



XRootD

Workshop Summary and Future Plans

FTS/XRootD Joint Workshop at Abingdon, UK

- ▶ 2nd edition of the FTS/XRootD Joint Workshop
 - <https://indico.cern.ch/event/1386888/overview>
- ▶ Held at Cosener's House, Abingdon UK
 - FTS: 9–11 September 2024
 - XRootD: 11–13 September 2024
- ▶ ~40 Attendees on site (+10 Online)
- ▶ 14 FTS presentations (11 Speakers)
- ▶ 25 XRootD presentations (18 Speakers)
- ▶ Some time allocated for technical discussions
 - Good feedback, likely to continue
- ▶ Thanks to the local STFC/RAL organizers!
- ▶ Will cover XRootD part in this presentation



Catching Up With XRootD

FTS/XRootD Workshop
September 9-13, 2023

Andrew Hanushevsky, SLAC
<http://xrootd.org>



SLAC



SLAC

What's Happened Since 2023

- ‡ 27-March-23 **XRootD Workshop @ JSI**
- ‡ 08-May-23 Patch Release 5.5.5
- ‡ **30-June-23 Feature Release 5.6.0**
- ‡ 11-July-23 Patch Release 5.6.1
- ‡ 15-September-23 Patch Release 5.6.2
- ‡ 27-October-23 Patch Release 5.6.3
- ‡ 11-December-23 Patch Release 5.6.4
- ‡ 22-January-24 Patch Release 5.6.5
- ‡ 25-January-24 Patch Release 5.6.6 (Oops)
- ‡ 06-February-24 Patch Release 5.6.7
- ‡ 23-February-24 Patch Release 5.6.8
- ‡ 08-March-24 Patch Release 5.6.9
- ‡ **01-July-24 Feature Release 5.7.0**
- ‡ **04-July-24 Patch Release 5.7.1**
- ‡ 09-September-24 **XRootD Workshop @ RAL**

The largest number of patch releases for a feature release

XRootD Workshop @ STFC UK

September 9-13 2024

3



Performance Features 5.6.0 @ 06-30-23

- ‡ Allow **Xcache** file evictions
 - ‡ Can be done using the **xrdprep** command
 - ‡ `xrdprep -E host[:port] path [...]`
 - ‡ Can be done using **xrdfs** command
 - ‡ `xrdfs host[:port] {evict | fevict} path`
 - ‡ `evict` means evict only if file is not in use
 - ‡ `fevict` means forced eviction (i.e. evict no matter what)
 - ‡ All clients using the file get an I/O error
- ‡ Allow **Xcache** origin to be a locally mounted filesystem
 - ‡ `pss.origin path`
 - ‡ This allows fronting file systems with a smarter cache
 - ‡ E.g. Lustre when workload uses small reads.

XRootD Workshop @ STFC UK

September 9-13 2024

4



Security Features 5.6.0 @ 06-30-23

- ‡ **gsi** option to display DN when it differs from entity name
 - ‡ `-showdn:{true|false}`
 - ‡ `false` is the default
- ‡ Allow **Xcache** origin to be a locally mounted file system
 - ‡ `pss.origin path`
 - ‡ This allows fronting file systems with a smarter cache
 - ‡ E.g. Lustre when workload uses small reads.
- ‡ Implement ability to have the token username as a separate claim
 - ‡ This allows for identity tokens
- ‡ Use SHA-256 for signatures, and message digest algorithm
 - ‡ This is now the minimum required for any new OS

XRootD Workshop @ STFC UK

September 9-13 2024

6



Other Features 5.6.0 @ 06-30-23

- ‡ Move to CMake 3.16
- ‡ Support `musl` libc
 - ‡ Rich Felker library released under MIT license
 - ‡ Used by Alpine Linux, Dragora 3, and optionally Gentoo Linux
- ‡ General modernization of build system
 - ‡ More from our ReleaseManager later
- ‡ Better support for creating python binary wheels
 - ‡ More from our ReleaseManager later
- ‡ Improved HTTP protocol conformance
 - ‡ Supply caching object information
 - ‡ Accept-Ranges in HEAD response
 - ‡ Enabling trailer information for better error handling

XRootD Workshop @ STFC UK

September 9-13 2024

11



Catching Up With XRootD

FTS/XRootD Workshop
September 9-13, 2023

Andrew Hanushevsky, SLAC
<http://xrootd.org>



SLAC



SLAC

Performance Features 5.7.0 @ 07-01-24

- ❖ **cmsd** load balancing algorithm with randomized affinity
 - This avoids creating hot spots in DFS type environments
 - Learn about it in Jyothish Thomas' talk later in the week
- ❖ Implement the **kXR_seqio** open option for sequential I/O
 - FS plug-in can use `fadvise()` to tell kernel to optimize for `xrdcp`
 - Actual implementation awaiting code contribution
- ❖ Avoid some repeated calls of `EVP_PKEY_check`
 - Avoid `gsi` performance degradation in OpenSSL 3.0
- ❖ Option to force the destination IP address for HTTP-TPC
 - **tpc.fixed_route** {`true` | `false`}
 - True uses same interface that requested the TPC
 - Used by SENSE project for controlling net throughput

