Advances in Service and Operations for ATLAS Data Management

Graeme Stewart
for the ATLAS Collaboration

PH-ADP-CO

Overview

- DDM Project Overview
- DQ2 Current Design
- ATLAS Data
- Scaling and Performance
- New Services and Features
- Conclusions and Futures

DDM Project

- The ATLAS Distributed Data Management project is charged with managing ATLAS data on the grid
- Requirements:
  - Register and catalog data
  - Transfer data to/from sites
  - Delete data from sites
  - Ensure data consistency
  - Enforce ATLAS computing model requirements

DDM Contributors

- DDM is a collaborative effort, with many contributions from ATLAS institutes

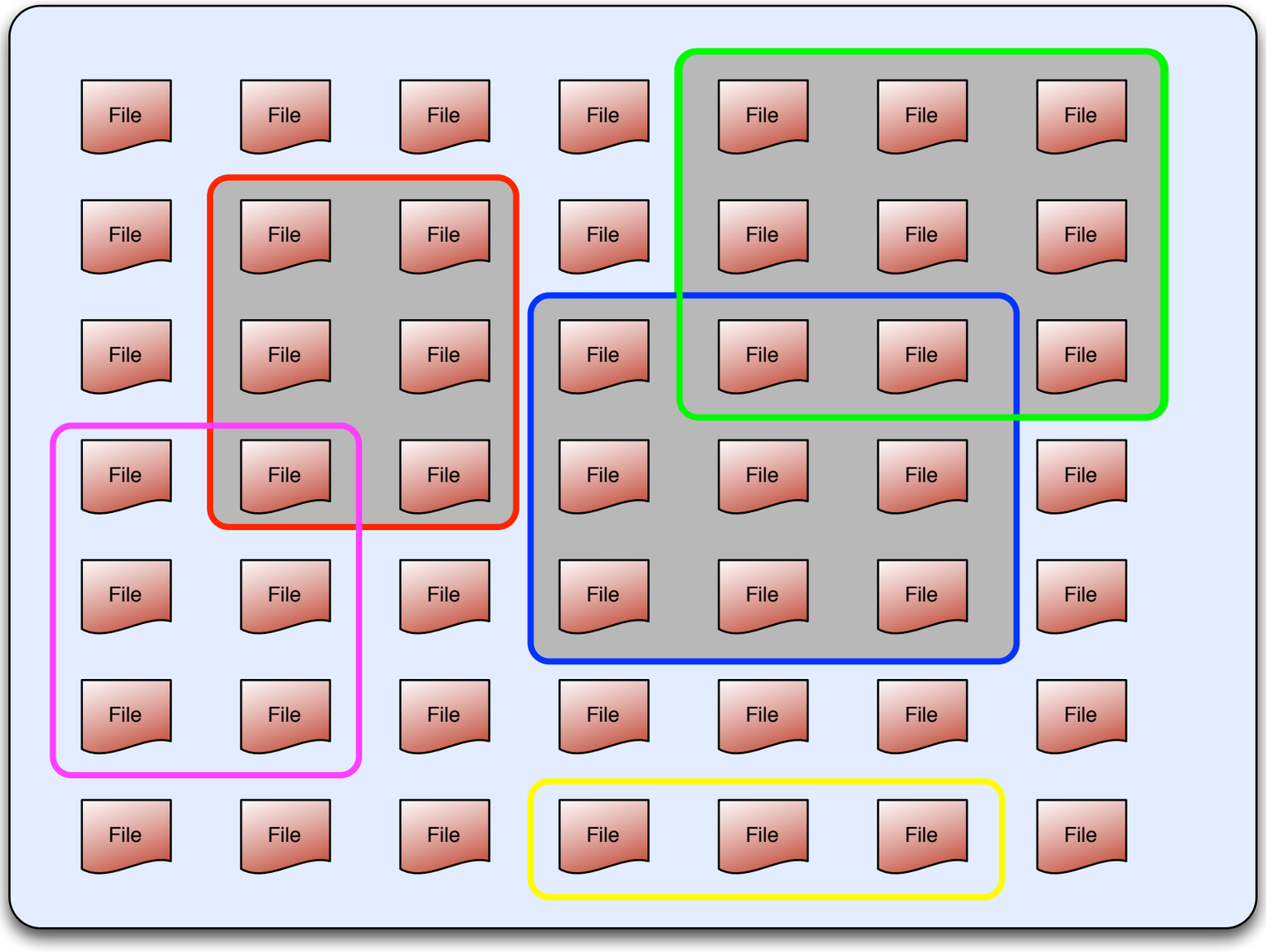
Jožef Stefan Institute

Institute of High Energy Physics
Chinese Academy of Sciences

ATLAS Data Concepts

- At the heart of everything is a file, of course
- Files in ATLAS are collected into datasets
  - Datasets live in a flat namespace
    - Naming convention: e.g., data11_7TeV.00184130.physics_Muons.recon.ESD.r2603_tid491184_00
  - All files must be in at least one dataset
  - But overlapping datasets are supported, i.e., files may be in multiple datasets
- Datasets are the units of replication
- Note in particular that ATLAS central catalogs do not organise data in a filesystem-like way
- Datasets can be aggregated into container objects

