European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2004
P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer (eds.)
Jyväskylä, 24–28 July 2004

SCIENCE ON NORDUGRID

Paula Eerola[⋆], Tord Ekelöf[†], Mattias Ellert[†], John R. Hansen[•], Aleksandr Konstantinov[◇◁], Balázs Kónya[⋆], Jakob L. Nielsen[•], Farid Ould-Saada[◁], Oxana Smirnova[⋆], Anders Wäänänen[•]

[⋆]Particle Physics, Institute of Physics, Lund University
Box 118, 22100 Lund, Sweden

[†]Department of Radiation Sciences, Uppsala University
Box 535, 75121 Uppsala, Sweden

[•]Niels Bohr Institutet for Astronomi, Fysik og Geofysik
Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark

[◇]University of Oslo, Department of Physics
P. O. Box 1048, Blindern, 0316 Oslo, Norway

[◁]Vilnius University, Institute of Materials Science and Applied Research
Saulėtekio al. 9, Vilnius 2040, Lithuania

Key words: Grid, NorduGrid, Middleware, Atlas, Atlas Data-Challenges

Abstract. *The NorduGrid Collaboration has deployed a Grid architecture with the goal to meet the requirements of scientists in the Nordic Countries. Starting out as a grid-project with emphasis on batch processing suitable for problems encountered in High Energy Physics, it has now developped into a rather generic Grid system used by a growing number of scientists in the Nordic Countries. The NorduGrid architecture is a light-weight, non-invasive and dynamic one, but robust and scalable enough to meet the most challenging tasks in science. In the following article, we will describe the NorduGrid ARC middleware design with emphasis on the performance, stability and useability that eases the job of scientists. We will also describe some of the science that is being done at the moment on NorduGrid.*

1 INTRODUCTION

In order to face the computing challenge of the LHC and other similar problems emerging from the science communities, the NorduGrid project was initiated by Nordic scientists in May 2001. The goal was to create a Grid-infrastructure in the Nordic countries (Denmark, Norway, Sweden and Finland) that could be used to evaluate Grid tools by scientists and find out if this new technology could help in solving the emerging massive data and computing intensive tasks.

It was quickly decided that the NorduGrid project would implement a Grid architecture from scratch. The implementation would use existing low-level code from other projects and implement new higher level services almost from scratch to create a working Grid middleware in order to create a production-level Grid-infrastructure.

The now existing middleware solution called ARC (Advanced Resource Connector) collects reliable implementations of fundamental grid services, such as information services, resource discovery and monitoring, job submission and management, brokering and data management and resource management. The middleware builds upon standard open source solutions like the OpenLDAP, OpenSSL, SASL and the Globus [1] Toolkit 2 (GT2) libraries and provides innovative solutions essential for a production quality middleware.

The middleware has been deployed and used since July 2002 as the NorduGrid production grid which at the moment is one of the largest production Grid facilities in the world with a peak of up to 1500 CPU's available 24 hours a day, 7 days a week. The resources range from original small test-clusters to some of the biggest supercomputer clusters in Scandinavia including the recently deployed SweGrid clusters.

In this article, we will give a comprehensive description of the architecture and middleware with focus on the solutions that makes the NorduGrid production grid a production-level Grid solution.

Since the Fall of 2002, the NorduGrid production grid has been used by a number of scientists to perform research. In this article, we will also describe a number of example of such research. In particular, we will focus on the so-called Atlas Data-Challenges which have been running in several stages on NorduGrid.

2 ARCHITECTURE AND MIDDLEWARE

The NorduGrid architecture and middleware were planned and designed from the beginning to satisfy the needs of users and system administrators simultaneously. In short these needs can be outlined by the following general philosophy which has been applied in all stages of the development and deployment of the middleware:

- Start with simple things that work and proceed from there.
- Avoid architectural single points of failure.
- The solutions should be scalable.

- As few requirements as possible on the clusters:
  - Resource owners retain full control of their resources.
  - No dictation of cluster configuration or install method.
  - No dependence on a particular operating system or version.
  - Clusters need not be dedicated to Grid jobs.
  - Computing nodes are not required to be on the public network.
- Reuse existing system installations as much as possible.

