Rucio, the next-generation Data Management system in ATLAS

ICHEP 2014, Valencia

Cédric Serfon for the ATLAS collaboration

CERN, PH-ADP-CO

July 4th 2014

Outline

- Introduction
- Rucio concepts and architecture
- Scaling tests
- Deployment in production
- Opening to other experiments/communities
- Conclusion

Introduction

- The current Distributed Data Management (DDM) system of ATLAS called DQ2 (Don Quixote 2) successfully managed to serve ATLAS data during Run-1 :
  - 160 PB.
  - 640M files.
  - 130 grid sites.
  - 1000 users.
- But DQ2 will not scale for Run-2 :
  - Heavy operational burden.
  - Difficult to add new features and technologies.
  - Many lessons learned during Run-1.

$\rightarrow$ New DDM framework needed : Rucio.

Introduction

Don Quijote y Sancho Panza (Plaza de España de Madrid.)

Rucio main features

- Better handling of users, groups, activities (multiple replicas ownership, quotas...) than DQ2.
- Data discovery based on name and meta-data versus pattern/wilcard searches in DQ2.
- No dependencies on an external file catalog (LFC) : The Physical File Names (PFN) can be obtained from the Logical File Names (LFN) via a deterministic function.
- Whereas DQ2 supported only SRM to interact physically with the files, Rucio supports multiple protocols, e.g. WebDAV, xrootd, S3, posix, gridftp.
- Smarter and more automated data placement tools (rules, subscriptions).