XRootD Workshop @ STFC UK

September 9-13 2024

12



Security Features 5.7.0 @ 07-01-24

- ❖ Update min/default RSA bits to 2048
 - This is the currently accepted minimum
 - It will likely increase as time goes on
- ❖ Option to allow for http tpc unrestricted redirection
 - **http.auth tpc fcreds**
 - Thus allows curl to forward credentials upon redirect
 - https://xrootd.slac.stanford.edu/doc/dev57/xrd_config.htm#_Toc171719991
- ❖ Enable ability to have token groups as a separate claim
 - Feature used by certain token issuers
 - This is configured via the token issuer profile
- ❖ HTTP external handlers can now be loaded without TLS
 - **http.external_handler name [+notls] path [token]**
 - https://xrootd.slac.stanford.edu/doc/dev57/xrd_config.htm#_Toc171720007

XRootD Workshop @ STFC UK

September 9-13 2024

13



Enhancements 5.7.1 @ 09-04-24

- ❖ Accommodate API endpoints
 - New directive: **pss.hostarena *text***
 - Inserts *text* between origin URL and client path
 - Assume **pss.origin *xrootd://myhost/***
 - Assume URL: ***xrootd://proxysrv//a/b/c***
 - Result: ***xrootd://myhost/text/a/b/c***
 - Used to access the requested data
 - Thus limiting the scope of client URLs
 - The *text* is exported via `envvar XRDXROOTD_PROXYARENA`
 - Composite origin exported as `XRDXROOTD_PROXYURL`

XRootD Workshop @ STFC UK

September 9-13 2024

18



Enhancements 5.7.1 @ 09-04-24

- ❖ Implement Read/Only redirector option
 - Motivation: Simplify CMS AAA redirection when R/W sites join a redirector
 - Standard solution requires AAA sites to use the **globalro** export option
 - Not easily enforced and is an XRootD-only option
 - New redirector directive centralizes enforcement
 - **cms.mode** {`r/o` | `readonly` | `r/w` | `readwrite`}
 - Works for any XRootD protocol storage provider
 - Will be documented in R6 reference

XRootD Workshop @ STFC UK

September 9-13 2024

19





XRootD

Catching Up with XRootD (Part 2)

G. Amadio

FTS/XRootD Workshop 2024

9–13 Sep 2024

XRootD GitHub Statistics

- ▶ Releases:
 - From 5.5.4 to 5.7.1 – 14 releases in total
- ▶ Pull Requests:
 - 204 pull requests since Jan 2023, 4 currently open
- ▶ Issues:
 - Total: 89 open, 1239 closed issues
 - Stats for 2023/2024: 254 issues opened, 194 closed (~78%)
 - Many issues are really just questions
 - Average of ~2 weeks to close
- ▶ Contributors:
 - From 2 to 12 contributors each month (2023 – 2024)
 - Average around 5~10 contributors per release
 - Total contributors: ~100
- ▶ Most contributions by CERN, SLAC, UCSD

| Domains | Total (%) |
|-------------------|---------------|
| cern.ch | 6578 (51.58%) |
| slac.stanford.edu | 4643 (36.40%) |
| ucsd.edu | 740 (5.80%) |
| cse.unl.edu | 285 (2.23%) |
| physics.uu.se | 123 (0.96%) |

2

XRootD 5.7.x Highlights

- ▶ XRootD 5.7.0
 - Moved baseline C++ standard to C++17
 - Updated min/default RSA bits to 2048
 - Restrict renegotiation for TLSv1.2 and earlier
 - Redact authz tokens from server logs
 - HTTP header parsing now case insensitive for better compatibility
 - Make more client errors recoverable to allow retry in more situations
 - Several improvements in XrdScitokens plugin (new auth strategies)
 - Reintegrated XrdCeph into main repository, now uses -DENABLE_CEPH=ON option
 - Better performance for GSI authentication by avoiding duplicate work (needed on Alma 9)
 - Enable SSL_OP_IGNORE_UNEXPECTED_EOF option if available, increase default timeouts
 - XRootD now includes a sample shell script for third-party copy (TPC) configuration
 - Erasure coding plugin now enabled by default and using OS-provided isa-l library
 - Completed migration of test suite to GoogleTest
 - Updated Doxygen documentation

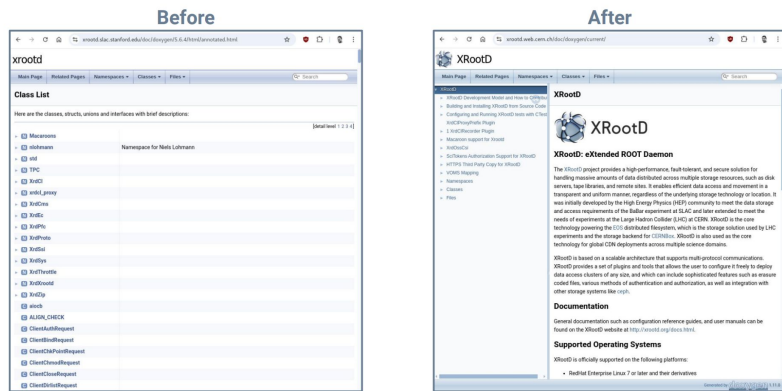
6

XRootD 5.7.x Highlights

- ▶ XRootD 5.7.1
 - Allow `cconfig` to write out combined config file
 - Harden systemd service units for better security
 - Fix memory leaks in Python bindings
 - XRootD security policy described in `SECURITY.md`
 - Ensure correct certificate is used when passed via cgi with `xrd.gsiusrproxy=...`
 - Finished work on redacting token information from logs (client-side)
 - Increased test coverage with more server/client setups
 - Enabled CodeQL code scanning tool on GitHub
 - Introduced abi-tracker reports to ensure full ABI backward compatibility
- ▶ XRootD on Alma 9 / RHEL 9
 - Need to pay attention to ulimit settings to avoid running out of memory
 - HTTP 1.1 vs HTTP 2.0 incompatibilities (case-insensitive headers from curl)
 - Bad performance from OpenSSL 3.0 affects GSI + VOMS extraction configurations

