Rucio

The next generation of large scale distributed system for ATLAS Data Management

Vincent Garonne, CERN

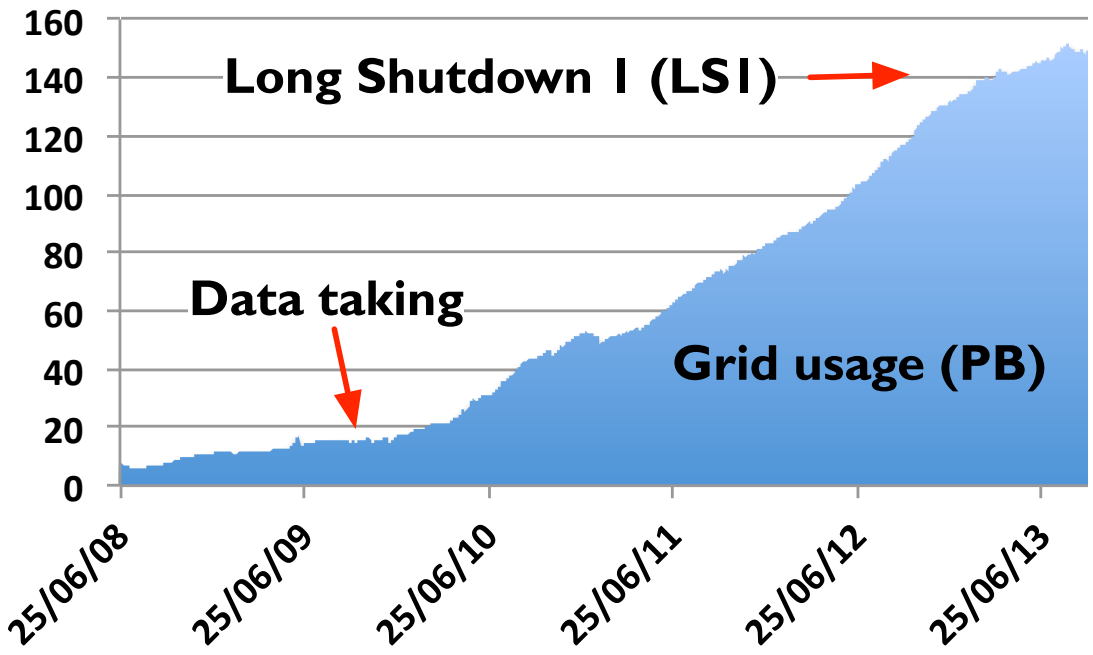
On behalf of the ATLAS Collaboration

Computing in High Energy and Nuclear Physics, Amsterdam, 2013

Background

The current DDM system Don Quijote 2 (DQ2) has demonstrated very large scale data management

- 150 PB
- +40 PB per year
- +1 M files per year
- 130 grid sites
- 800 users
- 0.6 M downloaded files per day



DQ2 will simply not continue to scale for LHC Run-2

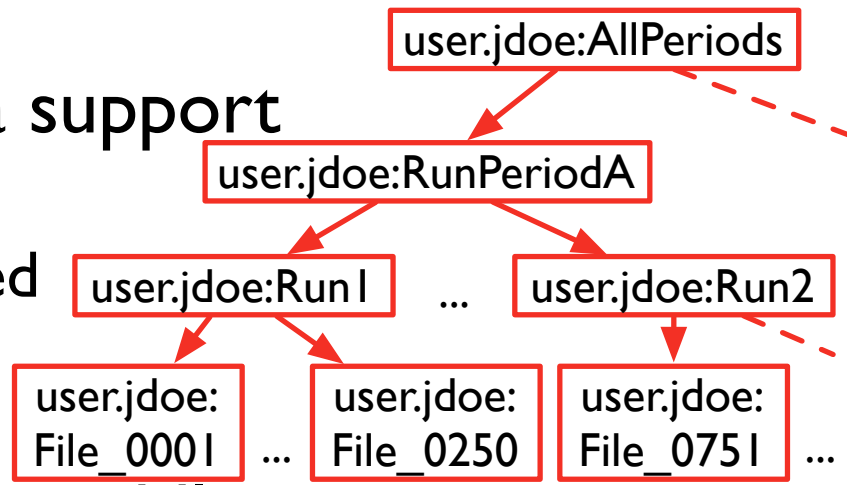
DDM LS1 Plans: Rucio

Rucio is an evolution from DQ2 designed to ensure system scalability, reduce operational overhead and support new ATLAS use cases