Data Model



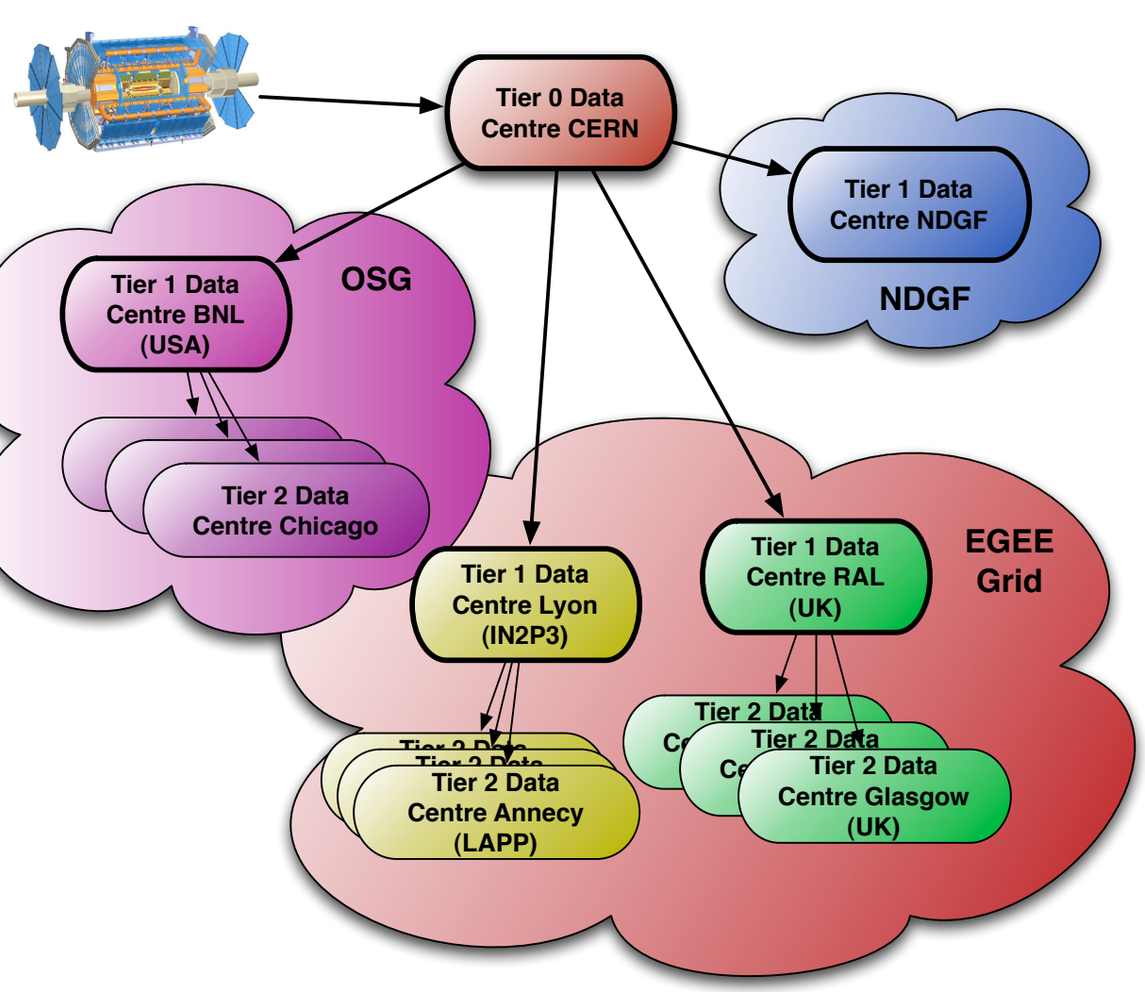
Dataset U Dataset U Dataset = Container

Sites, Regions and Grids

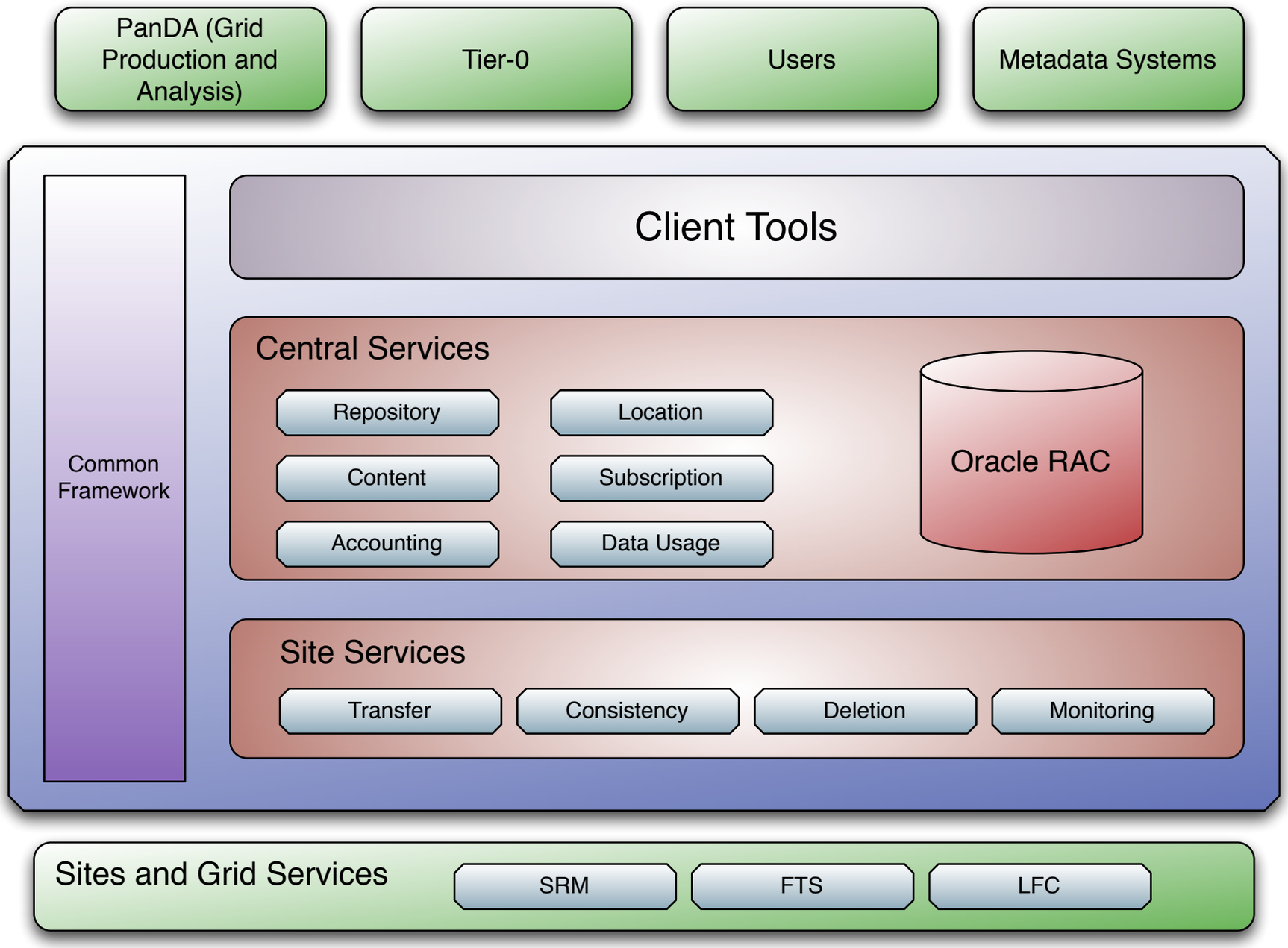
- DDM places data at ‘endpoints’
  - This is host, service, path and service specifier
    - svr018.gla.scotgrid.ac.uk
    - SRMv2
    - /dpm/gla.scotgrid.ac.uk/home/atlas
    - ATLASDATADISK
- So each grid site can host multiple endpoints
- ATLAS organises sites into regions, e.g., UK
- We are pretty agnostic to grid middleware flavours, but use resources in EGI, OSG and NDGF
  - NDGF is special - distributed storage but this is ‘hidden’ from DDM

Cloud Model

- Associations of sites, offering distinct QoS
  - Used to be rather strict, but now evolving



