



XRootD

XRootD Workshop 2023 Summary

G. Amadio (IT-SD-PDS)

12 Apr 2023

XRootD Workshop

- ▶ XRootD Workshop @ Josef Stefan Institut (JSI), Ljubljana, Slovenia
- ▶ From Wednesday (29 Mar 2023) to Friday (31 Mar 2023)
- ▶ Attendance: 45 registered
 - 35 in person + 10 online
 - FTS and XRootD together



Wednesday

| | | |
|-------|--|-----------------------------|
| 14:00 | Welcome | Jan Jona Javorek |
| | Jozef Stefan Institute | 14:00 - 14:10 |
| | XRootD Features | Andrew Bohdan Hanushevsky |
| | Jozef Stefan Institute | 14:10 - 15:00 |
| 15:00 | What's up with the XRootD client | Michal Kamil Simon |
| | Jozef Stefan Institute | 15:00 - 15:30 |
| | Coffee break | |
| | Jozef Stefan Institute | 15:30 - 16:00 |
| 16:00 | XRootD Release Schedule and Future Plans | Guilherme Amadio |
| | Jozef Stefan Institute | 16:00 - 16:20 |
| | Evolution of XRootD Testing and CI Infrastructure | Guilherme Amadio |
| | Jozef Stefan Institute | 16:20 - 16:40 |
| | OU XRootD Site Report | Horst Severini |
| | Jozef Stefan Institute | 16:40 - 16:55 |
| 17:00 | XRootD usage at GSI | Soren Lars Gerald Fleischer |
| | Jozef Stefan Institute | 16:55 - 17:15 |
| | Analysis of data usage at BNL | Hironori Ito |
| | Jozef Stefan Institute | 17:15 - 17:30 |
| | XRootD in the UK: ECHO at RAL-LCG2 and developments at Tier-2 sites | James William Walder |
| | Jozef Stefan Institute | 17:30 - 17:45 |

18:00

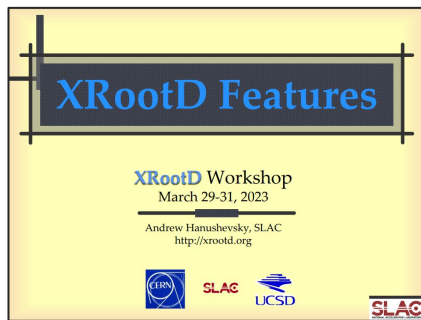
Thursday

| | | |
|-------|---|---------------------------|
| 09:00 | Open Science Data Federation - OSDF | Fabio Andrijauskas |
| | Jozef Stefan Institute | 09:30 - 10:10 |
| 10:00 | To the OSPool and Beyond: The guts of the OSDF client | Brian Bockelman |
| | Jozef Stefan Institute | 10:10 - 10:30 |
| | Coffee break | |
| | Jozef Stefan Institute | 10:30 - 11:00 |
| 11:00 | XCACHE Developments & Plans | Matevz Tadel |
| | Jozef Stefan Institute | 11:00 - 11:30 |
| | Experience with XCache in Virtual Placement | Ilija Vukotic |
| | Jozef Stefan Institute | 11:30 - 12:00 |
| 12:00 | Experience deploying xCache for CMS in Spain | Carlos Perez Dengra |
| | Jozef Stefan Institute | 12:00 - 12:20 |
| | Getting the most out of XCache | Ilija Vukotic |
| | Jozef Stefan Institute | 12:20 - 12:30 |
| | Lunch | |
| | Jozef Stefan Institute | 12:30 - 14:00 |
| 14:00 | Data-Aware Scheduling for Opportunistic Resources (with XRootD and HTCondor) | Robin Hofsaess |
| | Jozef Stefan Institute | 14:00 - 14:30 |
| | Kingfisher: Storage Management for Data Federations | Brian Paul Bockelman |
| | Jozef Stefan Institute | 14:30 - 14:50 |
| 15:00 | XRootD pgRead & pgWrite | Andrew Bohdan Hanushevsky |
| | Jozef Stefan Institute | 14:50 - 15:10 |
| | XrdEc: the whole story | Michal Kamil Simon |
| | Jozef Stefan Institute | 15:10 - 15:30 |
| | Coffee break | |
| | Jozef Stefan Institute | 15:30 - 16:00 |
| 16:00 | XRootD Plugins | Andrew Bohdan Hanushevsky |
| | Jozef Stefan Institute | 16:00 - 16:30 |
| | Porting of XRootD to Windows as a part of EOS-wnc | Gregor Molan |
| | Jozef Stefan Institute | 16:30 - 17:00 |
| 17:00 | A Brief History of the dCache Xroot Implementation (virtual) | ALBERT ROSSI |
| | Jozef Stefan Institute | 17:00 - 17:30 |

