Managing ATLAS data on a petabyte-scale with DQ2

Mario Lassnig
on behalf of ATLAS DDM

CHEP2007, Victoria B.C.

CERN, Switzerland
University of Innsbruck, Austria

Outline

- **System overview**

  - *Concepts and principles*
  - *Components of the system*

- **Improvements on the system**

  - *Changes to the architecture*
  - *Implementation details*

- **Conclusions**

  "The throughput peaked at 200MB/s for 2 hours at the end of the exercise, our largest average daily rate was just over 90MB/s."

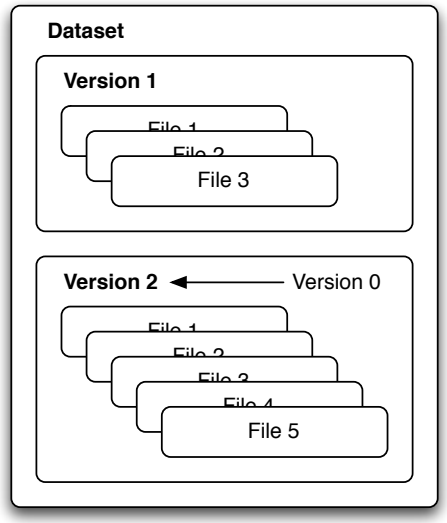
  -CHEP2006

Scope

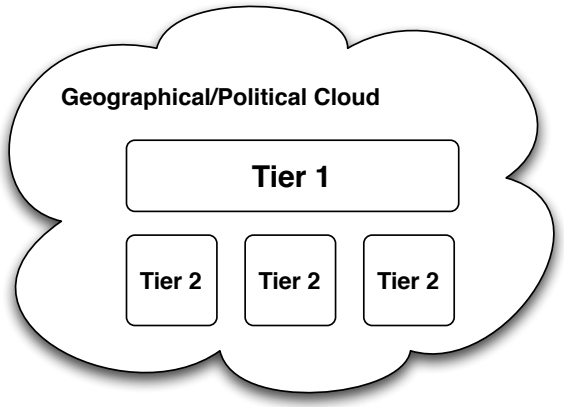
- **Responsibilities of ATLAS Distributed Data Management**
  - bookkeeping of all ATLAS file-based experiment and user data
  - managing movement of data across sites
  - enforcing access control and quotas
- **Objective of ATLAS Distributed Data Management**
  - manage the ATLAS dataflow
  - according to the ATLAS Computing Model (see talk #200, Jones)
  - with a single entry point to all distributed data

