon the job completion by the GM to a specified location. If no output location is specified, the files are expected to be retrieved by a user via the UI. Any of the input files can be given an executable permission by listing it in the executables attribute. Output file location can be specified as a Replica Catalog pseudo-URL, such that the file will be moved to a location specified in a logical collection, and registered in the RC.

It should be mentioned that the most recent implementation of the Globus GRAM adds some new attributes to the RSL, similar to those introduced in xRSL. However, they were not available at the time of the GM and xRSL development.

Several other attributes were added in xRSL, for convenience of users. Using the *cluster* attribute, a user can explicitly select or exclude a list of clusters on which the resource matching will be performed. The complete GM job log can be retrieved by specifying the *stdlog* attribute. Standard output and standard error can be merged into one file by requesting it in the *join* attribute. In order to enable matching based on a specific cluster configuration and setup, such attributes can be used as *middleware*, *architecture*, *runTimeEnvironment* etc. Since most file movement is done via GridFTP, a user can request a specific number of parallel FTP threads by using the *ftpThreads* attribute.

Such an extended RSL appears to be sufficient for job description of desired complexity. The ease of adding new attributes is particularly appealing, and NorduGrid is committed to use xRSL in further development.

E) MONITORING

The NorduGrid provides an easy-to-use monitoring tool, realized as a Web interface to the NorduGrid Information System. This Grid Monitor allows browsing through all the published information about the system, providing thus a realtime monitoring and a primary debugging for the NorduGrid.

The MDS-based NorduGrid Information System provides a robust and dynamic model for accessing not only quasi-static information about resources and services, but also about such rapidly changing parameters such as queue and job status. Being based on OpenLDAP, it can easily be interfaced to any browsing or monitoring tool, giving thus a user-friendly overview of all the resources.

The Grid Monitor makes use of the LDAP module of PHP4 to provide a Web interface to the information infrastructure. The structure of the Grid Monitor to great extent follows that of the NorduGrid Information System. The basic objects are defined by the following schema s objectclasses:

- *nordugrid-cluster*: a cluster
- *nordugrid-queue*: a queue at the cluster, accessible by the NorduGrid users
- *nordugrid-job*: a NorduGrid job, associated with a queue

- *nordugrid-authuser*: a user, authorized to submit jobs to a given queue

The Grid Monitor also uses the NorduGrid Virtual Organisation (VO) objectclass *organisationalPerson* and the corresponding attributes.

For each objectclass, either an essential subset of attributes, or the whole list of them, is presented in an easily accessible inter-linked manner. This is realized as a set of windows, each being associated with a corresponding module. There are seven major modules in the Grid Monitor:

1. An overall Grid Monitor
2. Cluster Description
3. Queue Details
4. Job Information
5. User Information
6. Attributes Overview
7. List of Users

The screen-shot of the main Grid Monitor window, as available from the NorduGrid Web page, is shown in Figure 5. Most of the displayed objects are linked to appropriate modules, such that with a simple mouse click, a user can launch another module window, expanding the information about the corresponding object or attribute. Each such window gives access to other modules in turn, providing thus a rather intuitive browsing.

The Grid Monitor proves to be a very useful and reliable tool, being the most frequently requested service of the NorduGrid, both by users and developers. The Web server that provides the Grid Monitor access runs on a completely independent machine, therefore imposing no extra load on the NorduGrid, apart of very frequent LDAP queries (default refresh time for a single window is 30 seconds).

