



WMS baseline issues in Atlas

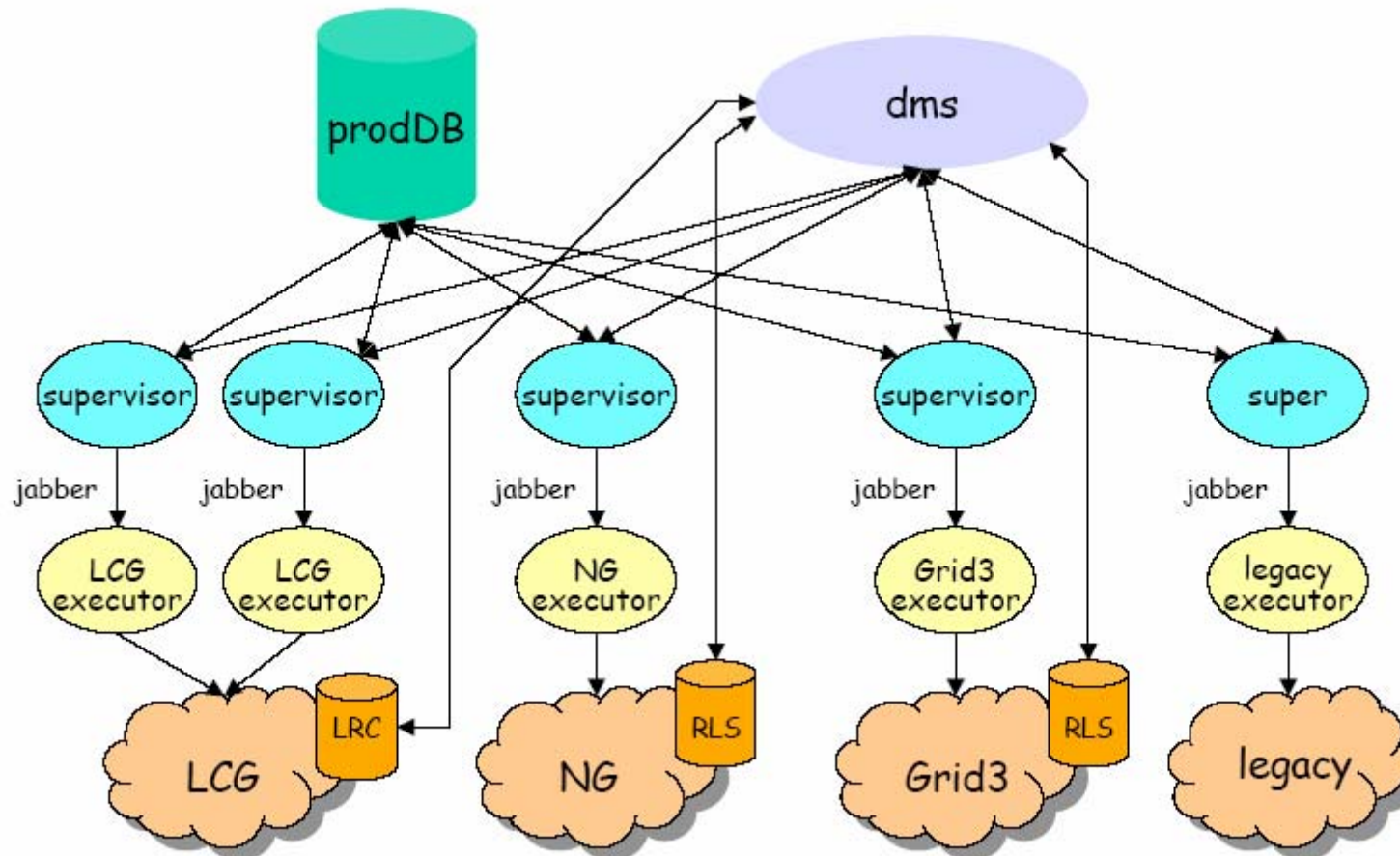
Miguel Branco
Alessandro De Salvo
15-4-2005

Outline

- **The Atlas Production System**
- **WMS baseline issues in Atlas**



The Atlas Production System (1)