Basic Concepts

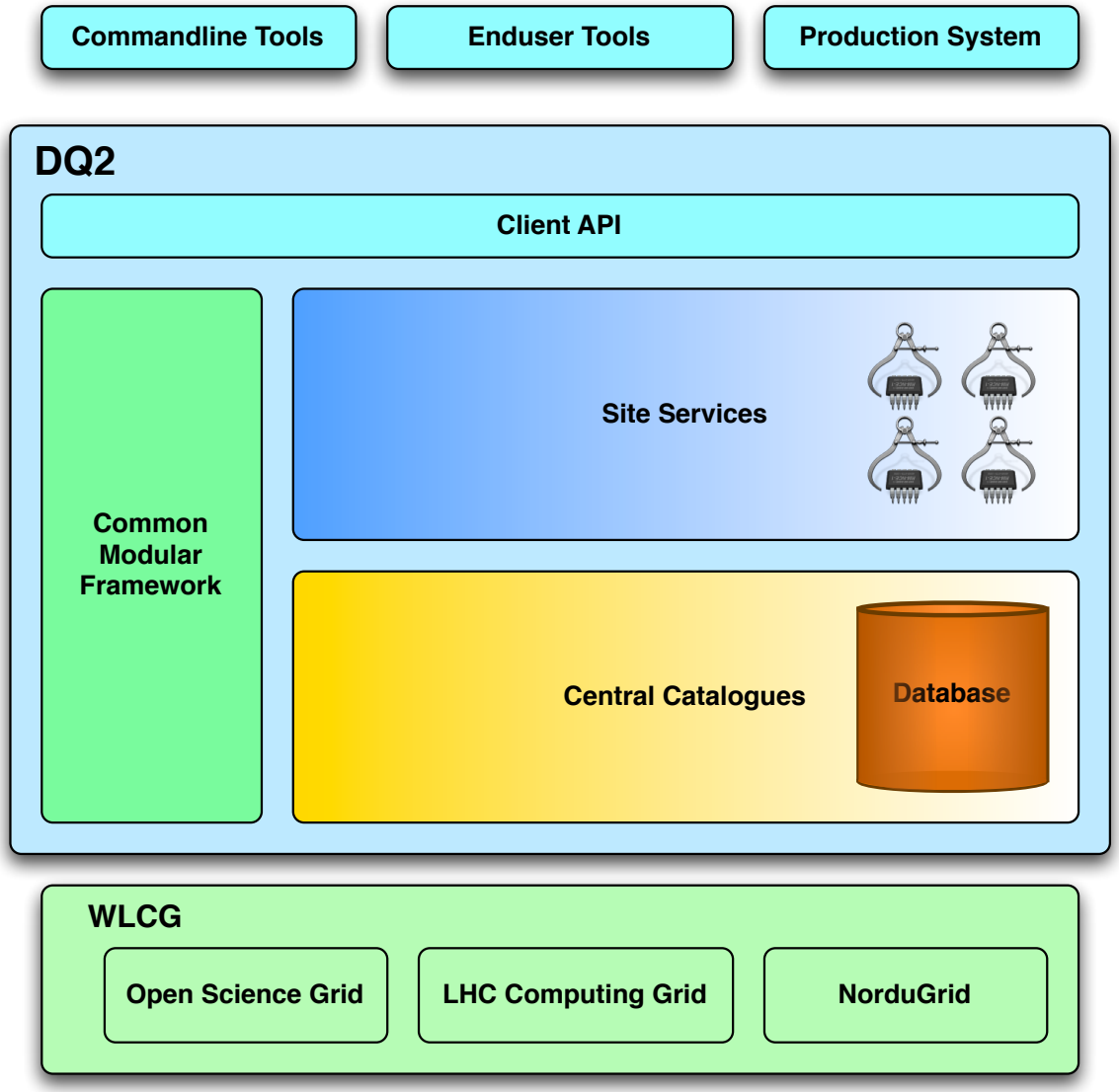
- **Experiment data**
  - Files belonging together are grouped in datasets
  - Datasets serve as primary metadata for grouped files
  - Extended features (versions, immutability, overlapping)
  - Cloud-based architecture provides sites organised in hierarchical Tiers



- **Transferring data**
  - Pull-based Subscription model
  - Sites subscribe to a dataset (version)
  - Local site service agents satisfy the subscriptions



System Architecture Overview



- **Common Framework**
  - Python 2.2/2.3
  - Transfertools, ...
- **Client API**
  - Tools
  - Data Acquisition
- **Local Site Services**
  - Autonomous agents
- **Central Catalogues**
  - Location
  - Subscription
  - Repository
  - Content

Issues addressed

- **Central Catalogues**

  - Rapid content increase in the last 6 months (since 0.3-dev)
    - Files by 15% to ~19.200.000
    - Datasets by 45% to ~290.000

  - Emergence of very large datasets with 20.000+ files

- **Site Services**

  - Performance Improvements (Priorities and Fairshare Scheduling)

  - Operations, Maintainability, Deployment

- **DQ2 API**

File catalogues

- **DQ2 provides a common interface to all Grid flavours**

- **Grid specific implementation of file catalogues**

  - LCG File Catalogue
  - OSG uses MySQL / POOL
  - NorduGrid uses Globus-RLS with LRC interface (OSG like)

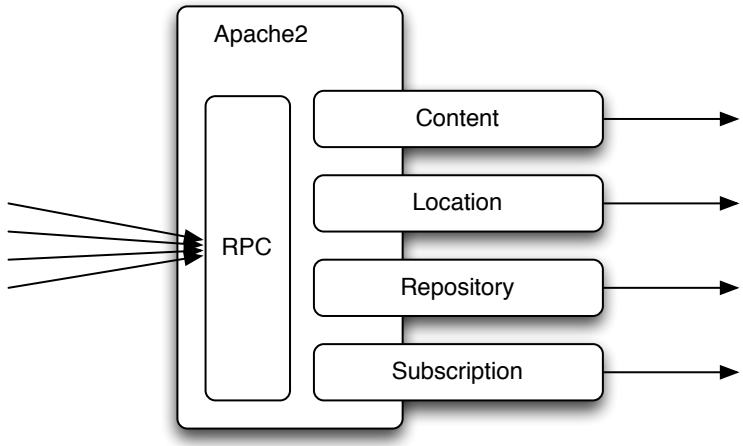
- **NorduGrid is now fully integrated in DQ2 (see poster #149, Cameron)**

