NorduGrid Tutorial

On the Testbed

overview of a Grid session

- user formulates the job requirements by editing an xrsl file
- having a valid proxy submits the job with ngsub
- the broker from the UI selects the target cluster, passes the job over to the GridManager and uploads the requested files from the submission machine
- after successful submission, a job handle (ID) is returned
  `gsiftp://seth.hpc2n.umu.se:2811/jobs/8632436256385296`
- from now on the GM takes care of the job
  - collects the requested input datafiles from the Storage Elements
  - submits the job to the Cluster Management System (PBS)
  - after job execution the GM uploads (if requested) the files to an SE
- Meanwhile the user may continously monitor the status of the job & Grid

Nordic Testbed for Wide Area Computing and Data Handling

Grid session cont.

- after job completion the user retrieves the output from the cluster (only those files which were already not uploaded to an SE)

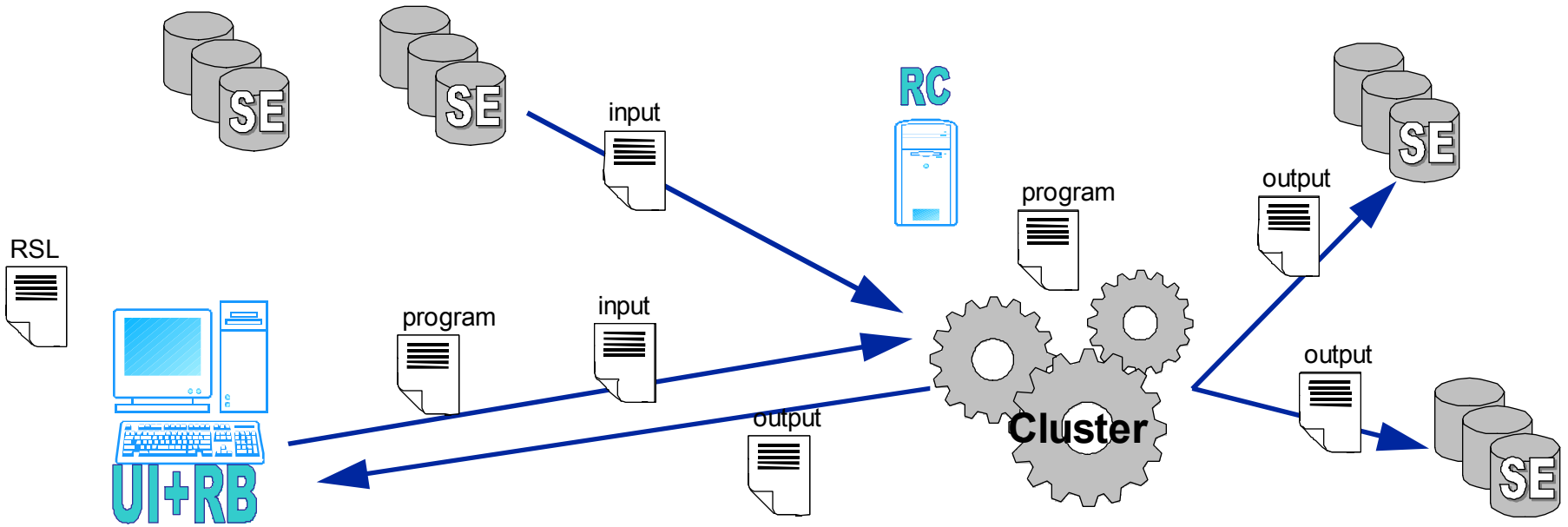
Grid session cont.

running on your desktop PC

- input is on your disk
- the program is on your disk,
- the output is kept on the disk

running on the Grid

- input is collected from the Grid (UI+SE)
- the program (binary) can be preinstalled (runtimeenvironment) or uploaded
- the output is distributed over the Grid or downloaded to the UI



what is there on the Grid?

- browse the NorduGrid LDAP Information Tree with the Map-based interface
  - locate the resources in different countries, cities
  - look into entries, check attributes, walk the tree
- Fire up the Loadmonitor
  - entries are clickable, clicking an entry performs an LDAP search over the Grid with respect to that attribute
  - check out the free resources for a particular user
- Use the `ngstat -c -l` UI command for getting information on clusters
- try out an `ldapsearch` command:

```
ldapsearch -h grid.quark.lu.se -p 2135 \
-b "mds-vo-name=sweden,o=grid" 'objectclass=nordugrid-cluster' -x dn
```

Nordic Testbed for Wide Area Computing and Data Handling

the “Hello Grid” XRSL

```
 &(executable=/bin/echo)(arguments="Hello Grid" )
(stdout="hello.txt")
(stderr="hello.err")
(stdlog="grid.debug")
(jobname="My Hello Grid")
(maxcputime=300)
(middleware="nordugrid-0.3.9")
```

Nordic Testbed for Wide Area Computing and Data Handling

“hello Grid” exercise

- download the hello_grid.tgz
- submit your first “hello Grid” job
  - ngsub -f hello_grid.xrsl
- check the job status
  - ngstat <jobID>
- submit your second “hello Grid job”
  - ngsub -d 1 -f hello_grids.xrsl
- check all your jobs, get the output
  - ngstat -a; ngget <jobid>
- submit the modified “say_hello.xrsl”, request a cluster from Uppsala
  - ngsub -f say_hello.xrsl -c uppsala

the sleepy job exercise

- get the dream.tgz package
- run the sleepy_script.sh locally
  - ./sleepy_script.sh
- upload the job to the Grid
  - ngsub -f dream.xrsl
- play with the ng commands
  - ngstat, ngkill, ngclean,ngget
- Use the information system for monitoring the Grid
  - www.nordugrid.org -> Loadmonitor, or InformationSystem

The Mandelbrot exercise

- download the mandel.tgz
- run the small program locally on your machine
  - ./generate_mandel.bin < parameters.inp
  - check out the generated figure
- look at the generated figure:
  - kview figure.ppm
- submit the same job to the Grid
  - ngsub -f mandel.xrsl -d 1
- monitor, your job, peek into the stdout
  - ngstat <jobid> ; ngget <jobid>
- submit several jobs, try to kill some, clean up the mess
  - ngkill <jobid>; ngclean -a

data access

- obtain the data_access.tgz
- submit the xrsl
  - ngsub -f test_replica.xrsl -d 1
- use the gsincftp client to peek into the sessiondir
  - gsincftp <the machine where your job runs>

brokering exercise

- imitate the jobsubmission, play with the UI without submitting real jobs (the UI performs a fake jobsubmission)
  - ngsub -f <one_of_the_previous.xrsl> -d 1 -dumpxrsl
- try to follow the brokering steps described in the brokering.txt file

Nordic Testbed for Wide Area Computing and Data Handling

a real life example

- run the validation test job of the High Energy Physics Atlas Data Challenge (exercises/HEP directory):
  - ngsub -f dc1_test.xrsl
- try to understand
  - where are the inputfiles taken from?
  - where comes the binary from?
  - what is the role of the “runtimeenvironment”
  - what happens to the results?