Distributed Storage from the WLCG/ATLAS Perspective

Shawn McKee/University of Michigan/ATLAS

MAGIC Meeting

March 6th, 2013

LHC Storage Management

- The LHC experiments have developed their own distributed data management tools and infrastructure
  - All have some form of meta-data catalog which is critical for managing the distributed data for effective use
  - The basic unit is a dataset which is a grouping of 1 or more files
  - Information about which storage locations have which datasets and the status of the datasets (incomplete, complete) is kept
- Since I am an ATLAS Physicist I will provide details about how ATLAS provides and manages storage. The other experiments have similar middleware to provide equivalent functionality though there are significant differences

ATLAS DDM Components

- ATLAS currently uses a system called DQ2 to provide the meta-data function for its data. (See http://iopscience.iop.org/1742-6596/119/6/062017)

The scope of the system encompasses the management of file-based data of all types (event data, conditions data, user-defined filesets containing files of any type).

- DQ2 is responsible for:
  - Bookkeeping of all file-based data; providing a global namespace
  - Managing movement of data between end-points
  - Enforcing data-access and user quotas
- DQ2 relies upon other tools like FTS (File Transfer Service) or **PANDA** (workload management system) for moving data.
- ATLAS maintains a Logical File Name(LFN) Catalog (called LFC) for each storage site which maps LFNs to physical instances at that site.

ATLAS DDM Concepts

- **Datasets:** A dataset is an aggregation of data, typically spawning more than one physical file, that are processed together and serve collectively as input or output of a computation or data acquisition process.
  - Datasets typically have O(100) files but range from 1-10000 files
  - Datasets are the “unit” of data movement managed by the system
  - Datasets have
    - Versions – Version 1 created when dataset created. New versions represent changes (add/delete files). Version ‘0’ always points to current
    - Immutability – Datasets can be open, closed or frozen. Open at first while data is added: new content appended to current version. Closed means content for this version is fixed. Frozen means no new version allowed, content is immutable
- **Files:** Basic unit of data. File are immutable and identified by GUID and LFN (Logical File Name)

Current ATLAS DDM Architecture

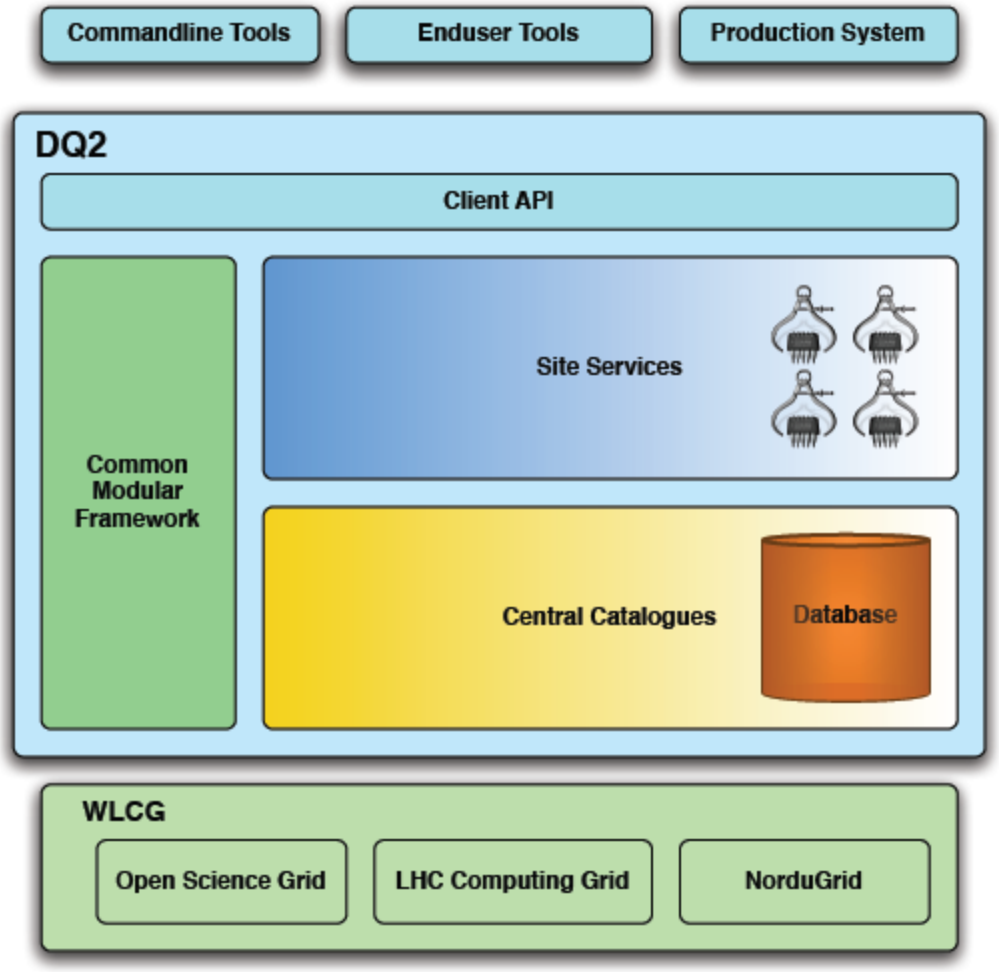
DQ2 is central for distributed data management