  - NDGF has a unique distributed T1 facility using dCache
  - Disk pools are physically distributed over Scandinavia
  - Provide single SRM endpoint to DQ2

- **Objective was to remove client dependency on ARC middleware**

Central Catalogues

- **Standalone web based API**
  - Apache with mod_python and mod_gridsite
  - Authentication based on grid certificates
  - Common endpoint to different catalogues
  - Client calls via CURL or pyCURL
  - Direct access via plaintext/HTTP



- **Relational database access pool**
  - Factory design pattern for easy access to different catalogues

Central Catalogues

- **Move to CERN centrally hosted Oracle instances**

- **Design changes**

  - Schema changes to accomodate partitioning

  - Unified webservice RPC endpoints for Central Catalogue access
    - Support for multiple API versions
    - Transactional security
    - Version control of dataset content
    - Provenance of database interaction

  - Aspect oriented programming for validation, logging, transaction, exception

- **(Minor) Hardware change**

Site Services

- **Autonomous Agent based framework**
  - Concurrent Python Agents acting on MySQL-InnoDB
  - One instance can serve multiple sites if needed
- **Objective is to satisfy subscriptions**
  - Scheduling file transfers into FTS (File Transfer Service)
  - Enabling high-level functionality (fairshare and priorities)
  - Handling dataset states
  - Callback architecture for monitoring (see talk #255, Rocha)

Site Services

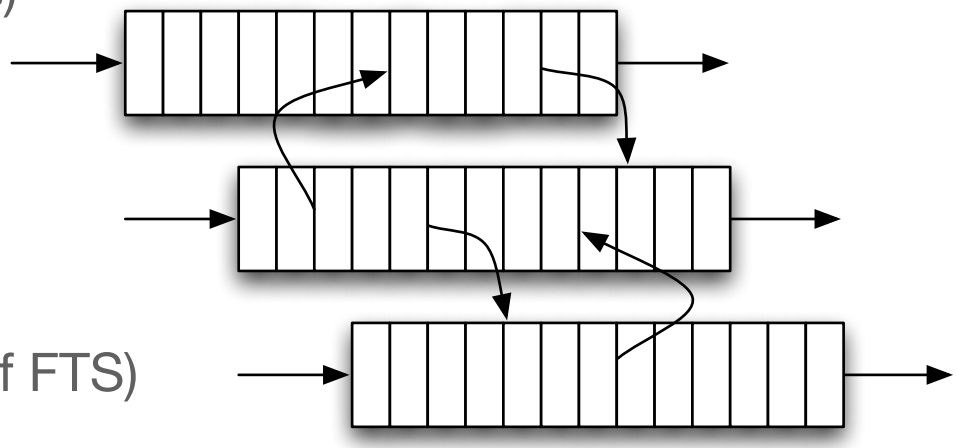
- **Parallel queue-based architecture**

Queue of subscriptions (of datasets)

Queue of files (in a dataset)

Queue of file replicas (to lookup)

Queues for FTS channels (in front of FTS)



Queue of files and datasets to register (in central catalogues)

- **Non-hierarchical Fairshare Algorithm**

  - Divides available slots according to file transfers (not filesize) into shares

  - Measure state of active transfers against queue for FTS channels

Site Services

- **Challenges**
  - Traditional queueing theory algorithms (like RED) not helpful
  - Late reshuffling of the queue (unique for ATLAS - lots of small files)
  - Sites can have over 2 million files in queue for late reshuffling
  - By contrast, idea is to keep timeouts short and queues short
- **Failover mechanism**
  - Retrial strategy
  - Element is taken off the queue with an exponential backoff
  - Then inserted again at its previous position

Site Services

- **Channel Allocation Algorithm**

  - We always try to perform on connected sites

  - Choose the source-destination channel that has the
    - least number of transfer attempts from that source
    - if multiple, take the one with the shortest queue

  - If that is not possible (after retrial) we fallback to multihop transfers

- **Special case in data integrity**

  - Dataset bookkeeping is tightly integrated with transferring

  - Ability to insert requests even before the data exists

  - This allows early optimisation of the queue

Tier0 Export Tests

- **First half of 2007 - ready for SC4**
  - At the beginning of March DQ2 0.3 is ready for testing
  - Multi-month exercise debugging Tier0 to Tier1 transfers
  - Improve transfer reliability (retrial strategies, transfer balancing)
  - Better subscription monitoring (with ARDA)
  - Concurrently, migration of datasets from DQ2 0.2 to new DQ2 0.3 schema
  - Reducing the load on central catalogues (SQL query optimisation)

