



Enabling Grids for E-science

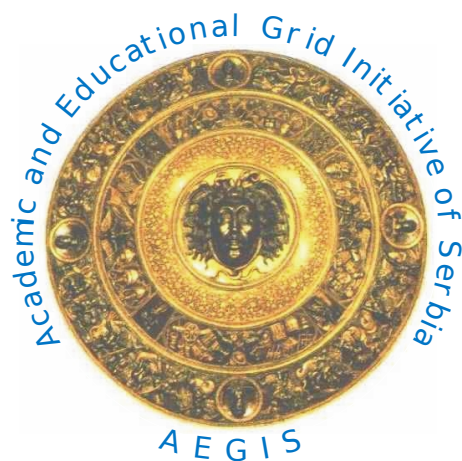
# gLite Job and Data Management

Nikola Grkić

ngrkic@ipb.ac.rs

Scientific Computing Laboratory

Institute of Physics Belgrade, Serbia



SEE-GRID-SCI  
SEE-GRID infrastructure for regional eScience



Information Society



Feb. 19, 2010

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

- **Job Description Language (JDL)**
- **Job management**
- **MPI example**
- **Data Management**

- **Programming languages (C, C++, Fortran, JAVA...)**
- **Script languages (Perl, Python, Bash...)**

- **JDL**

The Job Description Language (JDL) is a high-level script language used to describe jobs and aggregates of jobs with arbitrary dependency relations

- **JDL syntax**

attribute = expression;

- **Simple Job**

Simple jobs usually contains program, and do not require some specific resources or demands.

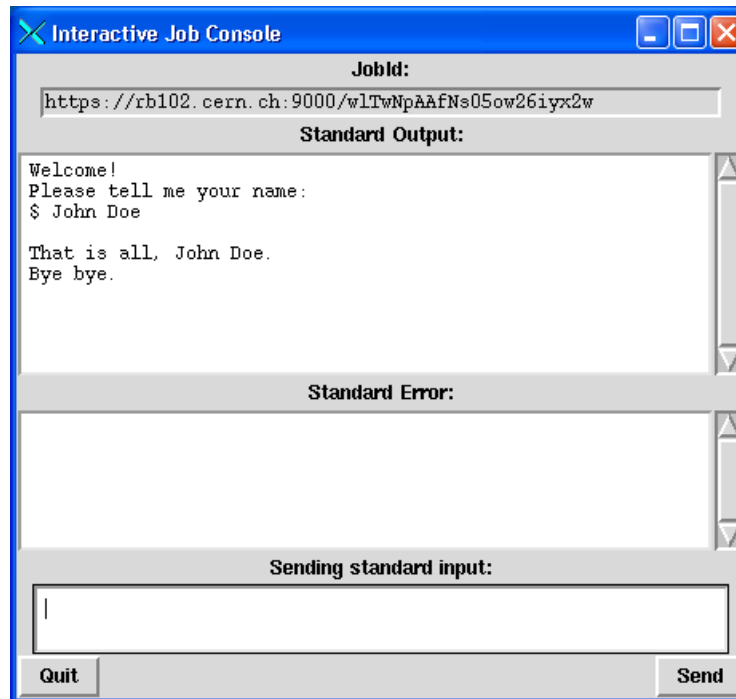
- **Job Collection**

Set of independent jobs which can be executed with one command.

- **Parametric Job**

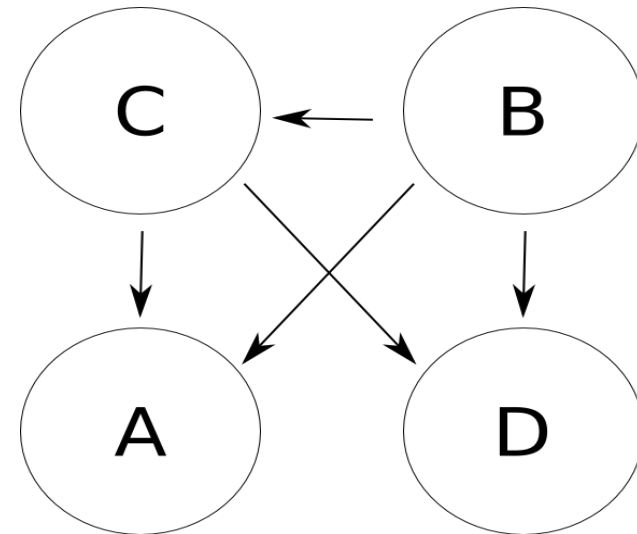
A parametric job is a job collection where the jobs are identical but for the value of a running parameter.