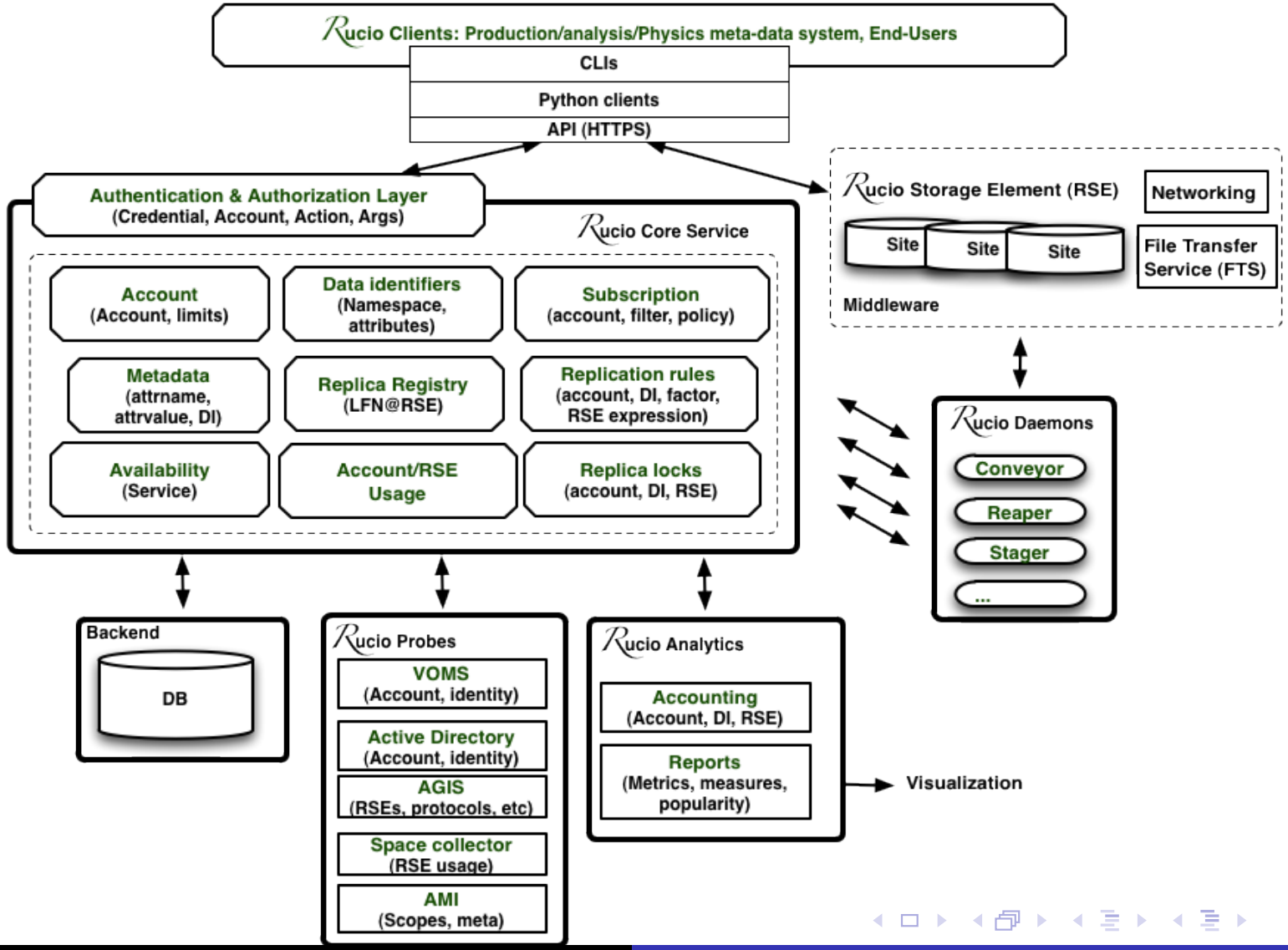
Rucio concepts

- Rucio account :
  - It can represent users (e.g. jdoe), groups (higgs), activities (tier0).
  - Quota, permissions tunable and associated to one account.
  - One can connect to a Rucio account using x509 certificate/proxy, kerberos, userpass.
- Rucio namespace :
  - 3 types of Data Identifiers (DIDs) : File, Datasets, Containers. Allows multiple hierarchy level for containers (only one level in DQ2).
  - All Data Identifier are identified by a scope and a name. A name is unique within a scope but can be used in other scopes (vs uniqness of the name in the whole DQ2 namespace).

Rucio concepts

- Rucio Storage Elements (RSE) :
  - Abstraction for storage end-point.
  - Can be grouped in various ways with tags (e.g. tier=1, cloud=DE).
- Replication rules :
  - Describe how a Data Identifier must be replicated on a list of Rucio Storage Elements.
  - e.g. : Make 2 replicas of dataset data12_8TeV:mydatasetname on tier=1&disk=1.
  - Rucio will create the minimum number of replicas to optimise storage space, minimise the number of transfers and automate data distribution.
- Subscriptions :
  - Replication policies based on Data Identifiers metadata, for Data Identifiers that will be produced in the future.
  - e.g. : Make 2 replicas of datasets with scope=data12_8TeV and datatype=AOD on tier=1&disk=1.

Design and architecture

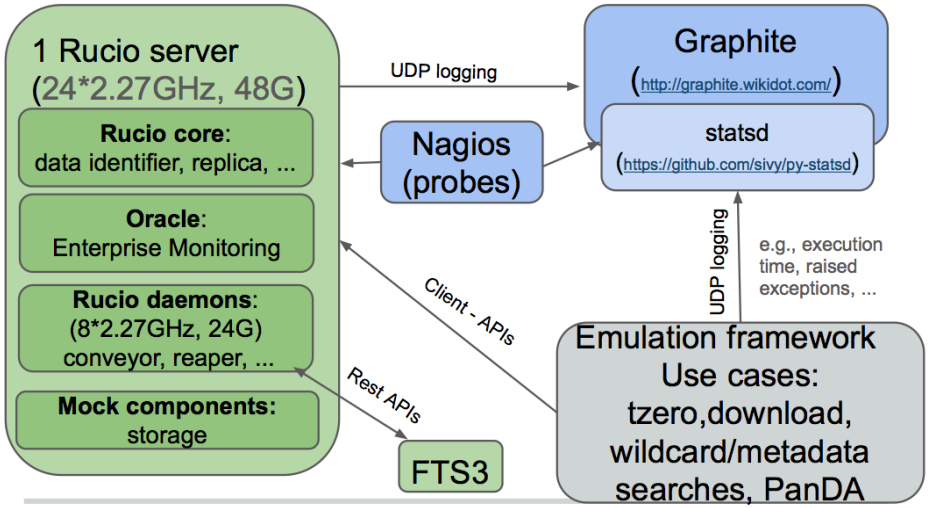


Design and architecture

- Rucio backend :
  - Based on Relational Database Management System (Oracle 11g).
  - Use of Object-Relational Mapper: SQLAlchemy.
  - Rucio supports Oracle, PostgreSQL, mysql...
- Rucio APIs:
  - RESTful APIs (HTTPS+json) : Can be used with curl or whatever HTTP client.
  - python CLI.
- Daemons : A list of lighweight, horizontally scalable agents :
  - Transmogrifier : Evaluates the subscriptions.
  - Judge : Evaluates the replication rules.
  - Conveyor : Submits and monitor the file transfers.
  - Reaper : Deletes expired replicas.
  - Undertaker : Deletes expired datasets.

