Authorization for XRootD and GridFTP gateways at RAL

Ian Johnson

Scientific Computing Department

STFC RAL

Recap – RAL Storage Service

- RAL are moving to replace CASTOR with a disk-only Storage Element using Ceph for the LHC VOs
- Need to provide GridFTP and XRootD gateway access, as not all LHC VO workflows can yet make use of S3/Swift
- Need to provide 5PB of usable storage to LHC VOs by April 2017

Echo Project at RAL

- Will provide ~ 60 disk servers, providing around 10PB raw storage space to Ceph pools
- Using 8+3 Erasure Coding, get about 7PB usable space
- Three gateway nodes, each running:
  - Rados GW (S3 and Swift access)
  - XRootD
  - GridFTP

Authorization Goal – Restrict Access to Ceph Storage

- Place access controls in XRootD and GridFTP gateways
  - Filter incoming requests before they hit the Ceph cluster (prevent unwanted traffic to the cluster)
  - Check that only authorized users can access the relevant Ceph pool according to their privileges
- XRootD and GridFTP gateways should use the same rule file to define access – AuthDB is the obvious choice

AuthDB and Pool Naming

- RAL plan to use a single Ceph pool for Atlas VO
- Currently, need to name the pool with a leading slash for AuthDB compatibility (otherwise, XrdAcc framework in XRootD tries to look up an AuthDB ‘template’)
- Pool for Atlas VO is called ‘/atlas’

Data and Scratch Areas in Pool

- Define two areas in the pool, distinguished by the leading component of the object path
  - ‘data’ only writable by ‘atlasprod’ (Production user from grid-mapfile)
  - ‘scratch’ writable by both ‘atlasprod’ and ‘atlasuser’ (Analysis user from grid-mapfile)

Ceph Path in AuthDB

- AuthDB paths (path prefixes) for Atlas pool are:
  - /atlas:data/ and /atlas:scratch/
- Privileges for atlasprod and atlasuser defined in AuthDB rules as:

  u atlasprod /atlas:/data/ a /atlas:scratch/ a

  u atlasuser /atlas:data/ r /atlas:scratch/ a

  (‘a’ = Read-Write access, ‘r’ Read-Only access)

Implementation

- In XRootD, enable XrdAcc framework:
  - ofs.authorize
  - acc.authdb /opt/xrd/etc/Authfile
- In GridFTP, call a C routine to check the tuple (VO user, operation, path) against rules in the AuthDB file
  - For STOR, DELE, and MKD commands, set operation to ‘wr’ (Write)
  - For RETR, MLST, and CKSM, operation is ‘rd’ (Read)

Testing

- The standalone ‘xrdacctest’ program tests candidate requests against rules in the AuthDB file
  - Small test suite (9 combinations) to exercise this
- End-to-end testing:
  - Using xrdcp and globus-url-copy commands, I’ve shown that AuthDB rules are processed as expected, and that permission is allowed/denied as appropriate

Summary

- XRootD and GridFTP gateways at RAL use AuthDB format to only allow the appropriate level of access, and only for authorized users
- GridFTP gateway uses a small C routine to parse the AuthDB file and check the requested operation against the rules.