The key components of the NorduGrid architecture consists of a set of **Computing Clusters**, the natural computing units in the architecture, the **Information System**, a dynamic information service serving information to other components and **Storage Elements** providing storage facilities for users in need of disk-space. These are supported by several components of the ARC middleware, the ARC **Grid Manager** that handles all grid-related tasks locally on the clusters, the ARC **User Interface**, a set of command-line tools and the ARC **Grid Monitor**, a monitoring agent providing easy-to-browse information to all users and system-administrators. In the following we will describe these components. A more detailed account of these can be found in [2].

2.1 Computing Clusters

The **Computing Clusters** consists of a front-end computer managing a back-end cluster of nodes. It is assumed that the cluster is equipped with a standard batch-system but ARC does not dictate any specific configuration and tries to be just another add-on component. In particular no middleware has to be installed on the back-end nodes.

ARC installs an extra layer which interfaces the local batch system to the Grid. This layer includes the ARC **Grid Manager**, a GridFTP server and the ARC information providers. The **Grid Manager** is a service taking care of jobs, file-staging and cache areas. Information services are efficient information gatherers providing information about the cluster and jobs to the information system.

2.2 Information System

The **Information System** [3] is based on MDS [4] and realized as a distributed service serving information for components such as the **User Interface** and monitoring. It consists of a dynamic set of local databases coupled to computing and storage resources building a hierarchical mesh of grid-connected sites. The local databases register themselves to a set of indexing services which are contacted by the **User Interface** and monitoring agents to find contact information of the databases for direct queries.

2.3 Storage Elements

The NorduGrid **Storage Elements** provide distributed storage facilities for users in need of extensive storage for their grid-jobs. At the moment **Storage Elements** are implemented as GridFTP servers with data access control based on the users Grid certificates. The data can be registered into the Globus Replica Catalog indexing service or the recently supported Globus Replica Location Service.

Development of a so-called **Smart Storage Element** and eventually a corresponding Indexing Service is underway. The main task is to create a service with a rich enough functionality that makes it possible to handle data management as autonomously as possible. In this way, data and their registration are integrated in a reliable way making automatic replication possible.

2.4 User Interface

The ARC **User Interface** is a set of command-line tools to submit, monitor and manage jobs on the grid, move data around and query resource information. The **User Interface** comes with a built-in broker, which is able to select the best matching resource for a job. The grid job specification is expressed in the extended Resource Specification Language (xRSL) [5] – an example of which is shown later. The complete list of commands can be seen in table 1 and detailed information about these can be found in the ARC **User Interface** manual [6].

command	action
`ngsub`	Job submission
`ngstat`	Show status of jobs
`ngcat`	Display standard output of running jobs
`ngget`	Retrieve output from finished jobs
`ngkill`	Kill running jobs
`ngclean`	Delete jobs from a cluster
`ngsync`	Update the User Interface's local information about running jobs
`ngresub`	Resubmits a job to another cluster
`ngcopy`	Copy files to, from and between Storage Elements
`ngremove`	Delete files from Storage Elements

Table 1: The NorduGrid User Interface commands.

2.5 Grid Monitor

ARC comes with an easy-to-use monitoring tool, the ARC **Grid Monitor** realized as a Web interface to the **Information System**. This **Grid Monitor**, which is available at *http://www.nordugrid.org/monitor.php*, allows browsing through all the published information about the system, jobs, users etc. providing real-time monitoring and giving an

