Critical (Missing) Middleware for LHC experiments

Dario Barberis

(CERN & Genoa University)

Middleware: what is missing?

- We are ~1 year from the beginning of data-taking
- Time to assess which critical components are still not available for distributed operations
  - Last chance to get anything new tested and used by the experiments
  - Also last chance for us to improve our systems and tune them before data-taking starts
- All experiments developed their own systems around the existing middleware, therefore it is <u>not</u> surprising that there is no new major development request (see later slides)
  - But we ask that a lot of effort be put into optimization and robustification of the existing middleware (code and services)

"Critically missing"?

- The criticality of each item depends on each experiment framework
  - As many developments became available much later than expected, in the meantime experiments developed their own solutions
  - Some of these solutions have reached a high level of maturity and experiments now rely on them
- It is evident that so far we have all been able to run scheduled productions on the Grid(s)
  - It is much less evident that in the current situation we would be able to support 1000s of analysis users (all experiments together) in addition to scheduled productions
- So we all have to work to improve:
  - Support for intra-VO allocations, priorities, monitoring, accounting
  - Stability, robustness and performance of existing tools
  - … on all Grid infrastructures we have to use!

Granularity within the VO

- ALICE and LHCb have developed a single VO task queue with job prioritization and optimization handling capabilities
- For ATLAS and CMS it is not practical to force everyone in the Collaboration to submit Grid jobs through the same central system
  - They are instead populating the VOMS database with groups and roles in order to have the possibility to implement intra-VO job fair share, storage quotas, accounting
- There is NOW no consistently implemented set of tools in deployed middleware that:
  - Defines job priorities according to the group/role of the submitter
  - Sends jobs where they have the highest probability to run faster (depending on their input data and local priorities/shares)
  - Stores the output files in the SE where the submitter (or his/her group) has an assigned quota
  - Transfers files or datasets with priorities that depend on the user group/role
  - Produces group-level monitoring and accounting of the user resources (CPU, storage, bandwidth)

Data Management (1)

- Everybody needs SRM 2.2 with a consistent implementation by all storage managers (Castor, DPM, dCache)
  - As agreed in the Storage Working Group meetings
  - Work is in progress, but there is no deployment yet
  - It may take some time before having efficient products
- More robust and performant FTS
  - Notification service (e.g. Jabber based) to avoid constant polling to find out FTS transfer status
  - Delegation service is important (coming with next release)
    - Avoid having to specify the myProxy password for FTS to retrieve a certificate.
      - When the certificate is uploaded to the myproxy-fts it should be possible to specify who is allowed to retrieve it, to avoid passwords
- Functional and complete Data Management client tools, lcg-utils
  - More functionality:
    - Look up physical file existence and properties
    - SURL to SURL copy
    - File removal with the same semantics for all the SE implementations; bulk file removal

Data Management (2)

- ATLAS needs absolutely a much faster and more robust LFC (>20 minutes to have back a list of 1000 files belonging to a given dataset is much longer than people are prepared to wait)
  - Bulk operations
  - Unsecure read access if needed for performance
  - File ownership assigned in the same way in the catalogue as in the SE
    - Not all replicas owned by the original production manager!
  - Automatic tools to check consistency between LFC, SRM, SE
- Robustness in the GFAL library (ATLAS/LHCb/CMS)
  - Better definition of "closest" SE
  - Working ROOT plug-in
  - Support for all access protocols
    - Rfio, rootd/castor, dcap, gsidcap
  - Separate release cycle for client libraries and binaries
- Alice would like to have the inclusion of xrootd in the SE with support for their authorisation plugin

Job Priorities

- Discussions on this topic are still heated… and there is no "obvious" conclusion in sight
  - Tests being done in the context of the EGEE Job Priorities Working Group are a good start, but far too late and too restrictive
    - We do not see how the system under test can ever be extended to support ~25 groups and ~5 roles within each VO
      - 3 queues and 2 priorities, even if deployed on each site, are far from the needed granularity
    - What we would like to have is something closer to a distributed fair share system
  - The EGEE development G-Pbox has been tested so far only on small scales by ATLAS and CMS since the beginning of 2006
    - But it has not yet been scheduled for certification
    - And we have not seen a reasonably large scale test yet (a few sites, many intra-VO groups/roles)
    - If/when it is deployed, it would be yet another service to support in each site!
  - US-ATLAS have their own central task queue (PanDA) for jobs that run on OSG
    - Not clear if it will scale to several hundred analysis users in addition to scheduled productions
- So far this problem has not become critical only because there are not that many Grid users, and the majority of resources are used by scheduled productions
  - But as soon as we really advertise Grid usage for everyone, people will fight for CPU by flooding the system with their jobs
    - And we have no handle to set relative priorities for activity groups and individuals

Information System & Job Management

- Full deployment of the improved information system
  - Enabling the the usage of VOView information in the job distribution
- ATLAS and CMS need a highly reliable gLite WMS, with high throughput and high availability
  - 50k jobs/day by end 2006 for each of ATLAS and CMS
  - 200k jobs/day by 2008 for each of ATLAS and CMS
- LHCb and Alice need the completion of the gLexec development and its deployment to support their job distribution model
  - If/when it shows to be performant, it could be adopted by others
  - Continue discussions with developers, security group and sites on
    - proxy delegation
    - users control
    - job traceability
  - This development was asked for by the sites to improve security and the traceability of job ownership

Other Components and Services

- GGUS responsiveness and efficiency needs to be much improved
- VO Box discussion has to come to an agreed conclusion on service levels
- Monitoring and accounting needs a quality step
  - Group and user level accounting must be made available to the VO management (in real time)
  - The ARDA dashboard is a useful tool but every information provider should make sure the inputs are correct and consistent
- Site service monitoring tools also need to be implemented and deployed consistently

Conclusions

- Time is an important factor, because data taking is getting closer
- The really critical points are:
  - FTS and completion of storage developments
  - Support for intra-VO allocations, priorities, monitoring, accounting
  - Stability, robustness and performance of existing tools