



User Analysis Workgroup Discussion

- Understand and document analysis models
 - Best in a way that allows to compare them easily
 - Answer the question how unpredictable Chaotic User Analysis will be
- Identify additional requirements coming from the analysis use case
 - Middleware
 - T1/T2 infrastructure
 - Load to be expected
- Total input received:
 - Several summary write ups and comments to questions and summaries
 - Roughly 100 reference documents and presentations



High Level View

- All experiments have well established analysis frameworks
 - They are documented and in use by end users
 - Users are shielded from most of the middleware
 - They look similar, but are different when it comes to resource access
 - The frameworks are currently in a “tuning” phase
 - Detailed performance measurements
I/O vs CPU, failure rates, network bandwidth
 - Stress tests
Automatic submission frameworks

- Communication channels between experiments and T2s are well organized
 - Different concepts, but communication works
 - Specific site stability monitoring for analysis use case
 - Some experiments have well defined resource requests
 - Storage per WG etc.



High Level View II

- T2s/T3s
 - Can be of almost any size
 - Several have less than 100 cores, one 6000
 - Storage:
 - „ZOO“ describes it best
 - Different storage systems with different characteristics
 - Same SE type does NOT guarantee same performance
 - Pool clustering, network, file systems, configuration
 - Experiments take an adaptive approach
 - ATLAS currently builds a „Storage Access Method Map“
 - Allows to pick best performing acces on every site
 - Without adaptation the CPU/Walltime varies 10-90%



High Level View II

- User activity
 - Large experiments each have > 1000 users who used the grid during 2008
 - Users per week on the order of 100-200/experiment
 - Expected increase during summer: Factor 2-3
 - Current resource usage: 15 – 30 % of the production use (30k/day)

- Computing resource access
 - Via submission frameworks
 - WMS, direct CE, pilot
 - Atlas uses both strategies
 - CMS investigates pilot option
 - LHCb and ALICE are pilot based



Main Issues and Requirements

- Reliability of sites
 - Important, but not part of the work group
- Access to computing resources
 - Some minor problems with scalability of WMS and lcgCEs
 - Can be addressed by adding more services
 - Must be planned now, overall job rate will double soon
 - Multi user pilot job support
 - Glxec/ gums in OSG (available)
 - glxec/SCAS (very, very soon available)
 - NDGF ???
 - Access control to shares and fair share balances
 - Fine grain control respecting groups and user locality is needed
 - Technical basis is available, but sites have to configure it
 - Share allocation monitoring needed (CMS)
 - End user job monitoring (Experiments develop specific dashboards)



Main Issues and Requirements

- Access to storage resources
 - Three main problem domains
 - I/O performance (including network)
 - Access Control
 - SRM commands
- Access Control
 - ACLs to control staging and data (re)moval
 - VOMS based ACL are in dCache, DPM, STORM
 - Why can't they be used? What is missing?
 - Quotas: Group based and user based are needed
 - Experiments monitor usage, but this will not scale
 - „Free Space for X“ tool wanted (LHCb)
 - To know how much space is available
 - Accounting



Main Issues and Requirements

- Storage I/O
 - Xrootd everywhere (Alice)
 - Milage varies based on backend
 - DPM uses 2 years old version
 - CASTOR moves to a major upgrade
 - dCache implemented subset of xrootd protocol
 - Xrootd native at most ALICE T2s
 - Access via “Posix” like clients: rfio, dcap, WN local (staged)
 - Very sensitive to file structure (number of ROOT trees)
 - Even local access varies from 2.2 MB/sec to 47 MB/sec
 - Very site dependent
 - ATLAS and ALICE performed specific measurements
 - CMS will do the same very soon
 - Network: Example ATLAS “Muon style analysis in HammerCloud)
 - 200CPUs -> 800MB/sec I/O -> 10 Gbit network



Main Issues and Requirements

- SRM commands
 - SRM client calls should not block SRM services
 - srmLs polling is a good example
 - Bulk operations for some commands
 - Which?



What next?

- Documentation of analysis models seems to be not an urgent task
- Detailed resource allocation at T2s are best coordinated by experiments directly
- In my view we should concentrate on the following issues:
 - I/O for analysis
 - Understanding the different access methods
 - How can we help the T2s to optimize their systems?
 - Can the ALICE analysis train model help?
 - Role of xrootd as an access protocol?
 - Clarification of requirements concerning storage ACLs and quotas
 - What can be done now
Protect staging, handle ownership, etc.
 - What is really missing?
Most SRM implementations have VOMS ACLs.
 - Short and longterm solutions for quotas
 - SRM calls



What next?

- CPU resource access
 - Specify for each experiment a share layout, including pilot queues
 - Pool middleware and batch system expertise to provide good templates for the most common batch systems
 - Share monitoring?
 - The infrastructure needs to understand clearly how many jobs will flow through the system
 - Analysis increases at least by factor 3 during the year
 - How many WMS are needed?
 - How many CEs should a T2 site with X cores provide?
Number of FQANs and jobs are the issue
- What else?



Discussion