

The gLite Workload Management System

M. Cecchi^(*), A. Dorise^(*), A. Ghiselli^(*), F. Giacomini^(*), A. Giannelle^(*), A. Maraschini^(**), M. Marzolla^(*), S. Monforte^(*), F. Pacini^(**), S. Pellegrini^(*),
L. Petronzio^(**), M. Sgaravatto^(*) - (*) Istituto Nazionale di Fisica Nucleare, (**) ElSag-Datamat s.p.a.

<http://egee-jra1-wm.mi.infn.it/egee-jra1-wm>



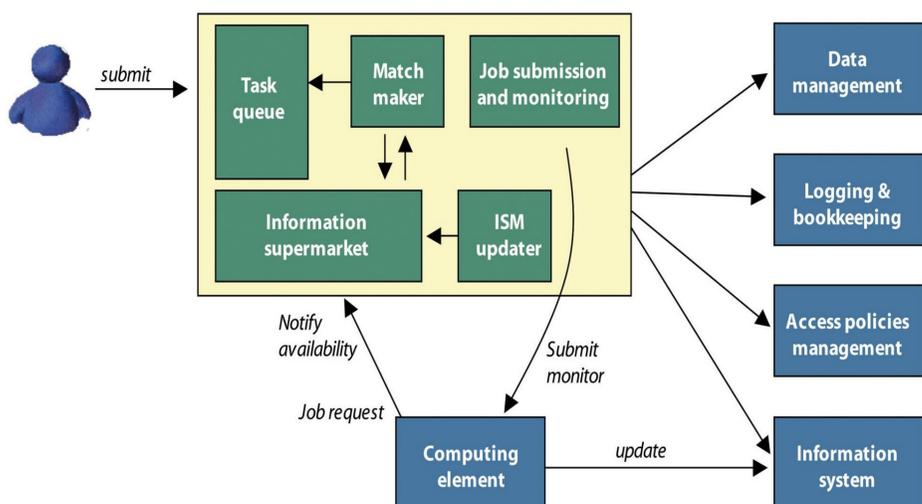
The gLite Workload Management System in a nutshell

The *gLite Workload Management System (WMS)* represents a **key entry point** to the computing services made available on a Grid. It provides a **reliable and efficient service** responsible for the **distribution and management of end-user computations** hiding both the inter-operation with a **highly heterogeneous and dynamic infrastructure** and the **prevention and recovery of faulty conditions**. This is accomplished without compromising performance and generality of approach; the provided level of abstraction of its design, in fact, has been kept generic enough to support applications coming from largely different domains.

The WMS receives requests concerning access to high-end Grid services to fulfill a demand basically for computation and storage, which is commonly referred to as 'job'. Such requests are described by the users, as a set of key/value pairs, in a flexible, high-level language. The WMS translates then this logical description into **concrete operations and decisions**, dictated by the overall status of the Grid services it interoperates with, taking responsibility to look after each and every incoming request on its way to **successful completion**. Several types of jobs are supported: ranging from simple jobs, batch or interactive, to a wide variety of compound jobs: intra-cluster MPI, collections (multiple jobs with a common description), parametrics (multiple jobs with one parametrized description), workflows in the form of DAGs and there is, moreover, on-going development of a generic workflow engine.

A key Grid service

- Hide the **complexity** of the Grid to end users
- Carry out the **Match-Making** process
- Interoperate** with other Grid services & other Grids
- Provide **added value** on top of Job Submission



Several job types

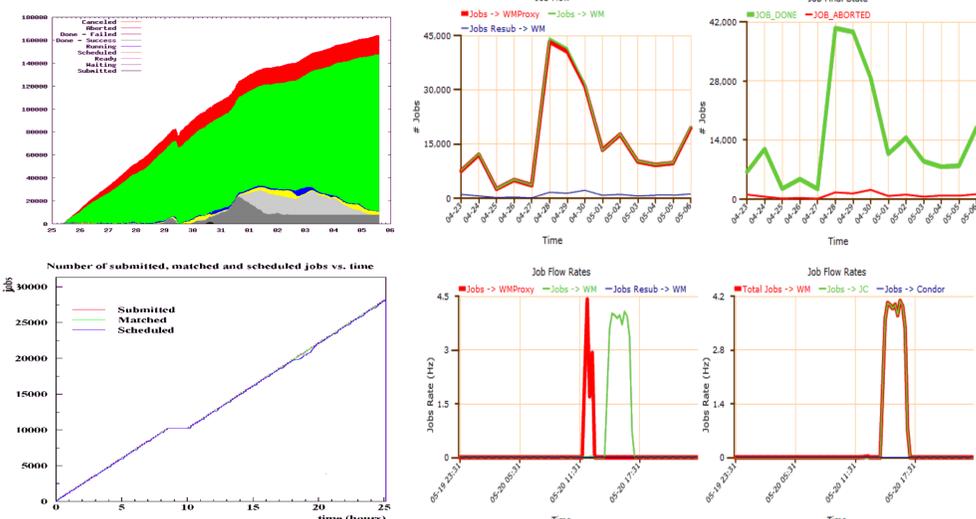
- Batch-like
- DAG workflow
- Collection
- Parametric
- MPI
- Interactive

Functionality

- Push (*eager*) and pull (*lazy*) scheduling policies by design
- Data-driven match-making: send jobs only where data are
- Automatic sandbox files handling - with support for multiple transfer protocols, compression and sharing
- Gang-matching – including storage elements in the MM
- Stochastic ranking for resource selection
- Automatic credential renewal
- Use of Service Discovery for obtaining new service endpoints
- Mechanisms for error prevention and recovery
- Load-limiting mechanism to prevent system congestion
- Bulk match-making: matching together jobs in clusters
- Faster authentication via explicit delegation
- Compliance to formal and *de-facto* standards, openness to forthcoming protocols thanks to its flexible design
- Interoperation with OSG and Nordugrid
- Interoperation with different Information Providers
- File peeking while output is being produced
- ...

Results & Performance

- The gLite WMS has been deployed in a number of different multi-user and multi-VO scenario
- A first successful acceptance test, accomplished at CERN as by Easter '07, was to be able to submit ~16K jobs/day over one whole week with no manual intervention on servers and stable memory usage. On a subsequent stress test a throughput of ~30Kjobs/day was reached (leftmost pictures below)
- Interoperation with Open Science Grid and Nordugrid has been achieved
- Recent stress tests at Imperial College and CNAF in a production-like environment showed a throughput of ~50 Kjobs/day (rightmost pictures).
- Ongoing redesign & optimisations are showing that a target throughput of ~100 Kjobs/day is within reach.



Future developments will be focused on such topics:

- Higher performance
- Stronger integration, stability and interoperability
- Load-balancing/high-availability
- Platform portability
- Reducing external software dependencies