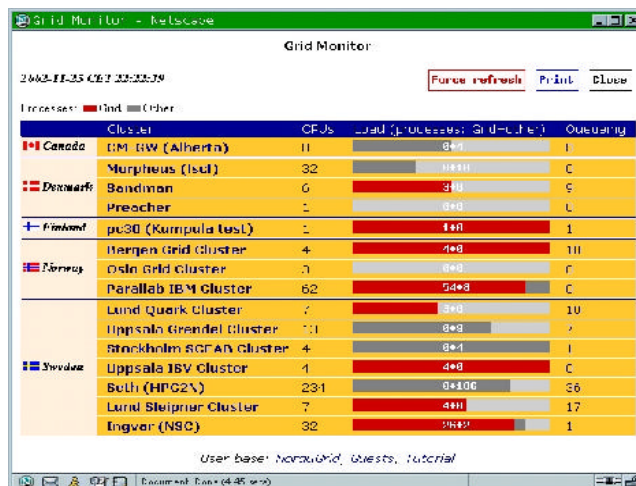


Figure 5. The NorduGrid Monitor

F) SOFTWARE CONFIGURATION

Since the Grid is supposed to be deployed on many sites, it implies involvement of many site administrators, of which not all may be Grid experts. Therefore the configuration of the NorduGrid Toolkit was made as simple as possible. It basically requires writing two configuration files: **globus.conf** and **nordugrid.conf**. The **globus.conf** is used by the *globus-config* package, which configures the information system of the Globus Toolkit™ from a single file. This configuration scheme is not specific to NorduGrid and is also used by other projects, such as the EDG. The **nordugrid.conf** file is used for configuring the various NorduGrid Toolkit components.

This approach proved to be very convenient and allowed to set up sites as remote from Scandinavia as Canada or Japan in a matter of hours, with little help from the NorduGrid developers.

G) DISTRIBUTION

The NorduGrid Toolkit is freely available via the Web site <http://www.nordugrid.org>. One can find there source tar-balls as well as CVS snapshots and nightly builds. There are binary RPM distributions for several Linux distributions. Furthermore, there is a stand-alone local client installation for users without root privileges. It is distributed as a tar-ball and is designed to be a NorduGrid entry point, working out-of-the-box. Since it contains the necessary Globus components, it can be used to interact with other Grids as well.

For completeness, NorduGrid software repository contains some required third-party software, such as the Globus Toolkit™.

4.3. USERS AND CERTIFICATION ISSUES

The user base of the NorduGrid consists predominantly of High Energy Physics researchers, who use the Grid in their work on a daily basis. By end-2002 there were around two dozens of active users in two Virtual Organizations.

User authentication is one of the key issues in a Grid environment. Globus Toolkit™ uses personal certificates, issued by a recognized certification authority, to identify each user. The NorduGrid Certification Authority (CA) is set up at NBI. It provides X.509 certificates for identification and authentication purposes. The scope is limited to people from the Nordic countries involved in Grid-related projects: primarily the NorduGrid and EU DataGrid, as well as DKGRID (Denmark). Contrary to most Certificate Authorities worldwide, the NorduGrid one is not a national, but a transnational virtual organization.

The certificates are meant to be used with the Globus Toolkit™, to provide user authentication. They are recognized not only by the sites, participating in the NorduGrid project, but also by the EU DataGrid.

In a Grid environment such as the NorduGrid, users usually don't have local password-protected user accounts on the computing resources they want to use. They hold instead a certificate issued by a Certification Authority. The access control for the computing resources (authorization) is a matter of a local policy: site administrators retain the full control of choosing which Grid user is allowed to use the resources. The local authorization process is done by mapping the accepted set of Grid users onto the

local user accounts. In the Grid terminology, the authorization is sometimes referred to as "adding a Grid user into the grid-mapfile". NorduGrid have set up a collective authorization method, the NorduGrid Virtual Organization (VO), following practices of other Grid testbeds. This VO maintains a list of people which are authorized to use the NorduGrid resources. The VO tools provide an automatic method for the sites to easily maintain the "VO member" ? "local Unix user" mappings. The tools periodically query the VO user database and automatically generate the local grid-mapfiles following the local policy formulated in the VO-tools configuration file. The automatic mapping does not violate the site autonomy, because the site administrators retain a full control over their systems thanks to the possibility of denying access to unwished Grid users in the NorduGrid VO-tools configuration file.

