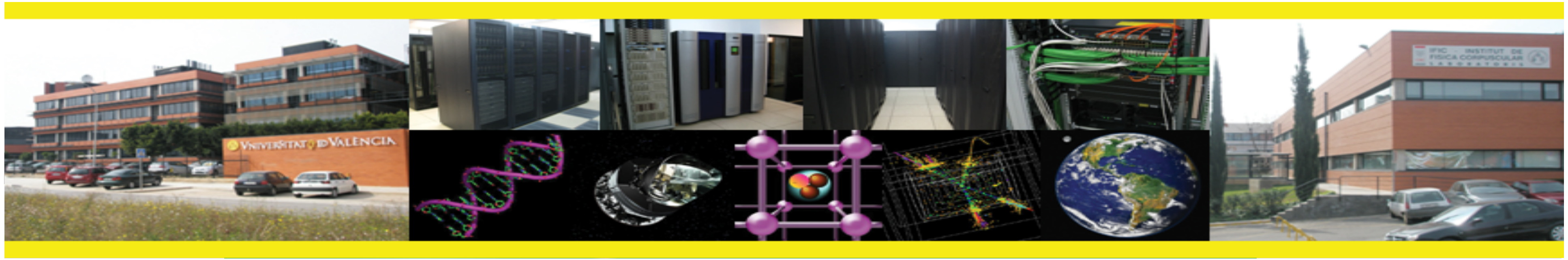
Curso de Postgrado del CSIC - 2010
GRID & e-CIENCIA
IFIC - Valencia 06-09 / 07 / 2010

Job Management

Carlos Escobar Ibáñez (IFIC)



Outline

- **Part I : Talk (20 min)**
    - Introduction
        - Basic concepts
    - JDL (Job Description Language)
    - Job Management
        - gLite commands
        - Few words about advance job management

- **Part II: Practices (1:40 h)**
    - Twiki: https://twiki.ific.uv.es/twiki/bin/view/ECiencia/JobManagement#Practices

Introduction

Why is the job management important?

• Assists users to interact with the Grid.

• Coordinates and manages the resources used/requested by the users, avoiding conflicts and optimizing the performance.

Basic Concepts

• **Job**: A user process.

• **Resources**: servers/nodes, storage servers/disks, data, network, etc…

• **Middleware Grid**: Unify the different components, proving management and access to the resources in a coherent way.

• **Planners**:

  • Low level planners: manage local workload.
  • High level planners (Grid): manage and send jobs to the local planners.

Basic concepts

- **User Interface (UI)**: is the "place" where users logon to the Grid. List resources suitable to execute a given job, submit and cancel jobs. Also retrieve job results. Copy, replicate or delete files from the Grid.

- **Workload Management System (WMS)**: manages and matches the user requirements with the available resources on the Grid.

- **Information System**: provides characteristics and status of CEs and SEs.
  - *The data published in the Information Service conforms to the **GLUE** (Grid Laboratory for a Uniform Environment) **Schema**, which defines a common data model to be used for resources monitoring and discovery.*

- **Computing Element (CE)**: is a batch queue on a site's computers where the user's jobs are executed. Accepts jobs sent by the RB and submits them to the Worker Nodes.

- **Worker Node (WN):** performs the computation and sends back job results to the RB or stores them in a Storage Element.

- **Storage Element (SE)**: provides access to (large-scale/performance) data storage resources. Can support different protocols and interfaces.

Preliminaries

User Interface:
First at all, one should be connected to a User Interface (UI).
The UIs which are available for this course via ssh (using your username) are:

```
cg01.ific.uv.es (SL4)
cg02.ific.uv.es (SL5)
```

Authentication and authorization:
Then, you should create a proxy with voms extension. Remember that this step is comparable to a login on the Grid:

```
voms-proxy-init -voms vo.formacion.es-ngi.eu
```

Job Description Language (JDL):
You must write a script in a special (and very simple) language in order to be able to submit a job. Let's take a look…

JDL (Job Description Language)

The JDL is the language (based on classAd) used for submitting jobs to the GRID. The following slides are in the practice twiki.

