

NorduGrid Tutorial: Installation and Examples

Tallinn, Estonia
January 23, 2004

Arto Teräs
arto.teras@csc.fi

Contents

- NorduGrid overview
- Grid identities and Certificate Authorities
- Installation of NorduGrid client software
- Obtaining a certificate
- NorduGrid user interface
- Running jobs on the NorduGrid (examples)
- Questions

CSC

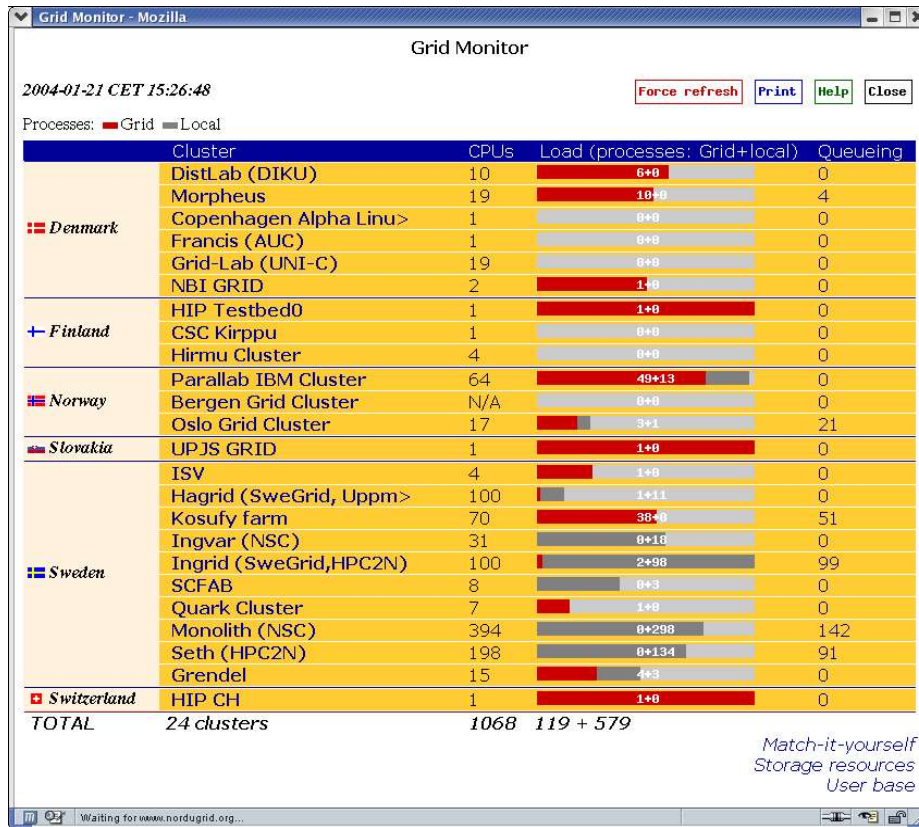
- The only major scientific computing center in Finland, owned by the Ministry of Education
- Not involved in the original NorduGrid project
- Now participating in Nordic DataGrid Facility (NDGF) project, also involved with other Grid initiatives
 - DEISA
 - National Grid for material science researchers planned for late 2004
- NorduGrid middleware is a good candidate for the material sciences Grid project and Nordic Grid collaboration
- <http://www.csc.fi>

NorduGrid



- Connects resources in Nordic countries
- Globus based middleware, some services replaced by own components
- Open for participation
- For a more complete overview, see the website or presentations later today
- <http://www.nordugrid.org>

