

# Grid Job Management with the gLite Middleware

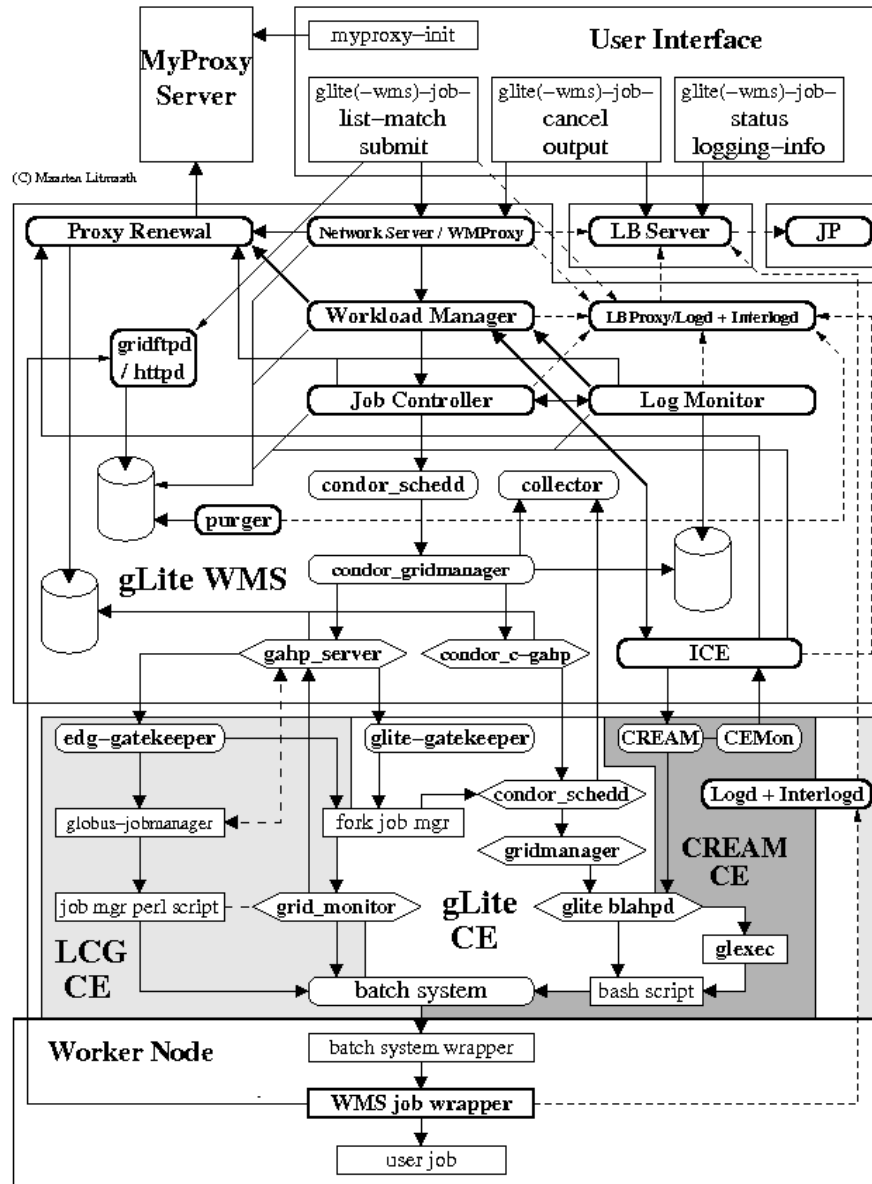
*CCR Workshop  
Rimini, 7-11 May 2007*

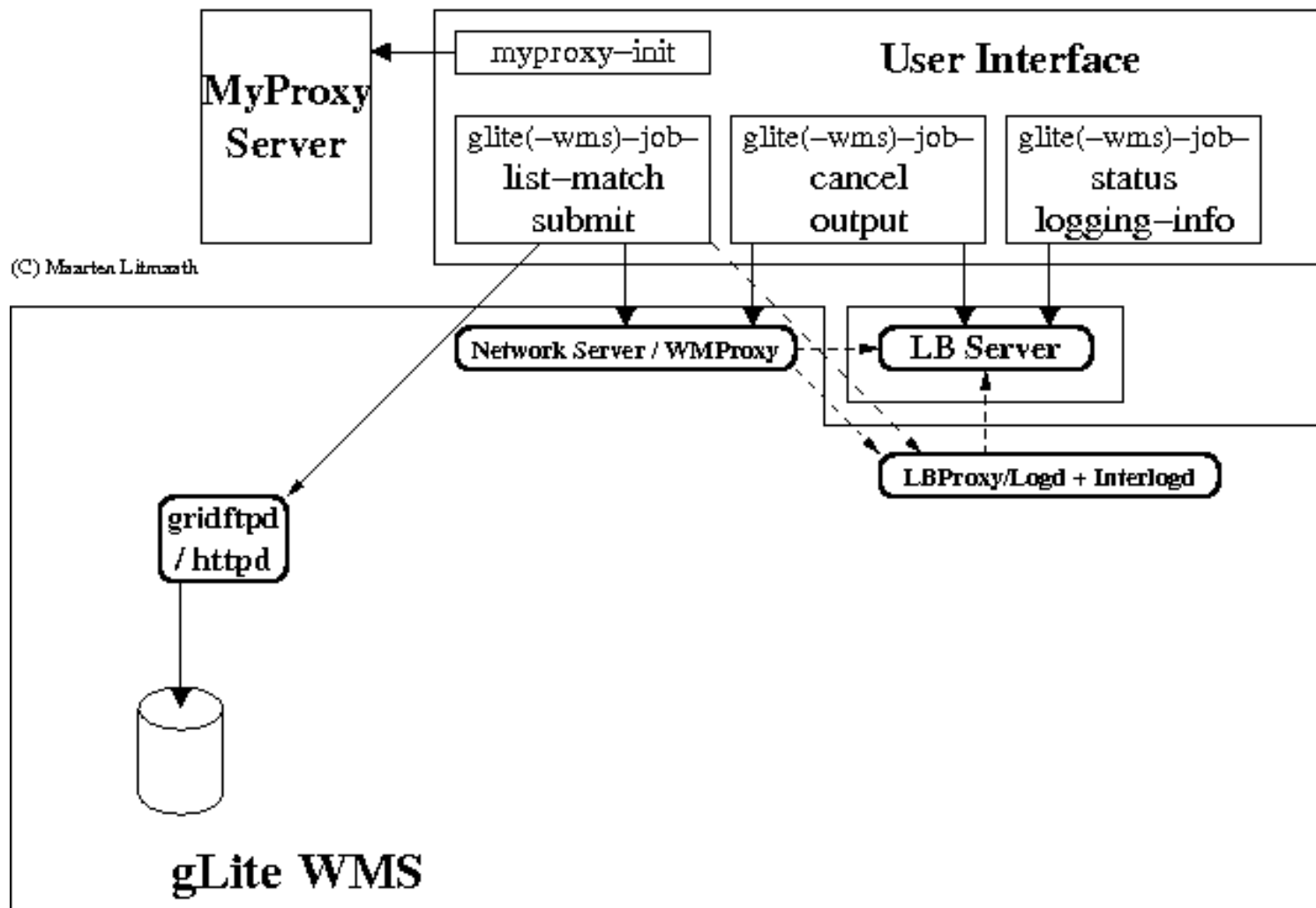
*Francesco Giacomini – INFN-CNAF*

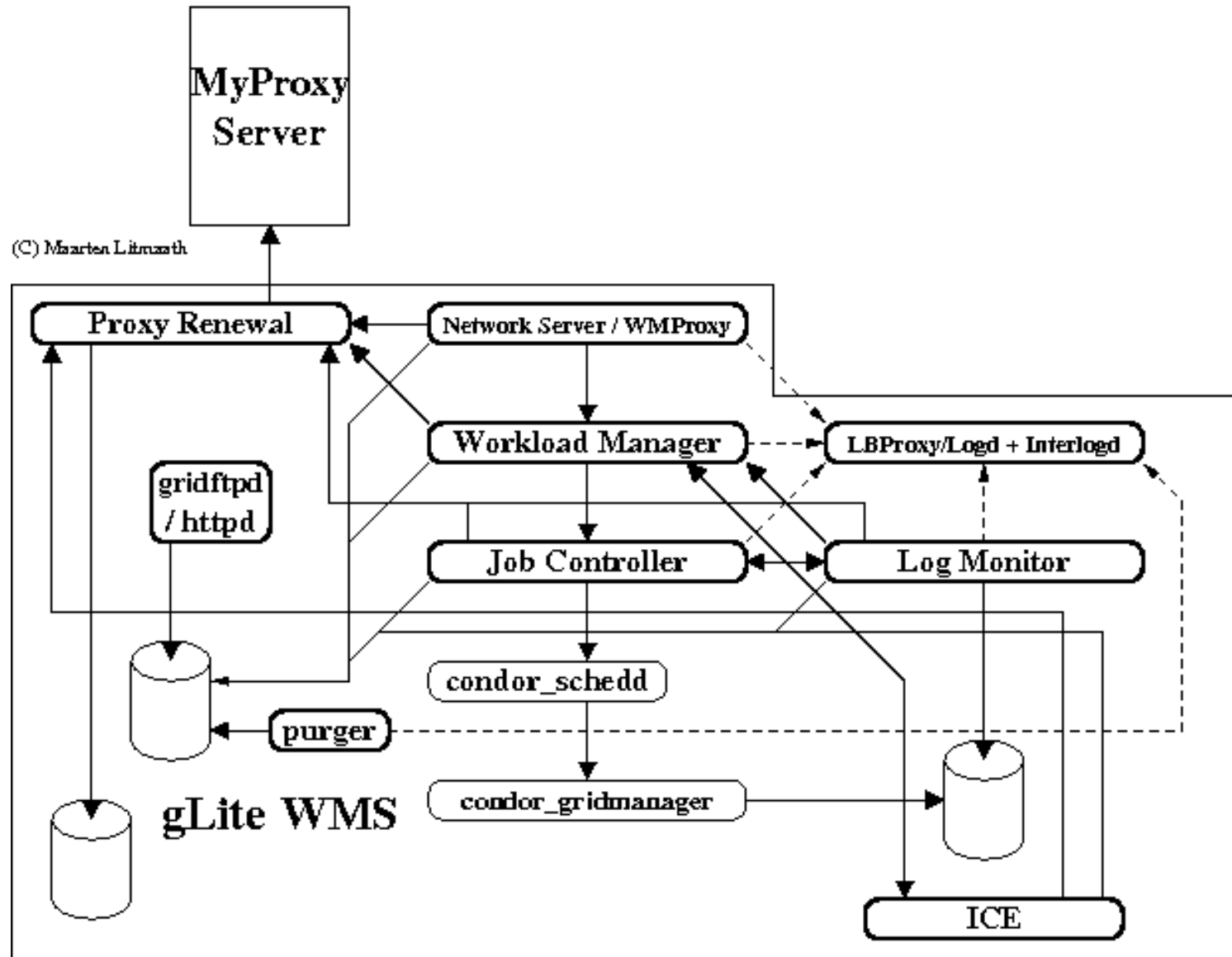
- **WMS/CE overview**
- **Current situation**
  - Recent improvements
  - Testing activity
- **Near-future plans**
  - gLite restructuring
- **Conclusions**

- The Workload Management System (WMS) comprises **a set of Grid middleware components** *responsible* for the *distribution and management* of tasks across Grid resources, in particular Computing Elements (CE), in such a way that applications are conveniently, efficiently and effectively executed
- Multiple processes on multiple machines
- Reliable communication, with persistency where/when needed
- Actions are done on behalf of the user, i.e. with delegated credentials

