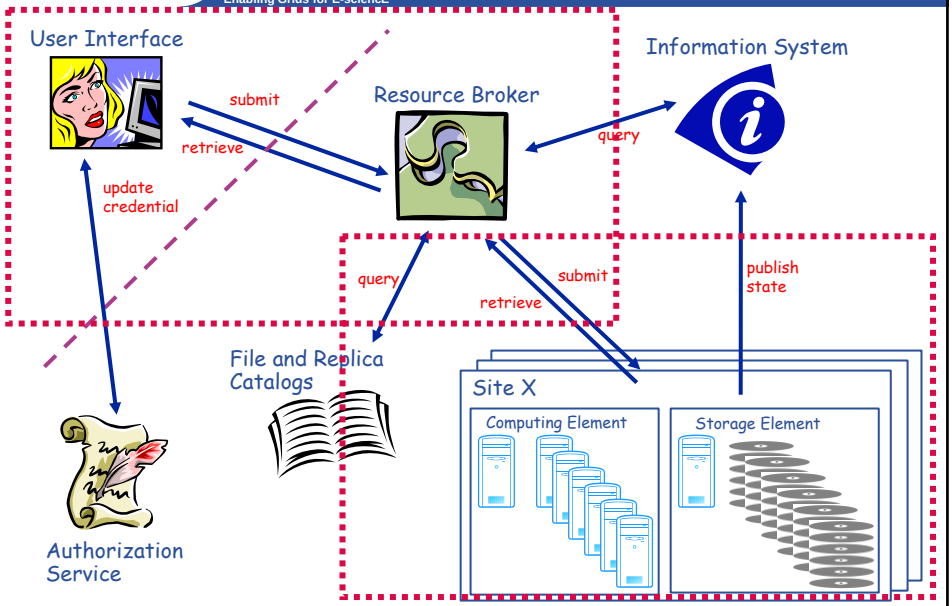
Workload Management

Heinz Stockinger
Swiss Institute of Bioinformatics

www.eu-egee.org

Information Society and Media

Reference Grid



Contents

General concepts of Grid Workload Management Systems

EGEE Workload Management System

Job Preparation

Architecture / Job submission and status monitoring

Matchmaking

Different job types

Workload Management - Definitions

- **Resource Management includes the efficient usage of computing and storage resources**
  - Processor time, memory, storage, network, etc.
- **Here, mainly referred to as "workload management system" since it deals with the distribution of user executables to Grid resources**
  - Do not put the load on one subsystem but distribute it
  - Manage the workload produced by end users
  - Workload management is partly referred to as "scheduling"
- **From the user's point of view, workload management should be transparent**
- **Workload consists of user jobs**
- **A job can be any kind of executable that requires CPU or storage**

General concepts of Grid Workload Management System

EGEE Workload Management Systems

- Job Preparation
- Architecture / Job submission and status monitoring
- Matchmaking
- Different job types

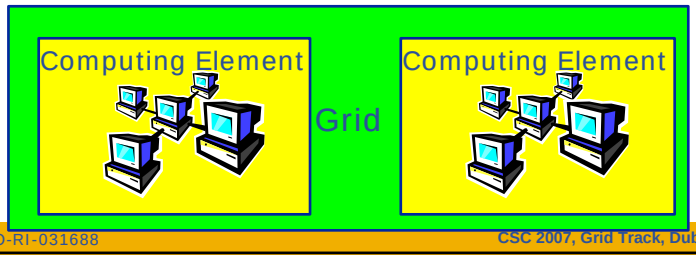
Grid Workload Management System

- **Scheduling consists of:**
  - Resource **Discovery/Brokering**
    - Find suitable resources
  - **Matchmaking**
    - Assign a job to a resource that satisfies job requirements
  - **Job execution**
    - Execute the jobs and retrieve output
    - Deal with error management
- **Need to interact with all major Grid services/components**
  - Information System
  - Grid security
  - Computing Element
  - Storage Element
- **Job execution requires to find the "right" Computing Element (computing resource)**

How to allow for execution of CPU intensive jobs?

Enabling Grids for E-sciencE

- **Use parallel and distributed computing environments**
  - Parallel machines
  - Clusters (set of workstations)
    - Sometimes also referred to as “farms” if they are loosely coupled
- **Grid = set of clusters or parallel machines**
- **Main focus here on clusters/farms and high throughput computing**
  - Rather than optimising the execution of a single program, allow for several user jobs that then efficiently use existing resources



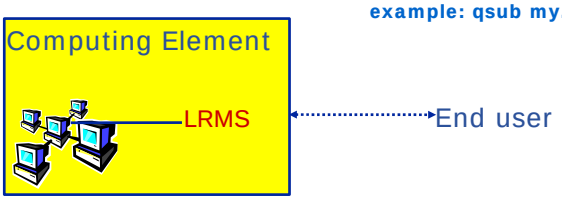
Local Resource Management System (LRMS)

