Enabling Grids for E-science in Europe

www.eu-egee.org

First Latin American Grid Workshop
18 November 2004

Job Services With Data Requirements

Elisabetta Ronchieri
INFN CNAF

EGEE is a project funded by the European Union under contract IST-2003-508833

Contents

Enabling Grids for E-science in Europe

- Job Description Language
- WMS Matchmaking
- The BrokerInfo

Job Description Language

Enabling Grids for E-science in Europe

- The supported attributes are grouped in two categories:
  - Job Attributes
    - Define the job itself
  - Resources
    - Taken into account by the RB for carrying out the matchmaking algorithm (to choose the “best” resource where to submit the job)
    - *Computing Resource*
      - Used to build expressions of Requirements and/or Rank attributes by the user
      - Have to be prefixed with “other.”
    - *Data and Storage resources (see talk Job Services With Data Requirements)*
      - Input data to process, SE where to store output data, protocols spoken by application when accessing SEs

JDL: Data attributes

Enabling Grids for E-science in Europe

- **InputData** (first version)
  - Refers to data used as input by the job: these data are published in the Replica Location Service (RLS) and stored in the SEs)
  - Contains LFNs and/or GUIDs
- **DataAccessProtocol** (mandatory if InputData has been specified)
  - Specifies the protocol or the list of protocols which the application is able to speak with for accessing *InputData* on a given SE
- **OutputSE**
  - Represents the Uniform Resource Identifier of the output SE
  - RB uses it to choose a CE that is compatible with the job and is close to SE

JDL: Data attributes

Enabling Grids for E-science in Europe

- **InputData** (second version)
  - Refers to data used as input by the job: these data are published in a dataset catalogue that is registered in MDS
  - Contains LDSs
- **ReplicaCatalog** (optional)
  - Specifies the dataset catalogue explicitly
- **Example 1**

```
InputData = "lds:test-file-dli";
ReplicaCatalog = "
  http://lxshare0203.cern.ch:8085/";
```

- **Example 2**

```
InputData = "lds:test-file-dli";
```

JDL: Data attributes

Enabling Grids for E-science in Europe

- **OutputData**
  - Allows the user for the automatic upload and registration in Grid of files produced by the job on the WN
  - Specifies several output files
  - Each OutputData consists of three elements:
    - OutputFile (mandatory)
      - Specifies the name of the generated file to be uploaded to the Grid
    - StorageElement (optional)
      - Indicates the SE where the file should be stored
      - If unspecified, the default SE is chosen using ($VO_<VO>_DEFAULT_SE)
    - LogicalFileName (optional)
      - Represents a LFN the user wants to be associated to the output file in Grid
      - If unspecified, it is set a GUID

JDL: Data attributes

Enabling Grids for E-science in Europe

- **Example OutputData** attribute in the JDL

```
OutputData = {
   [
      OutputFile = "toto.out" ;
      StorageElement = "adc0021.cern.ch";
      LogicalFileName = "lfn:theBestTotoEver"
   ],
   [
      OutputFile = "toto2.out" ;
      StorageElement = "adc0021.cern.ch" ;
   ]
};
```

Example of JDL file with InputData

Enabling Grids for E-science in Europe

```
[
JobType = "Normal";
Executable = "gridTest";
StdError = "sim.err";
StdOutput = "sim.out";
InputData = {"lfn:green", "guid:123454454"};
DataAccessProtocol = "gsiftp";
InputSandbox = "/home/joda/test/gridTest";
OutputSandbox = "sim.err", "sim.out"};
Requirements  = (other.GlueHostOperatingSystemName
== "linux") && (other. GlueCEStateFreeCPUs > 4);
Rank = other.GlueCEPolicyMaxCPUTime;
]
```

Automatic upload and Registration of output file

Enabling Grids for E-science in Europe

- The automatic upload of data from a job is done on the WN
- On the UI, the user has to specify in the JDL
  - OutputData attribute: it allows the user to automatically upload and register files generated by the job into the Grid

```
OutputFile = "dummy.dat";
StorageElement = "gilda-se-01.pd.infn.it";
LogicalFileName = "lfn:AntonioDelgado_outputData";
```

  - OutputSE attribute: it represents the SE where the job wants to save the files generated during its execution

```
OutputSE = "gilda-se-01.pd.infn.it";
```

- The automatically upload of data from a job with a default SE will work properly if that CE has at least one SE defined as close in the Information System
- Otherwise the StorageElement Attribute is used

Automatic upload and Registration of output file

Enabling Grids for E-science in Europe

- On the WN, it is generated an output file called DSUpload_<jobId>.out, where the JobWrapper save the result of the upload operations
  - It is automatically added in the files specified in the OutputSandbox attribute
- The user can check the result of the uploading simply doing
  - `cat DSUpload_<jobid>.out`

```
# <outputfile> <lfn|guid|Error>
dummy.dat    lfn:AntonioDelgado_outputData
```