Syntax based on statements ended by a semicolon (;): `attribute = value;`

Some rules:

• The JDL is sensitive to blank characters and tabs. No blank characters or tabs should follow the semicolon at the end of a line.

• Literal strings (for values) are enclosed in double quotes.
• If a string itself contains double quotes, they must be escaped with a backslash (e.g.: `Arguments = "\"event\" 10"`).
• Special characters (&, |, >, <) need to be preceded by an escaped backslash.
• Single quote cannot be specified in the JDL.

• Comments must be preceded by a sharp character (#) or have to follow the C++ syntax, i.e a double slash (//) at the beginning of each line or statements begun/ended respectively with /* and */.

JDL (Job Description Language)

Attribute	Mandatory?	Meaning	Example
Executable	**Yes**	**Specify executable**	**`Executable = “test.sh”;`**
Arguments	No	Supply arguments to the executable	`Arguments = “hello 10”;`
StdOutput	**Yes**	**Specify standard output**	**`StdOutput = “std.out”;`**
StdError	**Yes**	**Specify standard error** **(*Can be the same as StdOutput*)**	**`StdError = "std.err";`**
StdInput	No	Specify standard input	`StdInput = “std.in”;`
InputSandbox	No	Transfer input files from UI to WN	`InputSandbox = {"test.sh","std.in"};`
OutputSandbox	No	Transfer output files from WN to UI	`OutputSandbox = {"std.out","std.err"};`
Environment	No	Extend the environment	`Environment = {”MYPATH= $HOME/mypath};`
Requirements	No	Imposing Constraints on the CE	`Requirements = other.GlueCEInfoLRMSType == "PBS”;`
Rank	No	Apply a weight to select CE	`Rank = other.GlueCEStateFreeCPUs;`
PerusalFileEnable	No	Enable job perusal	`PerusalFileEnable = true;`
PerusalTimeInterval	No	Specify in seconds frequency that specified files are copied to WMS machine	`PerusalTimeInterval = 30;`
More information: Job Description Language (JDL) Attributes document			

JDL (Job Description Language)

Additional Notes

- **InputSandbox**
  - Wildcards allowed.
  - Files are relative to current directory.
  - The executable flag is not preserved for the files included in the Input Sandbox when transferred to the WN. Therefore, for any file needing execution permissions a chmod +x operation should be performed by the initial script specified as the Executable in the JDL file (the chmod +x operation is done automatically for this script).
  - The InputSandbox cannot contain two files with the same name (even if in different paths) as when transferred they would overwrite each other.

- **OutputSandbox**
  - No absolute file names
  - The OutputSandbox cannot contain two files with the same name (even if in different paths) as when transferred they would overwrite each other.

JDL (Job Description Language)

Additional Notes

• **Requirements**: The Requirements attributes can be used to express any kind of constraint on the resources where the job can run. Its value is a Boolean expression that must evaluate to true for a job to run on that specific CE.

They are based on the GLUE Schema.

- ***Forming expressions***

  To force a job to only run on a particular CE:

  ```
  Requirements = other.GlueCEUniqueID == "ce.iaa.csic.es:
  2119/jobmanager-lcgpbs-forngi";
  ```

  where the `other.` prefix is used to indicate that the GlueCEName attribute refers to the CE characteristics and not to those of the job. If `other.` is not specified, then the default `self.` is assumed, indicating that the attribute refers to the job characteristics description.

JDL (Job Description Language)

Additional Notes (Requirements cont'd)

Requirements can be ANDed together:

```
Requirements = other.GlueCEInfoHostName ==
"ce.iaa.csic.es" && other.GlueCEStateFreeCPUs > 10;
```

which ANDs in the requirement that there are at least 2 CPUs on the machine.
By default the system always ANDs in other requirement:

```
Requirements = other.GlueCEStateStatus == "Production" ;
```

A requirement can be negated:-

```
Requirements =  (!other.GlueCEInfoTotalCPUs < 10);
```

JDL (Job Description Language)

Additional Notes (Requirements cont'd)

- ***Functions***

• Member

The Member function is satisfied if the first argument (a scalar value) is a member of its second argument (a list). For example, one essential requirement for a production work is that the machine has the appropriate software installed:

```
Requirements = Member
(”SL”,other.GlueHostOperatingSystemVersion);
```

