The gLite Workload Management System

Elisabetta Molinari (INFN-Milan)
on behalf of the JRA1 IT-CZ cluster

Outline

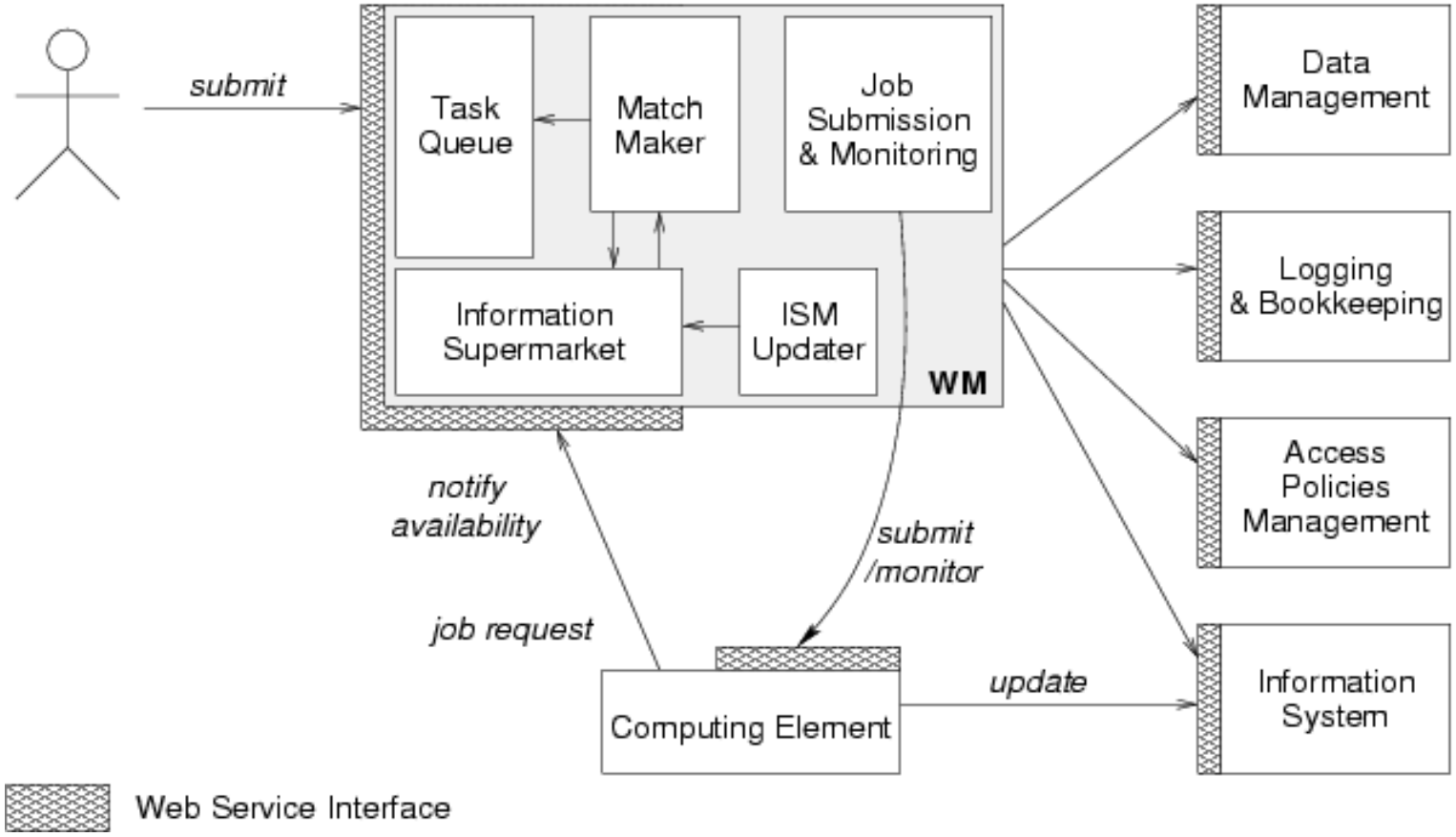
- The gLite Workload Management System
- Main differences to LCG-2
- File Catalogs Interfaces supported
- WMProxy and new Job Types supported:
  - Parametric jobs, collections, dags,...