Friday

| | | |
|-------|--|---------------------------|
| 09:00 | RNTuple: ROOT's Event Data I/O for HL-LHC | Jakob Blomer |
| | Jozef Stefan Institute | 09:30 - 10:00 |
| 10:00 | LHCOPNLHCONE Status and Updates | Edoardo Martelli et al. |
| | Jozef Stefan Institute | 10:00 - 10:20 |
| | Kubernetes and XrootD | Fabio Andrijauskas |
| | Jozef Stefan Institute | 10:20 - 10:30 |
| | Coffee break | |
| | Jozef Stefan Institute | 10:30 - 11:00 |
| 11:00 | Outlook on EOS, the CERN storage solution for LHC Run3 and beyond | Guilherme Amadio |
| | Jozef Stefan Institute | 11:00 - 11:30 |
| | A Prometheus XRootD exporter based on mpxstats (virtual) | Jan Kredlik |
| | Jozef Stefan Institute | 11:30 - 11:40 |
| | XRootD monitoring discussion | Matevz Tadel |
| | Jozef Stefan Institute | 11:40 - 12:00 |
| 12:00 | Don't be a Stranger, Please! :) | Michal Kamil Simon |
| | Jozef Stefan Institute | 12:00 - 12:15 |
| | Many Thanks & Future Outlook | Andrew Bohdan Hanushevsky |
| | Jozef Stefan Institute | 12:15 - 12:30 |
| | Lunch | |
| | Jozef Stefan Institute | 12:30 - 13:25 |

Day 1 (Wednesday)




Andy had a look back at the last four years of development in XRootD (since last workshop happened).

Many new features were introduced with major version 5, including data integrity features like `pgread`, `pgwrite`, and erasure coding, which has been used on EOS Alice O² instance with good performance.

Next version will be 5.6, and will increase number of redirectors, among other features, like better Python support in CMake, migration to modern CMake, support for musl based distros like Alpine Linux, etc.

Let's look back

- ✦ New Features Review
 - We'll start at the lockdown
 - It's going to take some time
 - You'll be surprised what you'll find



Security
Performance
Monitoring
Operational
Proxy & Xcache
SSI
Client

Categories

XRootD Workshop @ JSI March 27-31 2023 3 SLAC

Client Features 5.4.0 @ 12-10-21

- ✦ Data Integrity
 - Full support of `pgread` & `pgwrite`.
 - Declarative API, zip archives, unaligned requests.
 - On the fly correction of checksum errors.
 - Avoids retransmitting whole file.
 - XrdEC (erasure encoded parallel file system).
 - Checksum data (i.e., `pgread` & `pgwrite`).
 - Allow activation of XrdEC via config file.
 - Discover placement group in real time.
 - Full native support for XrdEC (i.e., no EOS).
 - `xrdcp`
 - Allow multiple checksums requests via `-cksum` option.