Enabling Grids for E-sciencE

- **Manage the local computing resources in the Computing Element**
- **Often, Batch Systems are used**
  - PBS (Portable Batch System)
  - LSF (Load Sharing Facility)

Interaction with local batch system through native batch client

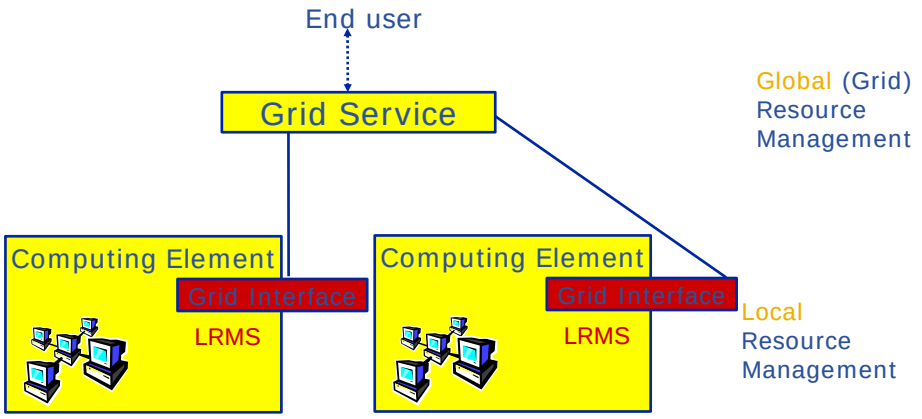
example: qsub my.executable



Grid Resource Management

Enabling Grids for E-sciencE

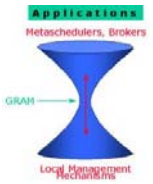
- **Manage several Computing Elements**

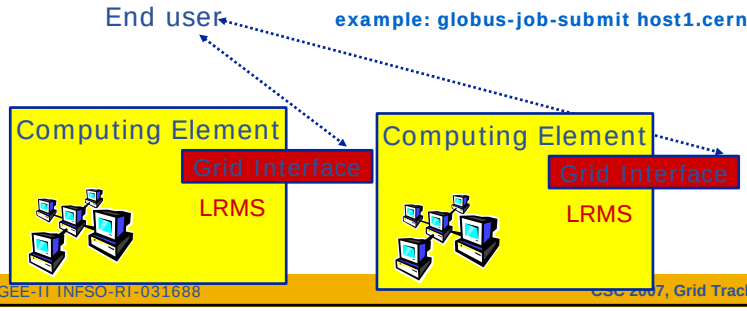


Globus Resource Management

Enabling Grids for E-sciencE

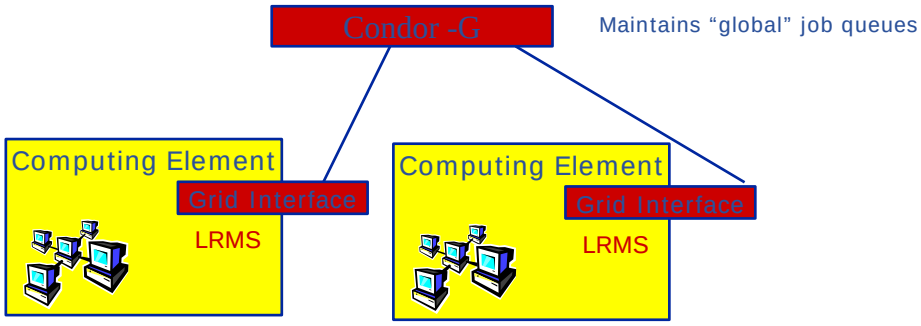
- **GRAM (Grid Resource Allocation Manager)**
- **Service that provides a Grid Interface to Local Resource Management System**
  - Also referred to as “Gatekeeper”
  - Provides a general interface to different batch systems like PBS, LSF
  - User needs to specify the exact hostname of Computing Element





Condor-G

- From Condor High-throughput computing project
  - http://www.cs.wisc.edu/condor
- Global Resource Management allows to submit to Globus GRAM managed resources
- No match-making/brokering



The EGEE Workload Management System (WMS)

- Builds on both (Globus and Condor) and provides a full Grid workload management system
- The Goal of WMS is the distributed scheduling and resource management in a Grid environment.
- What does it allow Grid users to do?
  - To submit their jobs
  - To execute them on the “best resources”
    - The WMS tries to optimise the usage of resources
  - To get information about their status
  - To retrieve their output

General concepts of Grid Workload Management Systems

EGEE Workload Management System

- Job Preparation
- Architecture /Job submission and status monitoring
- Matchmaking
- Different job types