- High level job control tools
- Some testing results

What the WMS Is

- The Workload Management System (WMS) is a collection of components providing a service **responsible** for the **distribution** and **management** of tasks across resources available on a Grid, in such a way that applications are conveniently, efficiently and effectively executed
- Tasks = Jobs to be submitted to the WMS are described via JDL (Job Description Language) and passed to the match-maker to find the best available resource that satisfies the requirements

WMS Internal Architecture



from “EGEE Middleware Architecture”, EU deliverable DJRA1.1, August 2004
https://edms.cern.ch/document/476451/

WMS Components Overview

- The WMS components handling jobs are:
  - **WMProxy: a service providing access to WMS functionality through a Web Services base interface. It validates, converts and prepares jobs and sends them to the WMS.**
  - **WorkLoad Manager**: the core component of the Workload Management System. Given a valid request it has to take the appropriate actions to satisfy it, among which finding the resources that best match the given requirements.
  - **Logging and Bookkeeping: provides support for the job monitoring functionality, it stores logging and bookkeeping information concerning events generated by the various components of the WMS. Using this information, the LB service keeps a state machine view of each job.**

Job Description Language

- Uniform language to express the characteristics, requirements and preferences of a job
  - https://edms.cern.ch/document/590869/1

```
[
  Executable = "my_exe";
  StdOutput = "out";
  Arguments = "a b c";
  InputSandbox = {"/home/user_1/my_exe"};
  OutputSandbox = {"out"};
  Requirements = Member(
    other.GlueHostApplicationSoftwareRunTimeEnvironment,
    "ALICE-3.07.01"
  );
  Rank = -other.GlueCEStateEstimatedResponseTime;
  RetryCount = 3
]
```

LCG2 – gLite Main differences

New components in gLite:

- **Task Queue:** the WM keeps a queue of pending submission requests, a list of jobs to be submitted together with their requirements. Non-matching requests will be retried periodically.
- **Information Supermarket:** read-only cached repository of information on available resources
- **WMProxy Server:** web server interface to submit jobs
- **LBProxy:** more efficient and reliable Logging and Bookeeping server.
- **CondorC:** reliable job submission between the WM and the CE (more reliable than Globus GRAM)

Lcg – gLite differences cont'd

New Features in gLite

- **Bulk submission:**
  - **DAGs, collections and parametric jobs**
- **Shallow resubmission:** re-submission of the job if it fails before the user job starts running(found to greatly improve success rate)
- **VoViews:** support for VoViews with VOMS FQAN-based access control rule
- **Automatic zipping of sandboxes, sandboxes from/to gridftp servers**
- **Voms extension renewal (via the glite-proxy-renewald service)**
- **Job perusal:** allows job's files content inspection while the job is running
- **Short Deadline Jobs:** the job is submitted to the right queue on the CE (ShortDeadlineJob = true)
- **High Load Limiter:** script to prevent new submissions in case of High Load of the WMS
- **Prolog and Epilog:** scripts that are run before and after the user job starts running

GLite File Catalogs

- Interface from the WMS to the following data catalogs (via the broker info component) is supported:
  - DLI – LCG Data Location Interface

```
DataRequirements = {
[
        DataCatalogType = "DLI";
        DataCatalog = "https://cms.org:8877/dli";
        InputData = {"lfn:/my/test/data1","guid:44rr44rr77hh77kkaa3",
    "lds:my.test.dataset",
    "query:my_query"};
]
```

  - Storage Index – gLite Storage Index

```
[
        DataCatalogType = "SI";
        DataCatalog = "https://glite.org:9443/StorageIndex";
        InputData = {"lfn:/eo/test.file", "guid:ddffrg5451"};
        ]
```

gLite File Catalogs cont'd

- RLS – LCG Replica Location Service

```
[
DataCatalogType = "RLS";
DataCatalog = "https://eu-datagrid.org/RLS";
InputData = {"lfn:/atlas/test.file", "guid:ggrgrg5656"};
]
```

- In the case of Storage Index and DLI the Data Catalog attribute is optional, if not specified the endpoint is found via Service Discovery.