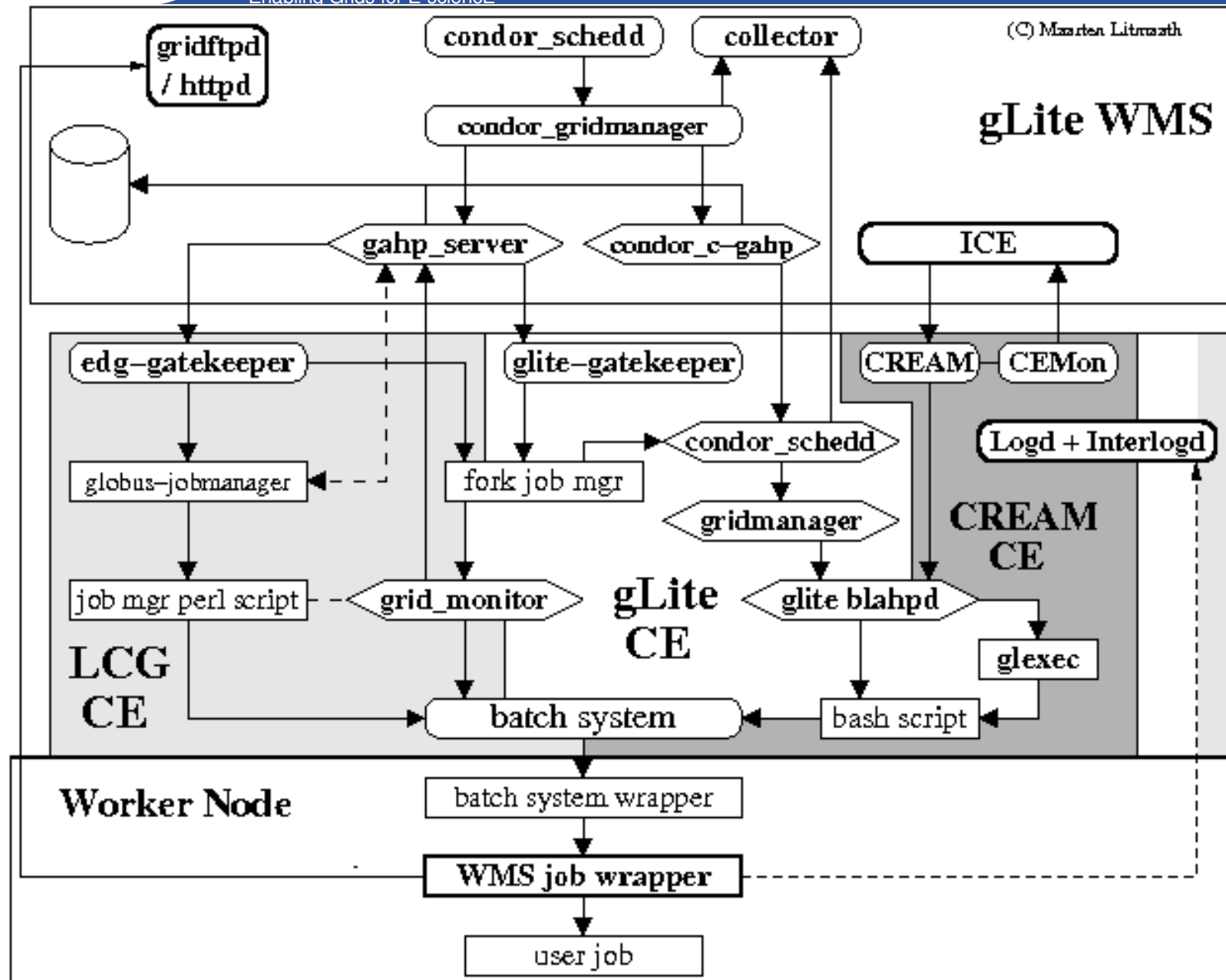
Enabling Grids for E-science







Enabling Grids for E-science



- **Towards adoption of standard specifications**
  - Web service interface both for WMS and CE
    - BES
  - JSDL
  - Multi-language (C++, Java, Python) APIs
- **Sandbox management**
  - Compression
  - Sharing
  - On remote server (i.e. not loaded on the WMS)
  - Support for multiple protocols, now gridftp and https
- **Bulk submission and matchmaking**
  - See later