Scaling test

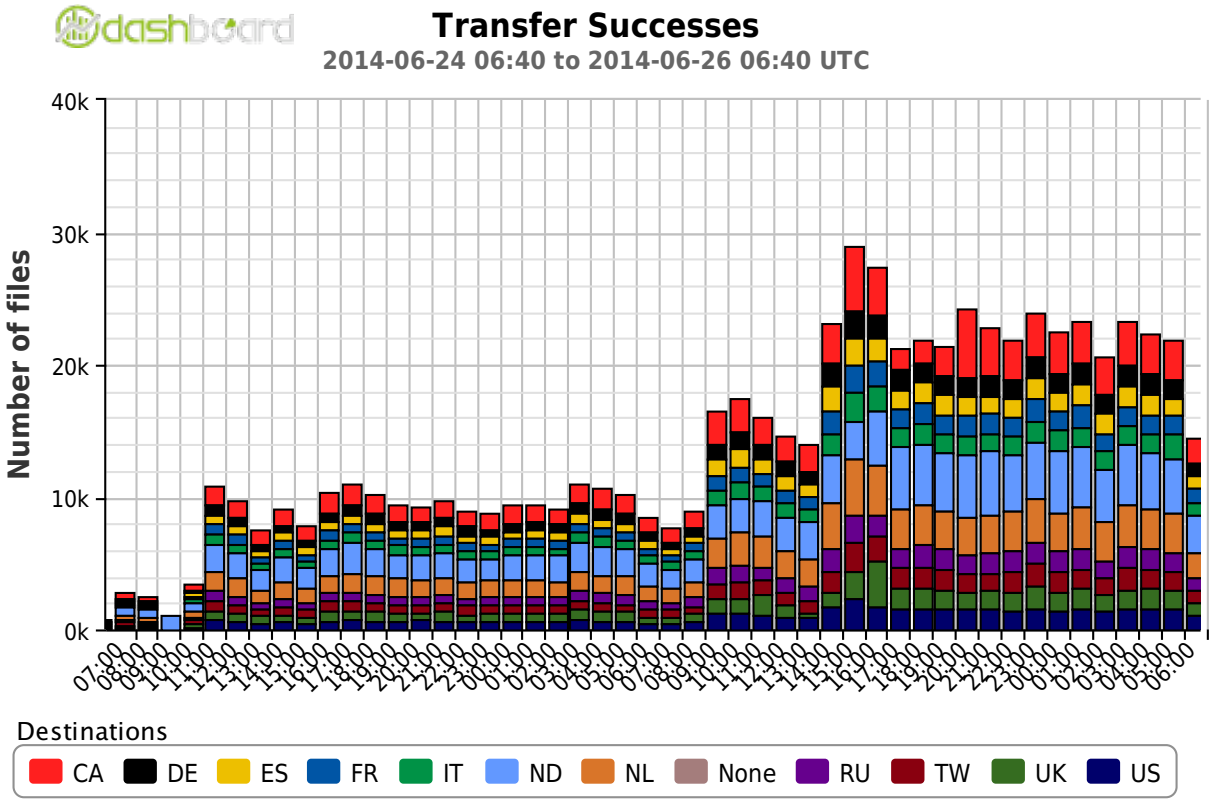
- An emulation framework has been developed and is running for more than 1 year:
  - Reproduces the main workflows of ATLAS.
  - Used to validate DB schema and new features.
  - Identify potential bottlenecks and concurrency issues as early as possible.
  - Explore system boundaries (tested up to 4 times the current load with 1B replicas).



- But this scaling tests only deals with fake files, fake sites.

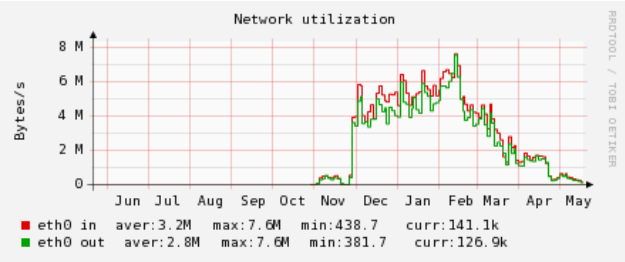
"Real life" scaling test

- Since 1.5 month, running a scaling test with real files, on real sites :
  - Generate datasets at CERN and export to T1s.
  - Started at 1000 files/day and ramping up to 1,000,000 files/day.