- Technical meetings and user surveys in 2011-2012
- Rucio conceptual model design (v2)
- Pilot service delivered in November 2012
- Performance framework established
- Functional testing, commissioning and migration
- Target: Rucio in production by beginning of 2014

Concepts - Highlights

- Better management of users, physics groups, ATLAS activities, data ownership, permission, quota, etc.
- Data hierarchy with metadata support
  - Files are grouped into datasets
  - Datasets/Containers are grouped in containers



- Concepts covering changes in middleware
  - Federations
  - Cloud storage
  - Move towards open and widely adopted protocols

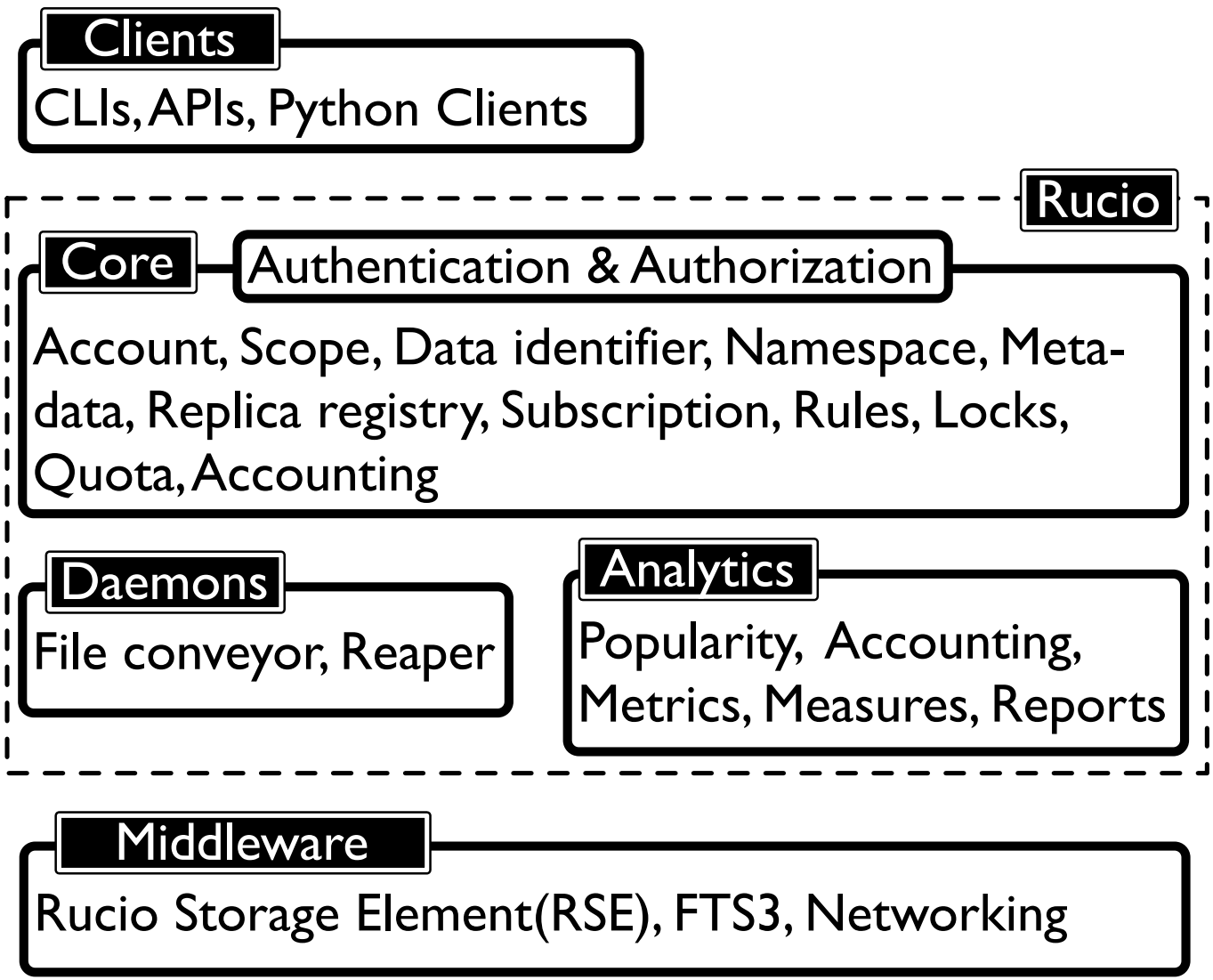
Replica Management

Replica management is based on replication rules and subcriptions to optimize storage space, minimize the number of transfers and automate data distribution

- A replication rule defines the minimal number of replicas to be stored on a set of Rucio storage elements, e.g., 2 replicas of file user.jdoe:file_001 on any Tier 1
