NorduGrid Tutorial Exercises

Arto Teräs <arto.teras@csc.fi>
Juha Lento <juha.lento@csc.fi>

CSC Grid Workshop
March 31, 2004

1   Introduction

This document presents the examples and exercises of the NorduGrid Tutorial at CSC Grid Workshop. The Unix command prompt is represented with the dollar sign and text which should be entered by the user is written in typewriter font as follows:

```
$ command
```

Please don't feel restricted by the order of how examples and exercises are presented — explore, edit the xRSL files, try out different commands and parameters and ask questions.

2   Getting Started

The NorduGrid client software and examples of this tutorial can be found in directory `/mnt/dogmi/gridtutorial/`. Please install the client and examples by uncompressing the tar packages and set up the environment [1]:

```
$ tar zxvf /mnt/dogmi/gridtutorial/nordugrid-standalone-rh73-0.3.36-1.i386.tar.gz
$ cd nordugrid-standalone-0.3.36
$ . ./setup.sh
$ cd ..
$ tar zxvf /mnt/dogmi/gridtutorial/examples.tar.gz
$ cd examples
```

Certificates have already been generated and installed for all tutorial users. Therefore requesting a certificate is not necessary, just login to the grid:

```
$ grid-proxy-init
(the password is ''csctutor'')
```

---

[1]A few of the computers are running Red Hat 9.0; we have already installed the NorduGrid client on them. In this case you can skip the first tar command.

In the tutorial (31.3.2004, Espoo) we are using temporary certificates signed by a temporary Certificate Authority for this tutorial only. This limits the number of the available clusters as these tutorial identities and certificates do not belong to any generally authorized Virtual Organizations. Also, the usefulness of the Grid Monitor www interface is somewhat limited for these tutorial identities.

Have the NorduGrid User Guide at hand: `http://www.nordugrid.org/documents/userguide.pdf`, also included in the examples tar package.

Exercises:

- Print the certificate in text form by typing `grid-cert-info`. What is your identity in the grid? Who has signed the certificate?

- Logging in to the grid actually means creating a temporary access token called grid proxy. Print information of your proxy by typing `grid-proxy-info`. How long is it valid?

3 Submitting a Simple Job

Take a look at file `hellogrid.sh`. It is a simple shell script which writes "Hello Grid" on the standard output and sleeps for a while before returning. You can try to run it locally by typing

```
$ ./hellogrid.sh
```

The job description file to submit this script to the grid is `hellogrid.xrsl`:

```
& (executable=hellogrid.sh)
(stdout=hello.out)
(stderr=hello.err)
(gmlog=gridlog)
(architecture=i686)
(cputime=10)
(memory=32)
(disk=1)
```

Try to submit the job to NorduGrid:

```
$ ngsub -f hellogrid.xrsl
```

This may take a while to complete as UI first contacts the root information server, asks for clusters connected at the moment and then queries all the available clusters for their attributes etc. Then it starts preparing a job by transferring the input files. When the job is submitted, you should receive a message such as

```
Job submitted with jobid
  gsiftp://benedict.aau.dk:2811/jobs/2837896291031006429
```

In this case, the job was submitted to `benedict.aau.dk` in Denmark and the url `gsiftp://benedict.aau.dk:2811/jobs/2837896291031006429` is the reference to the job. The last part is a session directory chosen randomly by the target system. It is possible to check the status of the job using the `ngstat` command:

```
$ ngstat gsiftp://benedict.aau.dk:2811/jobs/2837896291031006429
Job gsiftp://benedict.aau.dk:2811/jobs/2837896291031006429
  Jobname: hellogrid
  Status: FINISHED 2004-03-29 16:15:18
```

In this case the job has been successfully completed. Other stages that the job may be in are described in the NorduGrid User Guide. Retrieve the results by typing

```
$ ngget gsiftp://benedict.aau.dk:2811/jobs/2837896291031006429
```

This downloads the result files and some statistics in the directory `2837896291031006429`. Take a look at the output (files `stdout` and `stderr` and `diag` file in the `gridlog` directory. What can you see?

Notice, that we made no reference to which cluster the job should go. If you would like to specify the cluster (or exclude some), it can be described in the xRSL file or on the command line:

```
$ ngsub -f hellogrid.xrsl -c datagrid3.csc.fi
```

Exercises:

- Try submitting the job with command `ngsub -f hellogrid.xrsl -d 1` to see more information about the submission process. Even more info is available with `-d 2`.

- Specify a job name by adding line `(jobname=hellogrid_your_name)` to the file `hellogrid.xrsl`. Submit the job again. Now you can refer to the job with the name instead of url when using `ngstat` and `ngget` commands.

- Submit some more jobs and try commands `ngkill` and `ngclean`.

- Specify three alternative clusters as accepted targets in the hellogrid.xrsl file. Try submitting the job. (Hint: Use the "`cluster`" attribute, see the User Guide for details.)

- Add a "`notify`" attribute in the xRSL file to receive email notifications of job status changes. See the User Guide for details.

4 Monitoring Jobs

Command `ngstat` was introduced in the previous section. Take a look at available options by typing

```
$ ngstat -h
```

The status of jobs can also be seen via the graphical Grid Monitor, which can be found on the NorduGrid web site `http://www.nordugrid.org`. Click on the "Grid Monitor" link at the top of the page.

The main view of the monitor shows currently connected resources. Most of the elements are links, clicking on them opens a new window giving more information of that particular resource. For example, click on a cluster name to view more information about that cluster, on the process bar to view more information about jobs running on the cluster, and "User base" link at the bottom of the window to view a list of users authorized to run jobs in NorduGrid.

