Storage TEG “emerging” observations and recommendations

Wahid Bhimji

With contributions from the SM editors (listed in intro)

Responsible TEG (To aid organisation here	TOPICS: As grouped at F2F See Twiki for details	Editor. Experiment co-editor TBC
Data Management	Data placement (DM2) and Federation (DM3)	Andrew Hanushevsky + Dirk
	WAN Protocols (DM4) and FTS (DM5)	Markus Schulz
	Catalogues (DM9) and Namespaces (DM10)	[Brian Bockelman]
Storage Management	Security and Access Control (DM6/SM6)	Maarten Litmaath
	Separation of Disk and Tape (SM3)	Andrew Lahiff
	Storage Interfaces (SM4): SRM and Clouds	Paul Millar
	Management and operation of storage at sites (SM7)	Andreases Heiss and Petzold
	Storage I/O (SM1) , LAN Protocols (SM5) and Evolution of Storage (SM2)	Giacinto Donvito and Wahid

Data & Storage Management Security matters

To be continued in the Security TEG

Maarten Litmaath

Status quo

- SE + catalog configurations
  - Protect production data from users
  - Some experiments prevent tape access by users
  - User and group access regulated by expt frameworks
    - Including quotas
    - SE may be more permissive than desired
      - To be checked and fixed as needed
- X509 overhead
  - Use bulk methods, sessions, trusted hosts as needed
  - Cheap short-lived tokens may become desirable

Data protection

- Do different data classes need the same security model?
  - Custodial
  - Cached
  - User
- Access audit trail important for traceability
  - Security and performance investigations
- Protection needed against:
  - Information leakage (“Higgs-discovery.root”)
  - Accidental commands
  - Malicious outsider, insider

Issues with data ownership

- Missing concept: data owned by the whole VO or by a service
  - Use robot certificates for that?
- Mapping person ←→ credential
  - Changes → consequences for data ownership
    - Certificate might indicate “formerly known as”?
    - Make use of VOMS nicknames or generic attributes?
  - X509 vs. Kerberos access
- VO superuser concept desirable?
  - Avoid bothering SE admin for cleanups

More items

- CASTOR: RFIO/NS backdoors to be closed
- Not only data, but also SE itself needs protection
  - Against illegal data, DoS
- Storage quotas
  - On SE: conflict with replicas
  - Better handled by experiment framework
  - Can still be useful to SE admin
  - Low priority, available for some SE types
- Quotas on other resources e.g. bandwidth?
  - Prevent DoS

Separation of archives and caches

Andrew Lahiff

Current situation

- Two classes of workflows at the Tier-1 sites common to the experiments give the requirements:
  - **READ**
    - Keep defined data pinned on disk for reprocessing and redistribution
    - Ability to allow user analysis without negatively impacting tape system
  - **WRITE**
    - Ability to process data without writing immediately to archive
    - User analysis should not write to the archive
- **All LHC experiments seem to be working fine (or towards) splitting disk caches from tape archives**
  - ALICE, ATLAS, LHCb: split
  - CMS: work plan in progress
- Managing data movement between caches and archives
  - FTS controlled by experiment data distribution software (ATLAS, LHCb)

Discussion from face-to-face

- Accessing data on the tape archive
  - Some experiments want to directly read from the disk buffer in front of the tape system, e.g. for reprocessing
  - Alternative view:
    - pre-staging /pinning = copy from T1D0 to T0D1
- Internal Tier-1 data movement vs transfers between sites
  - Experiments prefer the idea of a single system (e.g. FTS) to manage both transfers internal to the Tier-1 as well as transfers with other sites
    - Interaction between disk and tape within a Tier-1 should not be considered differently from any other data transfer
    - Data resident at a Tier-2 or on a disk cache at a Tier-1 can therefore be archived in exactly the same way
  - FTS using 3rd party copy functions is like triggering the SE to do something
    - Change the directory/storage class rather than copy a file

Discussion from face-to-face

- Managing data movement between caches and archives
  - FTS seems to be the only tool available for scheduling and managing data placement
    - We can consider FTS as a system for moving data between caches and archives
  - Are there any other concepts or architecture that would fit the problem better?
    - **FTS is working well at the moment**

Storage operations and management at sites

Andreas Petzold, Vladimir Sapunenko, Andreas Heiss

SEs and storage access protocols

- Need common, agreed protocols which are fully and correctly implemented in SEs
- Sites choose type of SE based on requirements and their own environment and expertise

Monitoring of data access patterns

- Shall be done on the application or catalogue level
- Experiments shall provide this information to sites in a some standardized, machine readable form.
- Information can be used by site to optimize the storage system layout.