The database of the VO is maintained by the VO managers. Their responsibility is to add, delete or modify user entries. The NorduGrid VO supports the creation of groups they can be created or removed only by the VO managers. A group is a subset of the a VO and is maintained by an appointed group manager. The group manager selects members of the group out of the full VO database. With the existence of the user groups, the site administrators can implement group based mappings, such that all the members of a certain group are mapped to the same local Unix user, in addition to the default user-based mappings.

Technically, the VO database is stored in an LDAP database. For this purpose, a GSI-enabled OpenLDAP server is used. The built-in SASL (Cyrus Simple Authentication and Security Layer) and GSI-GSSAPI mechanisms of the OpenLDAP server provide an entry and attribute level access control, based on the Grid certificates. The database managers, being authenticated and authorized through their certificates, make use of the OpenLDAP command line tools in order to add, delete or modify entries in the VO database. The NorduGrid sites periodically run the *nordugridmap* utility in order to query the VO LDAP server and automatically create/update local user mappings according to a site policy (as defined in a **nordugridmap.conf** configuration file).

5. USE CASES

5.1. ATLAS DATA CHALLENGE

The ATLAS Experiment at the Large Hadron Collider is a huge High Energy Physics detector, at the moment under construction at the CERN physics laboratory in Geneva, Switzerland. Its goal is to explore the fundamental nature of the matter and forces that shape our Universe.

Vast quantities of data from the experiment are to be collected and interpreted by computers and stored on disks and tapes. The amount of data to be stored is expected to be around 3 PB per year. The computing challenge faced by ATLAS is thus enormous, so to be ready in 2007, when the experiment is planned to start operating, ATLAS has planned a series of computing challenges of increasing size and complexity to test its computing model and the complete software suite. The goal is also to incorporate the use of Grid middleware as soon as possible.

One of the stages of these challenges, the so-called ATLAS Data Challenge 1, ran from July to September 2002 and consisted of large-scale physics simulations based on requests from different physics communities. 39 different institutes from around

the world participated in the simulations, and NorduGrid were amongst them.

Participants were assigned different data sets, according to their capacities. The NorduGrid was assigned two different data sets: 10 percent of the so-called *Dataset 2000* and the whole of the *Dataset 2003*.

For the Dataset 2000, the input data volume was 18 GB, stored in 15 files. These were downloaded manually from CERN and distributed locally to the different sites in Nordu- Grid participating in this part of the challenge. A typical jobsubmission-script is shown in Figure 6.

```
& (executable="ds2000.sh") (arguments="1101")
{stdOut="dc1.002000.simul.01101.hlt.pythia_jet_17.log"} (join="yes")
{inputFiles={"ds2000.sh"
"http://www.nordugrid.org/applications/dc1/2000/dc1.002000.simul.NG.sh"}}
{outputFiles=
  ("dc1.002000.simul.01101.hlt.pythia_jet_17.log"
  "rc://dc1.uio.no/2000/log/dc1.002000.simul.01101.hlt.pythia_jet_17.log")
  ("atlas.01101.zebra"
  "rc://dc1.uio.no/2000/zebra/dc1.002000.simul.01101.hlt.pythia_jet_17.zebra")
  ("atlas.01101.his"
  "rc://dc1.uio.no/2000/his/dc1.002000.simul.01101.hlt.pythia_jet_17.his")
  ("dc1.002000.simul.01101.hlt.pythia_jet_17.AMI"
  "rc://dc1.uio.no/2000/ami/dc1.002000.simul.01101.hlt.pythia_jet_17.AMI")
  ("dc1.002000.simul.01101.hlt.pythia_jet_17.MAG"
  "rc://dc1.uio.no/2000/mag/dc1.002000.simul.01101.hlt.pythia_jet_17.MAG")
} (jobName="dc1.002000.simul.01101.hlt.pythia_jet_17")
{runtimeEnvironment="DC1-ATLAS"}
```