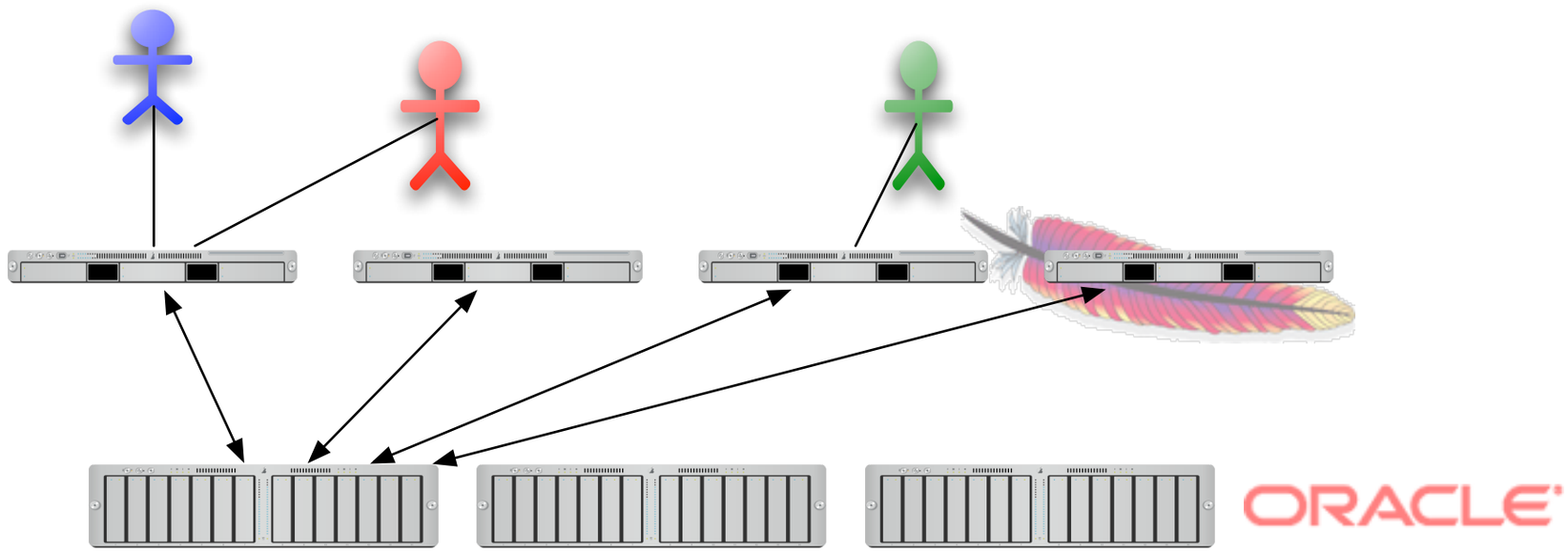
DQ2 Architecture



- Internally to DDM separation of
  - Clients
  - Central Services
  - Site Services
- But code base is common where possible

Central Catalogs

- High availability architecture
  - Multiple stateless front ends with apache + mod_python
  - Oracle RAC for backend database
    - So ACID compliant, with redundancy
    - Schema is optimised for high performance
      - Tuning and re-optimisation is frequent