- Subscriptions offer the on-demand creation of replication rules on new/existing data, e.g., data12_8TeV*physics_Egamma*AOD on T1s

➥ Cf. Martin Barisits's talk [link]

Software Stack



Open and standard technologies:

- RESTful APIs (https+json)
- http caching
- WSGI server
- Token-based authentication (X509, GSS)
- Open source data access protocols

Daemons

The Rucio daemons are active components that are orchestrating the collaborative work of all the system

- Conveyor - File transfers
- Reaper - Deletion
- Undertaker - Expired datasets
- Transmogrifier - Subscriptions
- Judge - Rule Engine

The daemons are lightweight, thread-safe and use sharding techniques to reduce concurrency and distribute the work

Persistence

Relational database management system

- Our main database is Oracle 11g
- Object-Relational Mapper: SQLAlchemy
- Rucio supports Oracle, PostgreSQL, MySQL, ...
- No hand-crafted SQL code in Rucio
- Use cases: Real-time data, transactional consistency

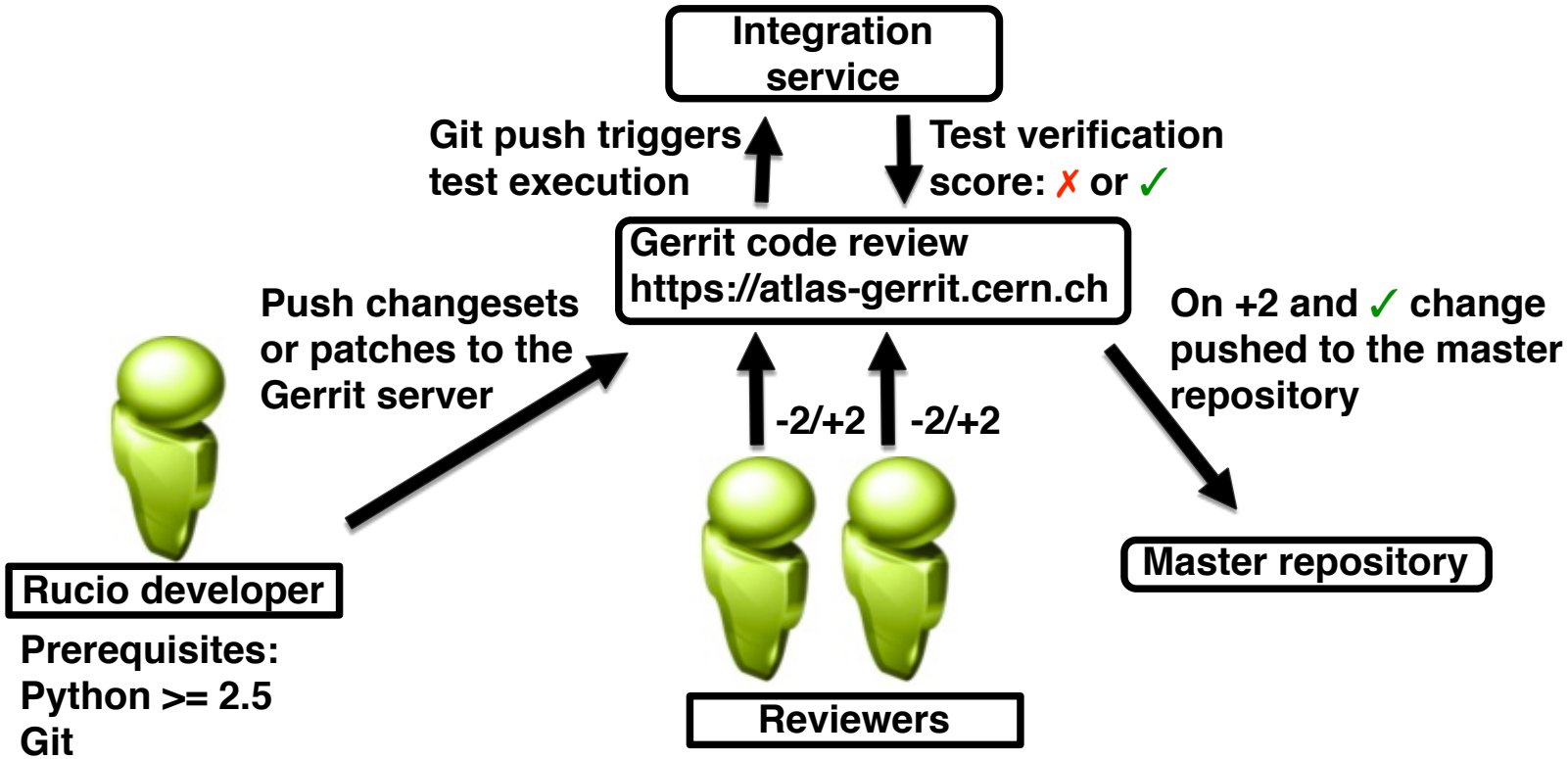
➥ Cf. Gancho Dimitrov's talk [link]