11

XRootD Doxygen Documentation



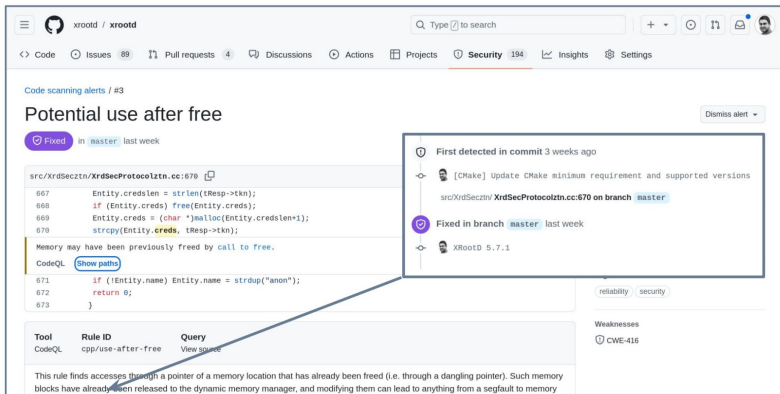
8

XRootD Python Bindings Improvements

- ▶ Improved Python 3.x support
 - No more need to use raw strings, fixed file iteration by line, bug fixes
- ▶ Python 2.x support to be dropped along with CentOS 7 support
- ▶ Rewritten Python packaging
 - Integrated into CMake build
 - Compliant with PEP 517
 - Customizable builds (CXXFLAGS)
- ▶ Tests converted to Python 3.x
 - TODO: Integrate running Python tests with CTest (partially done)
- ▶ Machinery to build Python wheels is ready, need to find solution for linking OpenSSL

12

Code Scanning with CodeQL on GitHub



16

XRootD ABI Tracker

ABI Tracker (XRootD)

API/ABI changes review

| Version | Date | Senase | Change Log | Backward Compatibility | Added Symbols | Removed Symbols | Headers Bk/FF | Package Bk/FF |
|---------|------------------|--------|------------|------------------------|--|-----------------|---------------|---------------|
| master | 2024-09-01 17:01 | 1/2/3 | changeLog | 100% | 0 | 0 | 1 | N/A |
| 5.7.1 | 2024-09-01 | 1/2/3 | changeLog | 99.97% | 4 new | 1 removed | 2 | N/A |
| 5.7.0 | 2024-07-01 | 1/2/3 | changeLog | 99.49% | added 1 object | 302 new | 1 removed | 20 |
| 5.6.9 | 2024-03-08 | 2/3 | changeLog | 99.72% | 0 | 0 | 22 | N/A |
| 5.6.8 | 2023-06-30 | 2/3 | changeLog | 96.56% | 7 new | 0 | 23 | N/A |
| 5.5.5 | 2023-05-07 | 2/3 | changeLog | 57.92% | changed 10 new added 6 objects removed 2 objects | 1212 new | 177 removed | 113 |
| 4.12.0 | 2022-03-21 | 1/2 | changeLog | N/A | N/A | N/A | N/A | N/A |

Maintained by [Guilherme Amadio](#). Last updated on Wed Sep 4 12:06 2024.
Generated by [ABI Tracker](#), [ABICC](#), and [ABIDumper](#) tools.

Problems with Data Types, High Severity 1

EBL308

[1] class XRootDs 1

| Change | Effect |
|---|---|
| 1) Size of this class has been increased from 704 bytes to 720 bytes. | 1) An object of this class can be allocated by the applications and old code will be validated at the compile time. Call at any expected constructor will break the memory of neighboring objects on the stack or heap. |
| [1] affected symbols: 31 (0.2%) | 2) The memory layout and size of subclasses will be changed. |

Problems with Data Types, Medium Severity 3

EBL308

[1] class XRootDs 2

| Change | Effect |
|--|--|
| 1) Field Mask has been added at the middle position of this structural type | 1) One of the inclusion type has been changed |
| 2) Layout of structure fields has been changed and therefore fields at higher positions of the structure definition may be incorrectly accessed by applications. | 2) Layout of structure fields has been changed and therefore fields at higher positions of the structure definition may be incorrectly accessed by applications. |
| 3) Field Mask has been added at the middle position of this structural type | 3) One of the inclusion type has been changed |
| 2) Layout of structure fields has been changed and therefore fields at higher positions of the structure definition may be incorrectly accessed by applications. | 2) Layout of structure fields has been changed and therefore fields at higher positions of the structure definition may be incorrectly accessed by applications. |

[1] affected symbols: 31 (0.2%)

