Grid Computing School
July 10-12, 2006.
Rio de Janeiro

Practice of EGEE Job execution

Goal: Practice job management in gLite.
You will solve a linear equation system A*x=B by matrix operations as it is described in
http://www.lpds.sztaki.hu/pgportal/v23/includes/Equation_Solver.html

The submissions of jobs Invert_A and Multip_B are needed to get result, which will be gained by command line operations.

Preparation:

You have received a two digit number between 30-60. (**XY**). Use this number to login to one of the machines:

Account: budapest**XY**
Password: GridBUD**XY**

Open a terminal window and login to the GILDA UI machine:

```
ssh budapestXY@glite-tutor.ct.infn.it
```

Password: GridBUD**XY**

1. **Controlling the existence of your certificate** - "usercert.pem" and "userkey.pem" must exist:

```
ls -al ~/.globus
```

2. **Gaining access right to work in the grid by creating a short term proxy certificate** **(voms-proxy-init, voms-proxy-info, voms-proxy-destroy)**

You will be asked for a "Grid pass phrase" (lock on your secret key file). It is: **BUDAPEST.**

2.1 Crating your proxy

Usage:

```
voms-proxy-init --voms <VO>
```

Example:

```
> voms-proxy-init --voms gilda
Your identity: /C=HU/O=NIIF CA/OU=GRID/OU=SZTAKI/CN=Hermann Gabor
```

```
Enter GRID pass phrase:
Creating temporary proxy
...................................................... Done
Contacting  voms.ct.infn.it:15001 [/C=IT/O=GILDA/OU=Host/L=INFN
Catania/CN=voms.ct.infn.it/Email=emidio.giorgio@ct.infn.it] "gilda"
Done
Creating proxy ...................................................
Done
Your proxy is valid until Tue Jun 27 19:53:18 2006
```

2.2 Controling your existing proxy

Usage:

```
voms-proxy-info [-all]
```

Example:

```
> voms-proxy-info -all
subject   : /C=HU/O=NIIF CA/OU=GRID/OU=SZTAKI/CN=Hermann
Gabor/CN=proxy
issuer    : /C=HU/O=NIIF CA/OU=GRID/OU=SZTAKI/CN=Hermann Gabor
identity  : /C=HU/O=NIIF CA/OU=GRID/OU=SZTAKI/CN=Hermann Gabor
type      : proxy
strength  : 512 bits
path      : /tmp/x509up_u505
timeleft  : 11:54:39
VO        : gilda
subject   : /C=HU/O=NIIF CA/OU=GRID/OU=SZTAKI/CN=Hermann Gabor
issuer    : /C=IT/O=GILDA/OU=Host/L=INFN
Catania/CN=voms.ct.infn.it/Email=emidio.giorgio@ct.infn.it
attribute : /gilda/Role=NULL/Capability=NULL
timeleft  : 11:54:39
```

Note:

The highlighted area shows the Virtual Organization Management Service (VOMS) extension of the proxy. (Details during the security lecture on Day 2.)

2.3 Removing an existing proxy

Usage:

```
voms-proxy-destroy
```

Note:

If you executed the voms-proxy-destroy command then the voms-proxy-init command must be repeated! (see 2.1)

3. Make and install a subdirectory for submit the Matrix inversion job.

Copy from the source repository the Input file “A_in_Ax_EQUAL_B.txt” as “INPUT1” ,
the matrix operation program “MatrixDemoNew” with the original name
and the job submission Program “Invert_A.jdl” also with the original name.
Example:

3.1 Make an own working directory:

```
mkdir <workdir>
```

3.2 Download and decompress the teaching material:

```
wget http://www.sztaki.hu/~ghermann/Szemelyes/SommerScool_06/SummerSchool_06.zip
unzip -x SummerSchool_06.zip
```

3.3 Copy the needed files

```
cp ~/SummerSchool_06/EquSolver/A_in_Ax_EQUAL_B.txt <workir>/INPUT1
cp ~/SummerSchool_06/EquSolver/MatrixDemoNew       <workir>
cp ~/SummerSchool_06/EquSolver/Invert_A.jdl        <workir>
```

At this point your current directory should look like this:

```
-rw-r--r--    1 budapest65 users         111 Jun 30 14:45 INPUT1
-rw-r--r--    1 budapest65 users         266 Jun 30 14:48 Invert_A.jdl
-rw-r--r--    1 budapest65 users       18816 Jun 30 14:48 MatrixDemoNew
```

Important!
The renaming of **A_in_Ax_EQUAL_B.txt** to **INPUT1** is needed as the executable **MatrixDemoNew** expects input file(s) as **INPUT**1 (and **INPUT2**)

4. Check the correctness of the JDL file

(glite-job-list-match)

4.1 Investigate the jdl file Invert_A.jdl with a text editor.

Observe the role of the InputSandbox, OutputSandbox,.