Non relational structured storage: Hadoop

- Complex analytical reports over large volumes
- Use cases: Popularity, accounting, log analysis

Development Process

Agile development with mandatory code reviews and prerequisite extensive unit and functional tests



➥ Cf. Mario Lassnig's talk [link]

Performance & Scaling

A complete framework has been established

- Instrumentation with core's internal workings exposed in graphite

A constant workload is being generated by the emulation framework

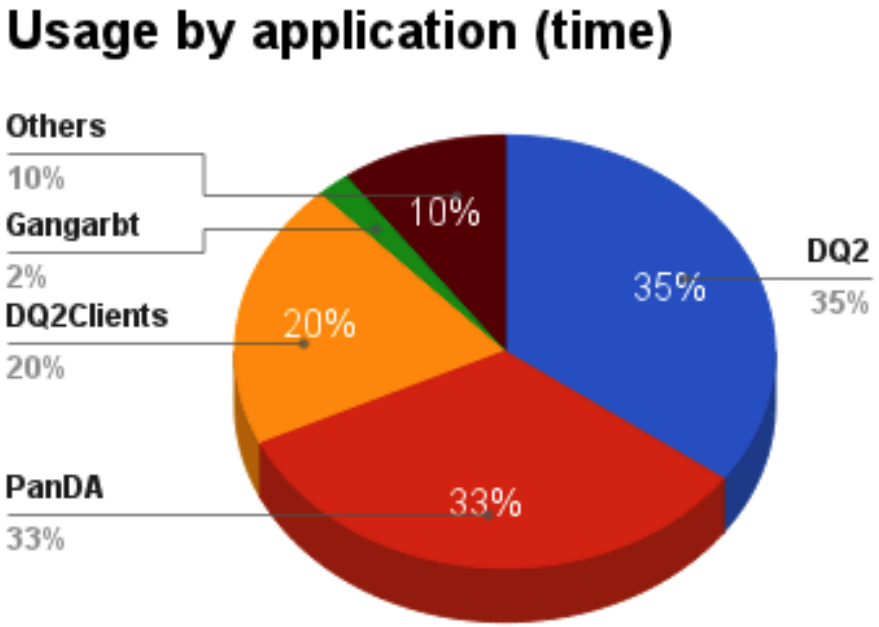
- Provide a continuous integration service
- See how Rucio behaves with increasing workload
- Identify potential bottlenecks and concurrency issues as early as possible
- Explore system boundaries

