The NorduGrid Toolkit
User Interface and Resource Broker

Mattias Ellert

Presented at the 4th NorduGrid workshop, Uppsala, 2002-11-11
The User Interface

- The NorduGrid toolkit user interface consists of a set of commands executed from the command line
  - `ngsub` — for job submission
  - `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 kill a running job
  - `ngclean` — to delete a job from a remote cluster
  - `ngsync` — to synchronize the local job info with the MDS
  - `ngmove` — to transfer files to, from and between clusters
The eXtended Resource Specification Language (XRSL)

- The Resource Specification Language (RSL) is used in the Globus Toolkit to describe a job and the resources needed for its execution
- Extendable by design
- The eXtended Resource Specification Language (XRSL) is RSL with NorduGrid additions
Simple Hello World Example

- The ngsub command is used to submit a job to a NorduGrid cluster
- The job and the resources needed to run it should be described using an XRSL string ...

```bash
ngsub '&(executable="/bin/echo") \ (arguments="Hello World") \ (stdout="stdout.txt") \ (stderr="stderr.txt")'
```
- ... or the XRSL can be read from a file

```bash
ngsub -f myjob.xrsl
```
An ATLAS example

&
  (executable="dc1.pileup.standard.v2.wrap4")
  (stdout="dc1.002099.lumi10.00001.NG.pyt_min_bias.log")
  (stdlog="gridlog.txt")
  (rsl_substitution=("GSIFTP" "gsiftp://lscf.nbi.dk/dc1/dc1-ds2099"))
  (join="yes")
  (executables=dc1.pileup.standard.v2.wrap4 dc1.pileup.standard.v2)
  (inputfiles=
    ("dc1.pileup.standard.v2.wrap4"
     "http://www.nbi.dk/~langgard/dc1.pileup.standard.v2.wrap4")
    ("dc1.pileup.standard.v2" "http://www.nbi.dk/~langgard/dc1.pileup.standard.v2")
    ("dc1.002099.simul.0001.hlt.pyt_min_bias.zebra"
     $(GSIFTP)/dc1.002099.simul.0001.hlt.pyt_min_bias.zebra)
    [ . . . ]
    ("dc1.002099.simul.0010.hlt.pyt_min_bias.zebra"
     $(GSIFTP)/dc1.002099.simul.0010.hlt.pyt_min_bias.zebra)
  )
  (outputFiles=
    ("dc1.002099.lumi10.00001.NG.pyt_min_bias.log"
     "gsiftp://grid.tsl.uu.se/test/dc1.002099.lumi10.00001.NG-Uppsala.pyt_min_bias.log")
    ("dc1.002099.lumi10.00001.NG.pyt_min_bias.zebra"
     "gsiftp://grid.tsl.uu.se/test/dc1.002099.lumi10.00001.NG-Uppsala.pyt_min_bias.zebra")
  )
  (jobname="dc1.002099.lumi10.00001.NG.pyt_min_bias")
  (runTimeEnvironment="TEST-ATLAS-4.0.1")
  (middleware="nordugrid-0.3.10")
  (maxCPUTime=1800)
  (maxDisk=20000)
  (ftpThreads=10)
Brokering — where do jobs go?

- The user interface contains a broker that decides to which queue at which cluster a job should be submitted.
- The decision is made based on information about the different clusters obtained from the MDS servers on the clusters, and from the Replica Catalog.
Brokering algorithm

- The status of the cluster and the queue must be ‘active’.
- The user must be authorized to use the cluster and the queue.
- The cluster’s and queue’s characteristics must match the requirements specified in the XRSL string (max CPU time, required free disk space, installed software etc).
Brokering algorithm (cont)

- If the job requires a file that is registered in a replica catalog, the brokering gives priority to clusters where there already is a copy of the file present.
- From all queues that fulfill the criteria one is chosen randomly, with a weight proportional to the number of free CPUs available for the user in each queue.
- If there are no available CPUs in any of the queues, the job is submitted to the queue with the lowest number of queued job per processor.
• A user’s manual for the NorduGrid toolkit user interface describing the commands and all their options, as well as a document containing the XRSL definition, are available from the NorduGrid website http://www.nordugrid.org

• The tutorial tomorrow will give plenty of examples on how to use the different user interface commands and how to construct an XRSL string