4.2 Determine where the job can run:

Usage:

```
glite-job-list-match [ -o <resource_list> ][--vo <VO>] <JDL_FILE>
```

Example:

4.2.1 Change to the working directory:

```
cd <workdir>
```

4.2.2 Execute the job submission

```
glite-job-list-match -o res.list --vo gilda Invert_A.jdl
   Selected Virtual Organisation name (from proxy certificate
extension): gilda
Connecting to host glite-rb.ct.infn.it, port 7772
=============== glite-job-list-match output ===================
 Computing Element(s) matching your job requirements have been
stored in the file:
/home/hermann/WrkForInvert/res.list
=================================================================
```

Hint:

You can see in the file `res.list` the CE-s where the program may run.

5. Submit the first job

(`glite-job-submit`)

Usage:

```
glite-job-submit [-i <resource_list>] [ -o <job_identifier_file>]][--vo
<VO>] <JDL_FILE>
```

Example:

glite-job-submit -i res.list -o InvertJobIdent --vo gilda Invert_A.jdl

```
Selected Virtual Organization name (from proxy certificate extension): gilda
--------------------------------------------------------------------------
1 : COMPUTING ELEMENT IDs LIST
2 : The following CE(s) matching your job requirements have been found:
3 : dgt01.ui.savba.sk:2119/jobmanager-lcgpbs-long
4 : dgt01.ui.savba.sk:2119/jobmanager-lcgpbs-short
5 : gilda01.ihep.ac.cn:2119/jobmanager-lcgpbs-infinite
6 : gilda01.ihep.ac.cn:2119/jobmanager-lcgpbs-long
7 : gilda01.ihep.ac.cn:2119/jobmanager-lcgpbs-short
8 : gildace01.roma3.infn.it:2119/jobmanager-lcgpbs-infinite
9 : gildace01.roma3.infn.it:2119/jobmanager-lcgpbs-long
10: gildace01.roma3.infn.it:2119/jobmanager-lcgpbs-short
11: grid004.iucc.ac.il:2119/jobmanager-lcgpbs-long
12: grid004.iucc.ac.il:2119/jobmanager-lcgpbs-short
13: grid010.ct.infn.it:2119/jobmanager-lcgpbs-long
14: grid010.ct.infn.it:2119/jobmanager-lcgpbs-short
15: trigrid-ce00.unime.it:2119/jobmanager-lcgpbs-infinite
16: trigrid-ce00.unime.it:2119/jobmanager-lcgpbs-long
17: trigrid-ce00.unime.it:2119/jobmanager-lcgpbs-short
18: trigriden00.unime.it:2119/jobmanager-lcgpbs-infinite
19: trigriden00.unime.it:2119/jobmanager-lcgpbs-long
20: grid011f.cnaf.infn.it:2119/jobmanager-lcgpbs-short
21: trigriden00.unime.it:2119/jobmanager-lcgpbs-short
22: grid011f.cnaf.infn.it:2119/jobmanager-lcgpbs-long
23: dgt01.ui.savba.sk:2119/jobmanager-lcgpbs-infinite
24: grid004.iucc.ac.il:2119/jobmanager-lcgpbs-infinite
25: grid011f.cnaf.infn.it:2119/jobmanager-lcgpbs-infinite
26: grid010.ct.infn.it:2119/jobmanager-lcgpbs-infinite
27: gildace.oact.inaf.it:2119/jobmanager-lcgpbs-infinite
28: gildace.oact.inaf.it:2119/jobmanager-lcgpbs-long
29: gildace.oact.inaf.it:2119/jobmanager-lcgpbs-short
q : quit
--------------------------------------------------------------------------

Choose a CEId in the list - [1-29]1:12


Connecting to host glite-rb.ct.infn.it, port 7772
Logging to host glite-rb.ct.infn.it, port 9002
```

```
============== glite-job-submit Success ============================================
 The job has been successfully submitted to the Network Server.
 Use glite-job-status command to check job current status. Your job identifier is:

 - https://glite-rb.ct.infn.it:9000/NMjN_CTMXEuP7dTApn_zjg

 The job identifier has been saved in the following file:
 /home/hermann/WrkForInvert/InvertJobIdent
=====================================================================================
```

6. Observe the status of the job

(glite-job-status,
glite-job-cancel)

Usage:

```
glite-job-status { -i <job_identifier_file> | < job_identifier>
```

Simple way:

Example:

```
glite-job-status https://glite-rb.ct.infn.it:9000/NMjN_CTMXEuP7dTApn_zjg
```

```
*************************************************************
BOOKKEEPING INFORMATION:

Status info for the Job : https://glite-
rb.ct.infn.it:9000/NMjN_CTMXEuP7dTApn_zjg
Current Status:     Done (Success)
Exit code:          0
Status Reason:      Job terminated successfully
Destination:        grid004.iucc.ac.il:2119/jobmanager-lcgpbs-short
Submitted:          Thu Jun 29 09:10:41 2006 CEST
*************************************************************
```

More intelligent way:

```
glite-job-status -i InvertJobIdent
```

The result is the same as in the case of **6.1**

Automatic polling:

```
watch “glite-job-status -i InvertJobIdent”
```

6.4 Canceling a running job

Usage:

```
glite-job-cancel [ -o <resource_list> ][--vo <VO>] <JDL_FILE>
```

Please do not apply this command if it is not necessary!