Atlas Production System (2)

- **4 major components**
 - **Common Central Database (ProdDB)**
 - **Common Data Management System (DQ)**
 - **Production Supervisor (Windmill)**
 - **Executor**
 - **One per grid-flavour**
 - **GRID3 → Capone**
 - **LCG → LEXOR**
 - **NORDUGRID → Dulcinea**
 - **One for legacy systems**
 - **Bequest**



Atlas Production System components (1)

- **Production Database (ProdDB)**
 - The ProdDB holds records for
 - job transformations
 - a transformation is an “executable” that transforms data from one state to another one, using a set of pre-defined actions, to be executed in sequence
 - job definitions
 - status of jobs
 - job executions
 - logical files
 - The bookkeeping database (AMI) is currently sincronized with ProdDB
 - Oracle database hosted at CERN

- **Data Management System**
 - allows global cataloguing of files
 - interfaced to existing replica catalog flavors
 - allows global file movement
 - See previous talks from Miguel about the Data Management



Atlas Production System components (2)

■ Supervisor

- consumes jobs from the production database
- submits them to one of the executors it is connected with
- follows up on the job
- validates presence of expected outputs
- takes care of final registration of output products in case of success
- possibly takes care of clean-up in case of failure
- will retry n times if necessary
- uses Jabber to communicate with executors

■ Executor

- one for each facility flavor
 - LCG (Lexor), NG (Dulcinea), GRID3 (Capone), PBS, LSF, BQS, Condor?, ...
- translates facility neutral job definition into facility specific language
- implements facility neutral interface



WMS problems seen in recent productions

■ Information System

■ Ranking

- some sites do not publish correctly all the informations in the IS
- the ranking expression based on "EstimatedResponseTime" had to be changed
 - *if there are no queued jobs, rank on the percentage of free CPU on the number of running jobs, else use the negative value of percentage of waiting jobs on the number of running jobs*
- The number of free CPUs for Atlas only is not available
- The value of Max running jobs (which is VO dependent) is not always set to a reasonable value

■ Sites that don't pass the daily test suite run by the CIC on duty were automatically excluded from the ATLAS BDII

- May exclude good sites for Atlas
 - Not all the tests are relevant to decide if a site is good for ATLAS or not
 - Some test could fail for a temporary problem, but currently there is no retry procedure implemented
 - Often excluded sites are not efficiently informed about the failure
- May excluded also Storage Resources from a site, also when only the Computing Resources are to be excluded

■ RB/Job submission

- Only single jobs may be submitted and a single RB cannot serve efficiently more than 2000-3000 jobs
 - Submission time reaching peaks of > 40 secs, while normal values are around 10 secs.
- Submission speed affected also by the load on the L&B server, in particular when querying job by job
 - A global query is impossible via CLI



WMS baseline requirements for Atlas (1)

- **Bulk submission**
 - **Submission of collections of jobs**
 - Sandbox sharing
 - Parametric jobs
- **Bulk status**
 - Status retrieval of collections of jobs
- **User tags in jobs**
 - Associating a tag to a job, in order to label it with its Production System name
 - Middleware jobID could be lost, if a crash occurs
- **Real-time job output inspection**
 - Ability to retrieve the partial output of a job



WMS baseline requirements for Atlas (2)

- **Data-driven matchmaking**
 - **The replica location informations should be used in the ranking expression**
 - For example by having a function that returns “1” if the file, passed as parameter, is available in the close SE)
 - **Provide metrics for distance CE-SE, to be used in matchmaking**
 - At least something like “local”, “close”, “same continent”, “far away”

- **Task queue in RB**
 - **Keeping an historical view of the submitted jobs**
 - Better optimization
 - Avoid the “black hole” effect, where a misconfigured site pulls many jobs and makes them failing
 - **Dynamical rescheduling when the state of the grid changes**

- **Better handling of sandboxes**
 - **RB should not crash if the space used by the sandboxes is too much**
 - **Input sandbox sharing**

- **Better error reporting**
 - **Avoid generic error messages (like “Max Retry Count exceeded”)**