Data Management TEG

not quite a summary yet...

Dirk Duellmann & Brian Bockelman
(with slides from Andy Hanushevsky and Markus Schulz)
Pre-GDB, 7. Feb 2012

Initial Discussions

- Often more “status quo / known defect” discussions than a strategy one so far
  - several long standing problems resurfaced with repeated conversation threads and frustration
    - Eg dark data, SE vs grid quota, archive security, security model
  - we need to document failure of previous goals to move on
    - experiments did manage to find workarounds to missing functionality
    - s/w projects did loose and will continue to loose available effort
- often we know better in which direction to leave than where we want to arrive..
  - lack of documented strategic direction increases risk of circular movements
  - large groups/committees make it very hard to maintain focussed strategy

A personal observation

- Most of the strategy proposals originate from s/w providers
  - but don't reach mass shell until one or more experiments start to deploy them
  - we are not really good at making this endorsement or even the plan towards that explicit
- Most of the concerns are in the area of sustainability within the shrinking effort in the WLCG community
  - do all experiments buy into the need to consolidate and optimise?
  - if yes, it should be easier to give up on divergent approaches or older strategies, which did not prove to be feasible
- Experiment involvement in defining and maintaining a common strategy will be key to further optimise the DM system we will have

POOL – from A. Valassi

- LHCb stopped using POOL (will collect statement for report)

- "ATLAS will continue to need support for POOL, including any relevant software patches and releases, for as long as the 2012 production version of the ATLAS software is actively used. It is based on the LCG61 configuration, which includes a version of POOL centrally built and maintained by the core team in IT-ES (using ROOT 5.30).

- ATLAS will no longer need support for POOL from IT-ES for their releases based on the LCG62 configuration (using ROOT 5.32), where a custom software package derived from POOL is built and maintained by ATLAS as part of their internal software. The first such release already exists and will be used as an ATLAS development release in 2012; this will eventually become the production version of the ATLAS software, by end 2012-beg 2013."

The {missing} Diagram..

- Data Management TEG operated for most of the time jointly with Storage TEG
  - while this was a good choice in our current situation, it may already point to a key problem
  - we can not reason about/evolve both areas independently
- There was (and still is) no upfront model defining the relation between data and storage management components and/or stake holder responsibilities as part of a layered system
  - with a bit of additional work a first straw man could/should be derived from the status quo discussion for the report

Placement Responsibilities

- Experiment – Data Management
  - Define geographical data (sets) distribution within available resources according to current priorities

- Site – Storage Management
  - Maintain stable storage for placed data
    - ..but Experiments regularly take over repair tasks
  - Support access from experiment jobs
    - Bandwidth, availability and access latency(eg tape) have directly impact on cpu/wall ratio

Eg Dark & Grey Data

- Mismatch between SE disk and SE catalogue - dark data
  - detection is site responsibility
  - repair / avoidance with help from SE s/w provider
- Mismatch between SE and experiment catalogue - grey data
  - detection is experiment responsibility
    - why? because the site has no reasonable way to determine a mismatch
  - repair / avoidance with help from SE s/w provider and experiment

Data Placement & Federation

- Direct placement (push via FTS / xroot) continues to be main data management tool of experiments
  - pre-placement is ideal wrt to latency and has significantly increased in efficiency using recent data popularity services
- Now complemented with federated access via xroot to reduce impact of (temporarily or permanently) unavailable data. Relevance increasing towards smaller sites.
  - Main source of information:
    - http://indico.in2p3.fr/conferenceTimeTable.py?confId=5527#20111121
- Large global federations have been established and successfully tested over the last year
  - Other federation mechanism are possible (eg http or nfs based) but have not reached similar level of maturity and experiment use yet.
- Experiments are increasing their efforts to optimise the existing data management infrastructure
  - site service problems are increasingly being absorbed when necessary
  - in other words - data management solutions are used to overcome storage management problems
  - service level assumptions are shrinking in view of reality

Federation in WLCG

- Most experiments are already federating data
  - Alice already doing production wide-scale federated storage
  - US CMS has pre-production federations
  - US Atlas has test federations
- All use xrootd as the federation infrastructure
  - Storage infrastructure is mixed bag
    - dCache, DPM, GPFS, HDFS, and native xrootd

Federation - Discussion

- The question was whether xrootd is sustainable
  - Specifically, would a broader based protocol be longer lived
- Technically, only two other viable open protocols
  - HTTP and NFS v4
- Broad agreement that NFS v4 not yet suitable
  - Still immature
  - Does not support x.509
  - Not designed for creating robust WAN federations
    - Though not really tested whether current elements are sufficient

Namespaces and Catalogs

- Each LHC experiment must maintain a namespace - i.e., must have a unique name for each file or dataset they produce.
  - The unique name may be a path, a filename (no hierarchy), or a GUID.
- The software that maintains the dataset and file namespace is unique to each VO.
  - There exists a mapping from dataset to a set of files is via some catalog. This software is also not shared across VOs. Different conceptual models exist; example: ATLAS heavily uses the fact that a file may be in multiple datasets.
  - No commonalities currently exist, and creating commonalities at this layer appears to be prohibitively difficult.