Problems with Data Types, Low Severity 3

EBL308

[1] class XRootDs 1

[1] class XRootSecEntity 1

[1] class XRootSecEntity 1

[1] class XRootSecEntity 1

[1] class XRootSecEntity 1

17

XRootD Future Feature Plans

FTS/XRootD Workshop
September 9-13, 2023

Andrew Hanushevsky, SLAC
<http://xrootd.org>



SLAC



SLAC

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 - 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

XRootD Future Feature Plans

FTS/XRootD Workshop
September 9-13, 2023

Andrew Hanushevsky, SLAC
<http://xrootd.org>



SLAC



SLAC

Beyond 6.0.0 - I

- ‡ Use kernel level TLS (kTLS) when available
 - Motivation: Increased performance
 - Requires OpenSSL \geq 3.0.1 & Linux \geq 4.13
 - Combination available in RH9 & Alma9
 - OpenSSL \geq 3.0.1 (3.2.0 recommended) & Linux \geq 5.4.164
 - However, not automatically enabled
 - OpenSSL must be rebuilt with enable-ktls or install 3.2
 - Always distributed that way for Debian \geq 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



XRootD

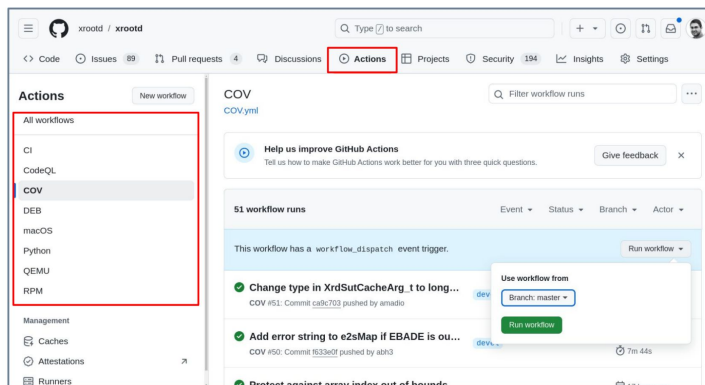
Contributing to XRootD

G. Amadio

FTS/XRootD Workshop 2024

9-13 Sep 2024

GitHub Actions Overview



How to use test.cmake to run the XRootD test suite

```

# Simplest case, configure, build, and test
$ ctest -VV -S test.cmake

# Build in Debug mode
$ ctest -VV -C Debug -S test.cmake

# Run static analysis with clang-tidy
$ ctest -VV -DSTATIC_ANALYSIS=1 -S test.cmake

# Perform memory checking with valgrind on all tests
$ ctest -VV -DMEMCHECK=1 -S test.cmake

# Build in Debug mode and create coverage report
$ ctest -VV -C Debug -DCOVERAGE=1 -S test.cmake

# Use clang compiler to build
$ env CC=clang CXX=clang++ ctest -VV -S test.cmake

# Build in Release mode and submit test results to CDash
$ ctest -VV -C Release -DCDASH=1 -S test.cmake

# Use custom configuration for CMake (build, but don't test)
$ env CMAKE_ARGS="-DENABLE_TESTS=0" ctest -VV -S test.cmake

Please see docs/TESTING.md on GitHub for more information.

```

```

~/XRootD $ ctest -VV -S test.cmake
-- Using cmake cache file cmake.cache
Run dashboard with mode Experimental
Source directory: /home/amadio@rc/xrootd/build
Build directory: /home/amadio@rc/xrootd/build
Group: Experiments1
Running CMake configuration file: /home/amadio@rc/xrootd/CTestConfig.cmake
Site: zen001-001-001
Build name: zen001 Linux GCC 14.2.0 1000000000
Use Experiments1 tag: 20240904-143
Updating the repository: /home/amadio@rc/xrootd
Use Git repository type
Git revision of repository is: 23761f4eb9d920885249a2d074e68b64628
New revision of repository is: 23761f4eb9d920885249a2d074e68b64628
Fetching version information (one per revision)

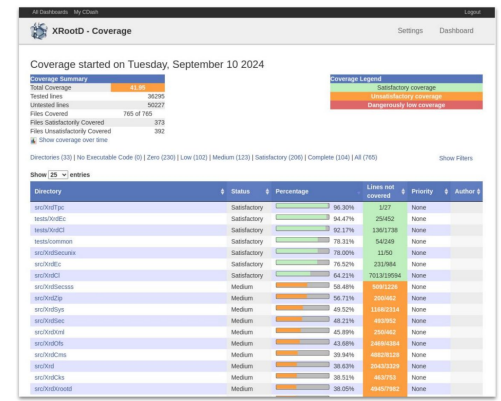
Configure project
  Each . represents 1024 bytes of output
Build project
  Each %0x01 represents 1024 bytes of output.
  Size of output: 57K
  Compiler settings
  B Compiler settings
test project: /home/amadio@rc/xrootd/build
Start 1: XRootD:Start
Start 2: XRootD:URLTest_LocalURLs
Start 3: XRootD:URLTest_InvalURLs
Start 4: XRootD:PollerTest_FunctionTest
Start 5: XRootD:PollerTest_PropertyTest
Start 6: XRootD:PollerTest_TestManagerTest
Start 7: XRootD:URLTest_TestManagerTest
Start 8: XRootD:URLTest_PropertyListTest
Start 9: XRootD:URLTest_PropertyListTest
Start 10: XRootD:URLTest_PropertyListTest
1/120 Test #15: XRootD:Start ..... Passed 0.02 sec
2/120 Test #1: XRootD:URLTest_LocalURLs ..... Passed 0.03 sec
3/120 Test #3: XRootD:URLTest_InvalURLs ..... Passed 0.02 sec
119/120 Test #10: XRootD:URLTest_PropertyListTest ..... Passed 0.02 sec
120/120 Test #120: XRootD:Submitter:stop ..... Passed 0.04 sec
All tests passed. 0 tests failed out of 120
Total Test Time (real) = 140.60 sec