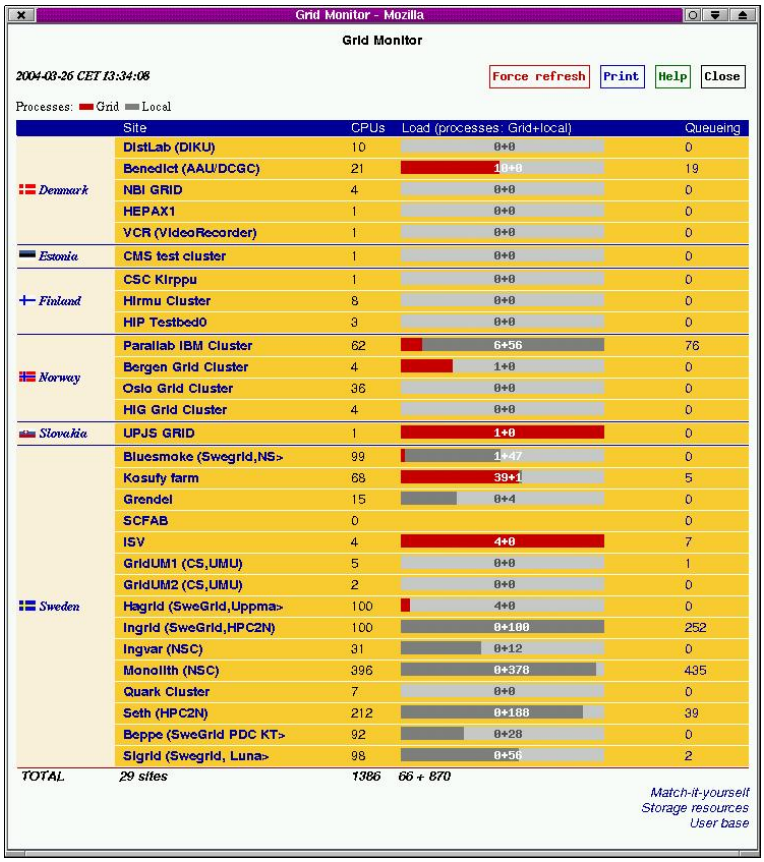

Figure 1: A typical screenshot of the GridMonitor. The number of CPU's for each cluster is given to the right of each cluster name. The number of running grid-jobs for each cluster is given by the red bar and the left number in the middle of the bar. The number of locally submitted (non-grid) jobs is given by the grey bar and the right number in the bar. The total number of clusters, CPU's, running grid-job and locally submitted job are given at the bottom.

excellent debugging tools for users as well as administrators on the NorduGrid production grid. A screen-shot of the main **Grid Monitor** window, as available from the NorduGrid Web page, is shown in Fig. 1. A user can with simple clicks launch other windows expanding the information about objects or attributes in the main window. Each such window gives access to object and attributes in turn providing a rather intuitive browsing.

3 TASK FLOW

The components described above are designed to support the job-submission task flow as follows:

1. A user prepares a job description using a NorduGrid extension of the Resource Specification Language (called xRSL). This description can include application specific requirements, such as input and output data descriptions as well as other options

used in resource matching, such as required cpu-time, required disk-space or an explicit **cluster**.

2. The job description is submitted to and interpreted by the **User Interface**, which performs the resource brokering using the **Information System** and input/output-data and uploads the interpreted job-description to the chosen cluster eventually uploading specified accompanying files.

3. The job request is received by the **Grid Manager** on the cluster front-end. It handles pre-processing, job submission to the local batch system and post-processing depending on the job specifications. Input and output data manipulations are made by the **Grid Manager** with the help of the **Storage Elements**.

4. A user can during job-execution use the **User Interface** or **Grid Monitor** to follow the job progress. When the job ends, the user can retrieve files specified in the job description.

4 Advantages of the NorduGrid architecture

As mentioned above, the architecture and middleware has been designed to provide end-user and system-administrator alike with a stable, scalable and reliable grid-solution.

The development has from the outset been user- and application-driven with emphasis on providing solutions to feature-requests and problems as they arise from users and system-administrators. This has allowed a rapid development of new features but always from a working and tested base. In that way users and system-administrator experiences have been tightly coupled to the middleware development.

Unlike other grid-solutions, the ARC middleware is highly non-instrusive. In the standard configuration, it is installed only on the cluster front-end and at the same time allows a high degree of configuration that should suit the needs of most system-administrators. A system-administrator installing the ARC middleware on a cluster need not to dedicate the cluster to Grid. He retains as before the full control over his cluster. In fact a common way of configuring the ARC middleware on a cluster is to run the exact same setup as before but to use the grid-jobs for backfilling. In this way, resource owners can maximize the usage of their clusters without losing their original custormers. The conclusion is that nothing is lost but a lot is gained.

This whole approach has led to the rapid adoption of the ARC middleware on several world-class production clusters in the Scandinavian countries and a growing number of active users even beyond the Nordic Countries.

5 Science on NorduGrid

The NorduGrid project was initiated by the experimental High Energy Physics community in the Nordic Countries. For this reason the majority of applications running on the NorduGrid production grid has been related to this field. However, a growing

number of scientists from other fields are now using the NorduGrid production grid to perform their research. In the following we will describe some of the grid-projects that have recently been using the NorduGrid production grid resources.