Job Preparation

- **Information to be specified when a job has to be submitted:**
  - Job characteristics
  - Job requirements and preferences on the computing resources
    - Also including software dependencies
  - Job data requirements
- **Information specified using a Job Description Language (JDL)**
  - Based upon **Condor's *CLASSified ADvertisement*** *language (ClassAd)*
    - Fully extensible language
    - A ClassAd
      - *Constructed with the classad construction operator []*
      - *It is a sequence of attributes separated by semi-colons.*
      - *An attribute is a pair (key, value), where value can be a Boolean, an Integer, a list of strings, …*
        - <attribute> = <value>;

Job Description Language (JDL)

Enabling Grids for E-sciencE

- **The supported attributes are grouped into two categories:**
  - Job Attributes
    - Define the job itself
  - Resources
    - Taken into account by the Workload Manager 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***
      - *Input data to process, Storage Element (SE) where to store output data, protocols spoken by application when accessing SEs*

JDL: Relevant Attributes (1)

Enabling Grids for E-sciencE

- JobType
  - *Normal* (simple, sequential job), *DAG*, *Interactive*, *MPICH*, *Checkpointable*
- Executable **(mandatory)**
  - The command name
- Arguments **(optional)**
  - Job command line arguments
- StdInput, StdOutput, StdError **(optional)**
  - Standard input/output/error of the job
- Environment
  - List of environment settings
- InputSandbox **(optional)**
  - List of files on the UI's local disk needed by the job for running
  - The listed files will be staged automatically to the remote resource
- OutputSandbox **(optional)**
  - List of files, generated by the job, which have to be retrieved

JDL: Relevant Attributes (2)

Enabling Grids for E-sciencE

- Requirements
  - Job **requirements on computing resources**
  - Specified using attributes of resources published in the Information Service
  - If not specified, default value defined in UI configuration file is considered
    - Default: *other.**GlueCEStateStatus*** == *"Production"* (the resource has to be able to accept jobs and dispatch them on WNs)
- Rank
  - **Expresses preference** (how to rank resources that have already met the Requirements expression)
  - Specified using attributes of resources published in the Information Service
  - If not specified, default value defined in the UI configuration file is considered
    - Default: - *other.**GlueCEStateEstimatedResponseTime*** (the lowest estimated traversal time)
    - Default: *other.**GlueCEStateFreeCPUs*** (the highest number of free CPUs) for parallel jobs (see later)

JDL: Relevant Attributes (3)

Enabling Grids for E-sciencE

- **InputData**
  - Refers to data used as input by the job: these data are published in the Replica Catlog and stored in the Storage Elements)
  - LFNs and/or GUIDs

Details in Data Management lecture

- **InputSandbox**
  - Execuable, files etc. that are sent to the job
- **DataAccessProtocol (mandatory if InputData has been specified)**
  - The protocol or the list of protocols which the application is able to speak with for accessing *InputData* on a given Storage Element
- **OutputSE**
  - The Uniform Resource Identifier of the output Storage Element
  - RB uses it to choose a Computing Element that is compatible with the job and is close to Storage Element

Example of JDL File

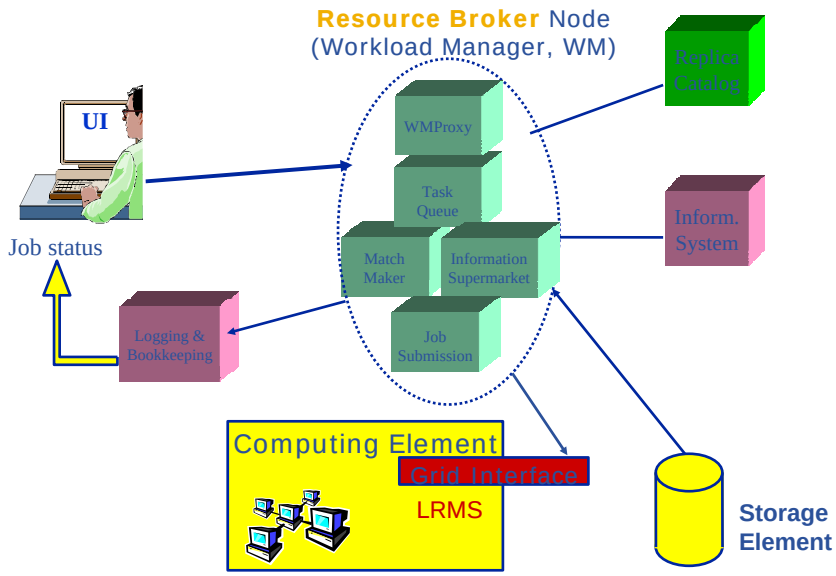
```
[
JobType="Normal";
Executable = "gridTest";
StdError = "stderr.log";
StdOutput = "stdout.log";
InputSandbox = {"/home/mydir/test/gridTest"};
OutputSandbox = {"stderr.log", "stdout.log"};
InputData = {"lfn:/grid/myvo/mylfn" };
DataAccessProtocol = "gridftp";
Requirements = other.GlueHostOperatingSystemNameOpSys
 == "LINUX"
              && other.GlueCEStateFreeCPUs>=4;
Rank = other.GlueCEPolicyMaxCPUTime;
]
```