Example of JDL file with OutputData

Enabling Grids for E-science in Europe

```
[
JobType = "Normal";
Executable = "gridTest";
StdError = "sim.err";
StdOutput = "sim.out";
OutputData = [
 OutputFile = "dummy.dat";
 StorageElement = "gilda-se-01.pd.infn.it";
 LogicalFileName =  "lfn:AntonioDelgado_outputData";
]
OutputSE = "se012.cnaf.infn.it";
]
```

The BrokerInfo

- The BrokerInfo file allows the user job to access at run time certain information related to the job, e.g. the name of CE, the file specified in the InputData attribute, etc
- It is always created in the job working directory on the WN
- It is called .BrokerInfo
- Its syntax is based on Condor Classads
- It can be parsed using a CLI:
  - directly from the job on the WN
  - from the user on the UI
    - IMPORTANT: the .BrokerInfo file must be retrieved with the rest of the generated output files when the job finishes

      OutputSandbox = {".BrokerInfo", ….};

The BrokerInfo command

- `edg-brokerinfo [-v] [-f <filename>] function [parameter] [parameter] …`
  - [-v] produces more verbose output
  - [-f <filename>] specifies the BrokerInfo file that has to be parsed
    - If unspecified, the BrokerInfo file is taken by the environment variable $EDG_WL_RB_BROKERINFO
  - function is one of the following:
    - getCE returns the name of the CE the job is running on
    - getVirtulOrganization returns the name of the VO specified in the VirtualOrganization JDL attribute
    - getDataAccessProtocol returns the protocol list specified in the DataAccessProtocol JDL attribute
    - getInputData returns the files specified in the InputData JDL attribute
    - getSEs returns the list of the SE with contain a copy of at least one file among those specified in the InputData JDL attribute

The BrokerInfo command

Enabling Grids for E-science in Europe

- function is one of the following:
  - getCE returns the name of the CE the job is running on
  - getVirtulOrganization returns the name of the VO specified in the VirtualOrganization JDL attribute
  - getDataAccessProtocol returns the protocol list specified in the DataAccessProtocol JDL attribute
  - getInputData returns the files specified in the InputData JDL attribute
  - getSEs returns the list of the SE with contain a copy of at least one file among those specified in the InputData JDL attribute
  - getCloseSEs returns a list of SEs close to the CE
  - getSEMountPoint <SE> returns the access point for the specified <SE>, if it is in the list of close SEs of the WN
  - getSEPort <SE> <Protocol> returns the port number used by <SE> for the data transfer protocol <Protocol>
  - getLFN2SFN <LFN> returns the storage file name of the file specified by <LFN>
    - <LFN> is a logical file name of a GUID specified in the InputData attribute

The BrokerInfo file

Enabling Grids for E-science in Europe

```
[
   ComputingElement =
   [
           CloseStorageElements =
            {
                    [
                          GlueSAStateAvailableSpace = 399645448;
                          GlueCESEBindCEAccesspoint = "/flatfiles/SE00";
                          mount = GlueCESEBindCEAccessPoint;
                          name = "grid009.to.infn.it";
                          freespace = GlueSAStateAvailableSpace
                     ]
            };
            name = "grid008.to.infn.it:2119/jobmanager-lcgpbs-long"
    ];
    InputFNs =
    {
    };
    StorageElements =
    {
    };
    VirtualOrganisation = "dteam"
]
```

The BrokerInfo file

```
 DataAccessProtocol =  {
             "gridftp" ,
             "file"
};
InputFNs =  {
  [
    name = "lfn:wp1-test-file-01-lfn";
    SFNs =  {
        "sfn://testbed013.cern.ch/wp1/wp1-test-file-01",
        "sfn://testbed113.cern.ch/wp1/wp1-test-file-01"
    }
  ],
  [
    name = "lfn:wp1-test-file-02-lfn";
    SFNs = {
        "sfn://testbed001.cern.ch/wp1/wp1-test-file-02",
        "sfn://testbed005.cern.ch/wp1/wp1-test-file-02"
    }
  ],
  [
    name = "lfn:wp1-test-file-03-lfn";
    SFNs =  {
        "sfn://testbed002.cern.ch/wp1/wp1-test-file-03"
    }
  ]
}
```

WMS API

Enabling Grids for E-science in Europe

- BrokerInfo CLI and APIs are described

  http://server11.infn.it/workload-grid/docs/edg-brokerinfo-user-guide-v2_2.pdf

Hands-on time!

Enabling Grids for E-science in Europe

- Create an executable that, using the ***BrokerInfo APIs***, lists all close SEs.
- Submit this job via a JDL file
- Retrieve the output of the job
- Check that the output contains the information you want