Functions can be also ANDed:

```
Requirements = Member
("ScientificSL",other.GlueHostOperatingSystemName)
            && Member
("SL",other.GlueHostOperatingSystemVersion)
            && Member
("5.3",other.GlueHostOperatingSystemRelease);
```

JDL (Job Description Language)

Additional Notes (Requirements cont'd)

- RegExp

  Another function RegExp can be used to see if a supplied matches as regular expression, for example:

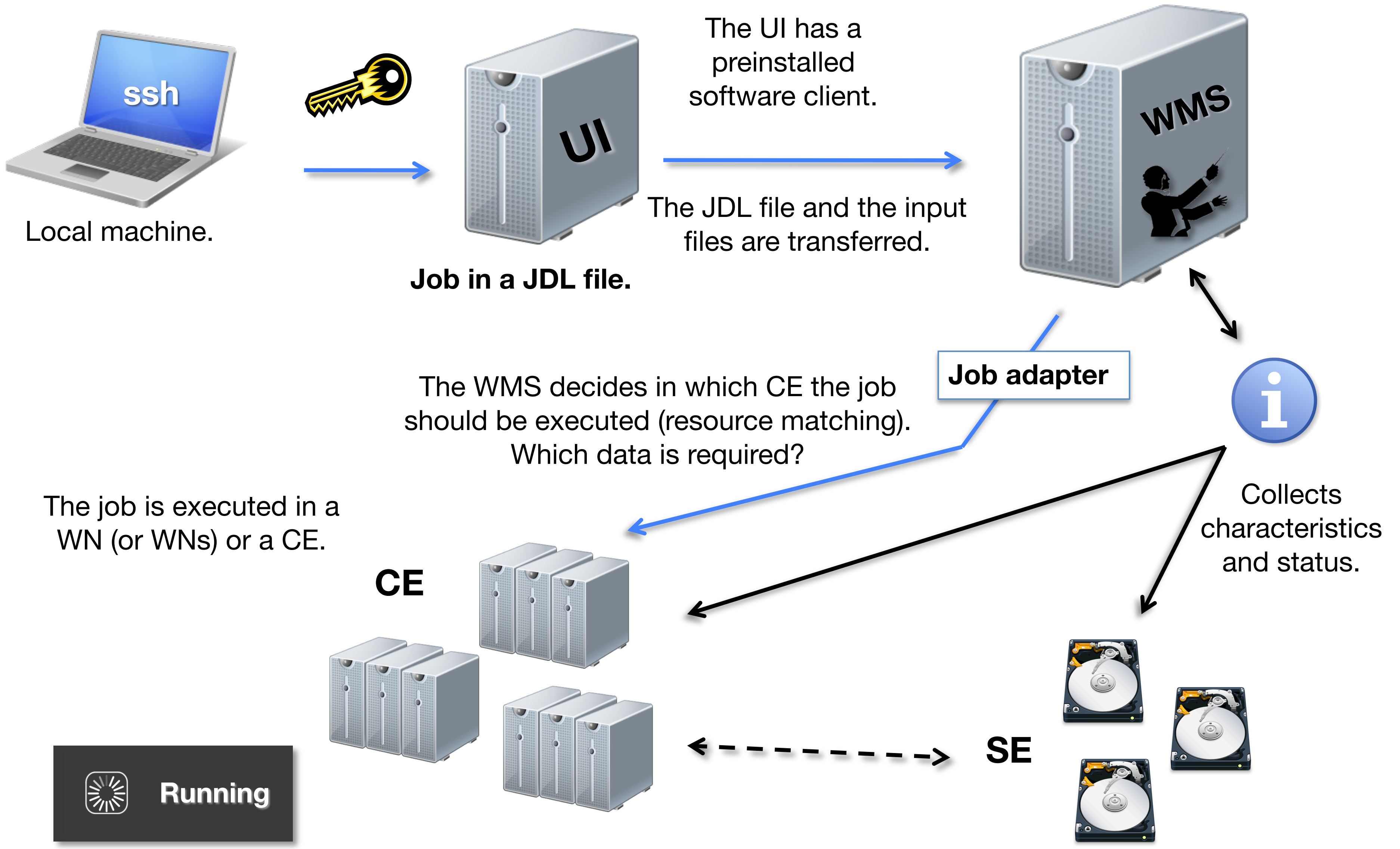
  ```
  Requirements = RegExp("ce.iaa.csic.es",
  other.GlueCEInfoHostName);
  ```