General concepts of Grid Workload Management Systems

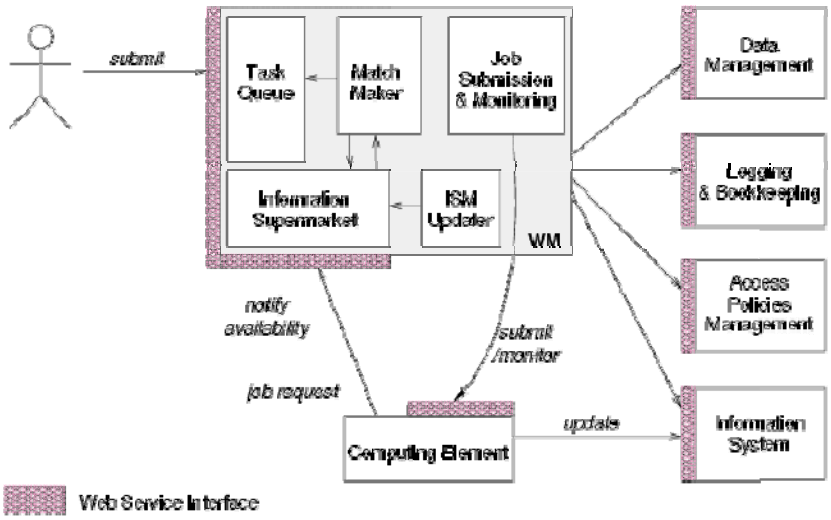
EGEE Workload Management System

- Job Preparation
- Architecture / Job submission and status monitoring
- Matchmaking
- Different job types

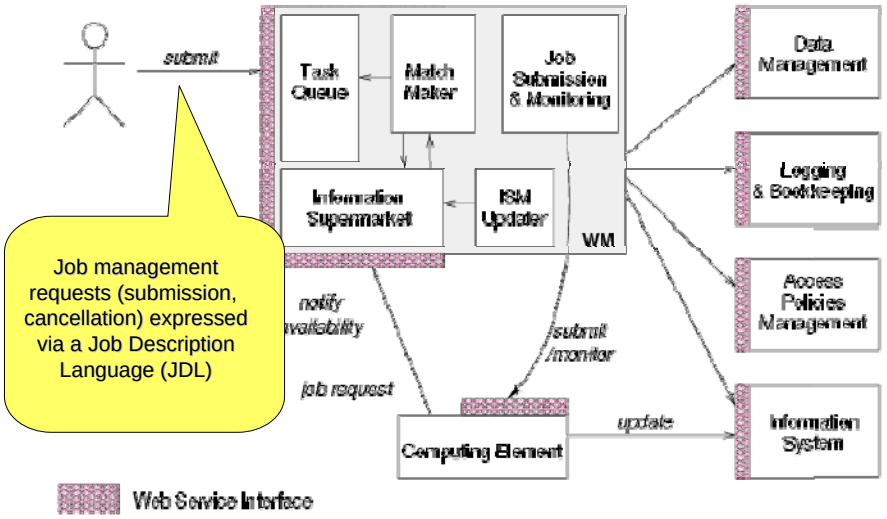
Simplified Architecture Overview



Architecture (1)

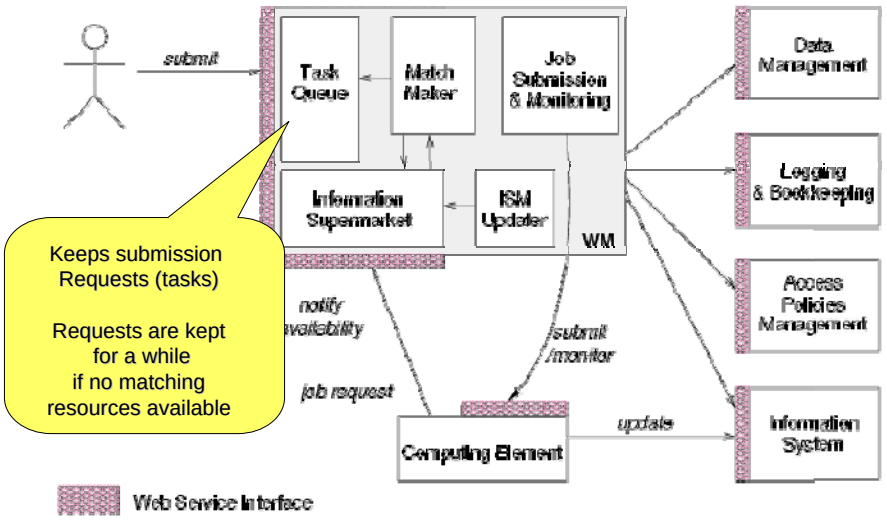


Architecture (2)



