XRootD Future Feature Plans

FTS/XRootD Workshop

September 9-13, 2023

Andrew Hanushevsky, SLAC
http://xrootd.org

SLAC

SLAC
NATIONAL ACCELERATOR LABORATORY

Future Features 6.0.0 - I

- Rucio aware dataset backup plug-in
  - libXrdOssArc.so (contributed by Vera Rubin Observatory)
  - Allows **xrootd** to become an archiver/restorer
    - Archive
      - Dataset tagged with meta-data to request backup
      - Archiver combines all dataset files into a single zip file
      - Stages zip file for backup to tape or other media
      - Optionally, registers zip file in Rucio
    - Restore
      - Client copies any zip file member out to restore
        - Can also copy out full zip file

Future Features 6.0.0 - II

- Improved curl error reporting to client
  - Motivation: WLCG/DOMA BDT WebDAV Error Message Improvement Project
    - https://twiki.cern.ch/twiki/bin/view/LCG/WebdavErrorImprovement
  - This is an **XRootD** initiated project
    - Make WebDav errors consistent across providers
  - Slow progress but progress nonetheless

Future Features 6.0.0 - III

- Allow timeouts > 65535 seconds
  - Motivation: Copying large files and specifying reasonable values that don't wrap
  - This is a substantial API change and care is being taken to provide compatibility.
- Related issue is number of copied files
  - Use of uint16_t limited it to 65535
    - Will change to uint64_t

Future Features 6.0.0 - IV

- Make XRD_LOCALMETALINKFILE the default
  - Motivation: Allows ROOT users to transparently open remote files on disk
    - File must have a ".meta4" or ".metalink" suffix

Future Features 6.0.0 - V

- Implement un-features
  - Motivation: Fed up and can't take it anymore
  - Drop python2 support
    - 'nuff said
  - Drop CentOS 7 support
    - Support will drop as of 1/1/2025

Future Features 6.0.0 - VI

- Add additional context to errors
  - Motivation: Proxy and caching server errors are often mysterious or misleading
  - Originating error messages fed upstream
    - Downstream plug-ins indicate context capability
    - If enabled, upstream plug-ins reap the context
      - The context of the error message is reported to the client
        - Will likely require a couple of iterations to get right
    - This breaks ABI
      - Plug-in re-compilation will be needed

Future Features 6.0.0 - VII

- Ease summary reporting for plug-ins
- Motivation: Make monitoring easier
  - Currently, plug-ins can use gStream to report
    - Used for relatively low frequency periodic reports
  - New interface to register statistical counters
    - Counters included in summary reporting
      - xrdfs query stats *what*
      - https://xrootd.slac.stanford.edu/doc/dev57/xrd_monitoring.htm#_Toc138968495
    - Will have xml (default) and json options
      - Exploring mechanisms for backward compatibility

Beyond 6.0.0 - I

- Use kernel level TLS (kTLS) when available
  - Motivation: Increased performance
  - Requires OpenSSL >= 3.0.1 & Linux >= 4.13
    - Combination available in RH9 & Alma9
      - OpenSSL >= 3.0.1 (3.2.0 recommended) & Linux >= 5.4.164
    - However, not automatically enabled
      - OpenSSL must be rebuilt with enable-ktls or install 3.2
        - Always distributed that way for Debian >= 12
      - Linux ktls must be enabled via sudo modprobe tls
    - So, some operational roadblocks for now

Beyond 6.0.0 - II

- Use io_uring (liburing.so) for asyc I/O
  - Motivation: Improved async performance
  - Available in RH9 / Alma9
    - Phased in approach
      - Disk I/O followed by Network I/O in server
      - Client will likely use it for selective Network I/O first
        - Only benefits **Xcache** and Proxy servers
          - Epoll() is better than io_uring for < 1000 sockets
            - https://www.alibabacloud.com/blog/io-uring-vs--epoll-which-is-better-in-network-programming_599544

Beyond 6.0.0 - III

- Use RDMA for network I/O when needed
- Motivation: Better integration with HPC's
  - Implementation will be based on libfabric
    - OpenFabrics Interfaces (OFI) Working Group
      - Available in practically every distribution
  - This is a significant project with high impact

Beyond 6.0.0 - IV

- Increase nodes per cmsd redirector
- Motivation: ease large cluster deployment
  - 64 node limit to increase to 128/redirector
    - Do we need more???
  - We have a prototype but need a volunteer
    - To test in an actual environment

Beyond 6.0.0 - V

- Implement client affinity
  - Motivation: sites that use batch node **xrootd**'s
    - This essentially redirects a client to the local **xrootd**
      - Only if the batch node has a working **xrootd**
    - This is largely relevant to DFS deployments

Conclusion

- **XRootD** future looks bright
  - Novel development is happening at a rapid pace
    - Framework remains relevant
      - The tagline – "It's **XRootD** Inside!" applies
- Our core partners
  - CERN SLAC UCSD
- Community & funding partners (*not a complete list*)
  - OSG WLCG ATLAS EXPERIMENT ALICE CMS LSST dCache GSI NSF

Funding from US Department of Energy contract DE-AC02-76SF00515 with Stanford University