7. Fetch the result

(glite-job-output)

Usage:

```
     glite-job-output [−-dir <result_directory>]
                      { -i <job_identifier_file> | < job_identifier> }
```

Example:

If the job reached `Status Done (Succes)` the result can be fetched.

7.1 Let us make a subdirectory for the result:

```
mkdir result
```

7.2 Fetching the result:

```
glite-job-output --dir ./result -i InvertJobIdent
```

```
Retrieving files from host: glite-rb.ct.infn.it ( for https://glite-
rb.ct.infn.it:9000/NMjN_CTMXEuP7dTApn_zjg )

********************************************************************************
                        JOB GET OUTPUT OUTCOME

 Output sandbox files for the job:
 - https://glite-rb.ct.infn.it:9000/NMjN_CTMXEuP7dTApn_zjg
 have been successfully retrieved and stored in the directory:
 /home/hermann/WrkForInvert/result/budapest65_NMjN_CTMXEuP7dTApn_zjg

********************************************************************************
```

7.3 Evaluation

In the subdirectory `result` you find the machine generated subdirectory of form `budapest`**`XY`**`_<machinegenerated>` which contains files with the names have been determined by the fragment **`OutputSandbox`** of the `<JDL_FILE>.`
See **std.out** as the message of the job and the OUTPUT the result of the matrix inversion operation. As this value will be one of the inputs of the next job it must be copied.

Example:

```
cat result/budapest65_NU0VmU2iZMH4sCVs376GgA/std.out
```

You should see the listing of the inverted matrix.

8. Prepare the next –matrix multiplication - job and copy the result of the first job as “INPUT1”:

Note that `INPUT2` (**B**) is copied from the source repository, and `INPUT1` ($\mathbf{A}^{-1}$) copied from the result part of the previous job.

Example:

8.1 Make a different working directory <new_working_directory>:

Example:

```
cd ~
mkdir <new_working_directory>
```

```
cd <new_working_directory>
```

8.2 Copy the result (“OUTPUT”) of the recent job with a new name “INPUT” in the new directory and copy the common executable “MatrixDemoNew” in the same directory.

```
cp ../<workdir>/<result_directory>/<machine_generated_name>/OUTPUT  INPUT1
cp ../<workdir>/MatrixDemoNew .
```

8.3 Copy the file of the vector B (“B_in_Ax_EQUAL_B.txt”) as INPUT2 and copy the proper jdl file of the new operation from SummerSchool_06/EquSolver into the new directory:

```
cp ~/SummerSchool_06/EquSolver/B_in_Ax_EQUAL_B.txt INPUT2
cp ~/SummerSchool_06/EquSolver/Multip_B.jdl .
```

8.4 Check the result of copies

```
ls –l
```

At this point you must see something like:

```
-rw-r--r--    1 budapest65 users         242 Jun 30 15:11 INPUT1
-rw-r--r--    1 budapest65 users          31 Jun 30 15:16 INPUT2
-rw-r--r--    1 budapest65 users       18816 Jun 30 15:11 MatrixDemoNew
-rw-r--r--    1 budapest65 users         278 Jun 30 15:16 Multip_B.jdl
```

9. Submission the second job

Example:

```
glite-job-submit  -o MultipJobIdent --vo gilda Multip_B.jdl
```

10. Watch the status:

Example:

```
watch "glite-job-status -i MultipJobIdent"
```

The output can be similar to:

```
Every 2s: glite-job-status -i MultipJobIdent
Fri Jun 30 15:35:54 2006


*************************************************************
BOOKKEEPING INFORMATION:

Status info for the Job : https://glite-rb.ct.infn.it:9000/uk477Cd0Ysl-
SSCl9kBSYw
Current Status:     Scheduled
Status Reason:      Job successfully submitted to Globus
Destination:        dgt01.ui.savba.sk:2119/jobmanager-lcgpbs-short
Submitted:          Fri Jun 30 15:34:45 2006 CEST
*************************************************************
```

Hint:

`CTRL c` can interrupt the command.

11. Fetch the results after success

Example:

```
mkdir result
glite-job-output --dir ./result -i MultipJobIdent
```

And let us print the result matrix , which is a vector corresponding to our expectation (1, 2, 3, 4, 5)

```
cat result/budapestXY_<machine_generated>/OUTPUT
```

```
5 1 0.999756 2.000208 3.000052 3.999896 4.999768
```