The ATLAS and CMS Experience with the gLite Workload Management System

Andrea Sciabà

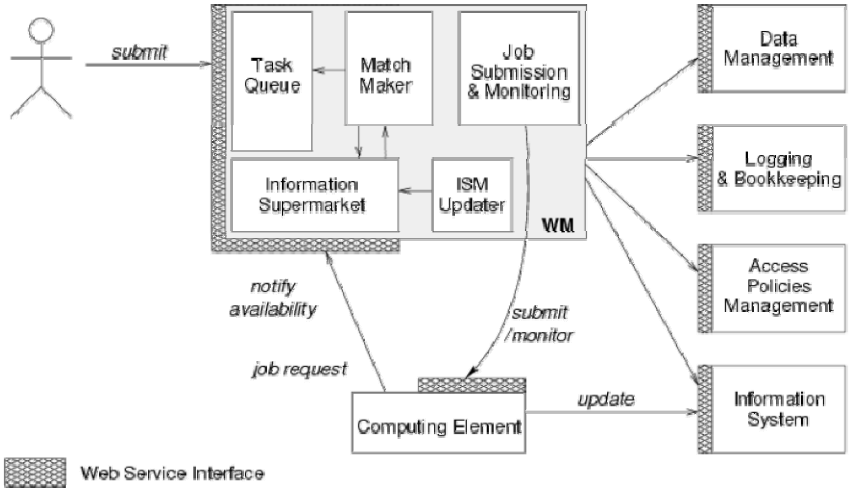
Simone Campana

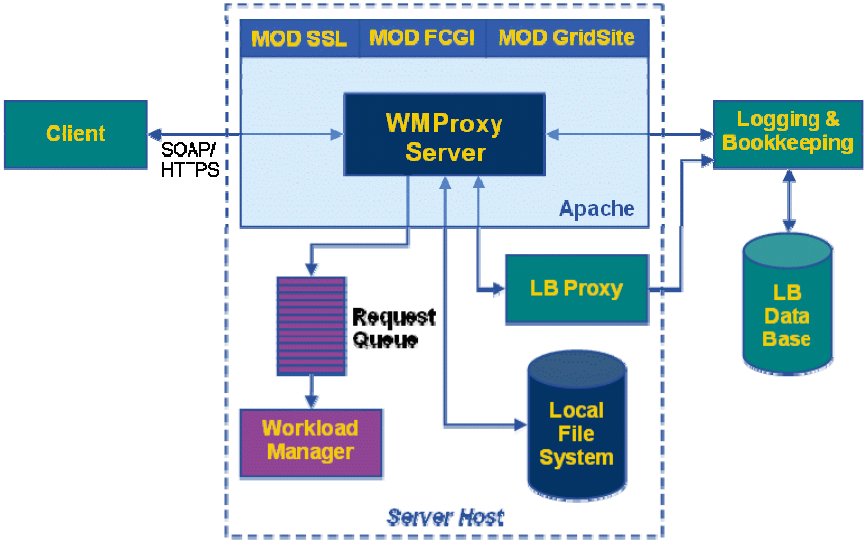
Introduction

- **The gLite Workload Management System**
- **The experiment applications**
  - CMS analysis
  - ATLAS Monte Carlo production
- **Tests of the WMS**
- **Results**
- **Conclusions**

gLite WMS architecture

- **The service to submit and manage jobs**
  - Task queue: holds jobs not yet dispatched
  - Information SuperMarket: caches all information about Grid resources
  - Match Maker: selects the best resource for each job
  - Job Submission & Monitoring
  - Interacts with Data Management, Logging & Bookkeeping, etc.