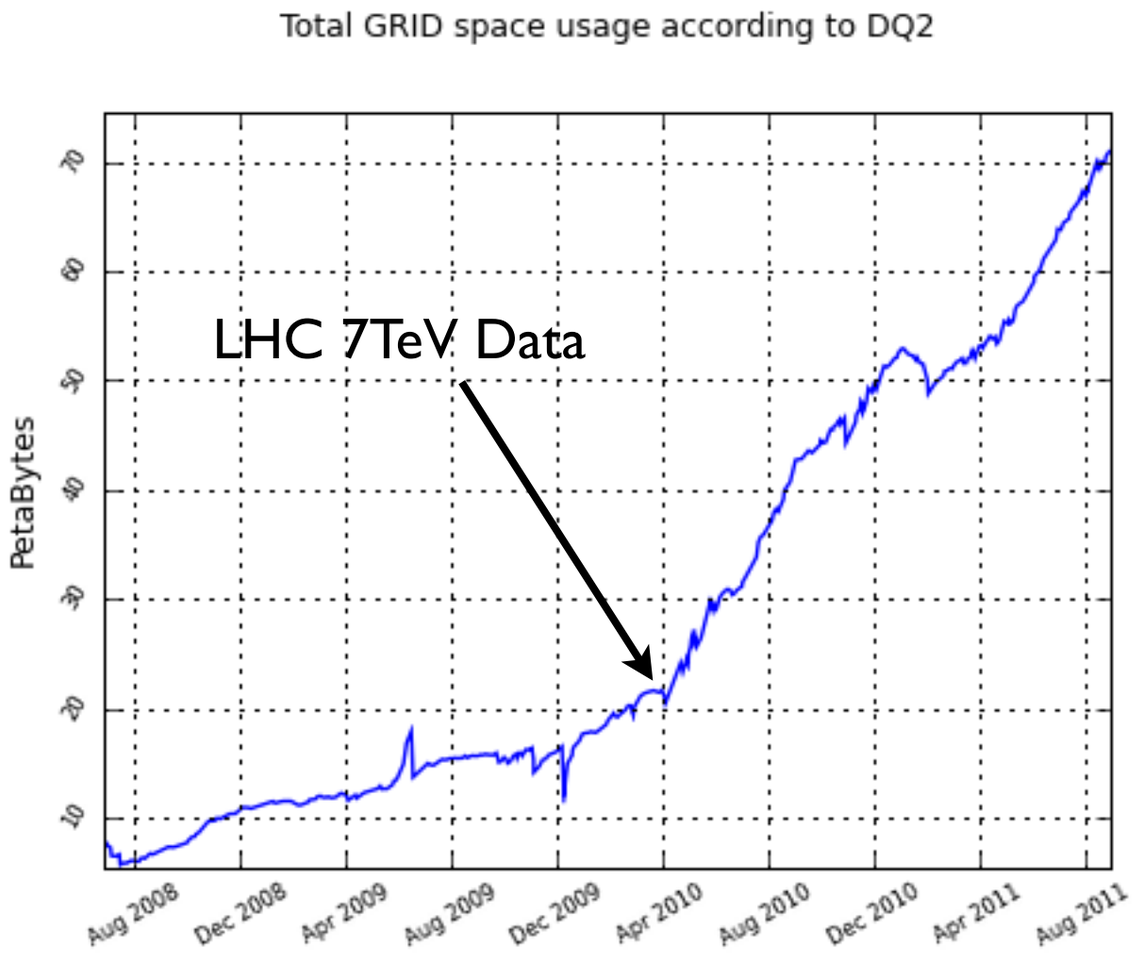


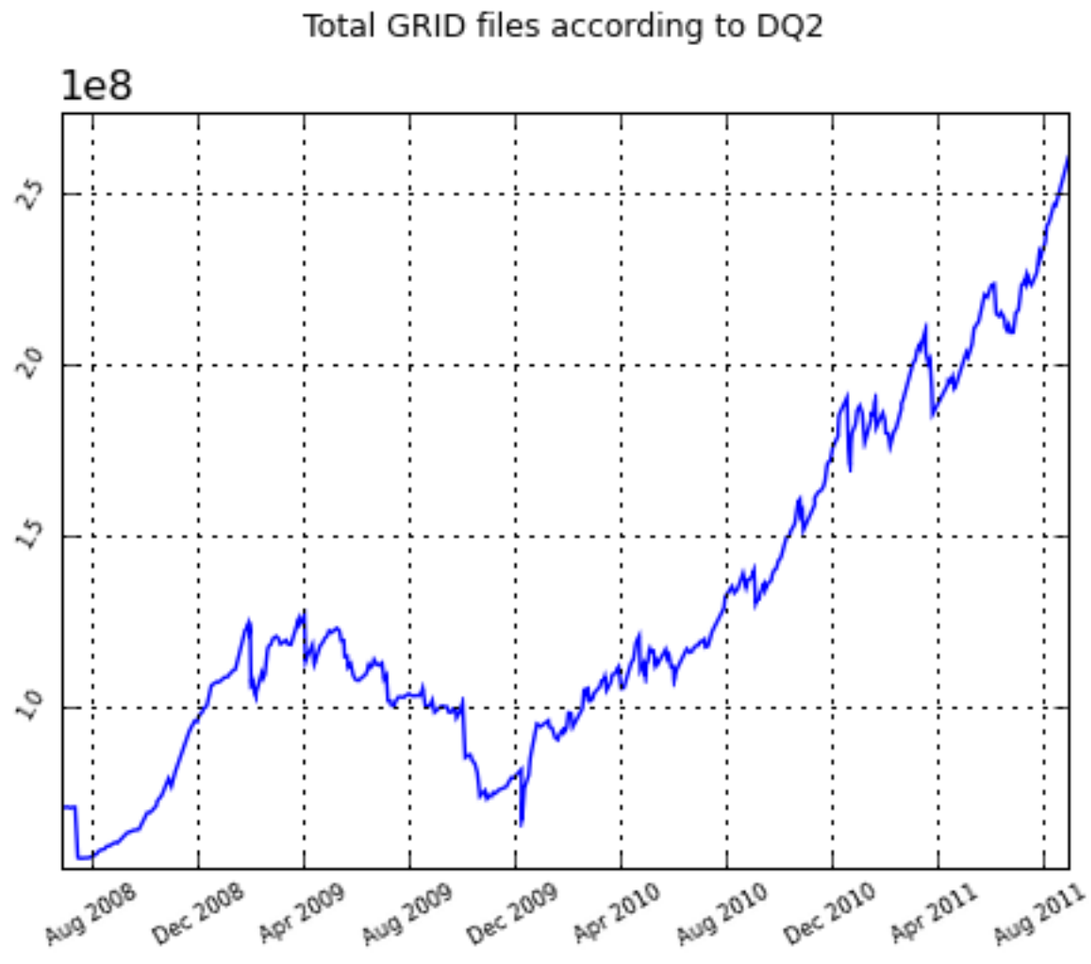
Site Services

- Interactions with
  - Central services
  - Grid resources and services
- Reuse of technical solutions
  - Define state machine for distributed transactions
  - Check-point centrally pending requests
  - Throttle interactions with remote grid servers
  - Split requests into chunks for efficient bulk execution
  - Retry, ... retry, ... and ... retry