```

XRootD on GitHub

- ▶ New README in Markdown
- ▶ GitHub Actions
 - Continuous Integration
 - RPM / DEB Packages
 - Python wheels
 - QEMU cross-platform
- ▶ CTest script
- ▶ CDash Dashboard

<https://my.cdash.org/index.php?project=XRootD>



Pelican and the OSDF: An Overview

OSDF & Pelican

- You may have seen prior presentations about the OSDF – it (or predecessors) have existed for ~10 years.
- We split out the technology powering the OSDF and christened it the “**Pelican Platform**”.
 - Same components as before, just integrated into a standalone platform.

OSDF (Software) → Pelican
OSDF (Service) → XRootD

OSDF Architecture - Vision

Long-term vision: We want OSDF to be an “all-science” CDN.

Requires:

- Connect many repositories to the distribution fabric.
- Provide clients that enable as many use cases.

And benefit from the network effects.

A slide for the XRootD people out there...

Note: pelican plugin is a modest wrapper around libcurl.

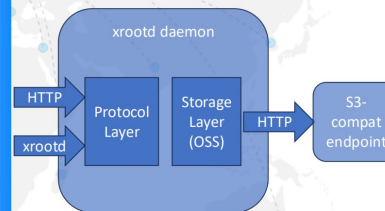


Pelican: Backends to Globus, HTTP, and S3



S3 Plugin

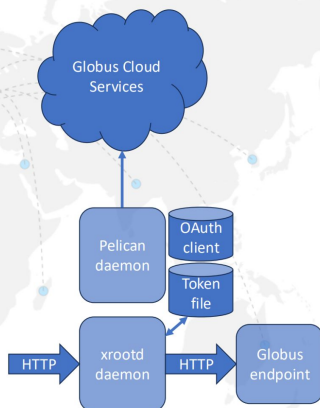
- Sharing the common libcurl-based infrastructure, we have the S3 OSS plugin.
- Similar concept: translate the XRootD OSS API into a sequence of S3 commands.
 - Includes support for “directory” listing!
- Tested with AWS S3, Ceph’s RGW, and MinIO.



Globus Integration

- To contact a Globus endpoint, you need a valid Globus token.
 - Globus uses traditional OAuth2 flows to hand tokens to web applications.
 - **Idea:** The Pelican daemon exports a web interface – use that as the OAuth2 client!
- We then use libXrdHttpServer.so to communicate with Globus.
 - No Globus-specific code!

httpserver.token_file /tmp/foo



Conclusions

- For Pelican, it’s important to integrate as many data repositories as possible.
 - This has led us into developing a series of HTTP-esque OSS plugins.
 - Often, we run the origin ‘nearby’ on behalf of the site by using the NRP’s Kubernetes cluster.
- The S3 backend is seeing production usage.
- The Globus backend is more “tech preview” but potentially has large impact in extending Pelican’s reach.
- Next up? More specialized data repositories and their APIs, such as the **DataVerse software**.



XCache: The new Resource Monitoring & Purging

XRootD Workshop @ STFC UK, Abingdon

September 12, 2024

Alja & Matevž Tadel, UCSD

OU XROOTD SITE REPORT

HORST SEVERINI

XROOTD AND FTS WORKSHOP 2024

SEPTEMBER 2024

Outline

- Computing and Storage Hardware
- Network
- XRootD Configuration
- CephFS Testing and Migration Plans



CERN Central Monitoring Overview

Borja Garrido Bear (On behalf of the monitoring team)

12.09.2024

XRootD Monitoring at Lancaster

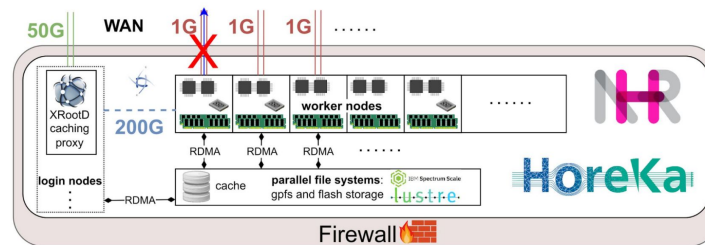
Steven Simpson, Gerard Hand, Matt Doidge, Pete Love,
Roger Jones

Experience with XCache on HPC in Germany

Robin Hofsaess for the GridKa R&D team

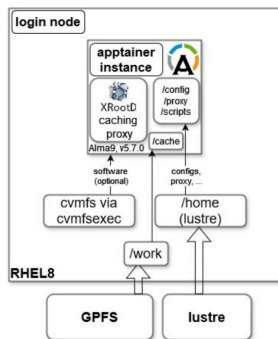


Data-Access Bottleneck Mitigation with XRootD



Setup and Configuration: Overview

- Host: RHEL8
- namespaces, CGroups v2, systemd user services
- Currently running: XRootD v5.7.0 as Alma9 aptainer instance (image bootstrapped from [docker](#))
- In principle up to 76c and 500GB RAM, 50G WAN
 - But shared with other users (limitation via aptainer instance with CGv2 possible)
 - Usage: 32t, 64GB memory
 - 250T quota on gpfs (via IB)