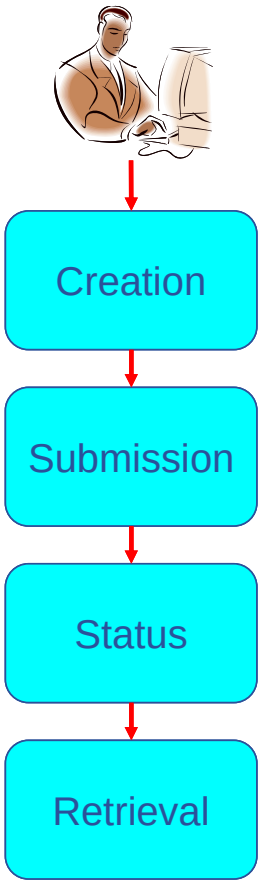


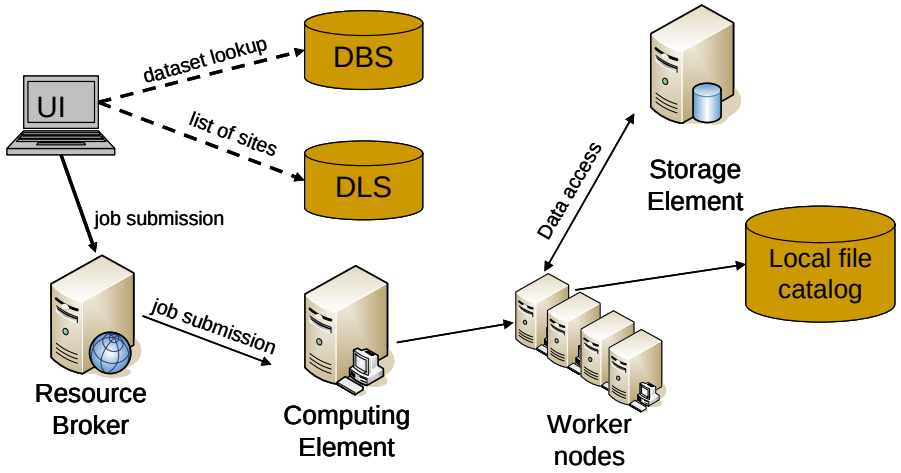
- **WMProxy service optimizes job management and stands between the user and the real WMS**
  - Service Oriented Architecture (SOA) compliant
    - Implemented as a SOAP Web service
  - Validates, converts and prepares jobs and sends them to the WM
  - Interacts with the L&B via LBProxy (a state storage of active jobs)
  - Implements most new features

Advantages of the gLite WMS

- **The gLite WMS offers several advantages over the old LCG WMS**
  - Bulk submission
    - Direct Acrylic Graphs (DAG): sets of jobs with dependencies among them
    - Collections: sets of independent jobs
    - Parametric jobs: sets of jobs with running parameters in the JDL
  - Job sandboxes
    - Shared input sandboxes for a collection
    - Download/upload of sandboxes via GridFTP, https, http
  - Faster authentication via WMProxy
  - Faster match-making
  - Faster response time for users
  - Higher job throughput
  - “Shallow” resubmission of failed jobs
    - a job is resubmitted if failed before reaching the Worker Node
    - Greatly improves the job success rates
  - Job File Perusal
    - To inspect the job output while it is running