Figure 6. Example of an xRSL script

The notation of the script in Figure 6 is almost self-explanatory. The executable, **ds2000.sh**, is a script downloaded from the specified URL, which takes as an argument the processed partition number, which in this case equals to 1101. **ds2000.sh** calls the ATLAS physics simulation program. To make sure that the ATLAS software is installed on the chosen cluster, the xRSL specifies that the simulation needs the runtime environment **DC1-ATLAS**. A small script, also called **DC1-ATLAS**, is then run locally at the chosen cluster, setting up the ATLAS software environment, together with a variable pointing to the local location of the input files. After simulation, all the output files are uploaded to the Storage Element **dc1.uio.no** and registered into the Replica Catalog.

In total there were 300 jobs. They used around 220 CPUdays for the simulations and produced 5×300 output files with a total size of about 320 GB. A Web page querying the Replica Catalog every 20 minutes for output files was set up to allow an easy browsing of the log-files of the simulations and an easy check if something had gone wrong during the physics simulation (see Figure 7).

NorduGrid: ATLAS DC1 - Netscape

NORDUGRID
Nordic Testbed For Wide Area
Computing and Data Handling

Dataset 2000: completed jobs

- Production script
- RSL-generating script

Partitions marked with ! are probably corrupt

Automatic refresh time is 20 minutes

#	Output partition	Node	Events	Time per event (sec)	Submitted on	ZEBRA file size (byte)	HIS file size (byte)	Log
1	J11C1	cnode1	450	188	Jul 27 11:53:43 CEST 2002	946468800	19472384	Log
2	J11C2	cnode2	454	185	Jul 27 14:04:28 CEST 2002	885006000	19505248	Log
3	J11C3	cnode1	455	177	Jul 27 10:54:46 CEST 2002	866538000	19361792	Log
4	J11C4	cnode2	403	180	Jul 25 14:00:27 CEST 2002	830386800	18915328	Log
5	J11C5	cnode1	424	180	Jul 25 14:39:58 CEST 2002	858762000	19517440	Log
6	J11C6	cnode2	456	177	Jul 25 14:45:16 CEST 2002	925246800	19111936	Log
7	J11C7	cnode1	419	185	Jul 25 20:45:57 CEST 2002	895503600	19521536	Log
8	J11C8	cnode2	458	188	Jul 26 10:53:13 CEST 2002	930884400	19402752	Log
9	J11C9	cnode1	456	185	Jul 26 12:08:41 CEST 2002	901238400	19230720	Log
10	J1110	cnode2	402	176	Jul 27 09:50:23 CEST 2002	786315600	19214336	Log
11	J1111	cnode2	446	190	Jul 26 14:35:35 CEST 2002	942742800	19206144	Log
12	J1112	cnode1	425	187	Jul 26 13:42:07 CEST 2002	878720400	19435520	Log
13	J1113	motstand	454	175	Jul 25 15:48:55 CEST 2002	872272800	18915424	Log
14	J1114	noce2	440	185	Jul 27 14:19:00 EDT 2002	938822400	19185664	Log

Document: Done (73.16 secs)

Figure 7. The Web page showing the production summary

Dataset 2003 was more challenging, because the input volume was a staggering 158 GB in 100 input files. Some sites in NorduGrid were not able to accommodate such a big data set, and it was therefore decided to distribute only subsets to these sites. Some sites did not have any files available locally and used features of the NorduGrid software to download and cache input data. All the distributed sets were then registered into the Replica Catalog.

During the job submission, the broker would query the Replica Catalog for sites with the required input file, and the job would then only be submitted to such sites. This means that the jobs would only go to those sites with the required data already present.

For this data set, there were 1000 jobs using about 300 CPU-days and producing a total output of 442 GB. All output files were successfully uploaded to the dc1.uio.no Storage Element and registered into the Replica Catalog.