Setup and Configuration: RDMA

- We currently don't use RDMA natively, but [IPoIB](#) for the transfers
- Currently, only IPv4 is possible (no link local v6 addresses in xrootd)
- The cache is also mounted via IPoIB (RDMA)
- We got some complaints from the GPFS team because of the many, many IOPS
- Reason: small blocksizes when caching is enabled
- ideally (FS PoV): pfc.blocksize == FS blocksize, or in general: as big as possible
- dca:
 - Tested, but problematic with containerization
 - Dependent of the campaigns/datasets, e.g. premix rarely completely cached -> in production rather pointless
 - *Would be very useful, if made possible for partially cached files – if possible.*

UK Storage and XRootD

Alastair Dewhurst

Optimising Vector Read requests for RAL disk storage

Alexander Rogovskiy
STFC, RAL

XRootD+FTS workshop @ STFC

Benchmarking XRootD: A Versatile Testing Framework for the Grid Community

Mariam Demir
mariam.demir@stfc.ac.uk

Load balancing XRootD

Jyothish Thomas, James Walder, Thomas Byrne
jyothish.thomas@stfc.ac.uk,
james.walder@stfc.ac.uk, tom.byrne@stfc.ac.uk

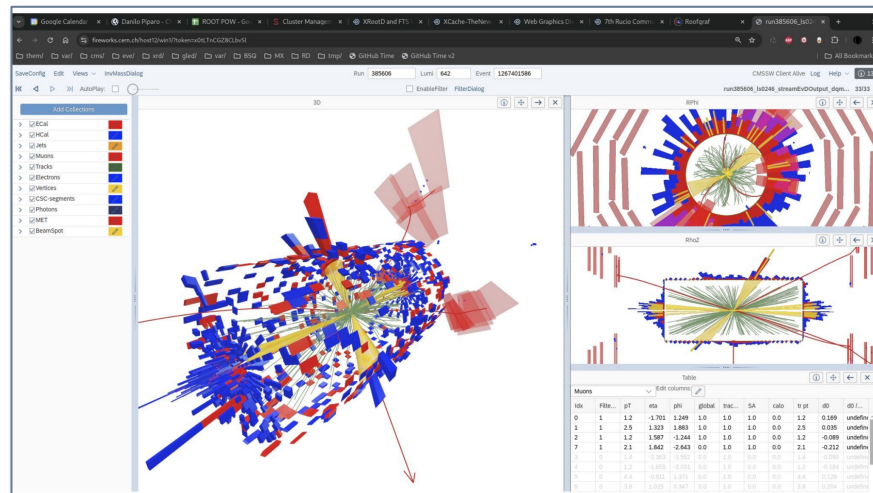
XRootD as a web server for ROOT-Eve and more

Disclaimer: super preliminary, early idea stage

XRootD Workshop @ SFTC UK, Abingdon

September 13, 2024

Matevž Tadel, UCSD



Initial motivation – introduction

- My other night job is ROOT-Eve and CMS Fireworks
 - physics analysis oriented event visualization (EVE – Event Visualization Environment)
- Transition to web-based client-server over the last couple of years
 - Client: JavaScript – JSRoot, OpenUI5, RenderCore
 - Server: ROOT – THttpServer, RWebWindow
 - uses wrapped up CivetWeb – embeddable web server with SSL support
 - The new web-based ROOT GUI also uses this (THREE.js instead of RenderCore).
- FireworksWeb – Event Display as a Service
 - CMS members can access dedicated servers at CERN & UCSD
 - Access data from eoscms, AAA (through XCache), CERNBox (*share with cms-vis-access*)
 - Proto-app with preloaded data-formats forks off an instance (*fast!*, can serve multiple users/tabs)
- Setting this up at CERN was a major PITA
 - Apache frontend with SSO & OAuth (*good*, also at UCSD), full proxying through Apache
 - redirect to instance (still proxied), support upgrade to WebSockets
 - oh, and you need Let's Encrypt certs (*well*, you need those anyway, unless you hate your users)

XrdHttp for REve (and, other ROOT graphics)

- Easier setup of services:
 - Trusted – they run EOS, don't they? Does not need Apache fronting / proxying
 - Web certs still needed
 - Open ports – or local SOCKS proxy
- Authentication & Authorization
 - Use standard mechanisms we're using anyway
 - XrdCI support for CERN SSO is planned
- "Active directories", e.g., */win1/*, */3DView/*, */RPhiView/*, */MuonTable/*, ...
 - Those get upgraded to WebSockets, permanent connection – connect to C++ handlers.
 - REveManager manages connections and knows which connection is connected to each view.
- The rest are served as normal files / directories, e.g.:
 - */ui5/distribution/resources/sap-ui-core.js*
 - */ui5/eve7/rcore/REveRenderCore-min.mjs*
 - */ui5/eve7/sdf-fonts/LiberationSerif-Regular.png*