Data Scales

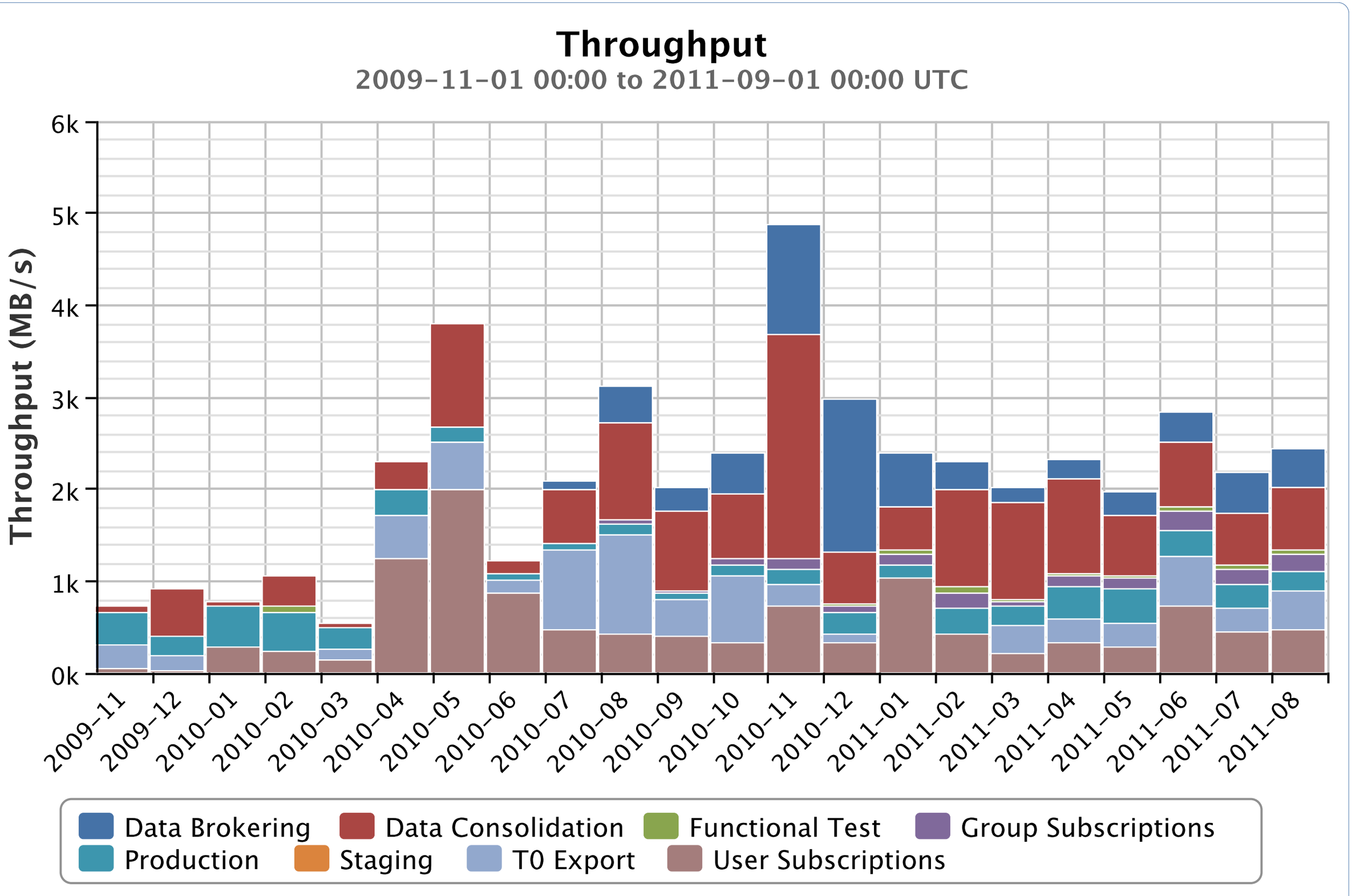
- All this was in place before significant LHC data arrived
- So we had some confidence it would work ... and it did!
- Scaling with LHC data arrival has been impressive





DQ2 Activity

- Central Catalogs:
  - 20M read queries/day
  - 1M write requests/day
- Grid File Accesses
  - 5M/day
- Transfer Rates
  - Peaks of 10GB/s globally, 2GB/s sustained

