Jamboree on Evolution of WLCG Storage and Data Management

(reader’s digest)

Tanya Levshina

Discussion Topics

- General requirement to storage
- SRM protocol
- Data Management and global namespace for data access
- File Catalogs
- Not covered here:
  - Network
  - Tape storage – true archive, optimization of tape management

All pre-jamboree docs and presentations are at
http://indico.cern.ch/conferenceDisplay.py?ovw=True&confId=92416

General requirements for storage

- Higher reliability and redundancy (CMS, J. Bakken):
  - No ACL or space token (VOs separation via hardware)
  - FTS fault tolerance and error propagation
- The main missing feature in accessing storage systems is a proper authorization framework. Needed in all storage systems: grained access control, storage quotas. (CERN, C. Grandi)
- Reliability, transparency for clients, usability by various applications (CMS, B. Bockelman)
- Now the WMS is storage-blind. A balanced load of compute and IO bound jobs makes for a more efficient computing environment (KIT, Van Vezel and many others)
- Cluster or parallel storage systems (e.g NFS) that are supported by open-source communities should be studied (KIT, Van Vezel and many others)
- Implement data aggregation (data sets, containers, directories, volumes, pools, …) and offer management interfaces allowing simple actions on the aggregated data (bring on line, move offline, replicate to another pool, manage the number of replicas, access control, quota, etc.) (CERN, A.Pace)
- Reduced number of protocols, clients (many speakers)

SRM Protocol

- Improve efficiency of SRM protocol– e.g srmPrepareToGet() should return TURL if file is there *(*GridPP Storage Group ,J. Jensen)
- Review the SRM specification to be less generic and more specific including identifying one or two file access protocols that all storage implementation must support. (CERN, A.Pace)
- File access protocols that all storage implementation must support could be XROOT for the short term (within one or two years) and NFSv4.1 in the longer term (5 years). (CERN,A.Pace)
- SRM protocol and its implementations can be improved (CERN, M. Schulz)
  - Further reduction of scope
  - Investment in monitoring and performance analysis
  - Improving error handling
  - Some of the scalability issues can be addressed
  - Handling non core functionality outside SRM
  - Storage – Catalogue synchronization
- Extend SRM protocol to handle metadata. (INFN23, G. Donvito)

Data Management and Global Namespace

- The interface layer between the application and the local storage should hide specifics of the underlying storage and data locality. (CMS, Ian Fisk)
- The remote access should allow applications to operate efficiency with incomplete datasets on the local site. The remaining files could be streamed over the WAN or downloaded dynamically. Peer-to-peer data serving and intelligent data placement should be investigated.(CMS, Ian Fisk)
- Wide-area protocol based on HTTP(S) and DNS, allowing sessions. Standard UNIX/POSIX Clients. One should be able to mount together several storage namespaces (possibly NFS4.1) to offer a global namespace and some centralized administrative tools for data organization (ATLAS, V. Garonne)
- GridData-FUSE driver/Global Grid Data Index (NIKEF, O. Koeroo)
- Hierarchy of Proxy Caches on the way between client application and SEs. These caches would accept the protocols used for user file access and maintain a local disk (and memory) area with copies of recently accessed data. will operate on read-only files with stable file names.(CERN, D. Duelmann)
- Requirement for global access (CERN, A. Pace):
  - Expose all the storage via mounts (POSIX) with strong authentication
  - Expose all data management via key-value pairs stored as XATTR
  - Worldwide P2P access with caching or persistent local storage
  - Asynchronous cache write-back to persistent storages for job output
  - NSFv4.1/FUSE/xroot
- Xrootd on top of various SEs (CMS, B. Bockelman)
  - xrootd interface HDFS, dcache (in development), CMS Trivial File Catalog