In conclusion, the ATLAS Data Challenge was successfully met by NorduGrid with the failure rate being negligibly small.

5.2. OTHER USE CASES

Although the described above case was the most challenging, the NorduGrid testbed provides resources for many other studies. In particular, various theoretical calculations in different fields of physics were performed in very short terms, impossible to be achieved otherwise. Several other calculations for the ATLAS experiment were also performed over the NorduGrid testbed. The results of those studies are published, with appropriate credits given to the NorduGrid project by the authors.

6. INTERNATIONAL COOPERATION AND DISSEMINATION

6.1. RELATION TO THE EU DATA GRID (EDG) PROJECT

The NorduGrid project participates in the EU DataGrid activities along two directions:

- being part of the DataGrid Testbed (coordinator: A.Wäänänen)
- development of applications (coordinator: O.Smirnova)

All the NorduGrid sites successfully took part in the Testbed0 (June 2001), main goal of which was Globus toolkit and basic services installation. The NorduGrid is an integral part of the Testbed, and the NorduGrid CA is one of those officially recognized by DataGrid.

To enable close cooperation with the DataGrid, representatives of the NorduGrid regularly attend corresponding meetings and Workshops. Presentations of the NorduGrid activities were regularly given at the DataGrid workshops.

6.2. OTHER ACTIVITIES

From the very beginning of the project, NorduGrid was actively interacting with the Globus project, primarily for the purposes to evaluate the Globus Toolkit™, and later - to suggest improvements and other modifications, as well as to share experience. In June 2002, a working meeting between the Globus and NorduGrid representatives took place in Copenhagen; since October 2002 NorduGrid participates regular phone conferences with the EDG, VDT and Globus projects. NorduGrid was invited to present their progress during the Globus World conference in January 2003.

Demonstrations and presentations of the NorduGrid toolkit and testbed were made during several international conferences, such as the 5th Global Grid Forum (July 2002), Conference on Applied Parallel Computing (June 2002), 8th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (June 2002) and other meetings.

The NorduGrid tutorial was given at the 3rd Annual Workshop on Linux Clusters for Super Computing in Linköping (October 2002). It was attended by 60 participants, and the materials were made available on the NorduGrid Web site.

In October 2002, the NorduGrid informational booklet was published and distributed worldwide.

7. SUMMARY

The NorduGrid project proved to be a very successful one, not only achieving the original goals, but moving beyond them.

In a very short time, and by a very small group of researchers, not only a thorough evaluation of existing Grid solutions was done, but a totally new philosophy was developed and realised as the NorduGrid toolkit.

The NorduGrid philosophy proves to be a valid one, as the NorduGrid environment spans many sites around the World and makes use of several different operation systems and distributions, such as RedHat, Mandrake, Slackware or Debian Linux. The whole system runs reliably 24/7 without the need for being manned around the clock. The NorduGrid thus operates a unique production environment, being used by academic researchers in their daily work.

APPENDIX A

List of Web sites containing further information about terms, tools and projects mentioned in the report

- <http://www.nordugrid.org> : The NorduGrid project
- <http://www.globus.org>: The Globus project
- <http://www.edg.org>: EU DataGrid project
- <http://www.openldap.org>: Open source implementation of the Lightweight Directory Access Protocol.
- <http://www.globus.org/security>: Grid Security Infrastructure (GSI).
- <http://asg.web.cmu.edu/sasl>: Simple Authentication and Security Layer (SASL)
- <http://www.globus.org/datagrid/deliverables/gsiftp-tools.html>: GSIFTP Tools.
- <http://www.openpbs.org>: The Portable Batch System (PBS)
- <http://supercluster.org/maui>: The Maui Scheduler
- <http://www-fp.globus.org/datagrid/replica-catalog.html>: Globus Replica Catalog Service.
- <http://grid-data-management.web.cern.ch/grid-data-management/replica-location-service/index.html>: Replica Location Service
- <http://www-fp.globus.org/gram/rs1%5Fspec1.html>: The Globus Resource Specification Language RSL v1.0
- <http://www.isi.edu/~flon/cgs-wg/index.htm>: The CIM-based Grid Schema Working Group
- <http://www.hicb.org/glue/glue-schema/schema.htm>,
<http://www.cnaf.infn.it/~sergio/datatag/glue/index.htm>: The GLUE schema effort
- <http://www.globus.org/mds>: The Monitoring and Discovery Service
- <http://www.globus.org/mds/hierarchical%5FGIIS.pdf>: Hierarchical GIIS, Globus documentation.
- <http://www.php.net>: PHP: Hypertext Preprocessor
- <http://atlasexperiment.org>: The ATLAS Experiment