Grid Monitor on NorduGrid Website

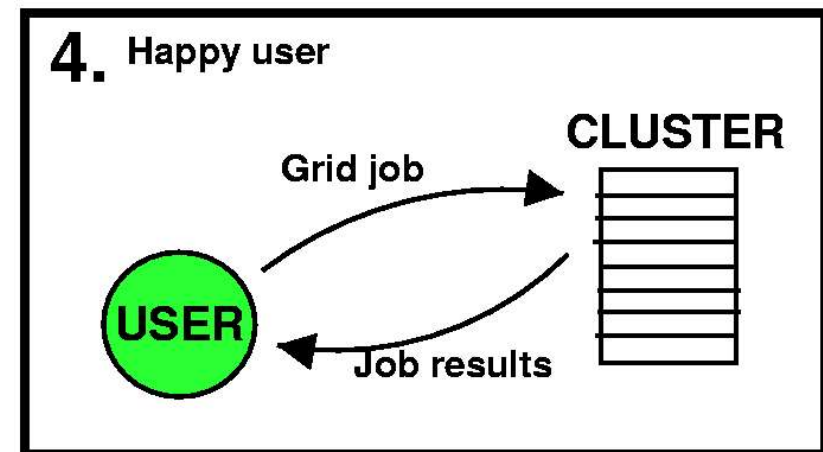
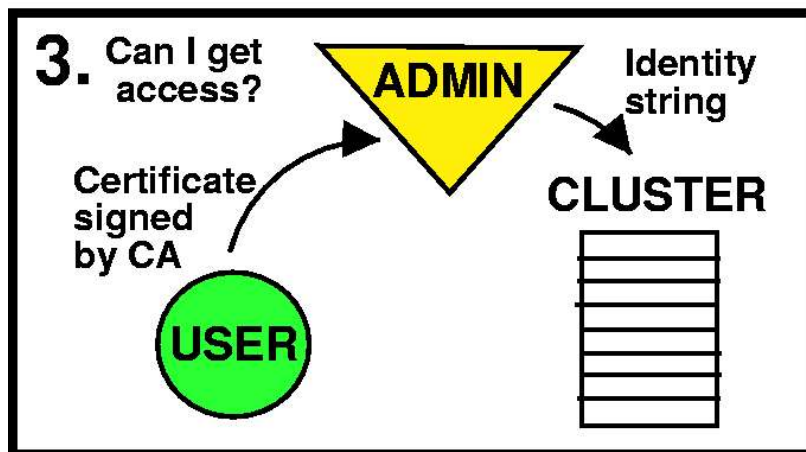
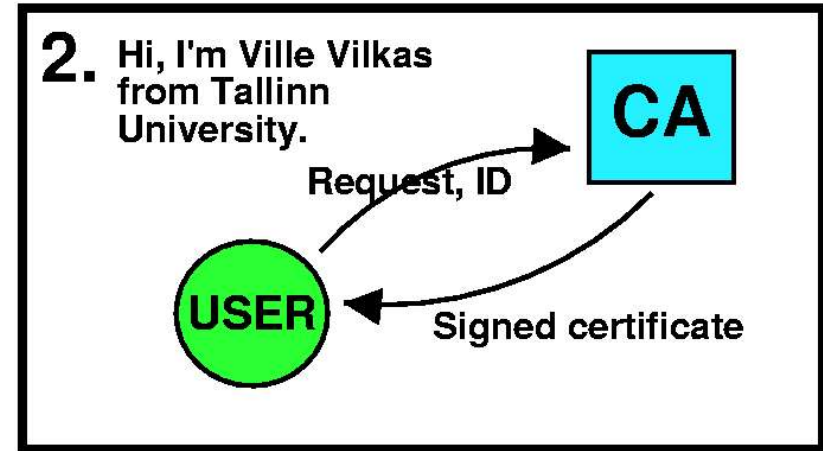
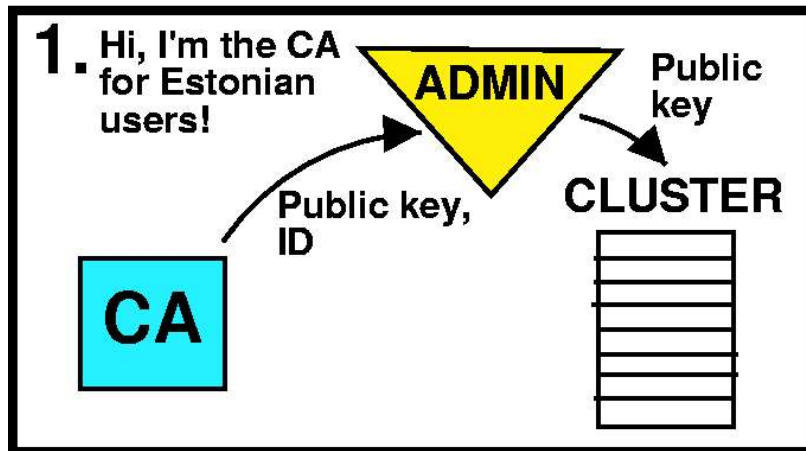