▪ ***Gangmatching***

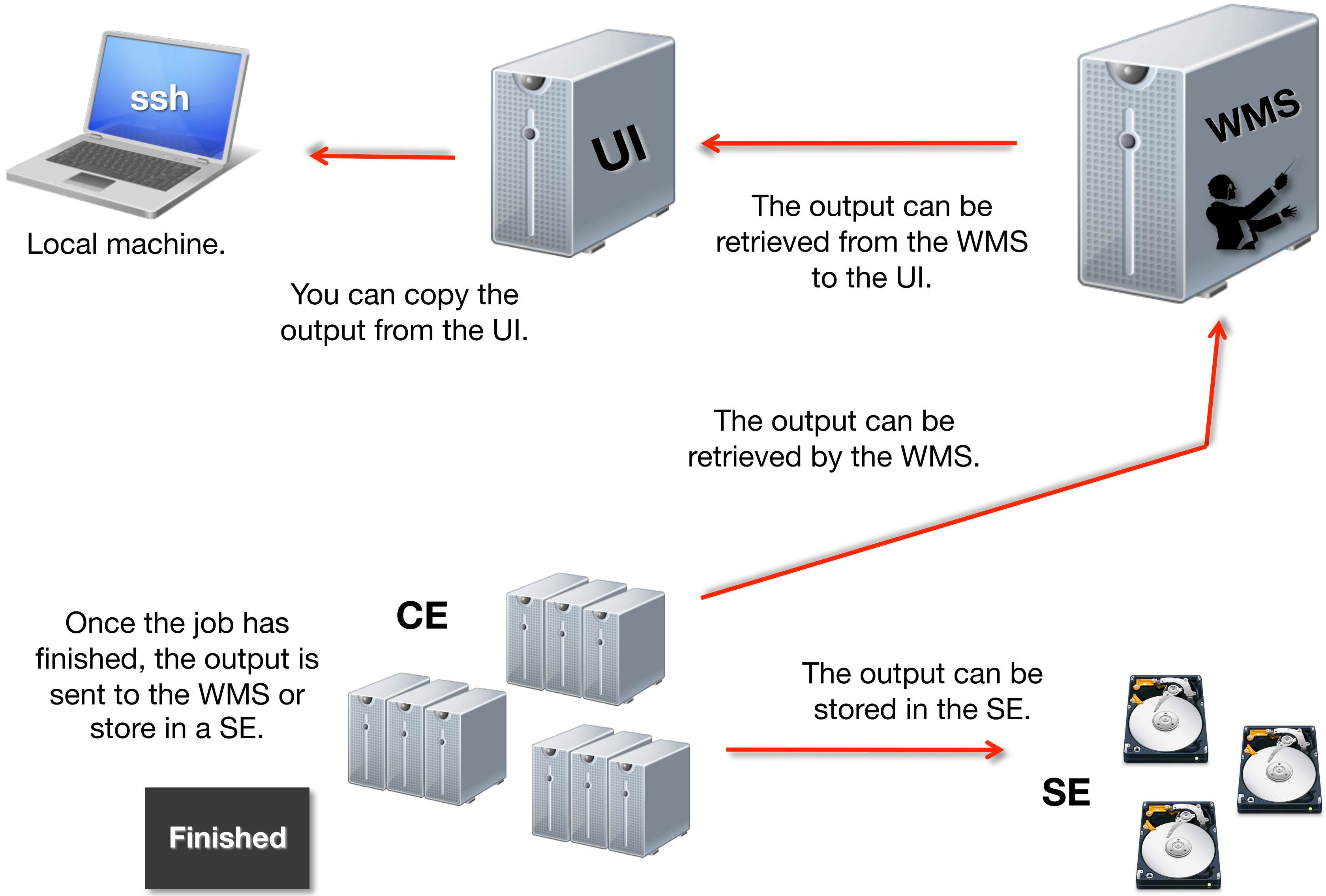
The previous requirements affected always two entities: the job and the CE. In order to specify requirements involving three entities (i.e., the job, the CE and a SE), the WMS uses a special match-making mechanism, called gangmatching. This is supported by some JDL functions: anyMatch, whichMatch, allMatch.