XRootD Workshop @ JSI March 27-31 2023 40 SLAC

Client Features 5.5.0 @ 08-26-22

- ✦ XrdEc (Erasure Encoded Parallel File System)
 - Add support in `proxy` server, `xrootdfs`, and `xrdadler32`.
 - Use free space as stripe server selection parameter.
 - Implement VectorRead.
 - Make remote config file more flexible.
- ✦ `xrdcp`
 - With `-server` option display IP stack information.
- ✦ `xrdfs`
 - List multiple files to be removed on command line.
- ✦ Record/Replay
 - Provide ability to record client execution.
 - Add `xrdreplay` command to replay recorded execution

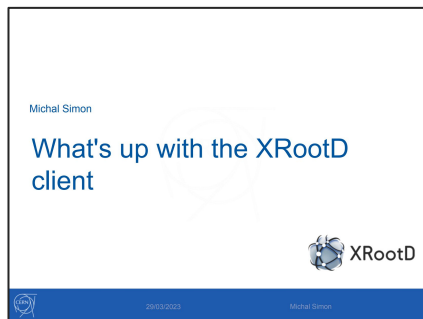
XRootD Workshop @ JSI March 27-31 2023 45 SLAC

Conclusion

- ✦ XRootD is facile, flexible, and sound
 - Applicable to a wide variety of problems
 - Framework widely used as core component
 - The tagline - "It's XRootD Inside!" applies
- ✦ Our core partners
 - Logos: CERN, SLAC, UCSD
- ✦ Community & funding partners (not a complete list)
 - Logos: ATLAS, CMS, LHC, SLAC, UCSD, GSI, etc.

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

XRootD Workshop @ JSI March 27-31 2023 47 SLAC



Michał gave us an overview of how to use the xrdcp client and had a look at recently added features, like the record & replay plugin.

The declarative API for the client is a big improvement that allows for more easily writing a sequence of asynchronous operations, for example opening, writing and then closing a file.

The record & replay plugin can be used to reproduce access patterns without the need to recreate a complex environment (e.g. a ROOT analysis). It has received several fixes in the latest release, XRootD 5.5.4.

Outline

- xrdcp primer
- Declarative API
- Lifting File API limitations
- Record & replay

Declarative API: Motivation

- Use case: erasure coding plug-in for EOS
 - Executing multiple operations on multiple **remote** files (stripes) in parallel
- Problem with **asynchronous operation composability and code readability**
 - Asynchronous `Open()` + `Write()` + `Close()` in the code is only visible as an `Open()` (rest of the workflow is in the callbacks)

Recorder plug-in

- User's actions are stored using **CSV file format**
 - We do **support quoting** so it is safe to use commas in URL opaque info
- By default the file is stored at: `/tmp/xrdrecord.csv`
 - This can be overwritten either in the config file **using the output key**, or
 - Using an **environment variable**: `XRD_RECORDERPATH`
- Introduces only **minimal or no overhead**

Summary

- Don't be afraid of **async APIs**
 - Declarative API makes it **much easier and readable**
 - The File object no longer needs to outlive the completion handlers
- Record / replay is **great for debugging and benchmarking** storage systems
- There is **lots of functionalities build into the xrdcp** tool
 - Be sure to know its capabilities before enhancing it with scripts



XRootD

Release Schedule and Future Plans

G. Amadio

XRootD Workshop Ljubljana, Slovenia

29 Mar 2023

Discussed the release process of XRootD, the last release and plans for the upcoming feature release and major release towards the end of the year.

The main items for the next major release are support for C++17, dropping support for Python 2.x, migration of tests from CppUnit to GoogleTest, changes to the interface of error objects to allow better error messages to be handled to the client, among other things.