➥ Cf. Ralph Vigne's poster [link]

Benchmarking

Monitoring infrastructure based on Hadoop to

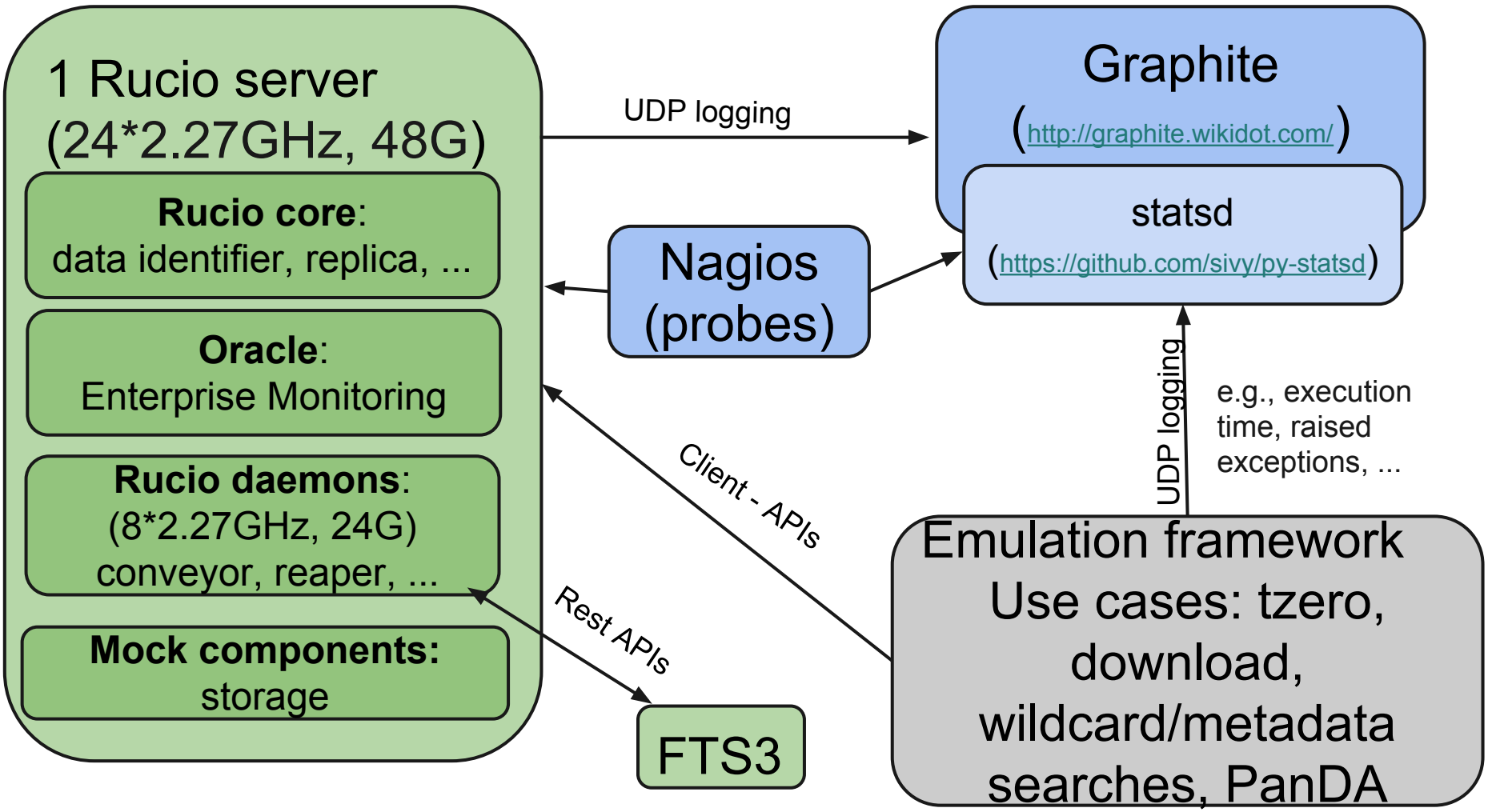
- analyse catalog traffic
- map the load to use cases
- optimize workflows