- **Interactive job**



- **DAG job**

Direct acyclic graphs (DAG) are sets of jobs linked by relative dependencies.



- **MPI jobs**

The Message Passing Interface (MPI) is a commonly used standard library for parallel programming.

- **Single Job Submission**

```
glite-wms-job-list-match -a <jdl file>
```

```
glite-wms-job-submit -a <jdl file>
```

```
glite-wms-job-status <jobID>
```

```
glite-wms-job-cancel <jobID>
```

```
glite-wms-job-output <jobID>
```

```
glite-wms-job-logginginfo <jobID>
```

- **Simple job example**

```
Executable = "/bin/hostname";
```

```
Arguments = "";
```

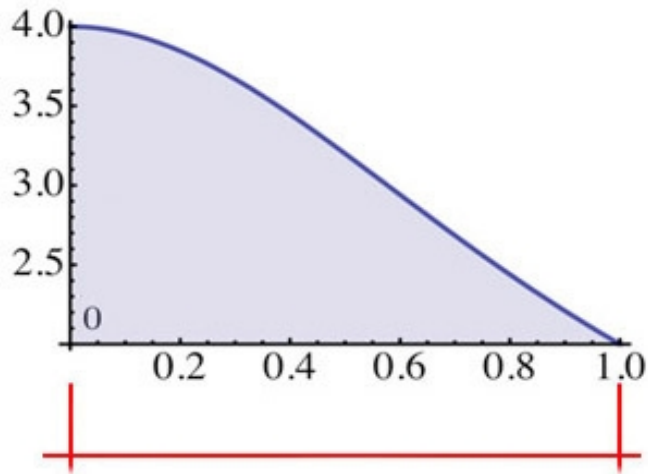
```
StdOutput = "std.out";
```

```
StdError = "std.err";
```

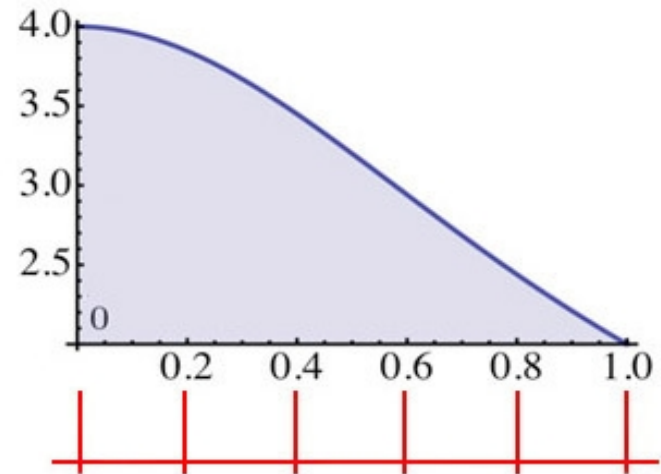
```
OutputSandbox = {"std.out", "std.err"};
```



Message Passing Interface (MPI) is a specification that allows many processors to communicate with one another.