Overview

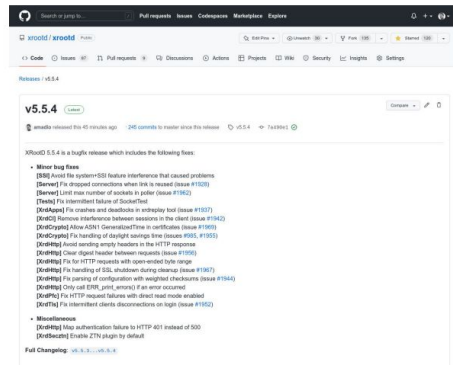
- ▶ Introduction
- ▶ Release process
- ▶ XRootD 5.5.4 patch release
- ▶ XRootD 5.6 feature release plans
- ▶ XRootD 6.0 major release plans
- ▶ Development workflow and project management
- ▶ XRootD Documentation
- ▶ XRootD Packaging

XRootD

2

XRootD 5.5.4 Release

- ▶ Released 24 Mar 2023
- ▶ 16 Minor bugs fixed
- ▶ ZTN plugin now always enabled



XRootD

8

Next Release: XRootD 5.6 Feature Release

- ▶ Migration to modern CMake (in progress, some of it already in master)
 - Update minimum required version to CMake 3.16 (many new features relative to 3.1)
 - Use upstream modules to reduce maintenance burden of build system
 - Use newer module `FindPython.cmake`, with better support for virtual environments, etc
 - Replace usage of variables with targets and target properties
 - For more information, we recommend the excellent talk [Effective CMake](#), by Daniel Pfeifer
- ▶ Initial migration of tests from CppUnit to GoogleTest (in progress as well)
 - CppUnit is no longer actively developed upstream
 - Terse output when tests fail, difficult to know what went wrong
 - GoogleTest is actively developed and widely used by C++ developers
- ▶ Support for distributions based on musl libc (Alpine Linux, Void Linux)

XRootD

9

XRootD 6.0 Major Release Plans

- ▶ Move to C++17 standard as baseline, ensure compilation with C++20
- ▶ Drop support for Python2.x
- ▶ File ownership based on uid/gid?
- ▶ Modernize Python bindings: packaging and implementation
- ▶ Full migration of testing infrastructure to GoogleTest
 - CERN Summer Student project to improve test coverage for authentication, etc
- ▶ Refactoring of event loop on the client
 - Run some event loop tasks on a separate thread pool to improve performance
- ▶ Planned for release in Q4 of 2023
 - Idea is to give time for inclusion of other potentially breaking changes
 - Use GitHub milestone to tag features/issues/developments for inclusion into XRootD 6.0

XRootD

10



XRootD

Evolution of Testing and CI Infrastructure

G. Amadio XRootD Workshop Ljubljana, Slovenia 29 Mar 2023

Talked also about the evolution we plan for the testing and CI infrastructure of XRootD, by making the current tests easy to run, and adding them back to run in GitHub Actions.

We have also a CERN Summer Student project for improving testing and CI, by testing authentication and other parts of the projects which are currently not tested, as well as making use of more QA tools like clang-tidy for static analysis.

Recent Developments

- Linux distribution recommendation changed from CentOS Stream to Alma
 - Added builds on Alma Linux 8 and Alma Linux 9 to XRootD GitHub Actions
 - Adapted docker based tests to work on Alma 8 and Alma 9 in addition to CentOS 7
- Added a build on Alpine Linux (musl-based Linux distribution)
- Dropped build on Ubuntu 18 (not supported anymore)
 - CMake is too old (3.15), we now require CMake 3.16 or newer (due to FindPython.cmake)
- Moved Fedora 35 build to Fedora 37
- Removed branch filters, now CI runs on all branches and pull requests
- Planned: builds with clang on Linux, static analysis with clang-tidy, coverage

XRootD

4

Docker Tests

- Existing tests in repository on GitLab
 - <https://gitlab.cern.ch/eos/xrootd-docker>
- Converted this setup into **xrd-docker** script
 - Subcommands
 - fetch - download data
 - package - create XRootD tarball
 - build - build docker images
 - setup - setup containers
 - run - run tests
 - clean - clean up running containers and drop testing network
 - Pull request with latest version: <https://github.com/xrootd/xrootd/pull/1974>
- Operating Systems: CentOS 7, Alma 8, Alma 9
- Planned to be included in XRootD 5.6 release

