
CERN

Michal Simon

XRootD development update

Outline

- XRootD Introduction
- New features
- XRootD Client overview
- Plans for 2018

XRootD Introduction

- Backbone of EOS
- Support / development in XrdCl
- Support / bug fixing in XrdSec
- Packaging and release management

New features: socket inheritance

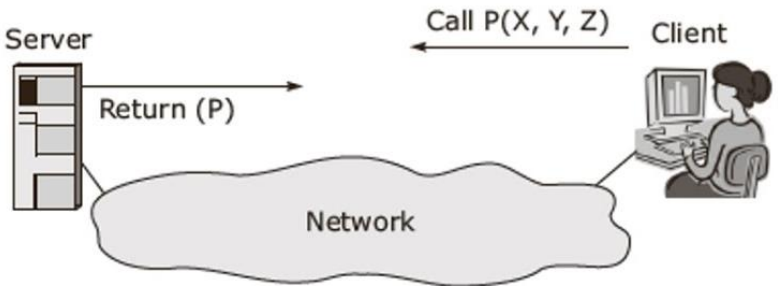
- Motivation:
  - For security reasons XRootD has to run as an unprivileged user and as a result has no access to low port ranges
  - As a result XrdHttp cannot run on port 80, which makes it strange HTTP server

New features: socket inheritance

- On systemd platforms the problem can be solved by socket activation / inheritance
- Systemd can reserve the socket, bind it, and allow the daemon to receive it via FD inheritance
- In addition, we can now benefit from socket activation

New features: XrdSsi v2

- Scalable Service Interface is a multi-threaded XRootD plug-in that implements a request-response framework
- XRootD RPC alternative
- Actions that may involve delay are asynchronous
- Used in CTA



New features: Caching Proxy v2

- Solution that allows to keep hot data near computational resources
- The existing proxy server implementation (memory caching) has been leveraged to provide a Disk Caching Proxy
- Full async IO support in Proxy Server
- The caching logic is also available through the posix API, making local caching possible

New features: Local files (file://)

- XRootD client supports now natively local files
- XRootD server allows full URL rewriting on redirect
- Redirections to local files:
  - use case: a distributed file system mounted on the client side
  - used at GSI

New features: Extreme copy

- Segmented file transfer (bit-torrent like)
- Supported through xrdcp and XrdCl::CopyProcess API
- User specifies only the number of sources
- The data servers are determined using deep locate or a metalling file

New features: Extreme copy

- The file is being partitioned into chunks
  - not too small so the source benefits from sequential reads
  - not too big so the destination is not overwhelmed with large sparse files
  - tunable through an env var (XRD_XCPBLOCKSIZE)
- Fast sources are allowed to steal work from slow ones

Client overview: bugfixes

- Request response mismatch
  - Client gets redirected to a data server and the requested file is not found so it retries at the load balancer
  - Bug: when redirected back to the manager, the client was reusing SID assigned during communication with data server
  - This validates the invariant that SID is unique per client - server connection
  - Fixed in 4.6.1

Client enhancements: mitigate deadlock risk

- Virtual redirections are handled in the thread-pool
- All error (socket / connection) and timeout handlers are delegated to the thread-pool
- In case of stateful operations user callback is not executed in the context of the XrdCl::FileStateHandler

Client enhancements: Nagle algorithm

- Write request header and body with single writev
- Disable TCP Nagle algorithm by default
- Tunable through an env var (XRD_NODELAY)

Client enhancements: IPv6/IPv4 retry

- By policy client first tries IPv6 and then IPv4 (user can change this with XRD_PREFERIPV4)
- Connection Window is now applied per IP address
- Posix connect errors are no longer considered as fatal (client moves to the next address)

Packaging enhancements

- On systemd platforms content of /var/run is managed by tmpfiles.d
- Python3 bindings
- Packaging for Debian (thanks to Jozsef)

Packaging overview: log rotate

- By default XRootD uses the default Linux log rotate mechanism
- In the XRootD config it is also possible to enable XRootD log rotate
- Now when XRootD log rotate is on it creates a lock file so there is no interference between the two log rotate mechanisms

Enhancements: miscellaneous

- xrdfs : recursive listing
- xrdfs : by default ls outputs only unique files
- XrdCl::File : posix like writev
- Server authorization : include X509 org and role and composite tests
- cmsd : Non-blocking message path

Build infrastructure

- Moved to GitLab CI @ CERN (thanks to Jozsef)

- C++0x/C++11 enabled by default
- Bi-weekly ‘experimental builds’, with yum friendly versioning

Plans for 2018

- XrdCl::File::VecorWrite : write at different offsets with one request
  - Potentially we could also support writing to different files within one request
- Implement extended attributes
  - Allow to set/get multiple attributes with one request

Plans for 2018

- Extended stat information (uid, gid, atime, mtime, mode, etc.)
- Support for opening files in append mode
- Http support in XRootD client:
  - Plugin or
  - New transport layer

Plans: SSL/TLS tunneling

- Separate port for encrypted traffic
- Openssl async API
- Redirections from unencrypted to encrypted
  - Single authentication
- Could be useful for CERN box

Plans: full ZIP support

- Appending files to ZIP archive
- Checksumming
- Compression

Plans: partial response handling

- Invoke user callback for partial results (with status ‘suPartial’)
- Use cases:
  - xrdcp streaming copy : request the whole file at the begging and handle the incoming chunks
  - xrdfs ls : listing of big directories (e.g. in case of Ceph)

Plans: bundled requests

- Provide an interface for bundled requests
- One response and handler per bundle
- Use cases:
  - Open + Read
  - Write + Close
  - Open + Read + Close
  - Open + Write + Close (problematic because of redirects)

Useful links

- http://xrootd.org
- https://github.com/xrootd/xrootd.git
- http://storage-ci.web.cern.ch/storage-ci/xrootd/experimental/
- xrootd-dev@slac.stanford.edu

Questions?