Job management requests (submission, cancellation) expressed via a Job Description Language (JDL)

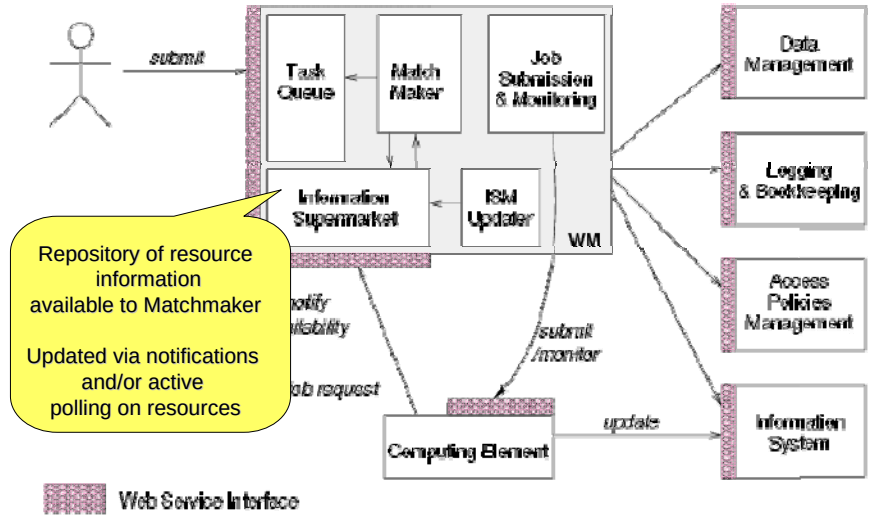
Architecture (3)



Keeps submission Requests (tasks)

Requests are kept for a while if no matching resources available

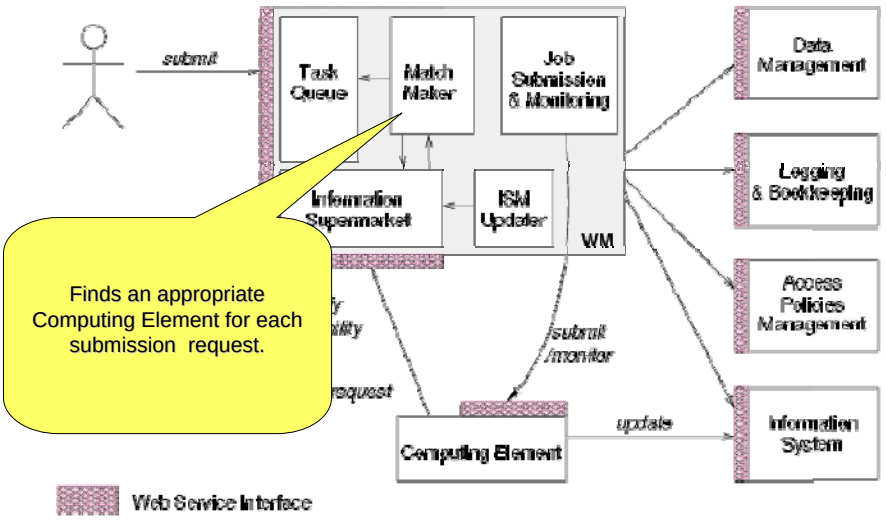
Architecture (4)



Repository of resource information available to Matchmaker

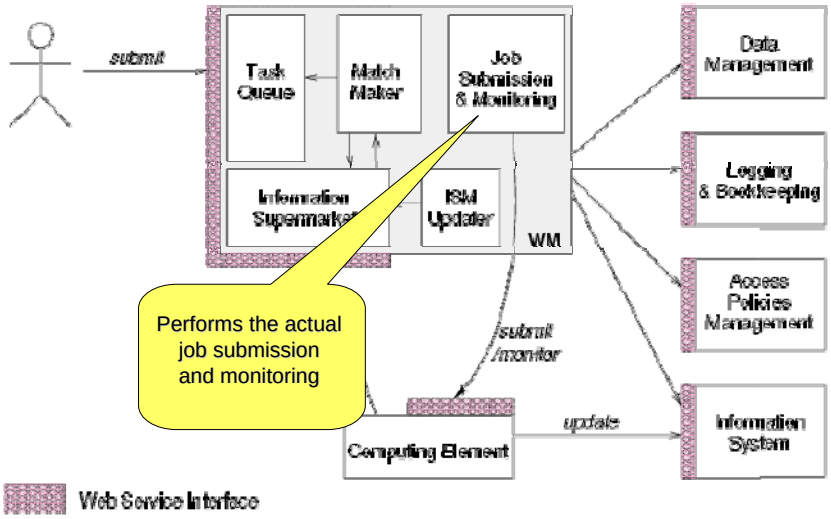
Updated via notifications and/or active polling on resources