XRootD

7

Testing with CMake/CTest

- Tests need to be easy to run
- No special knowledge should be required
- Gives confidence to external contributors that they are not breaking anything when making changes to the code
- Everyone knows the "standard" workflows
 - Autotools
 - configure && make && make test
- Provide similar experience with CMake
 - cmake && make && ctest

```
2 cmake -S xrootd -B xrootd_build
-- The C compiler identification is GNU 12.2.1
-- The CXX compiler identification is GNU 12.2.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/lib/clang/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Build files have been written to: xrootd_build

2 cmake --build xrootd_build --parallel $(nproc)
[1] src/XrdVersion.hh successfully generated
[0] Built target XrdVersion.hh
[0] Building CXX object ...

3 ctest
Test project xrootd_build
Start 1: XrdC1:URLTest.LocalURLs ..... Passed 0.01 sec
1/6 Test #1: XrdC1:URLTest.LocalURLs ..... Passed 0.01 sec
Start 2: XrdC1:URLTest.RemoteURLs ..... Passed 0.16 sec
2/6 Test #2: XrdC1:URLTest.RemoteURLs ..... Passed 0.16 sec
Start 3: XrdC1:URLTest.InvalURLs .....
3/6 Test #3: XrdC1:URLTest.InvalURLs .....
Start 4: XrdC1:URLTest.InvalURLs .....
4/6 Test #4: XrdC1:URLTest.InvalURLs .....
Start 5: XrdC1:URLTest.InvalURLs .....
5/6 Test #5: XrdC1:URLTest.InvalURLs .....
Start 6: XrdC1:URLTest.InvalURLs .....
6/6 Test #6: XrdC1:URLTest.InvalURLs ..... Passed 0.01 sec
100% tests passed, 0 tests failed out of 6
Total Test time (real) = 13.23 sec
```

XRootD

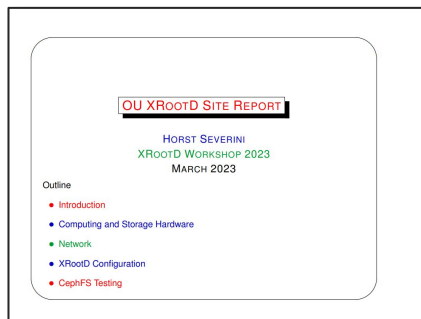
10

Supported Platforms

- Which platforms and compilers should we support?
 - Currently, we officially support CentOS 7, Alma 8, Alma 9, Ubuntu, and macOS
 - GitHub Actions now also covers Alpine Linux (due to recently added musl libc support)
- Supporting more compilers can be beneficial
 - More opportunity to find bugs via compiler warnings
 - Be more resilient against compiler-specific features/bugs
 - More tools to apply in development (e.g. clang-tidy, clang-format)
- Clang on Linux not currently supported, maybe good to add
- What hardware architectures to support?
 - No explicit support for anything other than x86_64 and arm64 (macOS)
 - Do our users run XRootD on unsupported architectures like PowerPC?
 - Plan to add other architectures via qemu to GitHub Actions (at least arm/arm64)

XRootD

11



The first site report was for Universities of Oklahoma and Texas Arlington.

Their plan is to move from a plain XRootD setup to cephfs.

Their hope is to ease maintenance burden as the cephfs storage will be provided by a third party.

There is also the hope that the new setup will provide better performance. They are, nevertheless close to their available hardware and network limit already with the current setup.

Horst Severini March 2023 OU XRootD Site Report

Introduction

- Planning to migrate from xrootd to ceph storage in the next year
- Unfortunately not as much to present on these tests as intended
- Had to spend much time on upgrading HTCondor-CE GK
- EL9 osg-36 brand new, installed from osg-upcoming-testing
- Took much longer to iron out teething bugs
- Most likely first ATLAS EL9 GK world wide
- Therefore this talk covers only first initial tests, done yesterday!