6 Atlas Data-Challenges

The European High Energy Physics community is in the final stages of the construction of the Large Hadron Collider (LHC) - the world biggest accelerator, being built at the European Particle Physics Laboratory (CERN) in Geneva. The challenges to be faced by physicists are unprecedented. Data collected by these experiments will allow for exploration of new frontiers of the fundamental laws of nature. One of the greatest challenges of the LHC project will be the acquisition and analysis of the data. At its design luminosity, each detector will observe particle-collisions at a rate of $4 \cdot 10^7$ per second. A set of filter algorithms aims to reduce the event rate to less than 1000 events per second for final storage and analysis. The equivalent data volume is approximately 10 PB pr. year. Atlas is one of the all-purpose experiments with almost 2000 physicists contributing to the development of hardware and software. Once running, they would expect to have almost instantaneous access to the data and to a set of up-to-date analysis tools.

In order to prepare for this data-taking, Atlas has planned a series of computing challenges [7] of increasing size and complexity. The goals include a test of its computing model and the integration of Grid-middleware as quickly as possible. The first of these Data-Challenges, Atlas Data-Challenge 1, was initiated in the second half of 2002 and ran in several stages throughout 2003 and it is planned that the second Data-Challenge will run over the summer of 2004. The Scandinavian contribution to the Atlas Data-Challenges consists of a set of resources all connected to the NorduGrid production grid. In the following we will describe the results and experiences from the first Atlas Data-Challenge and present the some of preparations leading up to the second Data-Challenge beginning summer 2004.

6.1 Atlas Data-Challenge 1

Atlas Data-Challenge 1 consisted of large-scale physics simulations running in several stages throughout 2002 and 2003. The simulations included 39 institutions from around the world including the present group of researchers using the NorduGrid production grid to process the tasks assigned to them. The tasks consisted of several sets of data to be simulated with the Atlas physics simulation programs.

A typical production xRSL-script is shown in figure 2. xRSL scripts consists of a set of tuples each specifying one component of the job-description. For example the `executable`-attribute specifies the executable that is run. In this case, the script `ds2000.sh` calls the Atlas physics simulation program, which performs the actual Atlas simulations. The script `ds2000.sh` is downloaded from the URL given under `inputFiles` and take the input-partition (in this case 1145) as `argument`.

```
&(executable="ds2000.sh")
(arguments="1145")
(stdout="dc1.002000.simul.01145.hlt.pythia_jet_17.log")
(join="yes")
(inputFiles=("ds2000.sh" "http://www.nordugrid.org/applications/dc1/2000/
dc1.002000.simul.NG.sh"))
(outputFiles=
   ("atlas.01145.zebra"
    "rc://dc1.uio.no/2000/zebra/dc1.002000.simul.01145.hlt.pythia_jet_17.zebra")
   ("atlas.01145.his"
    "rc://dc1.uio.no/2000/his/dc1.002000.simul.01145.hlt.pythia_jet_17.his")
   ("dc1.002000.simul.01145.hlt.pythia_jet_17.log"
    "rc://dc1.uio.no/2000/log/dc1.002000.simul.01145.hlt.pythia_jet_17.log")
   ("dc1.002000.simul.01145.hlt.pythia_jet_17.AMI"

    "rc://dc1.uio.no/2000/ami/dc1.002000.simul.01145.hlt.pythia_jet_17.AMI")
   ("dc1.002000.simul.01145.hlt.pythia_jet_17.MAG"
    "rc://dc1.uio.no/2000/mag/dc1.002000.simul.01145.hlt.pythia_jet_17.MAG")
)
(jobName="dc1.002000.simul.01145.hlt.pythia_jet_17")
(runTimeEnvironment="DC1-ATLAS")
(replicacollection="ldap://grid.uio.no:389/lc=ATLAS,rc=Nordugrid,
dc=nordugrid,dc=org")
(CPUTime=2000)
(Disk=1200)
```

Figure 2: A complete xrsl submission-script in Data-Challenge 1.

The xRSL requires the `runTimeEnvironment` DC1-ATLAS. This `runTimeEnvironment` is a small script run on the cluster to set necessary environment variables needed to run the Atlas-job. The `runTimeEnvironment` is specified to ensure that the job is only submitted to those clusters which have the corresponding Atlas software installed.

