



Enabling Grids for E-science

Practical using C++ WMPProxy API *advanced job submission*

Emidio Giorgio – emidio.giorgio@ct.infn.it

INFN Catania – EGEE NA3

CERN 23 Sep 2006

www.eu-egee.org



- Please bookmark the agenda page.
- You will need to refer to it during the practical.

- Browse to:

<http://agenda.cern.ch/fullAgenda.php?ida=a063196>

Follow the link “*More Information*” on the topic
Building blocks of reusable code

- **We are using the GILDA testbed today**
 - The production EGEE grid looks like this!
- **The practical exercises are to illustrate “how”**
 - Not using typical jobs for running on a grid!!
 - But to show how EGEE grid services are used, jobs are submitted, output retrieved...and how they can be accessed from your own programs
- **We will use the Application Programming Interfaces on a “User Interface” (UI) machine**
 - “UI” is your interface to the [GILDA] Grid
 - Where your digital credentials are held
 - Client tools and libraries are already installed

- **WMPProxy (Workload Manager Proxy)**
 - is a new service providing access to the gLite Workload Management System (WMS) functionality through a simple Web Services based interface.
 - has been designed to efficiently handle a large number of requests for job submission and control to the WMS
 - the service interface addresses the Web Services and SOA (Service Oriented Architecture) architecture standards
 - Beside the “classical” Command Line Interface has a complete set of API through which it can be accessed
 - APIs are available for Java, Python, C++

- A small set of “external” (to gLite) libraries are needed in order to successfully compile
- Everything needed for compilation and execution can be provided in the attached tarball
- Examples are just examples.....you may need to do some of operations and checks that are normally done automatically from CLI
- Not all the functions are implemented by the “official” API....has been necessary to add something....

- **Direct Acyclic Graphs of jobs (DAG):** set of jobs where the input, output, or execution of one or more jobs depends on one or more other jobs
- **Parametric Jobs:** they have one or more parametric attributes in the JDL, whose values vary according to a parameter
- **Job *Collection* :** a set of independent jobs that for some reason (known to the user) have to be submitted, monitored and controlled as a single request
- **JDL has been extended to allow specification of the input sandbox at the level of the compound request (i.e. DAGs, Collections and Parametric jobs)**
- **Input Sandbox can**
 - Be shared among nodes of collection/DAG → saving bandwidth use
 - Contain URI pointing to files on a remote gridFTP server

- **You will:**
 - Compile a source code which queries WMS on available resources
 - Compile a source code which submit a simple job (not compound) to the WMS
- **Please limit load on resources by reducing number of jobs submitted.**
- **Please work in pairs**

GLITE_WMPROXY_ENDPOINT=

https://glite-rb3.ct.infn.it:7443/glite_wms_wmproxy_server

- **WMS User's Guide**
 - <https://edms.cern.ch/file/572489/1/EGEE-JRA1-TEC-572489-WMS-guide-v0-2.pdf>
- **WM Proxy quick start**
 - <http://trinity.datamat.it/projects/EGEE/wiki/wiki.php?n=WMPProxyClient.QuickStart>
- **WM Proxy API documentation**
 - <http://trinity.datamat.it/projects/EGEE/wiki/wiki.php?n=WMPProxyAPI.APIDocumentation>
- **JDL Attributes Specification for WM Proxy**
 - <https://edms.cern.ch/file/590869/1/EGEE-JRA1-TEC-590869-JDL-Attributes-v0-8.pdf>