Architecture (5)



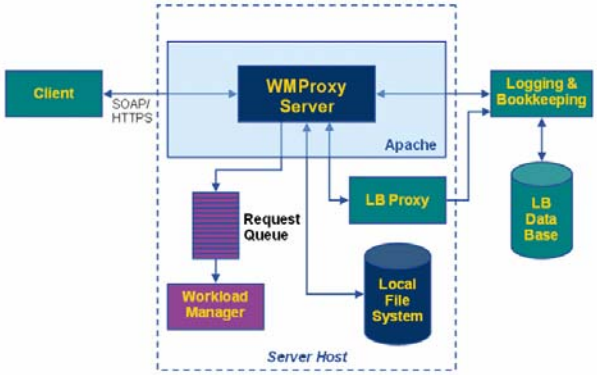
Finds an appropriate Computing Element for each submission request.

Architecture (6)

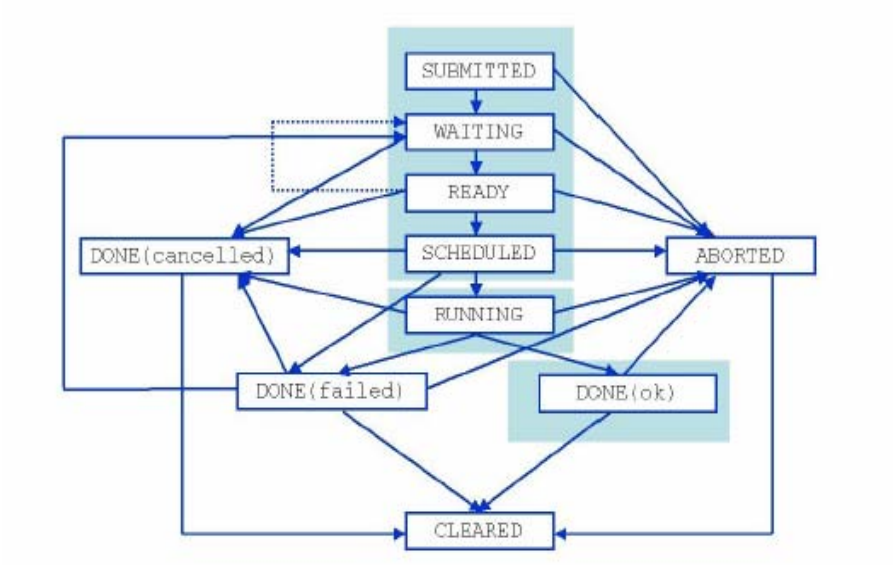


WMProxy service

- The Workload Manager Proxy is a component of the WMS
- Accessible through a Web Services based interface
- Handle more efficiently a large number of requests for job submission



Job States



Delegate a user proxy

```
glite-wms-job-delegate-proxy [-d <delegationID>]
 [-a autm_delegation][-c <config_file>][--vo <VO_name>]
 [-o <output_file>]
```

- **-d** the new delegated proxy is identified by the string *<delegationID>*
- **-a** the name of the new delegated proxy is automatically generated
- **-c** the configuration file *<config_file>* is pointed by the UI instead of the standard configuration file
- **--vo** the Virtual Organisation (if user is not happy with the one specified in the UI configuration file)
- **-o** information on the delegation operation is saved in this ouput file *<output_file>*

Job Submission

```
glite-wms-job-submit [-r <ce_id>] [-d <delegationID>]
 [-a autm_delegation][-c <config_file>][--vo <VO_name>]
 [-o <output_file>]  <job.jdl>
```

- **-r** the job is submitted directly to the computing element identified by *<ce_id>*
- **-d** submission is performed by a delegated proxy specified by *<delegationID>*
- **-a** submission is performed by a new delegated proxy, automatically generated
- **-c** the configuration file *<config_file>* is pointed by the UI instead of the standard configuration file
- **--vo** the Virtual Organisation (if user is not happy with the one specified in the UI configuration file)
- **-o** the generated edg_jobId is written in the *<output_file>*
  - Useful for other commands, e.g.:
    `glite-wms-job-status -i <input file> (or jobId)`

Job Resubmission

- **If something goes wrong, the WMS tries to reschedule and resubmit the job (possibly on a different resource satisfying all the requirements)**
- **Maximum number of resubmissions: min(RetryCount, MaxRetryCount)**
  - RetryCount: JDL attribute
  - MaxRetryCount: attribute in the “RB” configuration file
- **One can disable job resubmission for a particular job: *RetryCount=0;* in the JDL file**