APPENDIX B

List of the NorduGrid notes, publications and talks

Notes and publications:

- "An Overview of an Architecture Proposal for a High Energy Physics Grid". A.Wäänänen et al., PARA 2002, LNCS 2367, pp. 76, J. Fagerholm et al. (Eds.) Springer-Verlag Berlin Heidelberg 2002.
- "The NorduGrid project: Using Globus Toolkit for building Grid infrastructure". A.Konstantinov. To appear in Proc. of ACAT 2002, Nucl. Instr. and Methods A. Elsevier Science, 2002.
- "NorduGrid client installation instructions". Internal note.
- "The Grid Monitor", Usage Manual. Internal note.
- "The NorduGrid toolkit user interface", User Manual. Internal note.
- "The NorduGrid Grid Manager", Description and Administrator's Manual. Internal note.
- "An Overview of The NorduGrid Architecture Proposal". Internal note.
- "The NorduGrid Information System". Internal note.
- "Extended Resource Specification Language". Internal note.
- Grid certificate: "Mini How-to". Internal note.
- "Comments on Installation Procedure and Usage of Workload Management Software of EU DataGrid". Internal note.
- "Comments on the Computing Element Information Provider of the EU DataGrid Testbed 1". Internal note.
- "Performance evaluation of the GridFTP within the NorduGrid project". Internal note.

Talks:

- CMS Week, September 23-27, 2002, Geneva. "The NorduGrid: Project And Toolkit"
- ATLAS Software Week, September 16-19, 2002, London. "The NorduGrid Toolkit"
- GGF5, July 21-24, 2002, Edinburgh. "The NorduGrid Information System"
- ACAT 2002, 24 June 2002, Moscow. "The NorduGrid project: Using Globus toolkit for building Grid infrastructure"
- PARA'02, June 15-18, 2002, Espoo. "The NorduGrid Architecture and Toolkit"
- TREFpunkt vårmöte, March 14, 2002, Uppsala. "The NorduGrid Project: Technical Details "
- TREFpunkt vårmöte, March 13, 2002, Uppsala. "The NorduGrid Project: Building a Grid infrastructure in Scandinavia"
- LHC Computing Grid launching workshop, March 13, 2002, Geneva. "Grid

Deployment in the Nordic Countries"

- EDG WP6 meeting, 11 December 2001, Geneva. "NorduGrid Status"
- ATLAS Software Week, 04 December 2001, Geneva. "Installation and Evaluation of Grid Tools: Work in Progress"
- 4th Nordic LHC Workshop, 23 November 2001, Stockholm. "Quick Introduction to NorduGrid"
- NorduNet2 Project Managers Conference, 24 September 2001, Hurdalsjøen, Norway. "Nordic Testbed for Wide Area Computing and Data Handling"
- EDG WP8 meeting, 20 September 2001, Geneva. "Experience with GridFTP at NorduGrid"
- 2nd DataGrid Workshop, 4 July 2001, Oxford. "Status of NorduGrid testbed"
- 1st DataGrid Workshop, 7 March 2001, Amsterdam. "Nordic Grid Testbed"