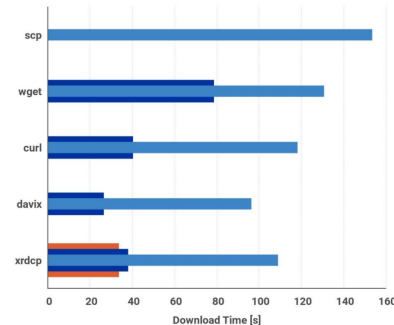
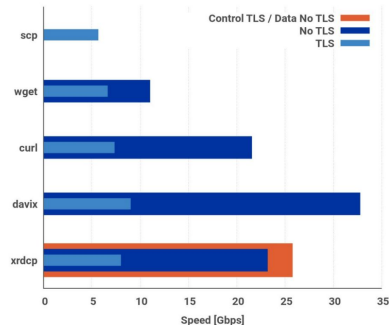


Physics Data Forge: Unveiling the Power of I/O Systems in CERN's Test Infrastructure

G. Amadio, A. Sciabà, A. Peters, D. Smith, L. Mascetti

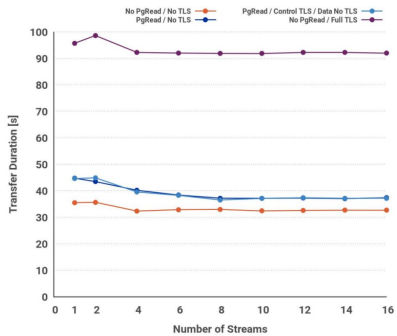
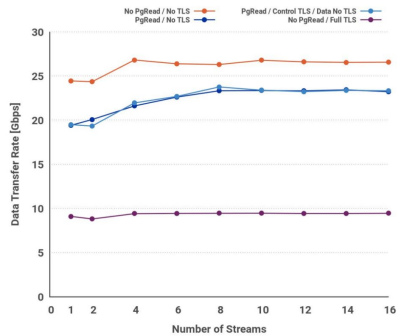
CHEP 2024, Kraków, Poland

Benchmark: Download 100GB File



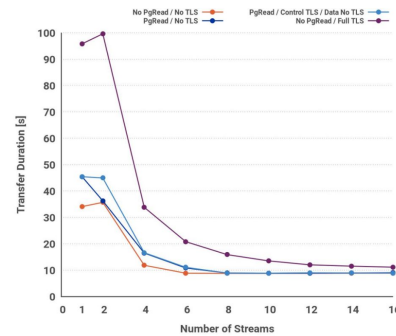
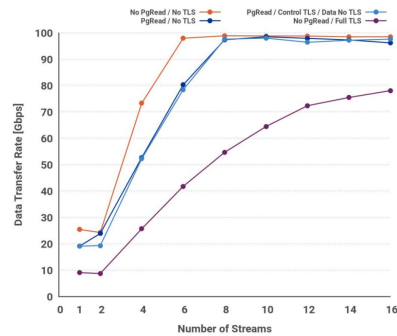
XRootD Client Performance with Multiple Streams

Download speed with xrdcp did not improve very much when adding streams.



XRootD Client Performance with New Architecture

Mapping sockets from same channel to different pollers improves performance.



XRootD Future Plans

- ▶ **Continuously improve overall security, stability, and performance**
 - Address problems shown by static analysis and other QA tools
 - Benchmark studies on 100G and 200G networks (CHEP 24)
 - Expand and improve current testing infrastructure
- ▶ **Plans for XRootD 6.0**
 - Planned for 2025, has not been necessary so far
 - Existing changes to timeout handling now require ABI breakage
 - Improved error handling strategy, make error messages clearer
 - Review long term stability of HTTP support in XrdCl
 - Move to C++20 standard as baseline (base language, ranges, co-routines, std::format)
- ▶ **Ideas for new features**
 - Native plugin for OAuth2 (for use with CERN SSO, with SWAN as main use case)
 - Native support for SSH-based authentication in addition to Kerberos, GSI, etc
 - Investigate support for remote direct memory access (RDMA) with Nvidia DPUs

XRootD in HEP Community

- ▶ **Core component of HEP software ecosystem**
 - Depended on by CTA, FTS, EOS, ROOT, Rucio, experiment frameworks, etc
- ▶ **Exabytes of data processed each year**
- ▶ **Needs security, stability, scalability, sustainability**
 - Code scanning (CodeQL), security policy setup on GitHub (allows private bug reporting)
 - Continuous effort to improve testing infrastructure
 - Measure and expand test coverage, use static analysis tools, ABI tracker, automatic testing in CI
 - Performance analysis of production workloads to guide performance optimizations
 - Lower barrier for contributors and users as much as possible
 - Make it as easy as possible to configure, build, run tests, and create packages



XRootD

XRootD 5.5.x

- ▶ **XRootD 5.5.2**
 - Support GCC 13, OpenSSL 3.x
 - Enable ZTN authentication with macaroons-based tokens
 - Extend number of parallel copies from 4 to 128 (--parallel option to xrscp)
- ▶ **XRootD 5.5.3**
 - Support user-provided script for computing checksums
- ▶ **XRootD 5.5.4**
 - ZTN plugin enabled by default
 - Fixes for authentication failures across daylight savings change
 - Support certificates with dates beyond year 2049
- ▶ **XRootD 5.5.5**
 - Enable XrdClProxy plugin to work with pgRead
 - Fix creation of zip archives with many entries
 - Fix for mixing of reused file handles coming from external table (seen on EOS AMS)