Other (most relevant) UI commands

- `glite-wms-job-list-match`
  - Lists resources matching a job description
  - Performs the matchmaking without submitting the job
- `glite-wms-job-cancel`
  - Cancels a given job
- `glite-wms-job-status`
  - Displays the status of the job
- `glite-wms-job-output`
  - Returns the job-output (the OutputSandbox files) to the user
- `glite-wms-job-logging-info`
  - Displays logging information about submitted jobs (all the events "pushed" by the various components of the WMS)
  - Very useful for debugging purpose

General concepts of Grid Workload Management Systems

EGEE Workload Management System

Job Preparation

Architecture / Job submission and status monitoring

Matchmaking

Different job types

WMS Matchmaking

- **The RB (Matchmaker) has to find the best suitable Computing Element (CE) where the job will be executed**
- **It interacts with data management services and Information Services**
  - They provide all the information required for the resolution of the matches
- **The CE chosen by RB has to match the job requirements (e.g. runtime environment, data access requirements, and so on)**
- **If *FuzzyRank=False* (default):**
  - If 2 or more CEs satisfy all the requirements, the one with the best Rank is chosen
  - If there are two or more CEs with the same best rank, the choice is done in a random way among them
- **If *FuzzyRank=True* in the JDL:**
  - Fuzziness in CE choice: the CE with highest rank has the highest probability to be chosen

WMS Matchmaking Scenarios

- **Possible scenarios for matchmaking:**
  1. **Direct** job submission
     - *glite-wms-job-submit –d proxyID –r <CEId>*
     - Corresponds to job submission with Globus clients (globus-job-submit)
  2. Job submission with **computational requirements only**
     - No InputData nor OutputSE specified in the JDL
  3. Job submission with **data access requirements**
     - InputData and/or OutputSE specified in the JDL
     - *Details will be given in the Data Management lecture*

Example of Job Submission (1)

Enabling Grids for E-sciencE

- **User logs in UI (User Interface) machine**
- **User issues a *voms-proxy-init* , enters her certificate's password and gets a valid Grid proxy**
- **User sets up his JDL file**
- **Example of Hello World JDL file :**

```
[
  Executable = "/bin/echo";
 Arguments = "Hello World";
 StdOutput = "Message.txt";
 StdError = "stderr.log";
 OutputSandbox = {"Message.txt","stderr.log"};
]
```


Example of Job Submission (2)

Enabling Grids for E-sciencE

- **User delegates his proxy to the WMProxy service:**
  glite-wms-job-delegate-proxy –d proxyID
- **User issues a:** ***glite-wms-job-submit –d proxyID HelloWorld.jdl***
  and gets back a unique Job Identifier (JobId)
- **User issues a:** ***glite-wms-job-status JobId***
  to get logging information about the current status of her Job
- **When the "Output" status is reached, the user can issue a**
  glite-wms-job-output JobId
  and the system returns the name of the temporary directory where the job output can be found on the UI machine.

Example of Job Submission (3)

Enabling Grids for E-sciencE

$ glite-wms-job-delegate-proxy *-d myproxy*

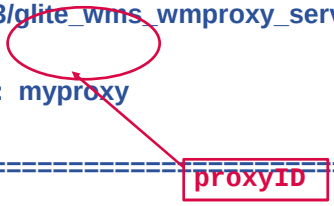
Connecting to the service https://glite-rb2.ct.infn.it:7443/glite_wms_wmproxy_server

========== glite-wms-job-delegate-proxy Success ==========

Your proxy has been successfully delegated to the WMProxy:
https://glite-rb2.ct.infn.it:7443/glite_wms_wmproxy_server

with the delegation identifier: myproxy

==========



Example of Job Submission (4)

Enabling Grids for E-sciencE

$ glite-wms-job-submit *–d myproxy* HelloWorld.jdl

Connecting to the service https://glite-rb2.ct.infn.it:7443/glite_wms_wmproxy_server

========== glite-wms-job-submit Success==========

The job has been successfully submitted to the WMProxy
Your job identifier is:

https://glite-rb2.ct.infn.it:9000/fiNwrsi2tJ-6rTn6thfioQ

==========

Example of Job Submission (5)

Enabling Grids for E-sciencE

$ glite-wms-job-status https://glite-rb2.ct.infn.it:9000/fiNwrsi2tJ-6rTn6thfioQ

```
**********************************************************
BOOKKEEPING INFORMATION:

Status info for the Job : https://glite-rb2.ct.infn.it:9000/fiNwrsi2tJ-6rTn6thfioQ
Current Status:     Done (Success)
Exit code:          0
Status Reason:      Job terminated successfully
Destination:        grid011f.cnaf.infn.it:2119/jobmanager-lcgpbs-infinite
Submitted:          Thu May 10 14:29:14 2007 CEST
**********************************************************
```