Catalogs

- All LHC VOs also maintain a catalog mapping the VO's file namespace (LFN) to a set of locations.
  - The mapping from LFN to PFN for a given location may-or-may-not be a separate step, depending on the VO.
  - The LFN->PFN translation is often a deterministic function instead of a DB lookup. The exception is ATLAS, and they plan on changing this.
- For 2 of 4 VOs (ATLAS and LHCb), this catalog is the LFC software from EMI.
  - Both VOs have stated they will be moving away from LFC in the next two-three years.
  - **Note**: ATLAS is still designing their next-generation data management system, so this could change!
  - **Note**: Implies a loss of commonality.

Recommendation

- The WLCG has no long-term need for LFC maintenance as a catalog for LHC experiments.
  - There is a small short-term need as ATLAS and LHCb transition.
  - **Alternate**: If ATLAS decides to keep the LFC, it could be treated as experiment-specific software
- We recommend EMI re-evaluate its work items with this in mind. Any project depending on the VO using the LFC is unlikely to gain broad adoption.
  - Projects may still depend on using the LFC, of course. The critical phrase in this recommendation is "*the VO using the LFC*".

WAN Protocols

- Expected functionality:
  - support for third party copy
  - support for data integrity (checksum as part of the transfer)
  - efficiency
    - partial file transfer/access
    - parallel streams
  - most important: stable and dependable client and server implementation
    - preference for established standards based clients
      - deployment, longterm cost, quality
  - WAN protocol should be compatible with Storage Federation
- List of candidates:
  - gridFTP, Xrootd, http(s)/WebDav, NFS-4.1/pNFS, S3
  - none of the candidates is perfect
    - need follow-up on remaining issues

Well Accepted Protocols

- gridFTP perceived issues
  - not a true standard based component:
    - needs to be maintained by the community
      - different error messages
      - authentication and authorization model
      - AA overhead limits efficiency for small files

      with move to openSSL and sessions this will improve
- xrootd perceived issues
  - HEP community specific, long term support needed
    - unlikely to profit from external contributions
  - functionality/quality/security of integration with storage systems varies
    - dCache, DPM, Castor, EOS, xrootd native
    - work plan to fix several of the issues for DPM
  - 3rd party copy
    - foreseen in the system and used in EOS, but not generally

Promising Protocols

- HTTP(s)/WebDav perceived issues
  - 3rd party copy
    - proof of principle solution in DPM
  - data integrity
    - md5 already supported
    - other checksums can be supported by appropriate http error codes
- NFS-4.1 / pNFS perceived issues
  - mechanism for federation and WAN access defined in the protocol
  - not clear if different (commercial) implementations will interoperate
- S3 / Clouds perceived issues
  - only basic functionality
  - non-posix, still unclear how it could be integrated
  - more experience needed

Conclusion Protocols

- gridFTP and xrootd are the current core protocols
  - working at scale
  - reference for the foreseeable future (including federations)
  - having some known issues
- http(s)
  - very promising
  - excellent support outside the community
  - future-proof, knowledge widely available
  - should be seriously considered in the medium term
- NFS-4.1, S3
  - to be watched closely, with S3 being likely to see growth

- Miron's recommendation:
  - We should be more worried about being able to easily change a protocol. We should identify which layers of our stack are making this task difficult!

FTS

- Not much discussed at the TEG F2F in Amsterdam
  - not much controversy
- Some additional use cases mentioned throughout the sessions

- FTS main use cases:
- As a file transfer batch system two use cases need to be supported
  - shallow queue with experiment framework support
  - deep queue with high level of autonomy
- Moving files at the same site between different storage classes
  - for those systems that use different endpoints (Castor/EOS)
  - SRM being used, but SRM less preferred

FTS-3 Input

- Support for additional protocols:
  - gridFTP (no SRM), xrootd, (should be flexible)
  - different Authentication and Authorization mechanisms
- If transfers fail due to source problems, replicas should be considered
- Individual channel configuration will not work for new storage hierarchy
  - FTS will replace channels by (dynamic) end-point pairs
- Miron: scheduling should rely on existing concepts and tools
- has to be a framework that can accept new protocols and policies
  - to adapt to the many changes in storage and DM that are coming