

# Rucio Cache and other Caching questions

Doug Benjamin  
Duke University



# What is a Rucio Cache



Rucio 0.1.0\_rc6-1-g3437023-dev1371019856 documentation »

## Rucio Storage Element

A Rucio Storage Element (RSE) is a container for physical files. It is the smallest unit of storage space addressable within Rucio. It has a unique identifier and a set of meta attributes describing properties such as supported protocols, e.g., file, https, srm; host/port address; quality of service; storage type, e.g., disk, tape, ...; physical space properties, e.g., used, available, non-pledged; and geographical zone.

Rucio Storage Elements can be grouped in many logical ways, e.g., the UK RSEs, the Tier-1 RSEs, or the 'good' RSEs. One can reference groups of RSEs by metadata attributes or by explicit enumeration of RSEs.

RSE tags are expanded at transfer time to enumerate target sites. Post-facto changes to the sites in an RSE tag list will not affect currently replicated files.

A cache is storage service which keeps additional copies of files to reduce response time and bandwidth usage. In Rucio, a cache is an RSE, tagged as volatile. The control of the cache content is usually handled by an external process or applications (e.g. Panda) and not by Rucio. Thus, as Rucio doesn't control all file movements on these RSEs, the application populating the cache must register and unregister these file replicas in Rucio. The information about replica location on volatile RSEs can have a lifetime. Replicas registered on volatile RSEs are excluded from the Rucio replica management system (replication rules, quota, replication locks) described in the section Replica management. Explicit transfer requests can be made to Rucio in order to populate the cache.

previous | next | modules | index

Rucio Storage Element - tagged as volatile -



# Other details on Rucio Cache



- Rucio does not manage the data transfers to/from the Cache
  - Possible data sources : Datri (or its successor), FT3 , Federated storage (ie FAX)
- Goal to provide lightweight way for small sites (ie Tier 3's) to get data from their PanDA jobs
- Data servers need to tell Central Rucio Servers when files are added and removed
- These RSE's are non deterministic – do not need to have run RSE consistency



# Is Rucio Cache vaporware?



- Wen Guan has written an alpha version of Rucio Cache Client Code.
  - Still not in Rucio Client Code – needs a code review
- Data server side code being written/tested by DB
  - Focusing on xrootd FRM and gridftp server development
  - File information (operation – write/remove, name, size, checksum) is “aggregated” on the data server and sent to Rucio Servers
  - Each xrootd FRM write process and ultimately gridftp file sends information to zmq broker running on the data server.

<http://zeromq.org>

<http://zeromq.org/bindings:python>

- Currently merging Rucio Cache Code into zmq broker code
- Working on gridftp “plugin” based on posix dsi plugin from Wei Yang



# Proposal for Read/Write “Depot” Distributed Read/write storage for users



- Dropbox like –
- Users manage the contents themselves
- Users decide how their files are “organized” (ie how they organize finding their files within this storage)
- Accessible from anywhere their analysis jobs might run (Campus clusters, Beyond Pledge resources , Cloud resources etc)
- Requirements:
  - Secure – User authentication – Can we use existing user authentication mechanisms (x509 proxies)
  - User Quotas
  - Distributed – for speed and resiliency
- Ceph might be a possible solution



# Questions for discussion



- **What about http federation (webDaV)?**
- SLAC has a T3 proof cluster with 740TB usage space (and 360 cores). They are extremely busy during the last 2 weeks. There are other big T3s. **How do they get data?**
- **How do we provide users a Read/Write Depot?**