

## Guidelines for porting applications to the EGEE Grid

*D. Weissenbach*

[www.eu-egee.org](http://www.eu-egee.org)



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. prepare execution subtree
3. unpack InputSandbox, compile
4. perform input file management
5. execute the code, control the execution
6. perform output file management

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 (except matlab).

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.

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

Two distinct environments:

1. new:

- Software tag: MPI-START
- Call of `mpirun` configured by middleware;
- user supplied pre- and post-execution scripts;
- Doc: [http://egee-uig.web.cern.ch/egee-uig/production\\_pages/MPIJobs.html](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>.

- Specify minimum wanted CPU & Wall times;
- Pass environment (Environment clause) for InputData matching;
- Specify MyProxyServer if different from default config <sup>1</sup>;
- 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>

---

<sup>1</sup>in 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