Tested CMS application

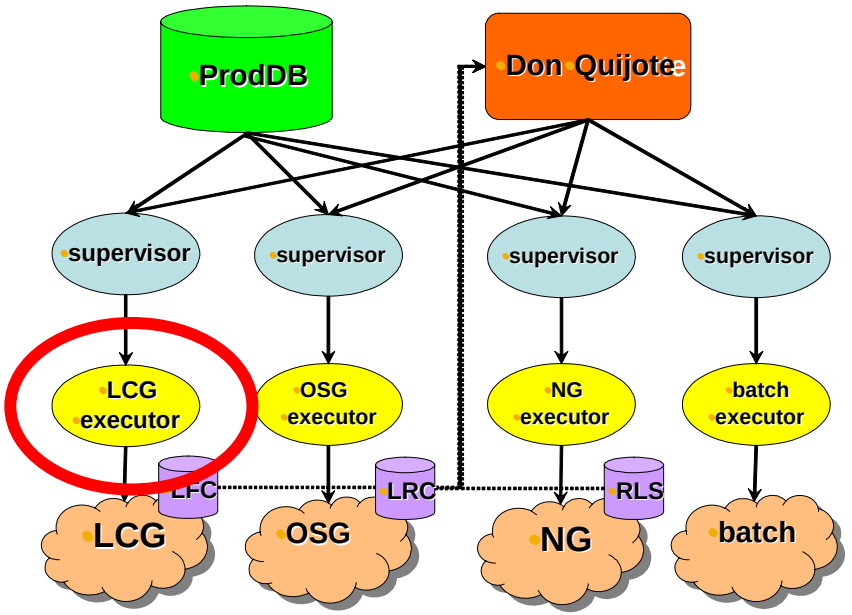




- **Analysis jobs with CRAB**
  - The user selects a dataset to analyze
  - The analysis task is split into many jobs
  - The jobs are submitted to sites hosting the data
  - The jobs run the locally installed CMS application on the specified data files
  - The user examines the status of the jobs and retrieves their output when they are finished

Tested ATLAS application

- **Production of simulated events**
  - A central database of jobs to be run
  - A “supervisor” for each Grid that takes jobs from the central database, submits them to the Grid, monitors them and checks their outcome
  - An “executor” acting as interface to the Grid middleware
    - EGEE/WLCG
      - *Lexor using the gLite WMS*
      - *Condor-G direct submission*

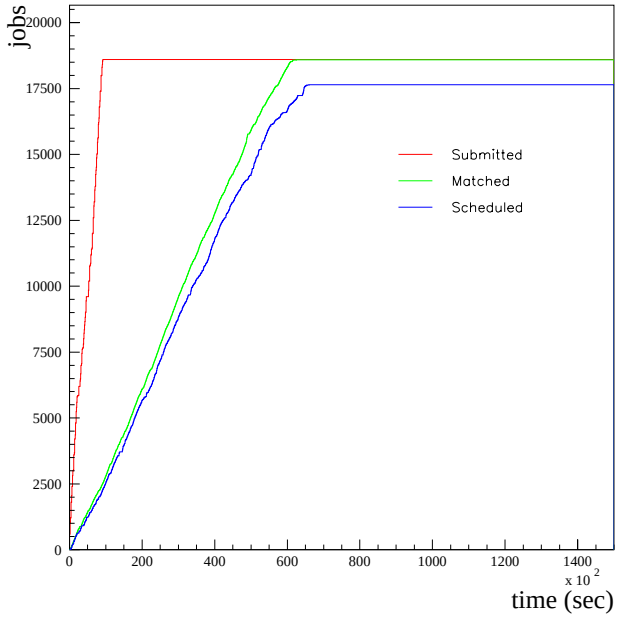


CMS Tests

- **Job characteristics**
  - Software: CMSSW 0.6.1
  - Data analyzed: test sample preinstalled at CMS sites
  - Approximate CPU time: 30’
- **Job submission**
  - Predefined number of jobs submitted at each CMS site
  - Various mechanisms tested
    - Network Server
      - *Extremely similar to the old LCG WMS*
    - WMProxy
      - *Faster submission rate than via NS*
    - Collections (“bulk submission”)
      - *Best possible submission speed*
  - Submission in parallel from up to three User Interfaces