• **Rank:** The choice of the CE where to execute the job, among all the ones satisfying the requirements, is based on the rank of the CE (a quantity expressed as a floating-point number). The CE with the highest rank is the one selected.
The user can define the rank with the Rank attribute as a function of the CE attributes.

Job Submission


Network daemon to accepts the incoming request.
ssh
Local machine.
UI
The UI has a preinstalled software client.
The JDL file and the input files are transferred.
Job in a JDL file.
WMS
Job adapter
The WMS decides in which CE the job should be executed (resource matching). Which data is required?
Collects characteristics and status.
The job is executed in a WN (or WNs) or a CE.
CE
SE
Running

Job Submission



Local machine.

You can copy the output from the UI.

The output can be retrieved from the WMS to the UI.

The output can be retrieved by the WMS.

Once the job has finished, the output is sent to the WMS or store in a SE.

The output can be stored in the SE.

gLite Commands

Job management by the users: gLite commands:

- Credentials delegation
- Job matching
- Job submission
- Job status
- Job logging info
- Job monitoring
- Job cancelation
- Job output retrieval

Credentials delegation

This command delegates your credentials (proxy) to the *Workload Management System* (WMS), assigns and holds an identification name (automatic or not), so that subsequent invocations of glite-wms-job-submit and glite-wms-job-list-match can be given that delegation name, bypassing the delegation of a new proxy.

Syntax: **glite-wms-job-delegate-proxy -d dID**
Options:
-a Creates an automatic identification name
-d dID Creates the identification name dID

The automatic delegation is not recommended for massive production.

Job matching

This command checks and lists which are the CEs that match the right requirements for our JDL file, assuring that our jobs will run successfully.

Syntax: **glite-wms-job-list-match script.jdl**
Options:
-a Automatic delegation
-d dID Uses a previous explicit delegation
Note that one of this options must be used.
-o file Stores the list of CEs in the file file

Job Submission

This command submits jobs to the GRID.

Syntax: **glite-wms-job-submit script.jdl**
Options:
-a Automatic delegation
-d dID Using an explicit delegation declared previously.
Una de estas opciones debe de ser utilizada.
-o jobId Añade el identificador del trabajo en el archivo jobId (lo crea si no existe)
-r CE Submits the job directly to a particular CE. With this option, the availability of the CE is not checked. The BrokerInfo is not created either.

During the job submission, a job identifier is created for this job which will be unique and it has to be used to check the job status or to retrieve the output.

The format is https://Lbserver_address[:port]/unique_string (be aware that it is not a web site).

Job status

This command gives information about the status of the job (or jobs).

Syntax: **glite-wms-job-status jobId1 ... jobIdN**
Options:
-i jobId Reads the file (or files) where the job identifiers are stored.
-o file Stores the output in the file file.
-v n Sets the output level (0, 1 or 2)

Job status

Flag	Meaning
SUBMITTED	submission logged in the Logging & Bookkeeping service
WAIT	job match making for resources
READY	job being sent to executing CE
SCHEDULED	job scheduled in the CE queue manager
RUNNING	job executing on a WN of the selected CE queue
DONE	job terminated without grid errors
CLEARED	job output retrieved
ABORT	job aborted by middleware, check *reason*
CANCEL	job cancelled by a user request

Job logging info

This command provides logging information of one or more submitted jobs. The syntax is as follows:

Syntax: **glite-wms-job-logging-info jobId1 ... jobIdN**
Options:
-i jobId Reads the file (or files) where the job identifiers are stored.
-o file Store the output in the file file.
-v n Sets the output level (0, 1 or 2).