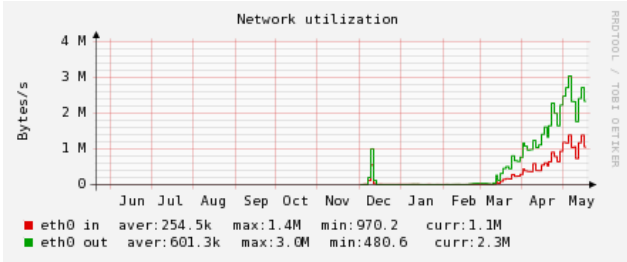


Deployment in production

- Deployment stategy steps :
  - From 1st January 2013 to February 2014 : Changing the path of all the files replicas (more than 300M files) on sites to follow the new deterministic convention : DONE
  - From mid-February 2014 to June 2014 : Migrating all the files replicas from LFC to Rucio : DONE



I/O on the LFC vs time



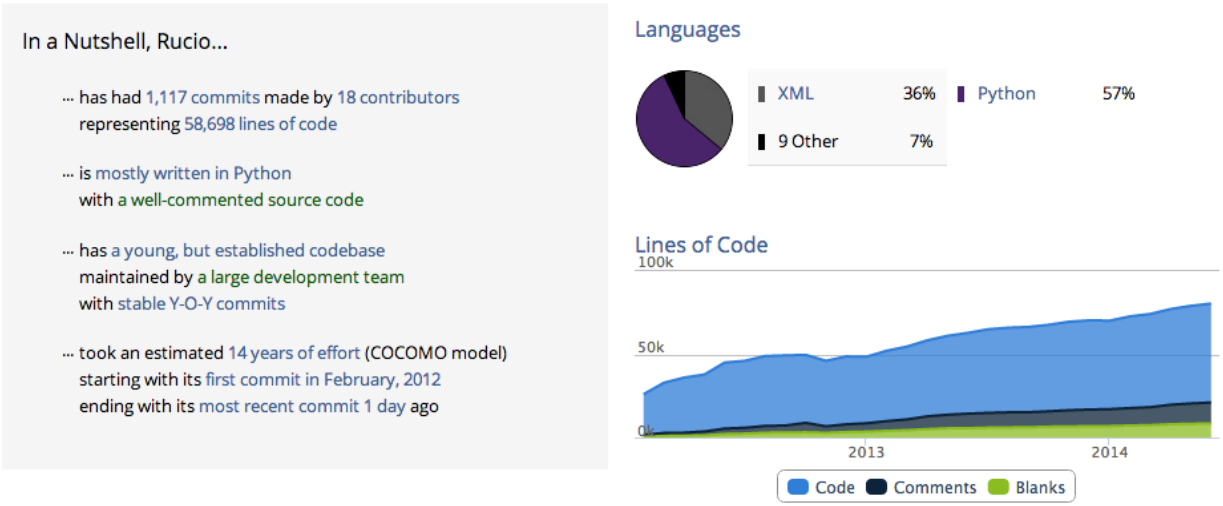
I/O on Rucio Servers vs time

  - Summer to Autumn 2014 : Migration of all datasets, containers from DQ2 to Rucio : BEING DONE

- All the changes were applied in a transparent way, without any disruption of ATLAS computing activities.

Opening to other experiments/communities

- Rucio is already used by other experiments/VO, e.g. : AMS.
  - If you are interested in using Rucio, contact `rucio-dev@cern.ch`
- Agile development with mandatory code reviews and prerequisite extensive unit and functional tests.



Statistics from Ohloh

- Rucio is an open-source project, if you want to contribute to it, your're welcome.

Conclusion

- Rucio has been developed to address the challenges of ATLAS Run-2.
- It has been successfully tested and validated on an integration testbed at high load (4 times the nominal) and in "real life".
- The service is now in production and we are completing the migration from DQ2 to Rucio.
- Rucio is also open to other experiments/communities. If you are interested to use it or to contribute :

http://rucio.cern.ch/

rucio-dev@cern.ch