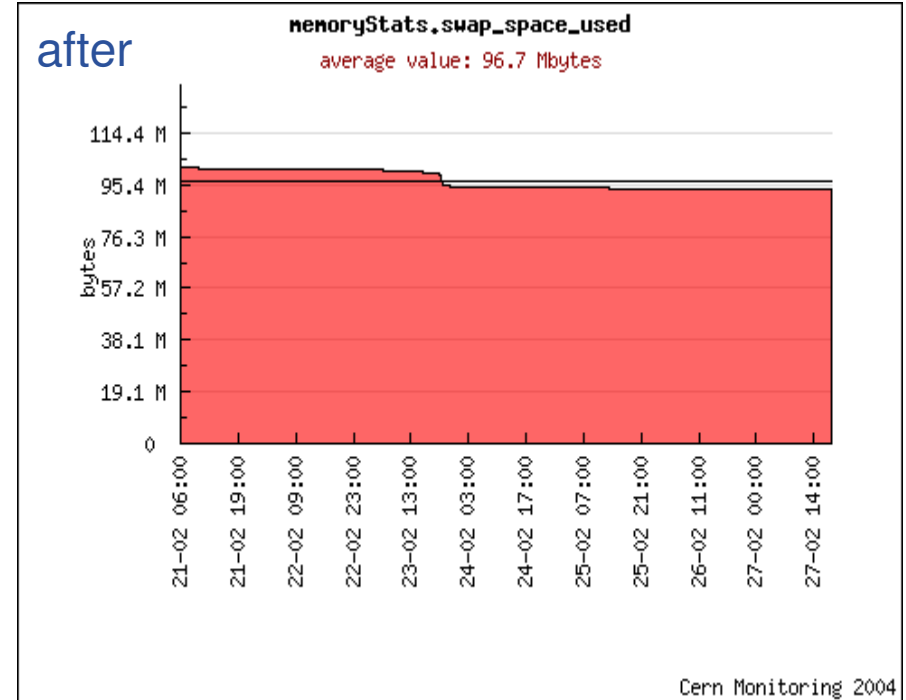
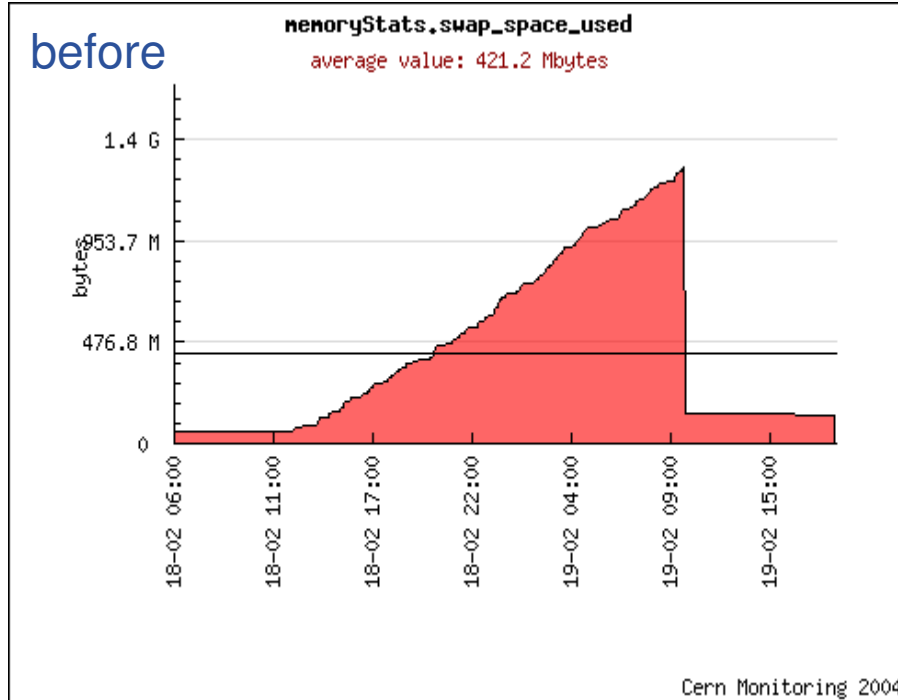
- **Service discovery**
  - To automatically find both WMS endpoints on the client side and LB endpoints on the server side
  - Some effort on leveraging the DNS is on-going
- **Job files perusal**
  - Look at the files produced by the job while it's running
- **Job Wrapper**
  - Interoperability with OSG
  - VO hook, i.e. allow applications to find middleware
- **NS deprecated**
- **Dismissed RLS support**