The standard output of the job will go into the file specified by the `stdout`-attribute. Furthermore the name of the job is specified using the `jobName`-attribute. The xRSL also requests a certain amount of CPU time and diskspace through the `CPUTime` and `Disk` attributes. The `CPUTime` attribute ensures e.g. that the **User Interface** chooses the right local batch queue to submit the job to the chosen cluster.

Each job produces a set of output-files. These are specified under `outputFiles` and will at the end of the job be automatically uploaded by the **Grid Manager** to the location specified under `outputFiles`. In this case, the output-files specified will be uploaded to a physical location registered in the NorduGrid Replica Catalog defined by the

`replicaCollection` attribute, so that on request from the **Grid Manager**, the Replica Catalog will resolve `rc://dc1.uio.no/log` to `gsiftp://dc1.uio.no/dc1/2000/log` and uploads the file to that **Storage Element**. It is completely similar with the other files.

In the whole of Data-Challenge 1, NorduGrid contributed with approximately 5 percent of the total production worldwide – measured in cpu-time. A total of 6050 jobs were running consuming about 5600 NCPU-days with more than 5 TB of input-data and producing more than 1.7 TB of output-data.

NorduGrids success in Atlas Data-Challenge 1 tasks showed the reliability of the architecture and the ARC middleware. For comparison, other grid-projects participating in Atlas Data-Challenge 1 could not perform more than just preliminary tests and were thus not used in the challenge at all.

6.2 Other science projects on NorduGrid

A substantial amount of scientists have started to use the NorduGrid production grid as their primary source of computer-power and storage-capacity. Their applications range from Atlas applications, through similar simulations for the HeraB experiment, quantum lattice models, quantum chemistry, supersymmetric simulations in theoretical high energy physics, genomics and bioinformatics studies to meteorology. A fairly complete list of ongoing projects is being updated at [8]. The science already done has resulted in a number of articles acknowledging the use of the NorduGrid production grid such as: [9],[10].

One recent project that has been running on NorduGrid is a Gene Regulation Bioinformatics project with the goal of providing a Grid platform for Gene Regulation Bioinformatics that allow predictions of involvement of genes in the pathogenesis of human diseases. This project consists of a large number of small, independent jobs running the same application with different input-data. For this kind of project, the NorduGrid production grid is ideally suited.

Another recent application that scientists have started to run on NorduGrid is the Molcas quantum chemistry software. Molcas [11] is developed by theoretical chemists at Lund University and the basic philosophy is to construct methods that will allow an accurate ab initio treatment of very general electronic structure problems for molecular systems in both ground and excited states. In collaboration with Grid personnel in Lund, this group is now able to run their simulations on NorduGrid.

7 Summary

The NorduGrid project has deployed a stable, reliable and scalable production Grid architecture in the Nordic Countries. This production Grid is increasingly utilized by scientists for their research. NorduGrid was the Scandinavian contribution to Atlas Data-Challenge 1. This was a big success with a contribution of about 5 percent of the total worldwide production.

REFERENCES

[1] "The Globus Project", **http://www.globus.org**.

[2] P. Eerola et. al., "The NorduGrid architecture and tools," in Proc. of the CHEP 2003, PSN: MOAT003 (2003)

[3] B. Kónya, "The NorduGrid Information System", [Online], **http://www.nordugrid.org/documents/ng-infosys.pdf**.

[4] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, "Grid Information Services for Distributed Resource Sharing", Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, 2001.

[5] O. Smirnova, "Extended Resource Specifikation Language", [Online], `http://www.nordugrid.org/documents/xrsl.pdf`

[6] M. Ellert, "The NorduGrid toolkit user interface, User's manual", [Online], `http://www.nordugrid.org/documents/NorduGrid-UI.pdf`

[7] "ATLAS Data Challenges", Document presented to the ATLAS Executive Board, `http://atlasinfo.cern.ch/Atlas/GROUPS/ SOFTWARE/DC/doc/AtlasDCs.pdf`

[8] O. Smirnova, "Applications gridified with NorduGrid", [Online], `http://www.nordugrid.org/applications/appdb.html`

[9] T. Sjöstrand and P. Z. Skands, "Baryon Number Violation and String Topologies", Nucl. Phys. B659 (2003) 243

[10] O. Syljausen, "Directed Loop Updates for Quantum Lattice Models", Phys. Rev. E67 (2003) 46701

[11] `http://www.teokem.lu.se/molcas/about.html`