2 XRootD Workshop 2023

Horst Severini March 2023 OU XRootD Site Report

US ATLAS SWT2 Center

- University of Oklahoma
 - Oklahoma Center for High Energy Physics (OCHEP)
 - OU Supercomputing Center for Education and Research (OSCCER)
- University of Texas Arlington
 - Chemistry and Physics Building (CPB)
 - Arlington Regional Data Center (ARDC) in Fort Worth

3 XRootD Workshop 2023

Horst Severini March 2023 OU XRootD Site Report

XRootD Configuration

- Currently, se1.osccer.ou.edu proxy server for 700 TB xrootd cluster
 - Pretty stable and performant
 - Occasionally, one or two of the 7 servers gets overloaded with open connections, causing transfer timeouts
 - Restart of xrootd (not cmsd) on these nodes eventually fixes this
 - Not fully understood
- Plan to migrate to cephfs file system after warranty expires
 - Part of 9.5 PB (and growing) OSCER ceph file system
 - 26 R740xd2 OSD nodes, 8+3 erasure coding
 - 14 GB/s total throughput
 - Very performant, reasonably priced
 - \$90 per usable TB, good for 7 years

6 XRootD Workshop 2023

Horst Severini March 2023 OU XRootD Site Report

Summary and Conclusions

- OU XRootD storage quite stable and performant
- Able to transfer 3+ GB/s, which is probably close to current available hardware/network limit
- Initial xrootd-ceph setup successful
- Looking forward to performance improvement with this new setup
- Hopefully get closer to 50 Gbps current network limit
- Possibly also test s3-connector, although current OSCER ceph configuration doesn't support any block operations
- Also still to do: Token Auth for XRootD (Working fine on new EL9 HT Condor-CE)

10 XRootD Workshop 2023

XRooD @ ALICE T2/AF
XRooD @ ESCAPE
XRooD @ PUNCHNFDI

XRooD usage at GSI

XRooD and FTS Workshop @ JSI, Ljubljana 2023

Sören Fleischer

2023-03-29

Sören Fleischer 1 / 22

Sören Fleischer reported on XRooD usage at GSI.

They use debian-based container images to run XRooD at their site (due to site policy), and currently have a custom procedure to build and distribute these images. They are planning on moving to an RPM-based distro in the future after site policies have been updated to be less restrictive. They are interested in potentially using a centrally provided docker image, should that become available. This was a recurring topic in the workshop (centrally provided docker images for XRooD).

XRooD @ ALICE T2/AF
XRooD @ ESCAPE
XRooD @ PUNCHNFDI

Overview
Local Redirect Plugin
alinat (Replaced XRooD Proxy)
Packaging Flow

ALICE T2/AF

- 3 2 XRooD data servers running Debian 10 (Bare Metal)
 - localroot on a shared filesystem (Lustre)
 - Lustre Quota Plugin
- 2 XRooD redirectors / data managers running CentOS 7 (VM)
 - Local Redirect Plugin
- 0 XRooD proxy servers (since 2019-02-13)

Sören Fleischer 2 / 22

XRooD @ ALICE T2/AF
XRooD @ ESCAPE
XRooD @ PUNCHNFDI

Overview
Local Redirect Plugin
alinat (Replaced XRooD Proxy)
Packaging Flow

alinat (Replaced XRooD Proxy)

Sören Fleischer 14 / 22

XRooD @ ALICE T2/AF
XRooD @ ESCAPE
XRooD @ PUNCHNFDI

Overview
Local Redirect Plugin
alinat (Replaced XRooD Proxy)
Packaging Flow

Current Packaging Flow

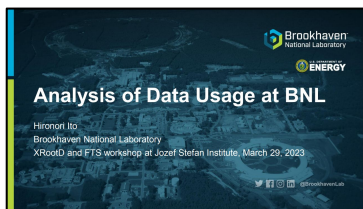
Sören Fleischer 17 / 22

XRooD @ ALICE T2/AF
XRooD @ ESCAPE
XRooD @ PUNCHNFDI

XRooD @ ESCAPE

- Data Lake
- Custom compiled XRooD
- Project ended/resurrected?
- Compilation/provisioning workflow developed by P. Kramp / M. Szuba a few years ago
 - Neither of them work at GSI anymore
 - Workflows broken, in the process of being repaired

Sören Fleischer 20 / 22



Hironori Ito provided an analysis of data usage via XRootD at BNL.

