

Rucio overview: Data access and caching

Mario.Lassnig@cern.ch

on behalf of ATLAS Distributed Computing

Data access

- Rucio provides full access paths to all available replicas of a file, takes care of
 - ... dataset contents and wildcards
 - ... priorities of site's protocol choices
 - ... priorities across WAN and LAN
 - ... client to storage distance (max throughput, avg. throughput, geographical, manual)
 - ... files in archives
 - ... URL signing (e.g., for clouds)
 - ... intermediate ROOT proxies where necessary
- Clients always know where all replicas are, does not query storage at runtime
- Exchange protocol of choice is cross-protocol *metalink*
- ROOT IO has been improved to understand cross-protocol metalink fallbacks
- Rucio-Clients / Ruciomover as the frontend

Dynamic data popularity and placement

- Efficient data access relies on good data placement
- File accesses are traced (including branches and events when possible)
- Workload system and network status are included

- Create additional transient replicas based on above metrics + state of queues
- Success rate measured as "transient replica used" at 65% to 78%
- Additional prediction model shows potential to increase success rate

- cf. Thomas Maier's CHEP talk

Volatile RSEs (== Caching RSEs)

- RSE is a definition of a storage endpoint (hostname, protocols, paths, ...)
- They can be marked as *volatile*, i.e., an external process can put data in and out
- Responsibility on external process to notify Rucio of storage updates
 - Directly calling the Rucio API
 - Placing messages in ActiveMQ and Rucio consumes
- Volatile replicas are outside the limits and quota enforcement

Upcoming

- Data access
 - US sites have developed LSMs (local site tools to move data) — integrate calling them transparently
- Dynamic data placement
 - Taking into account computing resources
- Volatile RSEs and replicas
 - Directly consume storage update events