- Shows currently connected resources
- Useful also as a user interface tool
 - browse queues and job states by cluster
 - list jobs belonging to a certain user
- No authentication in the current version

Certificates and Grid

- Almost all Grid middleware packages use certificate based authentication for logging in to the Grid
- Certificate represents the user's identity in the Grid
 - Similar to a passport in the physical world
 - Each user has her own personal certificate
 - Based on strong cryptography
 - Used for authentication (but does not automatically give access to resources)
 - Signed by a trusted third party called Certificate Authority (CA)

Certificate Trust Chain



Certification Authority (CA)

- Signs certificate requests coming from users
 - Similar to state official who grants passports
 - Checks the identity of the user before signing (for example meeting in person and requesting an id)
- CA gives his guarantee of the identity of the user
 - System administrators will know who is asking for access to resources - provided they trust the CA
- The NorduGrid CA signs only certificates for users in Scandinavia and Finland
 - Estonian CA established to serve Estonian users

What Does a Certificate Look Like?

- Consists of two files:
 - Private key is protected by a password and kept secret
 - Public key is given out to third parties
 - Certificate Authorities sign the public key, even they never see the private key
- Look like a string of random numbers and letters, but tools can be used to convert the information in readable form
 - \$ `grid-cert-info -file <certificate file>`
 - For example, my identity stored in my NorduGrid certificate is "O=Grid, O=NorduGrid, OU=csc.fi, CN=Arto Teras"

Virtual Organizations (VO)

- Lists of user identities in the Grid
 - Allow to manage users as groups
- Used for authorization
 - Similar to a group visa in physical world
 - Typically, access to a resource is granted to a virtual organization, meaning that all members of that VO can use the resource
- Implemented as simple text files, LDAP servers or databases specially designed for VO management

NorduGrid Access Policy

- All sites trust the NorduGrid Certification Authority
 - Some sites trust additional CAs
- Most resources are not dedicated to Grid use
 - Both local users and Grid users can submit jobs
- Members of the NorduGrid virtual organization have access to at least a part of the CPUs on most resources
- Users of certain applications have extra privileges
- Guest users have access to several test clusters
- No accounting or charges for cpu time or disk space, but no guarantees either

NorduGrid Client Software

- Required to submit jobs to NorduGrid
- Binaries available for various Linux distributions
 - Source code for developers and users of other Unix platforms (not tested, probably requires some modifications to compile)
- Standalone tar file or RPM packages?
 - The standalone client can be simply uncompressed in a directory without root privileges
 - RPM packages are recommended for multi-user installations
 - RPM packages are used in this tutorial
- <http://ftp.nordugrid.org/download/>

Installation Steps

- Install the RPM packages
- Create a certificate request
 - \$ `grid-cert-request -int`
- Send the request to a Certification Authority
 - Estonian users should send email to ca@nicpb.ee
- Copy the signed certificate to file `.globus/usercert.pem` in your home directory
- Check the installation with some test jobs
 - \$ `ngtest 1 -d 1` (submit test job 1, show debug output)
- Start using the Grid!

NorduGrid User Interface

- Set of command line utilities:
 - **ngsub** to submit a task
 - **ngstat** to obtain the status of jobs and clusters
 - **ngcat** to display the stdout or stderr of a running job
 - **ngget** to retrieve the result from a finished job
 - **ngkill** to cancel a job request
 - **ngclean** to delete a job from a remote cluster
 - **ngrenew** to renew user's proxy
 - **ngsync** to synchronize the local job info with the MDS
 - **ngcopy** to transfer files to, from and between clusters
 - **ngremove** to remove files

Submitting a Minimal Job

- Log in to the Grid

```
$ grid-proxy-init
```

- Submit a job

```
$ ngsb '&(executable='/bin/echo')(arguments='Hello Grid')(stdout='hello.txt)'
```

```
=> Job submitted with jobid gsiftp://morpheus.dcg.dk:2811/jobs/1757591474592630108
```