He tried to analyse usage patterns to identify how to optimize usage of the data infrastructure they provide and found that data reuse and total volume of data are usually very low.

The conclusion is that maybe it's possible to save costs by having a system backed mostly by tape with an appropriately sized cache in front, an interesting idea.

Analysis of data usage from storage logs

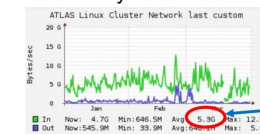
- There are more interests to see how much of the data are actively being used.
 - More attention has been given to how much data we can move between sites. But, once the data gets there, small attention is being given if the data is actively used.
- Experiments are coping the disk usage rate for now. But, that might not be easily attainable if no attention is being paid to the data usage on the storage.
- The analysis is done from the storage logs of ATLAS dCache at BNL using TimescaleDB.
 - At BNL, almost all "Read" access by jobs are by XRootD protocol while site-to-site data transfers are by WebDAV.



2

Checking logs

- dCache records all access information in logs.
- The recorded information includes path, file size, read size, access time, etc...
- Is it really accurate?



Network I/O seen from the worker nodes.
Values given by NIC on the hosts.

From dCache access logs and summing over all XRootD access during the same period, the average access rate is **5.16 GB/s**

Data volume received in the worker nodes are very similar to the data volume accessed in dCache, indicating the reasonable consistency between the two.

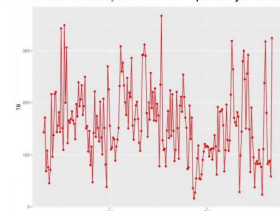


3

Daily data volume used by jobs at BNL

~ 0.8% of total data volume are used per day

Total size of **unique** used files per day ~ 150TB



BNL has
~60M files
~21PB

Note: it does not include the data out from BNL to the other sites.



6

Conclusion

- The production data see less reuse over the long period.
 - Large fractions of the data are not used frequently by local jobs.
 - Different data type might see different rate of reuse.
- The user data see more reuse.
 - Pure cache might work
 - Is it more efficient in terms of cost and labor?
 - What kind of cache?
 - Tape backed pure XCache?



10

XRootD experiences from the UK: ECHO and T2

James Walder
On behalf of UK storage community

XRootD + FTS Workshop, Ljubljana, Slovenia
27-31 March 2022

Hosts: Alexander, Tom Byrne, Rob Curtis, Andrew, Jonathan, Neil, Stephen, Andy Ellis, Gerd, Michael, Adam, Peter, Ben, Stephen, Jonathan, Thomas

James Walder presented experiences with XRootD in the UK.

They are the main user of the XRootD CEPH plugin, but they run a heterogeneous set of storage systems at various scales, many using XRootD.

Feedback to XRootD community is that documentation of “real world” best practises for non-experts would be nice to have (at the same time acknowledging that XRootD already has plenty of docs).

Outline

- Overview of Storage in the UK
- RAL-LCG2: ECHO
 - Object store: librados and Erasure Coding reminder
- Main architecture developments since last workshop
- Improvements for:
 - Deletes, Checksums, Davs, Reads/Writes, ReadV,
- Token support
- T2s:
 - XRootD + CephFS; Lancaster
 - Monitoring
 - Caches
- UK feedback / inputs
 - Summary