➥ Cf. Thomas Beermann's poster [link]

For each use case, we did the mapping of concepts between DQ2 and Rucio, check and implement the functional coverage of the use case

Emulation Framework



Scaling Test: Settings

The database has been pre-filled with the same characteristics as DQ2

- Data volume, e.g., 1 billion of file replicas
- Distribution according to a set of metadata and data types

The nominal load has been based on the usage and peak activities from 2012/2013

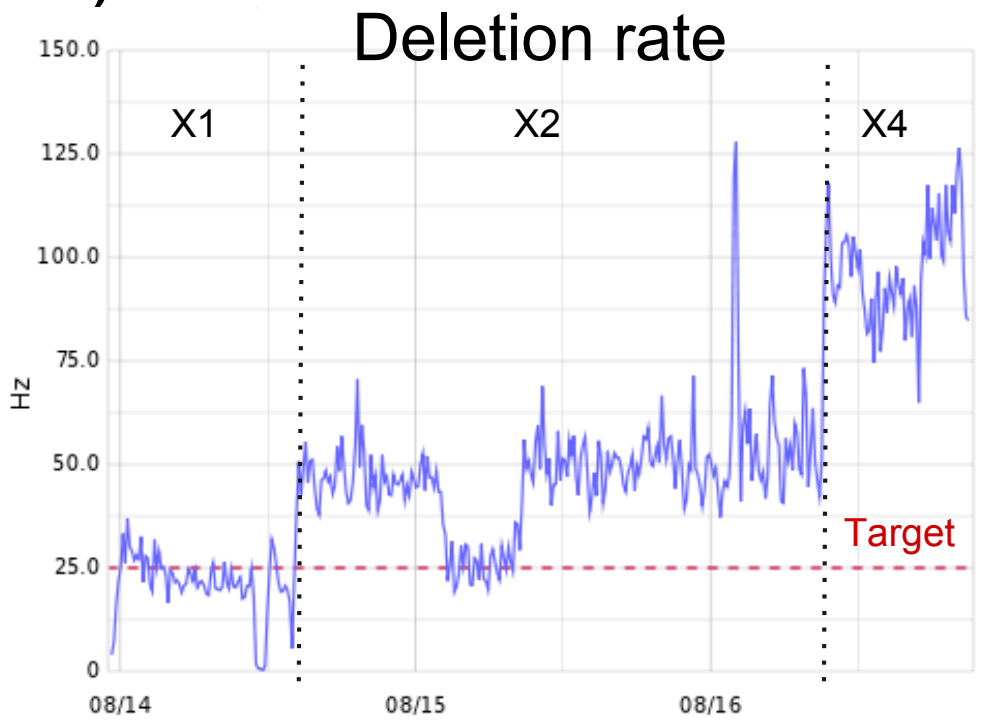
- Use cases: Tier-0 export, download, user upload, wildcard/metadata searches, PanDA ➡

Operations	Rates
Transfer	20 Hz
Deletion	25 Hz
Upload	0.04Hz
Download	7Hz

Modus Operandi

We increased the scaling factor by two for all use cases every 24 hours (ramp-up time)

- √ Nominal load: 1 → 3
  - No increase of database latency
  - No backlogs
  - No degradation of the response times
  - Rucio nodes behaved well
  - I/O bound DB application