DQ2 provides a global namespace for storage files

Includes command-line, end-user and production interfaces

Inter-operates with all three types of WLCG grids in ATLAS: OSG, LHC-CG and NorduGrid

Has worked well to-date but is reaching its limits



ATLAS DDM and "Grid Storage"

- The ATLAS DDM system uses DQ2 to manage dataset details (which sites/locations have which datasets, dataset status, composition[Files/GUID/Checksum/Size], ACLs, quotas), LFCs to map files to physical copies at specific sites and FTS to handle details of data movement.
- FTS (File Transfer Service) typically interacts with end-site storage through an abstraction: SRM (Storage Resource Manager: http://en.wikipedia.org/wiki/Storage_Resource_Manager)
  - SRM provides a generic interface to the sites local storage system, hiding differences between specific technologies like GPFS, Lustre, dCache, Hadoop and others
  - FTS relies upon other tools like GridFTP, RFT or SRM copy to move the bits. (http://wiki.chipp.ch/twiki/pub/LCGTier2/FTSlinks/transfer.pdf)

ATLAS and Space-Tokens

- Unlike CMS, ATLAS choose to manage end-site storage space via the concept of “space-tokens”.
  - DATADISK, PRODDISK, SCRATCHDISK, GROUPDISK, etc.
- Space-tokens provide a logical grouping and quota for specific types of data and represent a “location” at the end-site
- Transfers (writing data) must be allowed to access the destination space-token they are targeting and are limited by the available space in the space-token
- This has not proven to be that helpful in practice and ATLAS is merging/removing space-tokens as feasible
  - We do need a way to regulate space-usage within ATLAS
  - Central management is viewed as a possible solution (see Rucio)
  - Other future alternatives?

Updates to ATLAS DDM

- ATLAS is using the Long Shutdown 1 (LS1) period to update is software infrastructure:
  - DQ2 -> Rucio
  - FTS 2.0 -> FTS 3.0
  - PANDA -> PANDA+network awareness (same for CMS's PhEDEx)
- Rucio improves on DQ2 based upon our practical experience
  - PFN path is deterministic based upon LFN (no need for LFC)
  - File storage on end-sites balanced in terms of files/directory
  - Improvement in policy, accounting, replication, permissions, scalability
- FTS 3
  - Removes "overlay" channel model
  - More protocol support ((gsiftp, srm, http & xroot); simpler config
  - Better manageability, scalability, improved error messages

A New Option: Federated Xrootd

- ATLAS and CMS are both testing "Federated Xrootd"
  - Based upon the xrootd protocol and the global namespace of the experiments, files can be accessed where ever they are via the WAN
- Jobs are typically sent to the data in both experiments.
  - Sometimes data is missing at the site (bad catalog info)
  - Sometimes the data is on an offline element at the site
  - Sometimes sites with the data are "full" while others have job-slots
- Federating access to storage allows better resiliency and resource usage
- Protocol selected for this first implementation is Xrootd but it could also be any other protocol in the future (HTTP)
- Tests have shown this works. Remotely running via WAN access for large RTT has CPU efficiencies ~40-50% Better than a failure or 0%!

Some Thoughts

- "Storage" needs some attributes exposed to make it usable and manageable in a distributed system:
  - ACLs/authorization/quotas by user/group to control space use/access
  - Protocol discovery: What protocols are supported with this end-point?
  - Usage information: How much is available? Usable by X? Used by X?)
  - Reservations: Can I reserve space for my data?
  - I/O capabilities: What rate can this instance read/write?
- All issues are ultimately "end-to-end". Need to consider this when planning new middleware and research
  - Design with end-users in mind: how to make things work as well as possible with minimal user input and intervention?
- Storage is NOT separate from the network for distributed science
  - Networks and their behaviors are an integral part of any distributed storage system and need to be part of any design
  - New capabilities present in networks (SDN) need to be incorporated