Job monitoring

The job monitoring allows to see the job "output" while the job itself is running. To enable this feature you have to add the attributes PerusalFileEnable and PerusalTimeInterval to the JDL file before submitting the job. For example:

```
PerusalFileEnable = true;
PerusalTimeInterval = 120;
```

This does that the WN uploads regularly (el intervalo de tiempo se define con PerusalTimeInterval en segundos) a copy of the job "output" to the WMS when the command glite-wms-job-perusal is used. To request the monitoring of a file, you have:

Syntax: **glite-wms-job-perusal --set -f file_1 ... -f file_n -i jobId1 ... jobIdn**
Options:
-all Retrieve the full file. By default, just a part is retrieved since the last request.
--dir directory Stores the "output" in the directory directory. By default, it is done in /tmp.

Job monitoring

and to retrieve the requested file:

Syntax: **glite-wms-job-perusal --get -f file ... -i jobId**
Options:
-all Retrieve all the files.

Note: Obviously this feature has its impact over the performance. It is an option to debug your jobs and it is not recommended to use it in production because many transferred files can flood the WMS (WMS).

Once a job has been sent, the files you want to monitor have to be requested with the option –set.

The job monitoring can be disable with the option -unset.
Syntax: **glite-wms-job-perusal --unset jobId**

It is possible to use the command glite-wms-job-perusal to check the final status of the files once the job has finished. If you want to use this "post-mortem" feature, the monitoring has to be enable with the option glite-wms-job-perusal --set but leaving the file retrieval till the job has finished.

Job cancelation

This command cancels a job (or jobs) with job identifier jobId.

Syntax: **glite-wms-job-cancel jobId1 ... jobIdN**
Options:
-i jobId Reads the file (or files) where the job identifiers are stored.

Note: If the job has not been transferred to the CE (i.e. its status is WAITING or READY), the canl request can be ignored and therefore the job can run even if you can read "successful cancellation". In these cases, simply repeat your cancel request when the job status is SCHEDULED or RUNNING.

Job output retrieval

This command retrieves the job "output", obviously for finished jobs, i.e. with status DONE, to the UI. If he "output" is not retrieved, it would be removed from the WMS about one week after the job ending.

Syntax: **glite-wms-job-output jobId1 ... jobIdN**
Options:
-i jobId Reads the file (or files) where the job identifiers are stored.
--dir directory Stores the "output" in the directory directory. By default, it is done in /tmp.

Advance job management

Job Collections

A *job collection* is a set of mutually independent jobs, which, for some reason (for example, common input files) needs to be submitted, monitored and controlled as a single request.

DAG jobs

A *DAG* (directed acyclic graphs) *job* represents a set of jobs where the input, the output or the execution of one or more jobs depends on one or more jobs.

Parametric jobs

A *parametric job* causes a set of jobs to be generated from one JDL file. This is invaluable in cases where many similar (but not identical) jobs must be run.

Unfortunately, beyond the scope of this course!

Practical session

Now, you have all the ingredients to start your own receipts!

https://twiki.ific.uv.es/twiki/bin/view/ECiencia/JobManagement#Practices

Backup Slides

Job Management

Carlos Escobar Ibáñez

CE for vo.formacion.es-ngi.eu

Using the Information System to see the resourses:

lcg-info --list-ce --vo vo.formacion.es-ngi.eu

```
- CE: ce-iber.bifi.unizar.es:2119/jobmanager-lcgpbs-formangi
- CE: ce-ieg.bifi.unizar.es:2119/jobmanager-lcgpbs-formangi
- CE: ce-sge-ngi.ceta-ciemat.es:2119/jobmanager-lcgsge-ngiform
- CE: ce.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
- CE: ce.iaa.csic.es:2119/jobmanager-lcgpbs-forngi
- CE: ce01-tic.ciemat.es:2119/jobmanager-lcgpbs-training
- CE: ce01.macc.unican.es:2119/jobmanager-lcgpbs-grid
- CE: ce2.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
- CE: ce3.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
- CE: e-ce.iaa.es:2119/jobmanager-lcgpbs-forngi
- CE: gridce01.ifca.es:2119/jobmanager-sge-ngifor
- CE: ngiesce.i3m.upv.es:2119/jobmanager-pbs-ngies
- CE: test03.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
```