- **New components put in stand-by due to the prioritization of the certification and deployment activities**
  - Difficult to improve these components if not used by real users
  - CREAM (CE), G-PBox (policy mechanism)
- ***Preview test-bed* created by JRA1 to expose to users and site managers those components not yet considered for certification**
- **Slow start**
  - Problems identifying resources
  - Best-effort support

- **Scalability of some critical components needed improvements**
  - Stability as well
  - Workload Management System (WMS) and the Logging & Bookkeeping (LB)
  - gLite Computing Element (gLite CE)
- **The normal certification process is not adequate**
  - Tests need to be done at production scale
- **Direct and continuous involvement of developers cannot stop before the scalability tests**  
**⇒ organizational changes**

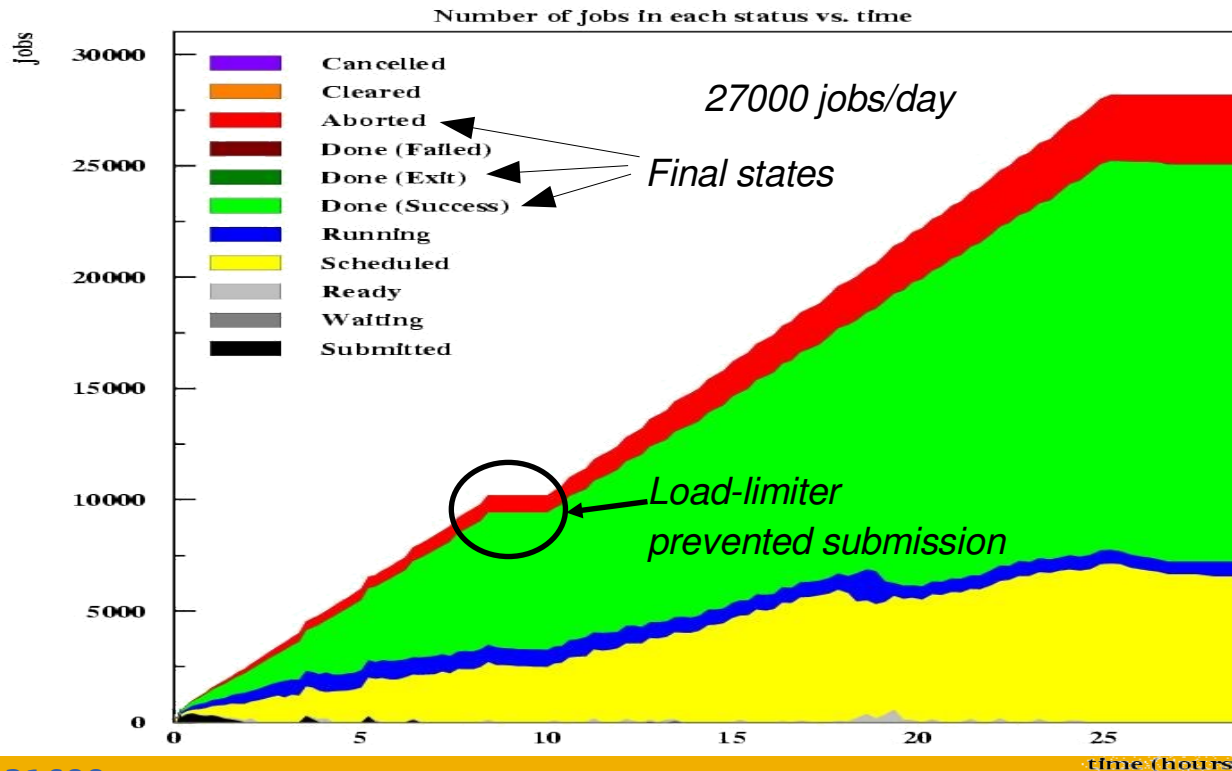
- ***Experimental Services were introduced***
  - WMS/LB at CNAF, gLite CE at CERN
  - Instances of the services attached to the production infrastructure
  - Maintained by SA1 and SA3
  - JRA1 patches are installed immediately (before the certification)
  - Testing done by selected application users
  - Process controlled by the EMT
- **The result was a rapid improvement of the components**
  - **WMS** and **LB** now ready for production
  - Most **gLiteCE** problems ironed out
    - But some not specific to the gLite CE, e.g. Proxy renewal didn't work at CERN because of a peculiar firewall setup between front end and WNs
  - Next will be the new **CREAM** Computing Element