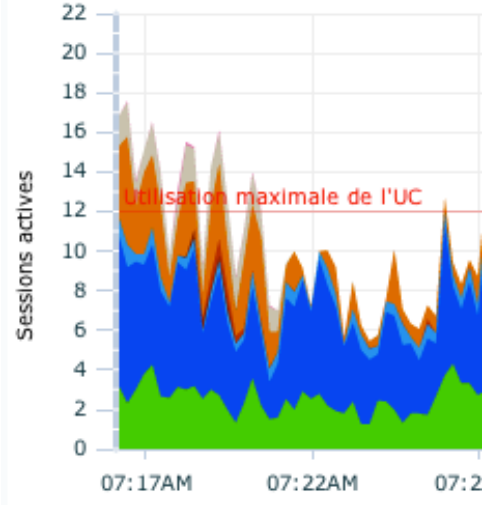


⇒ Rucio scales at nominal load

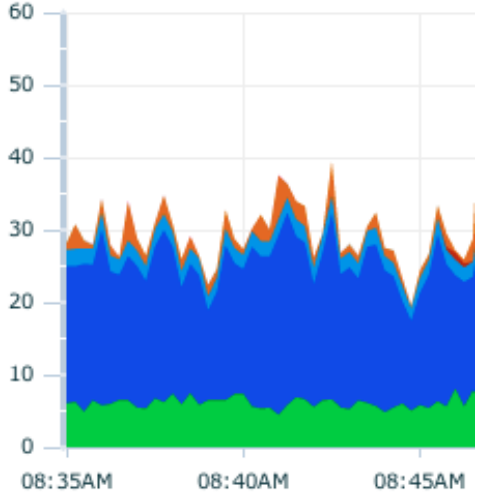
DB: Load Increase

Load

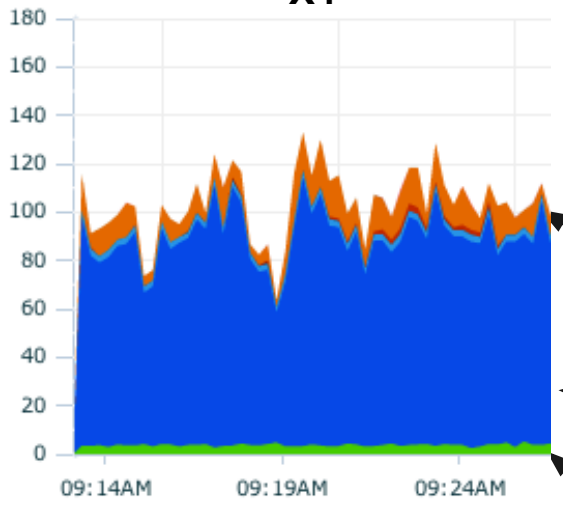
X1



X2



X4



Latency

Throughput

900 I/O ops.s-1
20 MB.s-1

2000 I/O ops.s-1
30 MB.s-1

3200 I/O ops.s-1
40 MB.s-1

Caveat: We learned that the DB was faulty since several weeks affecting the performance (latency) of commit

Migration

Many steps were defined to ease the transition from DQ2 to Rucio, e.g.,

- Mapping enforced between the CERN ATLAS accounts and the grid authentication system

A renaming infrastructure is being used to rename transparently via WebDAV all ATLAS files to a deterministic logical file name → path name scheme

- End of use of the LCG File Catalog
- 56% of Rucio paths in central CERN LFC
- Timeline: End of the year

➥ Cf. Cedric Serfon's talk [link]

Summary

- Rucio exploits commonalities between experiments and other data intensive sciences to address HEP experiments needs and scaling requirements
  - The AMS (Alpha Magnetic Spectrometer) experiment is evaluating Rucio
  - Build a broader support community
- We have a complete framework for continuous scale exercise
- Rucio with its concepts and technologies scales at nominal load
  - Nominal load is based on peak periods
  - Tests scheduled in November with the final DB hardware

What's Next

- A Rucio Integration testbed has been deployed for external applications and functional integration
  - To commission the "full chain" (upload, FTS3 transfers from/to, deletion)
  - To try multiple access protocols
- We are evaluating various migration strategies from DQ2 to Rucio
  - Per site, scope, frozen dataset, feature flags, etc.

CHEP Contribution

- ATLAS Replica Management in Rucio: Replication Rules and Subscriptions
- The ATLAS Data Management Software Engineering Process
- ATLAS DDM Workload Emulation
- Popularity Prediction Tool for ATLAS Distributed Data Management
- The DMLite Rucio Plugin: ATLAS data in a filesystem
- ATLAS DQ2 to Rucio renaming infrastructure

http://rucio.cern.ch