

Guidelines for porting applications to the EGEE Grid

D. Weissenbach

The general idea is to use a (shell, perl, python, tcl, ...) script as Executable of the job description.

This script will

1. check/set environment (`$LFC_HOST`, `$LCG_GFAL_VO`, `$PATH`, `$LD_LIBRARY_PATH`)
2. unpack InputSandbox
3. prepart execution subtree
4. perform input file management
5. compile
6. execute the code, control the execution
7. perform output file management
8. exit an informative value

- Specify minimum wanted CPU & Wall times;
- Pass environment (Environment clause) for InputData matching;
- Specify MyProxyServer if different from default config ¹;
- While debugging, Use RetryCount = 0;

WMPProxy (current version of WMS) manual:

<https://edms.cern.ch/document/674643>

JDL attributes description:

<https://edms.cern.ch/document/590869>

¹in this order: \$GLITE_LOCATION/etc/glite_wms.conf,
\$GLITE_LOCATION/etc/vo_name/glite_wms.conf,
\$HOME/.glite/vo_name/glite_wms.conf, or \$GLITE_WMS_CLIENT_CONFIG or
option -config (-c) to glite-wms-job-submit

Except the Executable, files from the InputSandbox generally come without execution bit set \Rightarrow use a tarball (or set modes).

Allow file management commands to fail:

```
lcg-cp lfn:$myFile ...  
  
for rep in $(lcg-lr lfn:$myFile) ; do  
    lcg-cp $rep file:$PWD/$myFile && break  
done  
[ -s $myFile ] || exit 1
```

No inbound connectivity on Worker Node \Rightarrow use passive ftp.

During porting, increase verbosity levels (bash: set -xv, csh: set verbose ; set echo)

Generally grid Worker Nodes have the linux software default installation (GNU make, gcc&g++ v.3 **but no fortran**, perl, python, tcl, ...), and SUN jdk.

Few libraries (< 50% sites with blas/lapack) or scientific software (scilab, octave, gdl), and **no** licensed code.

The SOFTWARE MANAGER of the VO has the possibility to install important/large programs and libraries under \$VO_XYZ_SW_DIR (\$VO_ASTRO_VO_EU_EGEE_ORG_SW_DIR, other VOs!)

```
Select CE with Requirement = Member( "tag", \
other.GlueHostApplicationSoftwareRunTimeEnvironment);
```

However it is generally a good idea to send **statically compiled** code with the job, especially if you use a licensed compiler.

If architecture specific optimisations are used, have several executables (why not storing them on SEs?). EGEE clusters today only run one OS version (32 or 64 bits) but may mix intel and AMD CPUs.

Interpreted languages have some advantages concerning porting issues.

However many of them (perl, python) are largely dependant on **compiled** modules, usually written in C.

If the appropriate modules have not been installed on the Worker Node, they must be brought by the job. Relevant factors:

- platform & OS arch.
- version of interpreter and module
- version of C compiler

Software Managers may also compile and install alternative versions for several interpreters.

matlab^(tm) has released a Grid capable toolkit to matlab 2009, available on some EGEE sites.

jd1 attributes: `JobType='MPI'; NodeNumber=N;`

Two distinct environments:

1. new:

- Software tags: MPI-START, MPI_SHARED_HOME — MPI_NO_SHARED_HOME, mpi flavors;
- Call of `mpirun` configured by installed `mpi-start.sh` script;
- MPI flavor and other parameters controlled by environment variables;
- user supplied pre- and post-execution scripts;
- Doc: http://egee-uig.web.cern.ch/egee-uig/production_pages/MPIJobs.html.

2. old:

- Software tag MPI flavor;
- wrapper script as usual, must call `mpirun` with good parameters;
- Doc: section 3 of <http://grid-it.cnaf.infn.it/fileadmin/sysadm/mpi-support/MPINotes.txt>.

Problems arise when identification/authorisation on the DB side is involved. Anonymous/passwordless DB queries are possible if the appropriate DB client is available on the WN.

For controlled DB access from Grids, several tools have been designed, performing a mapping between Grid users proxies and DB users. Non exhaustive list:

- AMGA <http://amga.web.cern.ch>: gLite official, voms, APIs & CLI, AMGA dialect
- OGSA-DAI <http://www.ogsadai.org.uk/>: no voms no VO, java API, SQL & xml
- GRelC <http://grelc.unile.it>: no voms but VO, APIs & CLI, SQL & xml

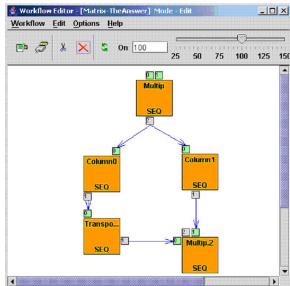
Act as a proxy service between the DB and the Grid.
Distributed queries or DB agregation capable.

The p-grade portal is a graphical tool that allows the user to build Grid workflows of jobs like the gLite DAGs.

Jobs may run scalar or parallel code.

It also has a ways to

- handle parametric studies;
- works with gLite, GT2 and GT4 grids;
- has an interface for non grid codes, seen as external services (GEM/LCA).



<http://p-grade.hu>

- Structures, heklpdesks
 - GGUS, <https://gus.fzk.de>;
 - NA4 <http://egeena4.lal.in2p3.fr>, RESPECT
<http://technical.eu-egee.org/index.php?id=290>;
 - GASuC, <http://www.lpds.sztaki.hu/gasuc>.
- Tools
 - VOBOX;
 - g-Eclipse, <http://www.geclipse.org>;
 - GANGA, DIANE;
- ... ???