- Evident memory leaks in some components



- Inadequate memory management in the C++ runtime
  - Use of alternative allocators (e.g. google-malloc)

- A single WMS/LB instance should demonstrate submission rates of at least 10 Kjobs/day sustained over 5 days, without the need to be restarted
- The number of stale jobs after 5 days must be  $< 5\%$



- **Bulk submission: possibility to submit a bunch of jobs in one single interaction with the WMS**
  - (possibly) heterogeneous → collection
  - Homogeneous → parametric
- **These jobs used to be transformed into a DAG and managed with Condor DAGMan**
  - Correct but overkill solution, because nodes don't have dependencies
  - Major source of instability and complexity of the system
  - Mitigation hacks, e.g. global limit on the number of planners
- **Now direct management**
  - Much smoother behavior

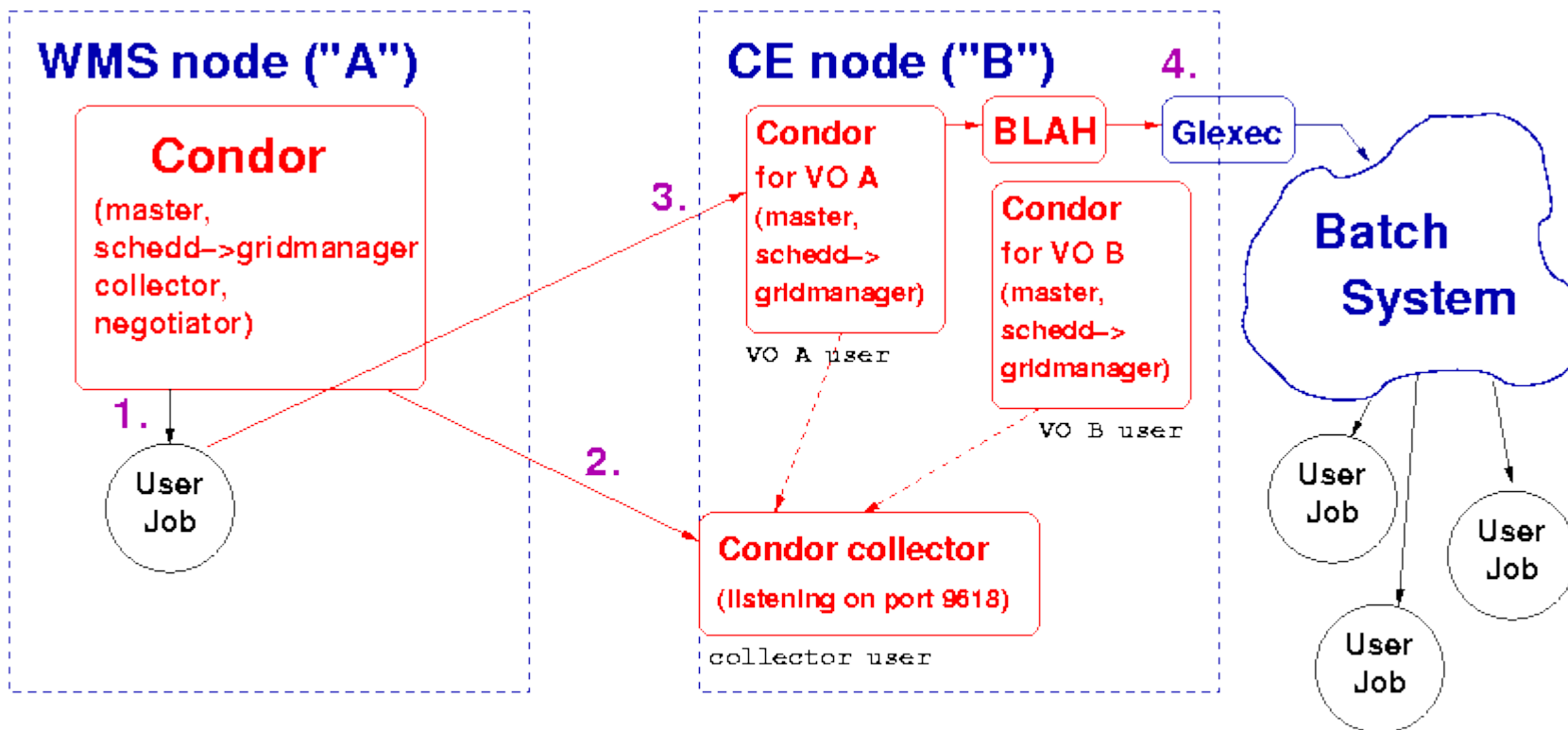
- **Natural completion of bulk submission**
  - Match “equivalent” jobs in one shot, i.e. with one single mm operation
- **Two jobs are equivalent if their *significant attributes* are literally the same**
- **The significant attributes are specified by the user**
  - Typically Requirements, Rank, FuzzyRank, ...