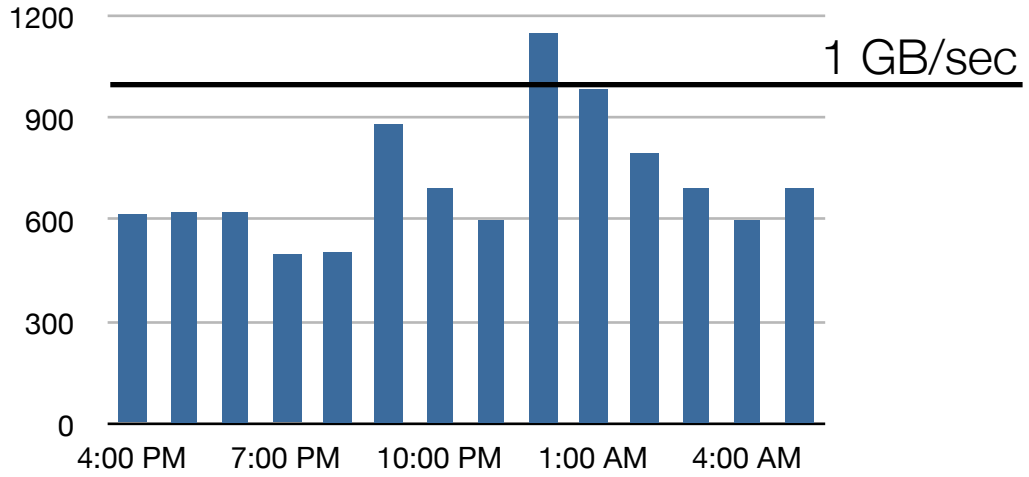
- **Special thanks**
  to all site admins, the ATLAS DDM operations team, the Tier0 Exporters, the ARDA people and the CASTOR team!

Tier0 Export Tests

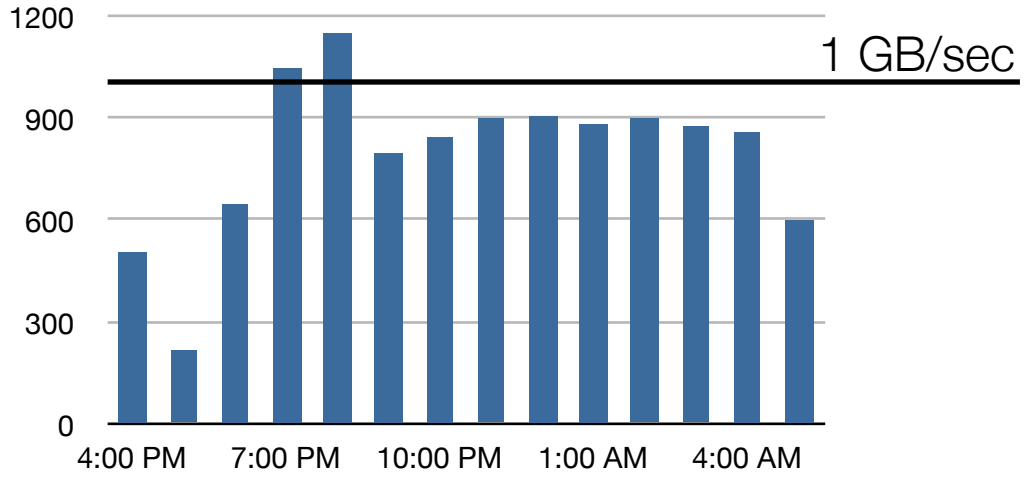
- **ATLAS T0M System provided 48 data export runs per day**
  - 960 new datasets
  - 14880 new files
  - 1037 GB new data
  - Computing Model expects 60% from TDAQ (28 runs per day)
- **All data movement is now done by FTS**
  - FTS2 pilot service to Lyon in parallel
- **Optimisation through micromanagement algorithms for FTS**