CE for vo.formacion.es-ngi.eu

Other useful command:

lcg-infosites --vo vo.formacion.es-ngi.eu ce

```
#CPU     Free     Total Jobs  Running  Waiting   ComputingElement
--------------------------------------------------------------------------------------------------------------
1360     530      0           0        0         ce05.pic.es:2119/jobmanager-lcgpbs-ngi
1360     501      0           0        0         ce07.pic.es:2119/jobmanager-lcgpbs-ngi
1360     511      0           0        0         ce06.pic.es:2119/jobmanager-lcgpbs-ngi
   8       8      0           0        0         e-ce.iaa.es:2119/jobmanager-lcgpbs-forngi
   1       1      0           0        0         ce.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
 164     164      0           0        0         ce2.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
 164     164      0           0        0         ce3.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
 448     448      0           0        0         ce.iaa.csic.es:2119/jobmanager-lcgpbs-forngi
  22      22      0           0        0         ce-ieg.bifi.unizar.es:2119/jobmanager-lcgpbs-formangi
 136     136      0           0        0         ce01-tic.ciemat.es:2119/jobmanager-lcgpbs-training
 248       1      1           0        1         ce-iber.bifi.unizar.es:2119/jobmanager-lcgpbs-formangi
  56      56      0           0        0         ce-sge-ngi.ceta-ciemat.es:2119/jobmanager-lcgsge-ngiform
 164     164      0           0        0         test03.egee.cesga.es:2119/jobmanager-lcgsge-GRID_ngifor
 151      45      2           2        0         ce01.macc.unican.es:2119/jobmanager-lcgpbs-grid
1616    1616      0           0        0         gridce01.ifca.es:2119/jobmanager-sge-ngifor
```

CE for vo.formacion.es-ngi.eu

lcg-infosites --vo vo.formacion.es-ngi.eu ce -v 2

```
RAMMemory    Operating System       System Version                 Processor          Subcluster name
--------------------------------------------------------------------------------------------------------------------
 16000            ScientificSL       Boron                           Xeon                                ce05.pic.es
 16000            ScientificSL       Boron                           Xeon                                ce07.pic.es
 16000            ScientificSL       Boron                           Xeon                                ce06.pic.es
  4096            ScientificSL       Beryllium                       xeon                                e-ce.iaa.es
  1024            ScientificSL       Beryllium                       Xeon                          ce.egee.cesga.es
  1024            ScientificSL       Beryllium                        PIV       ce2.egee.cesga.es
  1024            ScientificSL       Beryllium                       Xeon                         ce3.egee.cesga.es
  2048        ScientificCERNSLC       SL                            Intel         ce.iaa.csic.es
   513        ScientificCERNSLC      Beryllium                        PIV      ce-ieg.bifi.unizar.es
  1024            ScientificSL        SL                          Xeon5160                       ce01-tic.ciemat.es
 16384        ScientificCERNSLC      Boron                           Xeon                     ce-iber.bifi.unizar.es
  2048         ScientificSL 4.6       4.6                            Xeon                    ce-sge-ngi.ceta-
ciemat.es
     0                n.a             n.a                             n.a      test03.egee.cesga.es
  2048            ScientificSL        SLC                              PD       ce01.macc.unican.es
     0                n.a             n.a                             n.a         gridce01.ifca.es
```

The GLUE Schema

The data published in the Information Service conforms to the ***GLUE (Grid Laboratory for a Uniform Environment) Schema***, which defines a common data model to be used for resources monitoring and discovery.

• **General Attributes:** This group includes general attributes that are defined in entries of both CEs and SEs, the schema version, the URL of the IS server and finally the GlueKey.

• **Attributes for the Computing Element:** Information about a CE and its WNs.