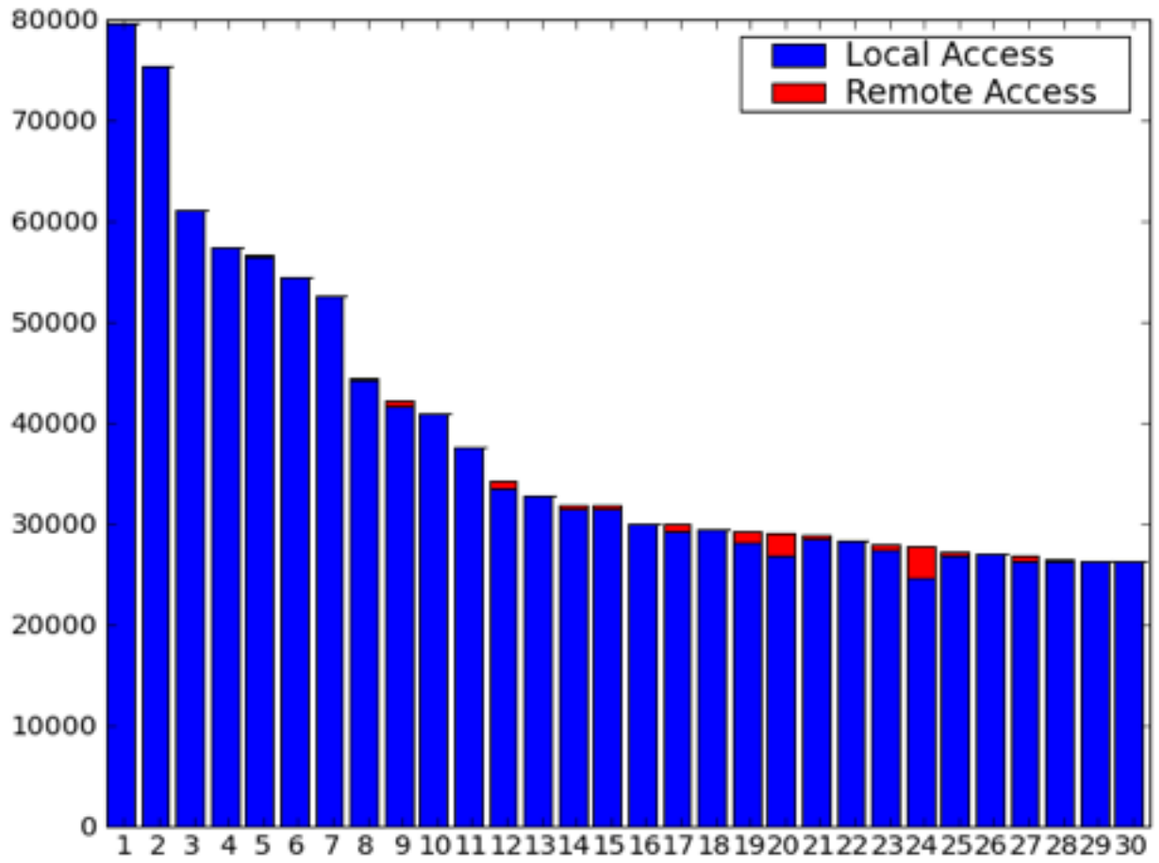

Throughput
2009-11-01 00:00 to 2011-09-01 00:00 UTC
Throughput (MB/s)
6k
5k
4k
3k
2k
1k
0k
2009-11
2009-12
2010-01
2010-02
2010-03
2010-04
2010-05
2010-06
2010-07
2010-08
2010-09
2010-10
2010-11
2010-12
2011-01
2011-02
2011-03
2011-04
2011-05
2011-06
2011-07
2011-08
Data Brokering
Data Consolidation
Functional Test
Group Subscriptions
Production
Staging
T0 Export
User Subscriptions

Tracer

- DDM tracer service records all data accesses on the grid
  - Data downloads, job accesses, etc.
    - Trace insertion rates are ~60Hz, peaks of 300Hz
- Recent advances have been to buffer traces using ActiveMQ
  - Enable bulk insertion into Oracle

Popularity

- Building on traces, summaries are made which enable queries of site (src and dst), user, dataset pattern to be made
- In particular very hot data can be identified
- And very cold data
  - This can be fed to the deletion service when sites become full and space needs to be freed
  - This is done by the 'Victor' deletion agent