Tier0 Export Rates

Throughput MB/sec
June 8, 2007



Throughput MB/sec
June 22, 2007



Open issues and future plans

- **No evident problems on the Central Catalogues**
- **Site Services**
  - Performance improvements on site services database
  - Continous improvements on failover strategies
  - Allocation algorithm improvements on partitioning and fairshare
  - Transfer reporting of partially complete datasets
- **Tracking Service of performance critical subsystems**

Open issues and future plans

- **Deletion of datasets is critical & non-trivial**
  - overlapping datasets cause heavy load on central catalogues
- **Redesigned integration with Production System**
  - due to now available LCG python libraries
- **We need users!**
  - Improvements based on access patterns
  - For an informal tutorial of DQ2 tools contact me this week

Conclusions

- **Improvements Tier0 Export**
  - Peaks increased by factor 5 to over 1000MB/s
  - Largest average daily increased by factor 7 to over 600MB/s
- **Improvements Production**
  - Number of files transferred as a new metric
- **Improvements in Interfacing, Deployment and Operations**
- **We are (almost) ready for data-taking!**

Managing ATLAS data on a petabyte-scale with DQ2

Mario Lassnig
on behalf of ATLAS DDM

CHEP2007, Victoria B.C.

CERN, Switzerland
University of Innsbruck, Austria

https://twiki.cern.ch/twiki/bin/view/Atlas/DistributedDataManagement

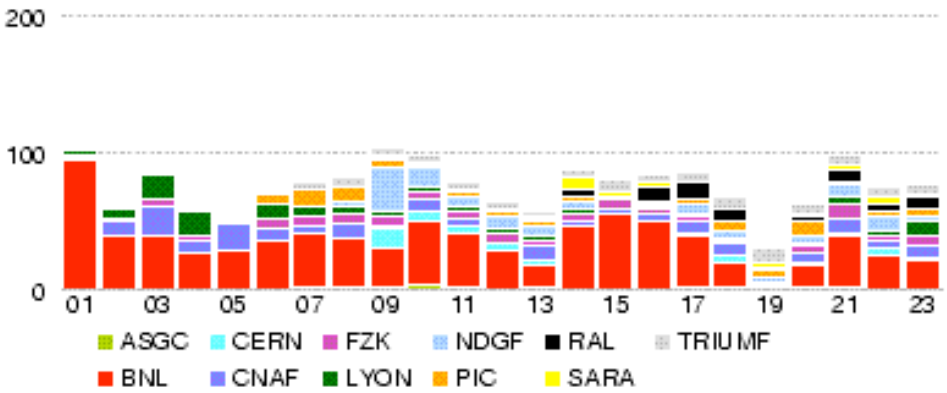
Backup Slides

Timeline

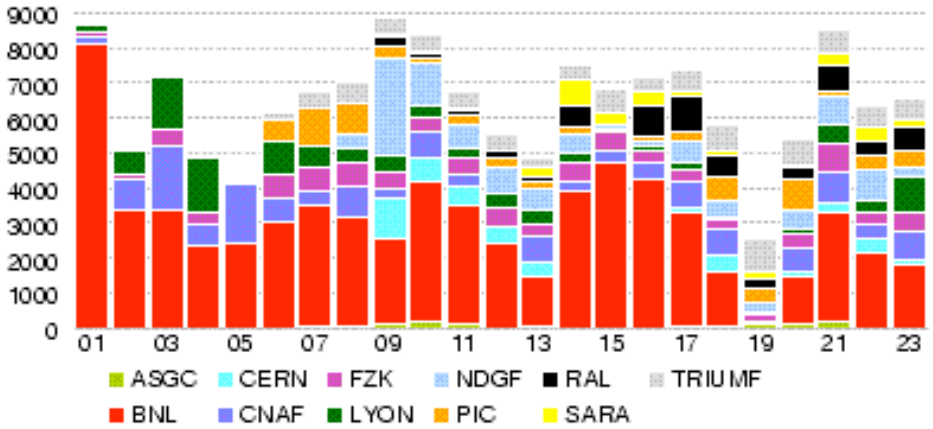
- February 2006, DQ2 Version v0.2.0 Production Release
- November 2006, DQ2 Version v0.2.12, Systems Review
- February 2007, DQ2 Version v0.3, Testing Release
- May - June 2007, DQ2 Version v0.3, Tier-0 Export Tests
- June 2007, DQ2 Version v0.3 (stable), Production Release
- September 2007, DQ2 Version v0.4 (unstable), Testing Release

Production Rates, August 2007
(please apply grain of salt to those numbers)

Throughput MB/sec



Transferred data GB



Completed files