- Fetch the results

```
$ ngget gsiftp://morpheus.dcg.dk:2811/jobs/1757591474592630108
```

xRSL files

- Resource Specification Language (RSL) files are used to specify job requirements and parameters for submission
 - NorduGrid uses an extended language (xRSL) based on the Globus RSL
- Similar to scripts for local queueing systems, but include some additional attributes
 - Job name
 - Executable location and parameters
 - Location of input and output files of the job
 - Architecture, memory, disk and CPU time requirements
 - Library dependencies and version requirements

xRSL example

- File sillyloops.xrsl:

```
& (executable=sillyloops.sh)
  (jobname=perftest)
  (arguments=' '10000' ')
  (stdout=sillyloops.out)
  (stderr=sillyloops.err)
  (gmlog=gridlog)
  (architecture=i686)
  (cputime=10)
  (memory=32)
  (disk=1)
```

```
$ ngsub -f sillyloops.xrsl
```

Using a Storage Element

- Storage Elements are disk servers accessible via the Grid
- Allows to store input files close to the cluster where the program is executed, on a high bandwidth network
- Possibility to upload output files at a desired place:

```
(inputFiles=  
  ('input1', '/home/user/myexperiment'  
  ('input2', 'gsiftp://se.somewhere.ee/files/commondata'))  
  
(outputFiles=  
  ('output', 'gsiftp://se.somewhere.ee/mydir/result1')  
  ('prog.out', 'gsiftp://se.somewhere.ee/mydir/stdout'))  
  
(stdout='prog.out')
```

ROT13 example

- Simple perl script doing ROT13 encoding (shifting letters by 13 positions):

```
#!/usr/bin/perl
open(INPUT, '< input');
open(OUTPUT, '> result');
while (<INPUT>) {
    $_ =~ (tr/a-mn-z/n-za-m/);
    print OUTPUT $_;
}
```

- This could be any binary file written using any programming language
 - Either statically linked or dynamically loaded libraries must be available at the computing nodes

xRSL Example Using a Storage Element

- xRSL file for the ROT13 encoder of previous slide, uploading the job results to a storage element:

```
&(executable=rot13.sh)
(jobname=rot13-se)
(inputFiles=
(input='tallinn-meeting.txt'))
(outputFiles=
(result 'gsiftp://grid.tsl.uu.se/tutorial/text.encoded')
(rot13.out 'gsiftp://grid.tsl.uu.se/tutorial/rot13.out')
(rot13.err 'gsiftp://grid.tsl.uu.se/tutorial/rot13.err'))
(stdout=rot13.out)
(stderr=rot13.err)
(gmlog=gridlog)
(architecture=i686)
(cputime=10)
(memory=32)
(disk=1)
```

Gsincftp

- Grid file transfer tool using certificate based authentication
 - Based on the popular `ncftp` ftp client
- Normally, uses secure authentication but unencrypted data transfer
- Can be used to transfer files to and from storage elements
- Example session:

```
$ gsincftp grid.tsl.uu.se  
...Logged in to grid.tsl.uu.se.
```

```
$ cd tutorial
```

```
$ get rot13.out
```

Runtime Environments

- Software packages which are preinstalled on a computing resource
- Avoid the need of sending the binary at the start of executing a job
- Allow local optimizations (e.g. compiling to the installed architecture using optimized compiler flags)
- Very useful if there are many users of the same software or if the same program is used frequently
- Required runtime environment(s) can be specified in the job description file (xRSL file):

```
(runtimeenvironment=povray-3.5)
```

Real Jobs

- Real jobs usually send several subjobs to the Grid to solve a larger problem
- It is possible to send a parallel MPI job to a single cluster, but not a good idea to do MPI communication between clusters
- Splitting the job to suitable parts and gathering the parts together is left to the user
 - More error prone environment than traditional local systems
=> error checking and recovery important
 - Fault reporting and debugging has room for improvements
- Leif Nixon's example: Rendering an image in slices using the **povray** tool

References

- NorduGrid website: <http://www.nordugrid.org>
- The NorduGrid User Guide:
<http://www.nordugrid.org/documents/userguide.pdf>
- Balazs Konya's presentation at the 4th International Workshop on Grid Computing:
<http://www.nordugrid.org/slides/20031117-balazs.pdf>
- Povray example by Leif Nixon:
<http://www.nsc.liu.se/~nixon/ng-povray/>

Thank you!