gqsub

Grid Computing at the Mesoscopic scale

http://www.scotgrid.ac.uk/gqsub/ Stuart Purdie, University of Glasgow

IEEE standard interface

IEEE 1003.2d defines the 'qsub' interface. Based off the NQS system, this interface was designed for distributed parallel batch systems, and as such, encompasses most of the concepts needed for Grid utilisation.
It is a standard that is widely deployed and readily familiar.

Mesoscopic scale computing

Not big enough to demand a Grid solution, but not small enough to fit easily onto local systems, meso scale computing occupies the murky middle ground between small and large.
A key way to identify these scenario is to note two things – that they use a local cluster already, and the cluster is nearly always full.
They can clearly benefit from the Grid, spreading around the jobs when the cluster is full, and working for the Grid when it's only partly full.
However, unlike the large VO's, they have recourse to give up on the Grid if it's too difficult – in the jargon of Human Computer Interaction, they have lower investment.
If the Grid is 'too difficult' to use, they can and will just wait on the local cluster instead.

Implementation details

gqsub is implemented as a set of python scripts, that act as a translation layer onto glite-wms-job*, voms-proxy* etc.
Accordingly it requires a gLite UI to be installed already.
Installation is as simple as putting the python scripts in a directory on your path.
It's been tested on gLite 3.1 and 3.2 UI's, and works on both.
It works well in combination with the TAR UI, placed on the head node of a cluster, to give easy selection between local and Grid jobs.

```
#!/bin/sh

#$ -l cput=0:30:00
#$ -l walltime=0:30:00

#GQSUB -W stagein=isinganneal
#GQSUB -W stagein=nio.geom

isinganneal --geometry nio.geom \
    --min-temp $1 --max-temp $2 \
    --temp-steps $SGE_TASK_LAST \
    --step $SGE_TASK_ID
```

Example of a submission script

In local mode a shared filesystem is used, hence the additional directives to gqsub to stage the needed executable and datafile for Grid jobs.
The CPU and wall time directive's are picked and used in both local and Grid modes.
Note that the semantics of the job are spread between bash, qsub, and the executable. This is a fairly typical blend, and supported by gqsub.

Key features

Implements the familiar POSIX qsub + qstat user interface
Submits via WMS – meta-scheduling by default
Handles proxies transparently – if needed, it will ask for password
Low cognitive load for existing cluster users

Future Work

gqsub is in use, and works well where there is a shared filesystem between the UI and the worker node (e.g. AFS).
There is experimental support for automatically shipping staged out files back to the user on job completion (where there is a GridFTP server on the UI).
There is also experimental support for collecting the output when qstat reports that a job is complete.

```
glite-ui$ gqsub -t 0-10 example.sh 10 160
File stagein requested - note this captures files at time of submission, rather than at execution time Torque does.
Proxy renewal required - looking for 36:45:00
Enter GRID pass phrase: ********
Submitting job 326 as:  https://svr023.gla.scotgrid.ac.uk:9000/L8-TVzIO2xq_urB6WoOzvA
glite-ui$ gqstat
Getting status for 315
Getting status for 326
Job id Name    User          WallTime State         Queue
------ ------- ------------- -------- ------------- ----------------------------------------------------
315            stuart_purdie 11:40:18 Done          dev010.gla.scotgrid.ac.uk:2119/jobmanager-lcgpbs-q3d
322            stuart_purdie 00:05:28 Exit Code !=0 svr026.gla.scotgrid.ac.uk:2119/jobmanager-lcgpbs-q30m
326            stuart_purdie          Waiting
326.0  Node_0  stuart_purdie          Submitted
326.1  Node_1  stuart_purdie          Submitted
326.10 Node_10 stuart_purdie          Submitted
326.2  Node_2  stuart_purdie          Ready         svr021.gla.scotgrid.ac.uk:2119/jobmanager-lcgpbs-q3d
326.3  Node_3  stuart_purdie          Submitted
326.4  Node_4  stuart_purdie          Submitted
326.5  Node_5  stuart_purdie          Submitted
326.6  Node_6  stuart_purdie          Submitted
326.7  Node_7  stuart_purdie          Ready         svr021.gla.scotgrid.ac.uk:2119/jobmanager-lcgpbs-q6h
326.8  Node_8  stuart_purdie          Ready         svr021.gla.scotgrid.ac.uk:2119/jobmanager-lcgpbs-q1d
326.9  Node_9  stuart_purdie          Submitted
```

Example of use

The output of qstat is familiar, and almost within the IEEE standard – extracting the currently used CPU time isn't possible in general, so the Wall time is reported in it's place.
The native identifier for the job is also reported, and can be used directly with gqstat, along with it's generated sequence number.

ScotGrid
Scottish Grid Service

Enabling Grids for E-sciencE