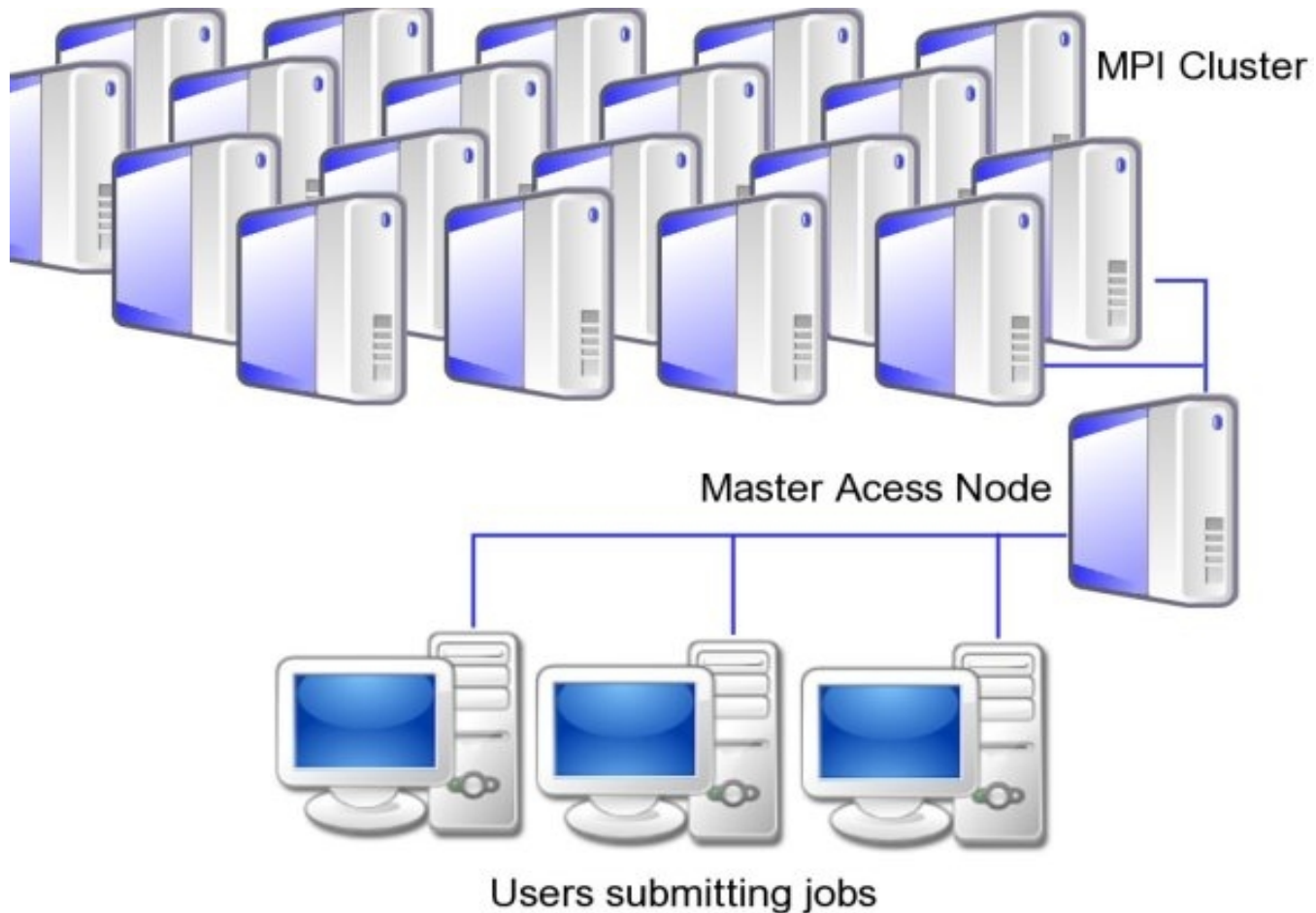
One CPU



CPU-01 CPU-02 CPU-03 CPU-04 CPU-05

- **MPI job**

```
JobType = "Normal";
CPUNumber = 50;
Executable = "mpi-start-wrapper.sh";
Arguments = "pi MPICH";
StdOutput = "mpi-start.out";
StdError = "mpi-start.err";
InputSandbox = {"mpi-start-wrapper.sh", "mpi-hooks.sh", "pi.c"};
OutputSandbox = {"mpi-start.err", "mpi-start.out"};
Requirements = Member("MPI-START",
    other.GlueHostApplicationSoftwareRunTimeEnvironment)
    && Member("MPICH",
    other.GlueHostApplicationSoftwareRunTimeEnvironment);
```



- **Listing folders**
- **Upload / download files**
- **Deleting files**
- **Adding and removing perm**
- **Replicate files**
- **etc...**

- **Directory structure**

```
/grid/<vo>/<you_create_it>
```

```
/grid/aegis/nikola
```

- **Unix-like commands**

## Windows

```
dir
```

```
mkdir
```

## Linux/UNIX

```
ls
```

```
mkdir
```

## GRID

```
lfc-ls
```

```
lfc-mkdir
```

- **Upload a file**

```
$ lcg-cr -l
```

```
lfn:/grid/aegis/petnica/nikola/mpi.tar.gz
```

```
file:/home/ngrkic/mpi.tar.gz
```

- **Copying files out of the Grid (download)**

```
$ lcg-cp lfn:/grid/aegis/petnica/nikola/  
mpi.tar.gz
```

```
file:/home/ngrkic/mpi.tar.gz
```

