Final hands on session

Stefano Cozzini/Riccardo di Meo

A first action:

- Cd ~/UIPnPcomb
- Create a directory
  glite/var/tmp/glite-ui
  (mkdir -p glite/var/tmp/glite-ui)

- Just to avoid problems with glite-job-submit weird options..

four different exercises

- Browse the ldap server with the help of graphical tools
- Submit bunchs of jobs on different Rbs
  - Glite-RB
  - Lcg-rb
  - (files are in: /afs/ictp/public/d/dimeo/WORLD/exercise2
- Submit a job that needs data in it
  - Files are in: /afs/ictp/public/d/dimeo/WORLD/exercise3
- Measure performance of CE's on the grid..

Ldap browser

- Download the tgz package from:

```
wget http://egrid-11.egrid.it/ldapbrowser.tgz
```

- 2) Untar the package:

```
tar –zxvf ldapbrowser.tgz
```

- 3) Enter the directory:

```
cd ldapbrowser
```

- 4) Run ldabrowser:

```
./lbe.sh
```

- 5) Select the GRID to browse

Jobs submissions on different RB

- Aim: evaluate how fast are different Rbs
- Glite-job-submit:
  - The glite resource broker:
    - glite-rb.ct.infn.it
- Edg-job-submit:
  - The lcg resource broker:
    - grid004.ct.infn.it

Actions

- Take the script 200_jdl.sh and fibo.sh.
  - modify 200_jdl.sh in in order to let it generate 10 jdls (instead of 200...) and run it.
  - Use glite to submit the jobs, saving the ids in a file called glite_id.txt and running the command as:
    - /usr/bin/time submit_bunch_glite.sh
  - Same thing, but using the edg-job-submit
    - /usr/bin/time submit_bunch_edg.sh
- Check the status of both simulations with (edg|glite)-job-status and time both commands.

Actions (II part)

- At the end, retrieve all data and verify how much time is needed for each group of command.
- What is the fast one ?
- Please summarize in your final report..

Job submission&data management

- AIM:
  - Submit a job that will find its input quite close where to run and it will store its output somewhere on the grid
- JDL parameters needed for input:
  - DataAccessProtocol = "gsiftp";
  - NB: option DataAccessProtocol = "gsiftp"; is needed once in the script...
  - InputData = "lfn:/grid/gilda/trieste/my name/original";
- Please note that JDL option above are just to identify the closest SE on the GRID that contains the input file.
- The input file should be loaded by you specifying the lcg-cp

The JDL (first part)

```
[
   Executable= "file_operations.sh";

   StdOutput = "stdout";
   StdError  = "stderr";

   InputSandbox  = {"file_operations.sh"};
   OutputSandbox = {"stderr","stdout"};


  # DataAccessProtocol="gsiftp";
  # InputData = {"lfn:/grid/gilda/trieste/MYNAME_original"};

....
```

The JDL (second part)

```
 ....

  OutputData={
     [
       OutputFile="WN-modified";
       StorageElement="gilda-se-01.pd.infn.it";
       LogicalFileName="lfn:/grid/gilda/trieste/MYNAME_modified";
     ]
  };

]
```

The script file_operations.sh

```
#!/bin/bash


# Check the content of the directory at job start.
ls -l

# Download the file from SE
lcg-cp --vo gridats lfn:/grid/gilda/trieste/MYNAME_original file:`pwd`/WN-copy

# Verify that the file have been correctly downloaded from the SE
# and print its content
ls -l WN-copy
cat WN-copy

# Let's create another file (which we will save on the grid) by
# substituting some contents of the old.
cat WN-copy \
    |sed s/Rules/Rules\ a\ lot/  \
    > WN-modified
# We don't need to manually upload the data on the SE!
# the grid will take care of this!
```

Actions (on your side)

- Take a look at JDL and shell scripts.
- Create a local file in your working dir:
  - echo 'grid Rules!' >local_file
- Create a file on the catalog:
  - lcg-cr –vo gilda -d se lnf:/grid/gilda/trieste/myname_original file:`pwd`/local_file/
- Check if the file is ok (created)
- Submit the job
  - edg-job-submit/glite-job-submit file_operations.jdl

Actions (part 2)

- At the end of job
  - Verify a new file is created on the catalog
  - Download the output on the file

Exercise 4

- Aim:
  - Run a CPU intensive application to check performances on different CE's
- The software:
  - A c small program to perform matrix multiplication .. (or any other program you would like to use..)
    - matrix.c
  - A jdl script
    - matrix.jdl
  - A small shell script
    - matrix.sh

Actions

- Take a look at matrix.c and compile it
  - Gcc -O2 -Wall -o matrix matrix.c -static
- Execute it on your local machine to estimate computing times:
  - /usr/bin/time ./matrix.x 500 (size) 1 (# of seeds)
- Submit the jdl job on the grid randomly..
  - edg-job-submit matrix.jdl
- Or Identify which are the “good ones”
  - lcg-infosite –vo gilda ce -v 2
- And submit accordingly..

Actions

- Rank CE accordingly to their speed...

Final questionnaire/report

1. Which are the queues' lenghts on CE grid004.iucc.ac.il (or any other CE on gilda grid)
   - (hint: play with the ldap browser)
2. Which is the largest storage element in Gilda grid?
3. How many different CPU types do we have ?
4. Do you have any estimate about the latency in job submission on glite/lcg WMS?
5. Give us the list of the most performing Wns on the gilda grid with respect to a test program (exercise 4)