Latest 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



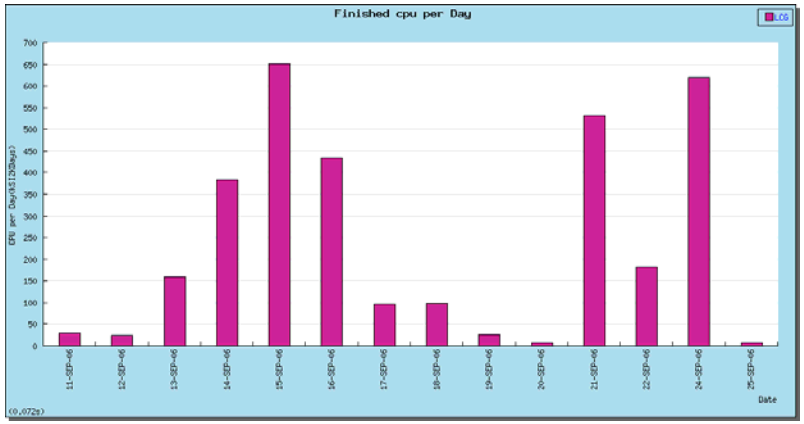
Failure reason	Job fraction (%)
Application error	28
Remote batch system	3.9
CRL expired	3.3
Worker Node problem	1.1
Gatekeeper down	0.2

ATLAS Tests

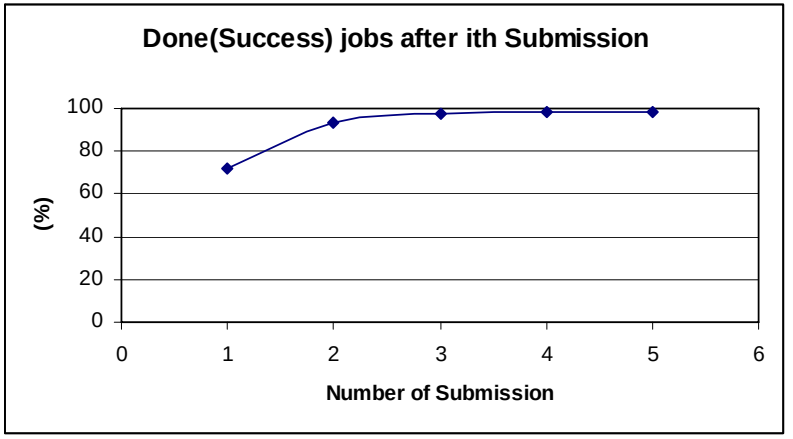
- **Used in real Monte Carlo production**
- **Job characteristics**
  - Simulation
    - Approximate CPU time: 3 h
  - Simulation
    - Approximate CPU time: 20 h
  - Reconstruction
    - Approximate CPU time: 3 h
- **Job submission**
  - Bulk submission
    - The supervisor groups jobs to be executed in collections of 100 jobs each
    - Each job in a collection can run on a different site
- **Also synthetic tests run**
  - Very simple jobs (“Hello world”) that can run anywhere
  - To study the impact of the shallow resubmission
  - To assess the reliability of the bulk submission

Latest ATLAS Results

- **Official Monte Carlo production**
  - Up to ~5000 jobs/day
  - Extremely low failure rate due to the gLite WMS
    - Over ~10000 jobs in the last 2 weeks, < 1% WMS-related failures



- **Synthetic tests**
  - gLite WMS at least as reliable as the LCG WMS
    - Confirmed by CMS tests
  - Shallow resubmission greatly improves the success rate for site-related problems
    - Efficiency =98% after at most 4 submissions



Conclusions

- **The gLite WMS has been seen so far to be as reliable as the LCG WMS**
  - The shallow resubmission actually improves the success probability
- **WMProxy allows to have a much better performance**
  - +20% in submission rate for single jobs compared to Network Server
  - 0.5 s/job for bulk submission, compared to ~5 s/job for single job submission via Network Server
  - ~3 s/job to dispatch jobs to CEs
  - ~ 26000 jobs/day for the tested CMS jobs
- **The performance and the reliability of the WMS has greatly improved over a short amount of time due to a very intense a fruitful collaboration among**
  - JRA1 developers
  - SA1 and SA3
  - The CERN fabric people
  - The ATLAS and CMS experiments