Data Management and Global Namespace

- NFSv4.1 an excellent core protocol (M.Crawford):
  - possible for making remote and local file accesses appear uniform
  - site-to-site transfers, a namespace that integrates multiple sites' namespaces directly yields a BitTorrent-like multisource transfer through pNFS
  - Full-file prefetching and write-through caching can be implemented
- PanDA distributes the data as needed, using a predictive brokering algorithm. A request to unavailable dataset will trigger replication to an optimal location chosen by PanDA. User jobs will continue to be brokered to Tier 1 untill dataset becomes available. If the number of requests to the same dataset is too high, replication starts. Important principle of data analysis: jobs go to data. (ATLAS, Panda group)
- ARC is a production middleware developed and maintained by members of the NorduGrid collaboration. The data management model in ARC differs significantly from the gLite model - all data transfer is performed on the CE front-end on each site before and after the job runs on the worker node. An essential feature of ARC is its ability to cache input data files on the front-end. Deficiency:
  - As knowledge of the job input is required before job submission, pilot job systems do not match well with ARC.
  - There are also no quotas in the cache, although separate caches on separate file systems can be set up for use by different users or VOs or groups within VOs

  (Alice, A. Cameron Smith)

Data Management and Global Namespace

- CHIRP allows a local disk area to be accessed securely from distributed hosts, without the need for root/kernel level modifications on those hosts; one can think of it as a user-level afs. A single server host with a lot of disk. The security is based on the X509 proxy with full acl's. Lack of true scalability and redundancy. (Rod Walker, LMU Munich)
- There are potential problems with using P2P for other data operations. Should use data prestaging in background for anticipated jobs (not suitable for pilots) (GridPP Storage Group , J. Jensen)
- The use of p2p technologies is not helpful for file movement (e.g. harvesting of job output, movement of private data, etc...). The data transfer infrastructure should be used, transferring requests asynchronously. FTS use should be extended to the data movement with the development of tools that make easier its use by e.g. running jobs. Another approach is by using a distributed (grid-wide) file system, with appropriate caches that are synchronized asynchronously with the storage servers. (CERN, C. Grandi)

Data Management and Global Namespace

- Analysis jobs access:
  - Object accessed/transferred on demand
  - Event on demand
  - File on demand
  - Dataset on demand
  - Dataset scheduled transfer based on measured demand
  - Dataset scheduled transfer based on imagined demand

  (SLAC, Richard P. Mount)

File Catalogues

- Problems:
  - Lack of integration in the management of metadata (i.e. catalogues) and data (srm end-points) causing the de-synchronization between storage and catalogues. (INFN23, G. Donvito)
  - unscalable and rigid channel architecture (Atlas, S.Campana)
  - a lack of awareness
    - of the network performance and the storage endpoints’ state and capacity
    - of multiple possible data sources
    - of the state of each endpoint, i.e., does it have available transfer capacity
    - of the current efficiency of any active transfer

    (Atlas, S.Campana)
  - Difficult error handling and propagation. Oversubscription lead to timeout. Incompatible with small files transfer. *(*GridPP Storage Group , Jensen)
- Solutions:
  - Data discovery service (Atlas, Garonne)
    - storage systems publish their contents and updates through notification system
    - converges to a model where storage becomes a key = value abstraction
  - Delegate the transport of cataloguing updates to a standard reliable messaging software stack and use this as the basis of the reliability. (J-P.Baud,G. McCance)
  - All transfers should go into a general queue, with specifiable priorities. Experiment/VO shares should be controllable, probably per endpoint, self configuring system. A generic transfer service as an integral part of the site storage system should be built (Atlas, S.Campana)

File Catalogues

- Solutions
  - AliEn
    - Global Unique name space: mapping from LFN to PFN
    - UNIX-like file system interface, ACLs
    - Powerful metadata catalogue
    - Automatic SE selection
    - Integrated quota system
    - Multiple storage protocols: xrootd, torrent, srm, file
    - Physical file archival (helps with MSS)
    - Used by several experiments

    (CERN, *P. Saiz)*

Summary

- Focus on analysis use cases
  - Concerns about performance and scalability
  - 2013 timescale for solution in production
  - Model:
    - network centric
    - tapes as pure archives
    - remote access
    - P2P
  - Data access layer (xroot/gfal,…)
  - Global namespace
  - Should work for Pilots, virtualization
- Milestones & Metrics
  - Important to compare with existing solutions and future needs
  - Update expected on the above at WLCG workshop in July, with regular updates at e.g. GDBs;
  - Tracking the progress (public wiki?)
  - Bheck-point – before end 2010 seen as essential
- Demonstrators for catalogues, file transfer system, data access, storage management building blocks are needed (there were some volunteers)