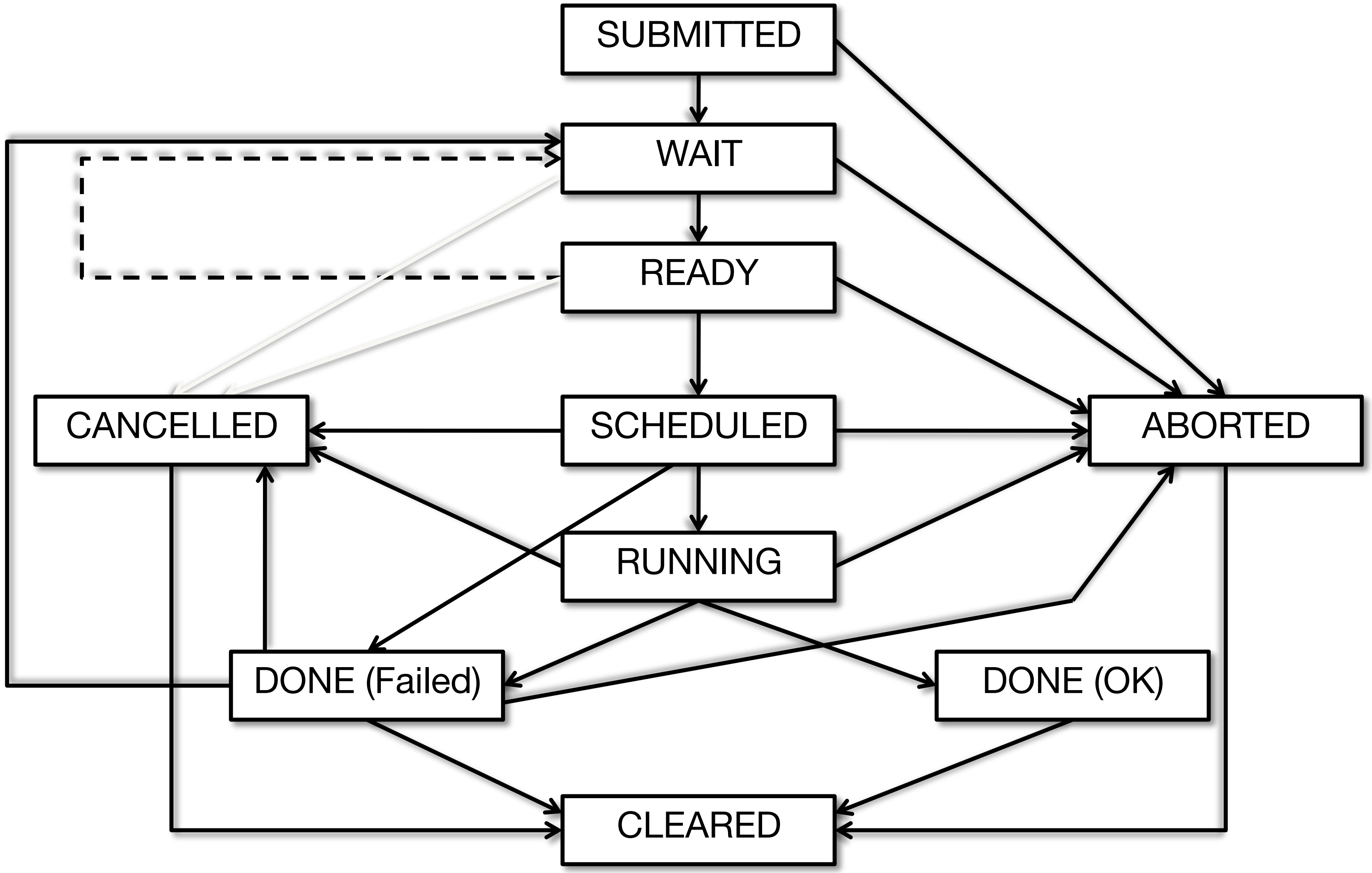
• **Attributes for the Storage Element:** Information about a SE and the corresponding storage space.

• **Attributes for the CE-SE Binding:** The CE-SE binding schema represents a mean for advertising relationships between a CE and a SE (or several SEs).

Try the command: *lcg-info --list-attrs --vo vo.formacion.es-ngi.eu*

Back to the JDL slide

Job Managing


SUBMITTED
WAIT
READY
CANCELLED
SCHEDULED
ABORTED
RUNNING
DONE (Failed)
DONE (OK)
CLEARED