Single point of failure (SPOF) in some D1T0 implementations requires many efforts (e.g. on-call service also at night) to operate, if non-scratch data is stored.

- Sites shall minimize the failure probability by using 'smart' techniques like
  - dual-tailed disks
  - distribute raid over multiple servers (example: RAID5 striped over 5(4+1) servers)
  - Disks separated from servers, high quality hardware etc.
  - Non-scratch datasets should be duplicated at another site
  - Applications level
    - access files at other sites if all or some files of a dataset are locally unavailable due to a SE failure. → Storage federations

Dark data – Consistency between catalogues and SE contents

- Consistency checks between catalogues and SE contents shall be done regularly by the experiments. SE metadata shall be provided by sites.
- Data on SE disks which does not appear in the SEs metadata database can only be found and removed by sites

Handling of data losses

- Site should inform the affected experiment(s) immediately and provide a list of lost files
- Site shall estimate the possibilities and efforts necessary to recover locally
- Experiment shall estimate effort for retransferring or reproducing data.
- Site and experiment should agree on the recovery procedure taking into account the estimated necessary time an possible costs.

Management of near-line and online storage (not discussed at Amsterdam F2F!)

- (In the long-term) local data management could be done by the sites, based on experiment requirements, e.g ”*we need access to data set A with latency not more than Y seconds and overall bandwidth of X MB/s for N days”*

Storage accounting

- Favoured solution/protocol is EMI StAR **(given that some outstanding issues are solved.)**
- **See http://cdsweb.cern.ch/record/1352472?ln=en**
- The release time scale is ok

Storage I/O, LAN Protocols and Requirements and evolution of storage

Giacinto Donvito

State of Play

- Magnetic disks are becoming bigger, but the performance is not increasing accordingly
  - This will highlight a problem in number of IOPS (per TB) available to the applications though different systems may have other bottlenecks
- In order to build the storage infrastructure it is important to take into account the “Total Cost of Ownership”
  - Not only hardware but man power needed to maintain and to operate it
- The experiments use every protocol supported by ROOT
  - But this is achieved by means of a deep knowledge of the system and several “tweaks” in the experiment framework
- The experiments see the storage services as poorly resilient and needing more detailed error handling

Discussion from face-to-face and some recommendations

- We need to find a plan to mitigate the performance problem:
  - Both at farm level and at the application level:
    - The computing centres could be optimized using new storage techniques
    - The application should be optimized in order to reduce the number of IOPS
    - Technologies such as SSDs should continue to be investigated in order to understand "how and if" they can help in improving the performance
- We need a benchmark that can "emulate" the analysis application
  - This will help in testing storage infrastructures without installing the experiment software
    - Could be generic but tuneable to specific cases.
    - Many things already exist but room for developing / publicising.
    - Could be a task for the ROOT I/O or other existing group…
  - We need a clear definition of the bandwidth, IOPS and latency required for experiment analysis workloads now and in future
    - This will be useful to configure the WN with the needed network bandwidth and the build the LAN infrastructure (e.g. 10Gbit/s WN networks)

Discussion and emerging recommendations

- LHC experiments are able to work with the range of current local protocols and that can continue:
  - Though in the future it looks likely that all storage providers will offer at least one of xrootd and file:// (e.g. nfs4.1 adoption)
  - Not essential but very welcome to simplify interaction.
  - File:// also helps users to interact with files interactively.
- Nobody likes "single point of failures",
  - But trying to get rid of those usually requires an increasing complexity of the software
- The storage service should aim to be more robust
  - "self healing" technologies are welcome
  - But also putting more intelligence at the application/library level is the easiest way to improve the fault tolerance
  - Need much more clear error handling and reporting – should aim to get more specific as to what that should be.

The end

Extras….

Separation of responsibilities (proposal, not discussed at the F2F in detail)

- Sites:
  - architectural and infrastructural solutions;
  - design and deploy storage solution based on exp requirements and site expertise;
  - define operational and support modes and models (24/7, best efforts, etc.);
- define policy for data placement and migration between on-line and near-line storage considering experiments' desire/requests for latency in data access;
  - populate and update data in the site catalog;
  - purge "dark data"

- Experiments:
  - consider Storage As A Service;
  - provide requirements on
    - capacity;
    - bandwidth;
    - high level protocols;
    - efficiency;
  - concept to use:
    - on-line storage (acceptable latency less then XX s)
    - near-line storage (acceptable latency less then YY s)

Proposal from Andreases