For the members of Virtual Organizations (VOs) the jobs belonging to a certain user can be monitored through the Grid Monitor. When you want to start real use of the NorduGrid, first request a real certificate and then ask for membership in some VO, for example NorduGrid VO. To see what kind of information is available, you can select the NorduGrid VO and click on the names of the instructors, for example.

Exercises:

- What is the processor type in the Monolith cluster? How much memory is installed in the nodes?

- Which version of NorduGrid software and which runtime environments are installed in the Benedict cluster in Denmark?

- On which clusters is user "Balazs Konya" (Balazs is a member of the NorduGrid VO) authorized run jobs?

- Which Storage Elements have more than a terabyte of free disk space?

- Using the "Match yourself" dialog, it is possible to see the amount of resources available for any user, including the tutorial identities which are not part of Virtual Organizations. In the "Match yourself" dialog, select "Resource / object: user" and click "Next". In the first row of the following dialog, select "Name", " " (tilde) and fill in your user name in the last field. Then select the types of resources you want to see on the subsequent rows, for example "Free CPUs" and "Free disk space".

5   More Examples

5.1   Rspace example

This example demonstrates how to run a simple serial computation on the grid. The application is a first-principles real-space electronic structure program calculating the electronic structure of the CH4 molecule. Thanks to Tuomas Torsti for providing the example. In this case the (statically linked) executable is submitted to the grid as one of the job input files and no reference to Runtime Environments (software packages installed on the target cluster) is required. Basically we request a single i386 compatible PC.

Go to directory containing the material:

```
$ cd rspace
$ ls
CH4_LUCKY.xrsl   INPUT   potentials   rspace-0.81_i386-linux_SERIAL
```

The job description is in the file `CH4_LUCKY.xrsl`:

```
$ cat CH4_LUCKY.xrsl
&(executable=rspace-0.81_i386-linux_SERIAL)
(JobName=CH4_LUCKY)
(inputFiles=(INPUT "")
            (potentials/C "")
            (potentials/H ""))
(outputFiles=(energies "")
             (forces "")
             (WAVES_1 "")
             (POTENTIAL ""))
(CpuTime=10)
(memory=64)
(disk=10)
(stdout=stdout.txt)
(stderr=stderr.txt)
(gmlog=debugdir)
(|(architecture=i386)
  (architecture=i686))
```

First line defines the name of the executable. If it is not specified in the list of input files, it is automatically appended there. Edit the job name from `CH4_LUCKY` to `CH4_LUCKY_YOUR_FIRST_NAME` so you can differentiate the instance submitted by you from the others in the tutorial more easily.

Read from the User Guide how the location of the input and output files is resolved. That can be tricky with all the available locations...

Next some of the requirements for the job are specified, so that the user interface can select a suitable platform (cluster).

Submit the job!

```
$ ngsub -f CH4_LUCKY.xrsl
INPUT->INPUT      1 s:            0 kB            0 kB/s           0 kB/s    . . .
rspace-0.81_i386-linux_SERIAL->rspace-0.81_i386-linux_SERIAL     1 s:    . . .
rspace-0.81_i386-linux_SERIAL->rspace-0.81_i386-linux_SERIAL     2 s:    . . .
C->C    1 s:           0 kB           0 kB/s           0 kB/s    . . .
C->C    2 s:          64 kB          31 kB/s          32 kB/s    . . .
H->H    1 s:           0 kB           0 kB/s           0 kB/s    . . .
Job submitted with jobid gsiftp://ingvar.nsc.liu.se:2811/jobs/7009965451436415513
```

Monitor the job with `ngstat` and when it is finished, fetch the results with `ngget`.

The default time that the output files are kept on the remote site is 24 hours. In practice one will like to transfer the results back to some storage server (Storage Element, SE) automatically after the

5

completion of the job. That's achieved by specifying the target location in the xRSL file. The files can then be moved between different SEs using `ngcopy` for example. See the User Guide for details. Interactive FTP clients with Grid authentication are also available.

Exercises:

- A Storage Element is available at `grid.tsl.uu.se`. Log in to the storage element by typing "`gsincftp grid.tsl.uu.se`" and create a directory named `tutorial/_your_name` there. Then modify the `CH4_LUCKY.xrsl` file so that the output files are uploaded to the storage element. See the User Guide for details. Submit the job using `ngsub` and fetch results using `ngget` when it is completed. Now `ngget` should only fetch log files, standard output and standard error. Log in to the storage element again to get the actual result files.

5.2 Povray example

Leif Nixon has written a nice demonstration of distributed image rendering using the `povray` program. It is an example how to split one large job in pieces, submit all of them to the grid and fetch the results. More information is available at `http://www.nsc.liu.se/~nixon/ng-povray/`.

Some of the Python modules needed by the demo are not installed on the CSC classroom computers. However, we have a couple of accounts on another system which can be accessed using ssh. If you want to try the `povray` example, please ask the instructors for access.

Exercises:

- Download the demo and try to run it. Monitor the progress of the subjobs with `ngstat`. Merge the pieces as one composite image using `merge.py`. Did all the subjobs succeed? Take a look at the log files in subjob session directories. How long did it take to execute subjobs in different clusters?

6 Submitting Your Own Application

You can also try your own application to NorduGrid. It is easiest if you have a statically linked executable (compiled for Linux x86 platform) which does not require any software to be installed beforehand in the target cluster. Write a job submission file by taking the xRSL file from the `rspace` example as basis and modifying it. Try to submit the job, monitor the progress and fetch the results. Ask help from the instructors if needed.