- **Batch-system Local Ascii Helper**
- **Interface to reliably transfer jobs to a batch system for execution**
- **From past production experience:**
  - It needs to be as simple and lightweight as reasonably achievable
  - It needs to keep as little state as reasonably achievable
- **Used by both the gLite CE and CREAM**

- **Pros:**
  - Practical and workable abstraction to map the job lifecycle onto a different system
  - Still a reasonably simple component with little dependencies
- **Cons:**
  - Hard to resist pressure to add more functionality to this supposedly thin layer
  - Needs some tracking with new versions of batch systems to synchronize with changes in client functionality and log formats

- Try replacing with more stable measures the (many) workarounds that had to be added in the Condor-G layer and the EDG RB to address GRAM troubles



1. Job is submitted to Condor on A
2. Condor on A queries collector on B
3. Job is transferred to Condor on B
4. Job is executed on batch system via BLAH and Glexec

- **INFN solution to the computing resource management problem**
- **Designed and implemented from scratch**
  - No legacy constraints
  - Adoption of WS standards
- **Thoroughly tested on the preview testbed**
  - Outperforms other CEs
- **Appeared on the experimental WMS**
  - ICE is the client side of CREAM, equivalent to JC + CondorG + LM
  - With the DAG-less collections, nodes can now be submitted to ICE
  - Direct support for bulk submission to ICE/CREAM is planned

- **Usual support, bug fixing, coordination, ...**
  - >50% of the time
- **The current functionality is reasonably complete**
  - Applications/frameworks can be built on top of the current system
    - e.g. Workflow management
  - New features will be added only on (strong) demand
- **Most development dedicated to improve stability, performance, scalability, ...**
  - The system is resource-hungry
- **Portability to other platforms**

- **User-friendliness**

- Review of error messages
- Is the provided interface adequate?

- **Sysadmin-friendliness**

- Simplify system installation/configuration/management
- Common Logging Format
  - security-driven
  - <https://edms.cern.ch/document/793208>
- Grid Monitoring Working Group
  - Started with a survey collecting information for each service: log files, files/directories, network ports, useful metrics, ...

- **The support of legacy components and the existing internal and external dependencies make the gLite stack too complex**
  - One of the causes of the delay of the gLite 3.1 release
- **Finally the need to stop the development of new functionality was recognized, in order to devote some time to the code base cleanup**
- **Activity in principle pursued with high priority, but the daily duties still take a lot of time**
  - support, bug fixing, important improvements decided by the TCG, adoption of ETICS, porting to SLC4, porting to 64bit archs, ...
- **The result will be a **simpler** system**

- **The job management system has finally reached an acceptable state**
  - It is usable
  - Feature-complete (more or less)
- **The system has not yet reached a stable state**
  - Still a lot of development needed to address non-functional aspects of the system
  - Need to address some architectural shortcomings
    - e.g. Policies and Information System
- **Organizational changes will be reflected in EGEE-III**
  - Closer collaboration between development and testing&certification
  - The responsibility of a component assigned to a single partner