Example of Job Submission (6)

Enabling Grids for E-sciencE

$ glite-wms-job-output --dir /tmp/mydir https://glite-rb2.ct.infn.it:9000/fiNwrsi2tJ-6rTn6thfioQ

```
Connecting to the service https://193.206.208.155:7443/glite_wms_wmproxy_server

================================================================================

                JOB GET OUTPUT OUTCOME

Output sandbox files for the job:
https://glite-rb2.ct.infn.it:9000/fiNwrsi2tJ-6rTn6thfioQ
have been successfully retrieved and stored in the directory:
/tmp/mydir

================================================================================

$ more /tmp/mydir/Message.txt

Hello World

$ more /tmp/mydir/stderr.log

$
```

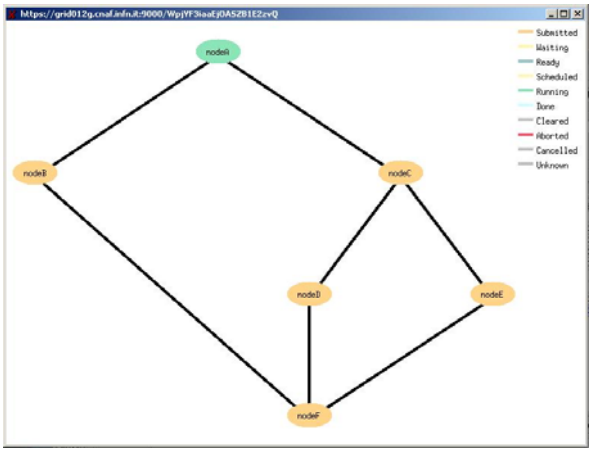
General concepts of Grid Workload Management Systems

EGEE Workload Management System

- Job Preparation
- Architecture/ Job submission and status monitoring
- Matchmaking
- Different job types

Job Dependencies

Condor's **DAGman** allows for job dependencies



DAG = Directed Acyclic Graph

DAG: JDL Structure

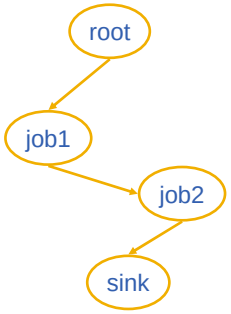
- JobType = “dag” ➡ *Mandatory*
- Requirements = “…” ➡ *Optional*
- Rank = “…” ➡ *Optional*
- InputSandbox = ➡ *Optional*
- ~~OutSandbox = “…”~~
- Nodes = … ➡ *Mandatory*
  Dependencies = … ➡ *Mandatory*

Attribute: Nodes

The *Nodes* attribute is the core of the DAG description;

```
[
   type = "dag";

   nodes = [
    nodeA = [
            file =“Job1.jdl";
           ];
    nodeB = [
            file =“Job2.jdl";
           ];
  ];
  dependencies = {{nodeA, nodeB}};
];
```

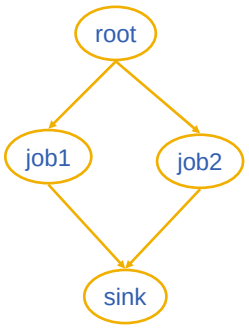


Attribute: Nodes

The *Nodes* attribute is the core of the DAG description;

```
[
  type = "dag";

  nodes = [
   nodeA = [
            file =“Job1.jdl";
           ];
   nodeB = [
            file =“Job2.jdl";
           ];
  ];
  dependencies = {};
];
```



Interactive Jobs

- **Specified setting JobType = “Interactive” in JDL**
- **When an interactive job is executed, a window for the stdin, stdout, stderr streams is opened**
  - Possibility to send the stdin to
  - the job
  - Possibility the have the stderr
  - and stdout of the job when it
  - is running
- **Possibility to start a window for**
- **the standard streams for a**
- **previously submitted interactiv**
- **job with command glite-wms-job-attach**

Further information

- Workload Management
  - http://www.glite.org/documentation/
  - In particular WMS User & Admin Guide and JDL docs
- Condor ClassAd
  - http://www.cs.wisc.edu/condor/classad
- Condor DAGman
  - http://www.cs.wisc.edu/condor/dagman/

Abbreviations

- JDL – Job Description Language
- RB – Resource Broker
- WMS – Workload Management System
- CE – Computing Element
- SE – Storage Element
- UI – User Interface
- WMProxy – Workload Manager Proxy

- Thanks to Workload Management Team for some slides:
  - Salvo Monforte, Marco Pappalardo, Valeria Ardizzone (INFN Catania)