XRootD 5.6.0

▶ XRootD 5.6.0

● Server

- Make `maxfd` configurable (default is 256k)
- Use `SHA-256` by default for signatures and message digests
- Switch to a fixed set of DH parameters (compatibility with OpenSSL 1.0.2)
- Allow specification of minimum and maximum creation modes
- Better detection of private IPV6 addresses (check also for `unique local address`)
- Include token information in the monitoring stream (subject, user, vorg, role, groups)

● XCache

- New function for file eviction
- Allow origin to be a locally mounted directory (e.g. XCache for Ceph/Lustre)

● Client

- New subcommand for `xrdfs cache` to allow for cache evictions
- Do not enforce TLS when `--notlsok` option is used in combination with `root://` URL
- Increase default number of parallel event loops to 10 (affects XCache)

XRootD 5.6.x

▶ XRootD 5.6.1

- Use kernel provided uuid on macOS
- Set `RPATH` that works for binaries and libraries on macOS

▶ XRootD 5.6.2

- HTTP: Fix chunked `PUT` creating empty files
- Scitokens: Update maximum header size and line length in INI files
- Fix template for default ZTN token location
- Change the thread-id returned to OpenSSL 1.0.x to improve performance
- Insert CRLs containing critical extensions at the end of the bundle
- XrdClHttp: Add `pgWrite` support to the HTTP client plugin
- Export `readv` comma separated limits via `XRD_READV_LIMITS` environment variable
- Python: Allow build customization via environment variables (e.g. `CXX`, `CXXFLAGS`)
- Fix promotion of `root://` URLs to use TLS encryption (bug introduced in 5.6.0)

XRootD 5.6.x

▶ XRootD 5.6.3

- Create environment file within `xrd.adminpath`
- Return an error if `xrd.fs.rm` fails to delete any file
- Initial packet marking support in HTTP TPC

▶ XRootD 5.6.4

- Use full certificate chain for verification
- Migrate tests to GoogleTest and run without containers
- Add integrity check for headers and fix header dependency issues
- Fixes on SPARC architecture and GNU/Hurd (external contributions)
- Fix crash on `pss.origin` directive without specifying a port (uses protocol default)

▶ XRootD 5.6.5

- Support GCC 14
- Export project version in `XRootDConfig.cmake` module

XRootD 5.6.x

▶ XRootD 5.6.6

- Use full certificate chain for verification
- Migrate tests to GoogleTest and run without containers
- Add integrity check for headers and fix header dependency issues
- Fixes on SPARC architecture and GNU/Hurd (external contributions)
- Fix crash on `pss.origin` directive without specifying a port (uses protocol default)

▶ XRootD 5.6.7

- Fix crash at teardown when using copies with multiple streams
- Fix TPC initialization to take into account control stream (was always using 2 streams before)

▶ XRootD 5.6.8

- Only claim to be TLS capable if TLS initialization succeeds (`--notlsok` no longer needed)
- Create `CDash` dashboard for XRootD and enable submissions in `test.cmake`
- Fix build on FreeBSD

XRootD 5.6.x

▶ XRootD 5.6.9

- Python
 - Fix iteration over a file with Python3
 - Fix crash with raw strings in prepare call
- HTTP TPC
 - Fix 500 server response code if X-Number-Of-Streams > 100
- XrdScitokens
 - Add stat permissions to create, modify and write operations
 - Allow creation of parent directories if necessary
 - Fix bug when scope includes basepath or /

XRootD 5.7.x

▶ XRootD 5.7.0

- Moved baseline C++ standard to C++17
- Updated min/default RSA bits to 2048
- Restrict renegotiation for TLSv1.2 and earlier
- Redact authz tokens from server logs
- HTTP header parsing now case insensitive for better compatibility
- Make more client errors recoverable to allow retry in more situations
- Several improvements in XrdScitokens plugin (new auth strategies)
- Reintegrated XrdCeph into main repository, now uses `-DENABLE_CEPH=ON` option
- Better performance for GSI authentication by avoiding duplicate work (needed on Alma 9)
- Enable `SSL_OP_IGNORE_UNEXPECTED_EOF` option if available, increase default timeouts
- XRootD now includes a sample shell script for third-party copy (TPC) configuration
- Erasure coding plugin now enabled by default and using OS-provided isa-l library
- Completed migration of test suite to GoogleTest
- Updated Doxygen documentation

XRootD 5.7.x

▶ XRootD 5.7.1

- Allow `cconfig` to write out combined config file
- Harden `systemd` service units for better security
- Fix memory leaks in Python bindings
- XRootD security policy described in [SECURITY.md](#)
- Ensure correct certificate is used when passed via `cgi` with `xrd.gsiusrproxy=...`
- Finished work on redacting token information from logs (client-side)
- Increased test coverage with more server/client setups
- Enabled CodeQL code scanning tool on GitHub
- Introduced abi-tracker reports to ensure full ABI backward compatibility

▶ XRootD on Alma 9 / RHEL 9

- Need to pay attention to `ulimit` settings to avoid running out of memory
- HTTP 1.1 vs HTTP 2.0 incompatibilities (case-insensitive headers from `curl`)
- Bad performance from OpenSSL 3.0 affects GSI + VOMS extraction configurations