Some supported job types

- **The gLite WMS supports some new job types:**
  - **Interactive Jobs:** jobs whose standard streams are forwarded to the submitting client
  - **MPICH:** a parallel application using MPICH-P4 implementation of MPI
  - **DAGs:** direct acyclic graphs, set of jobs with dependencies
  - **Collections:** group of jobs with no dependencies
  - **Parametric Jobs:** jobs having one or more attributes in the JDL that vary their values according to parameters, submitted jobs are instances of the same job, where a different value is assigned to parametric attributes

Parametric jobs example

```
[
JobType = "Parametric";
Executable = "cms_sim.exe";
StdInput = "input_PARAM_.txt";
StdOutput = "myoutput_PARAM_.txt";
StdError = "myerror_PARAM_.txt";
Parameters = 10000;
ParameterStart = 1000;
ParameterStep = 10;
InputSandbox = {
        "file:///home/cms/cms_sim.exe",
        "file:///home/cms/data/input_PARAM_.txt "
        };
OutputSandbox = {
"myoutput_PARAM_.txt",
"myerror_PARAM_.txt" };
OutputSandboxDestURI = "gsiftp://neo.datamat.it:5432/tmp";
Requirements = other.GlueCEInfoTotalCPUs > 2;
Rank = other.GlueCEStateFreeCPUs;
]
```

- The submission of the JDL will result in the generation of N jobs, where
- N = (Parameters – ParameterStart)/ParameterStep

High Level Job Control tools

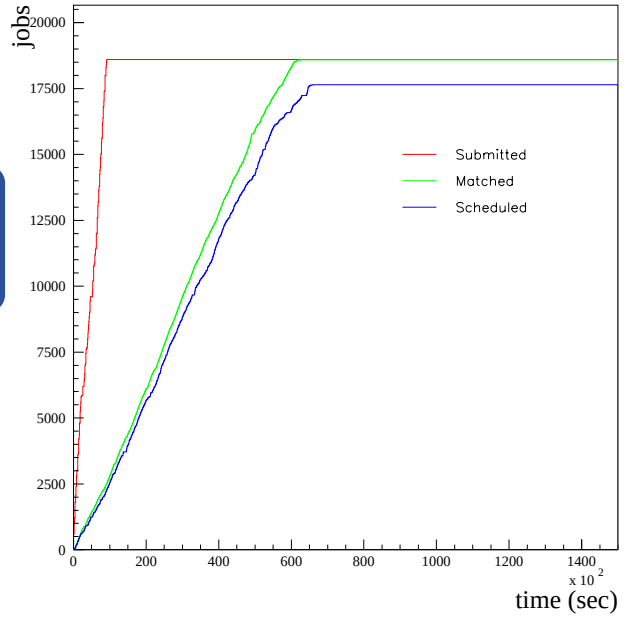
- **WMProxy** is a web service that lets the user to submit jobs to the WMS:
  - WSDL: web service based interface (client stubs can be genarted directly from it with the preferred tool/language)
  - API: C++, Java, Python bindings. Thin layer around the WS stubs
  - C++ command line interface
  - API Documentation:
- http://trinity.datamat.it/projects/EGEE/wiki/wiki.php?n=WMProxyAPI.JobSubmission