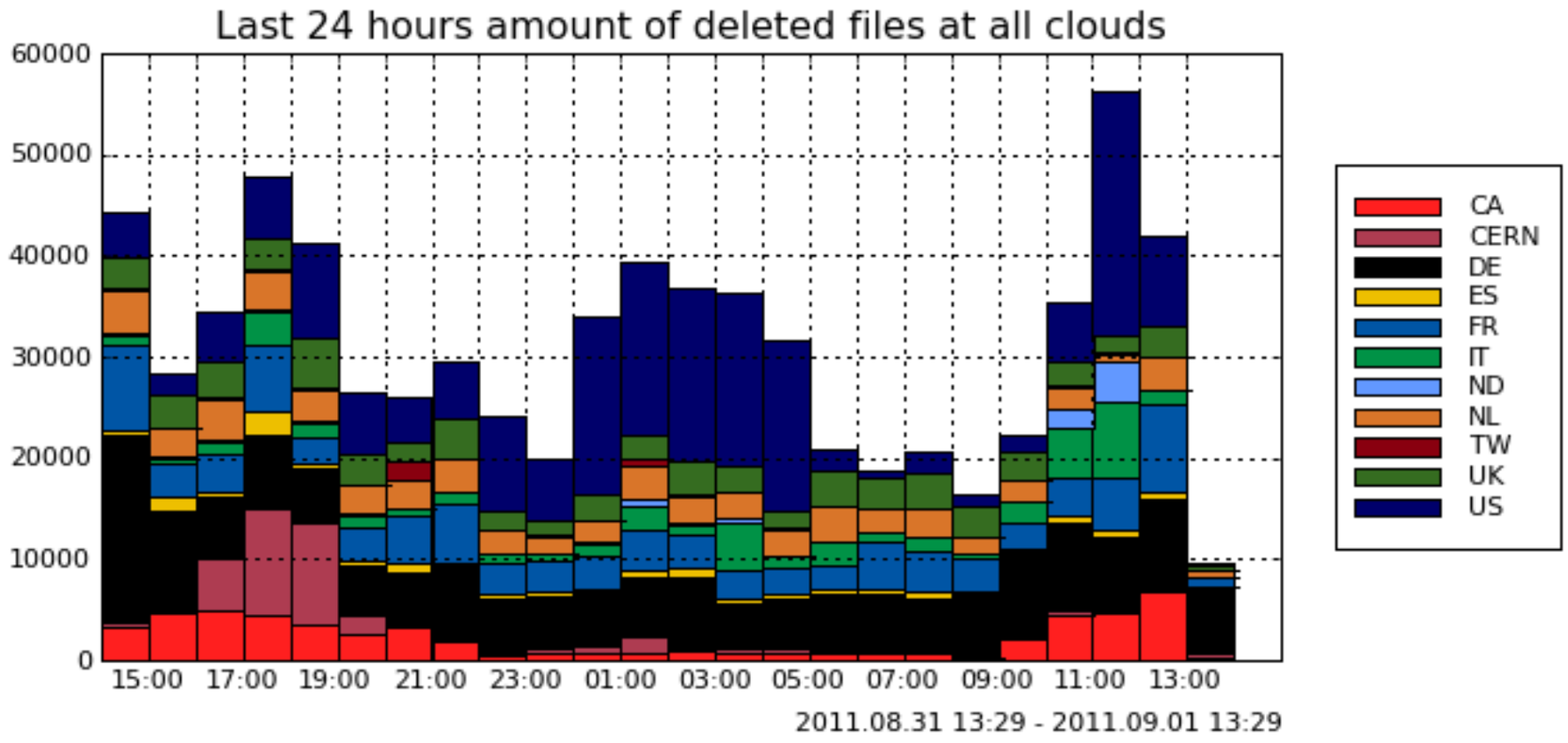


No	Dataset	Accesses	Local Access	Remote Access
1	data11_7TeV.00184130.physics_Muons.recon.ESD.r2603_tid491184_00	79541	79541	0
2	data11_7TeV.00186877.physics_JetTauEtmiss.recon.ESD.f394	75384	75362	22
3	data11_7TeV.00183780.physics_Muons.recon.ESD.r2603_tid491191_00	61108	61098	10
4	data11_7TeV.00184130.physics_JetTauEtmiss.merge.RAW	57366	57366	0
5	data11_7TeV.00184022.physics_Muons.recon.ESD.r2603_tid491189_00	56575	56573	2
6	data11_7TeV.00184169.physics_Muons.recon.ESD.r2603_tid491183_00	54486	54484	2

Deletion

- Deletion is part of the normal life cycle of data
- In particular ATLAS produces many transient datasets, which need deleted after use
- Deletion service must take care with overlapping datasets
  - Delete files only when their last dataset is removed from a site

Deletion Performance



- Typically deleting 1.5-2.0M files per day, with peaks up to 5M
- Recent new version optimises interactions with DDM and LFC catalogs

Consistency

- Data, sadly, gets lost
  - Disks die, RAID controllers die, data is corrupted, etc
- Consistency service is charged with re-establishing DQ2 consistency when data loss occurs
  - Files which exist elsewhere can be re-transferred
  - Files which are definitively lost need to be removed from dataset definitions

Consistency II

- Data can also be declared as ‘suspicious’
  - Marked as temporarily unavailable and subject to further testing
  - Recently instrumentation of grid jobs allows them to report data access failures directly to the consistency service

Locations	State 0	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State A	State B	State C
[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	3
[illegible]	0	57	0	93	0	0	0	0	0	2	17125	5	1018
[illegible]	0	124	0	91	0	0	0	0	16	4	29843	45	3750
[illegible]	0	98	0	30	0	0	0	0	0	2	37	1	49
[illegible]	0	0	0	0	0	0	0	0	0	0	197	0	2
[illegible]	0	2	0	0	0	0	0	0	0	0	0	0	0
[illegible]	0	6	0	0	0	0	0	0	0	0	0	0	0
[illegible]	0	93	0	0	0	0	0	0	0	0	0	0	0
[illegible]	0	684	0	0	0	0	0	0	0	0	0	0	0
[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	2

Accounting

- ATLAS needs to manage a large storage resource properly, which means that occupancy by site, dataset, etc. is important
- Old DQ2 accounting system based on fixed patterns proved too inflexible and hard to scale
- New system is based on metadata
  - Old: ‘data10.*.ESD.*’ + ‘CERN’
  - New: : {’project’:’data10’, ’type’:’ESD’, ‘location’:’CERN’ }
- Queries can be registered, to run periodically, then harvested to get historical data
- Both Oracle and Mongo DB supported as backends

Conclusions

- ATLAS Distributed Data Management delivered working scalable services to the collaboration in time for LHC data taking
- The systems are scaling and manage the current load
- New services, to manage the complete data life cycle, have been introduced
- We continue to optimise and tune the system
- We need to adapt to a changing landscape of distributed computing services

DDM Futures

- We have learned a lot from the current system
- However, the design is more than 5 years old and some conceptual limitations have been found
  - And the usage patterns are not quite those anticipated
- DDM team are currently re-considering the DQ2 design, for a new version *Rucio*, anticipated for 2013