- **LB:** Logging and Bookeeping server that provides support to the job monitoring functionality:
  - It has a wsdl that also can be used to generate java client stubs.
  - It also can be queried via c api:
    - http://egee.cesnet.cz/en/JRA1/LB-guide.pdf

gLite WMS - CMS Results

- ~20000 jobs submitted
  - 3 parallel UIs
  - 33 Computing Elements
  - 200 jobs/collection
    - Bulk submission
- Performances
  - ~ 2.5 h to submit all jobs
    - 0.5 seconds/job
  - ~ 17 hours to transfer all jobs to a CE
    - 3 seconds/job
    - 26000 jobs/day
- Job failures
  - Negligible fraction of failures due to the gLite WMS
    - Either application errors or site problems

~7000 jobs/day on the LCG RB

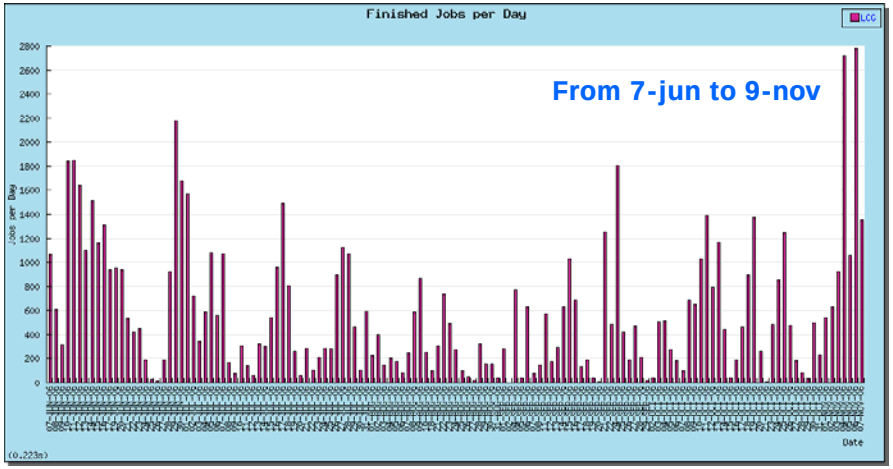


Failure reason	Job fraction (%)
Application error	28
Remote batch system	3.9
CRL expired	3.3
Worker Node problem	1.1
Gatekeeper down	0.2

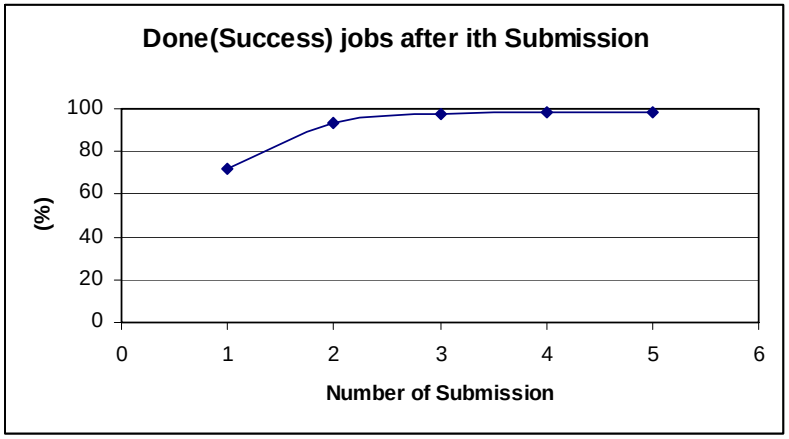
By A.Sciabà - 27 September 2006

gLite WMS - ATLAS Results

- Official Monte Carlo production
  - Up to ~3000 jobs/day
  - Less than 1% of jobs failed because of the WMS in a period of 24 hours



- Synthetic tests
  - Shallow resubmission greatly improves the success rate for site-related problems
    - Efficiency =98% after at most 4 submissions



By A.Sciabà – 10 November 2006

Forthcoming features

- High Availability of the RB: solution that makes the RB more robust and resistant to failures
- Job Provenance: Store and retain data on finished jobs, complementary to RB
- Implementation of collections not using DAGman
- ICE: interface to the new web service based CE (CREAM)
- **DGAS: storage accounting system that collects usage records on the CE**