2

Storage in the UK

- UK a heterogeneous source of storage technologies
- More recently, (significant) storage is being consolidated to 5 main T2 sites (+T1)
- With DPM EOL; smaller sites typically to become storageless:
 - Or, migrating to dCache with existing storage.
- XRootD+CephFS selected for some larger sites (see later slides)

| Site | Storage (now) | Storage (if changing) |
|-------------------------|-----------------------|------------------------------|
| RAL-LCG2 (T1) | | Echo (XRootD+Ceph) |
| UKI-LT2-Brunei | DPM | XRootD+CephFS |
| UKI-LT2-IC-HEP | | dCache |
| UKI-LT2-OMUL | | StoRM (Austrel) |
| UKI-LT2-RHUL | DPM | Storageless (SE - OMUL) |
| UKI-NORTHGRID-LANCS-HEP | XRootD+CephFS (+ DEM) | XRootD+CephFS (+ dCache) |
| UKI-NORTHGRID-LIW-HEP | DPM | dCache |
| UKI-NORTHGRID-MAN-HEP | DPM | XRootD+CephFS |
| UKI-NORTHGRID-SHEF-HEP | | Storageless (SE - RAL-LCG2) |
| UKI-SCOTGRID-DURHAM | DPM | (TBD) |
| UKI-SCOTGRID-ECDF | DPM | dCache |
| UKI-SCOTGRID-GLASGOW | | Echo (XRootD+Ceph) + |
| UKI-SOUTHGRID-BHAM-HEP | | Storageless (SE - MAN + VPI) |
| UKI-SOUTHGRID-BRIS-HEP | | (XRootD+HDFS) |
| UKI-SOUTHGRID-CA-HEP | | Storageless (SE - RAL-LCG2) |
| UKI-SOUTHGRID-RALP | | dCache |
| UKI-SOUTHGRID-SUSX | | Storageless (SE - PMM IL) |

T1 and large (storage) T2s highlighted

Object storage in ECHO

- XrdCeph (xrootd-ceph) OSS plugin interfaces XRootD to librados(stripes)
 - GridFTP plugin also successfully deployed
 - Significant effort and recently to develop XrdCeph for efficient usage in Run-3 and beyond
- Object store; i.e. no directory structure - the path is the name of the file/object
- So - putting it all together:
 - Objects on disk are made up of all the chunks for that object:

8

Feedback / Summary

- The UK runs a heterogeneous set of storage technologies at varying scales: many using XRootD
 - ECHO:
 - New dedicated effort for supporting the XrdCeph plugin.
 - Pivoting towards developments needed for the challenges of HL-LHCs (and non-WLCO VOs).
- Lancaster: Deploying CephFS takes a lot of effort:
 - Successful high-throughput XRootD deployments need to be built wide
 - Monitoring is key
- Recent releases have had some issues (particularly) for UK configurations;
 - Benefited from xrootd developer support / responses.
 - A suite of FTs using Rucio + FTS, against site test RSEs could be set up across the UK and beyond, to test our various use-cases.
- Many other activities, not mentioned here: Shovel, packet marking, ...
- The UK is gaining considerable expertise with XRootD and tends to propose it as a frontend for new users into HEP-like/large-scale data transfer orchestration and operations:
 - Improved documentation for non-experts in ‘real-world’ best-practice setups desirable;
 - Attempting to improve our feedback into the XRootD community.

21

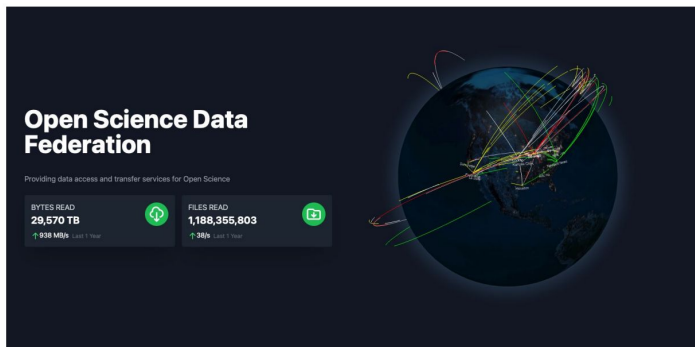
Day 2 (Thursday)

Fabio presented the OSDF, which is the data counterpart to the OSG, the Open Science Grid.

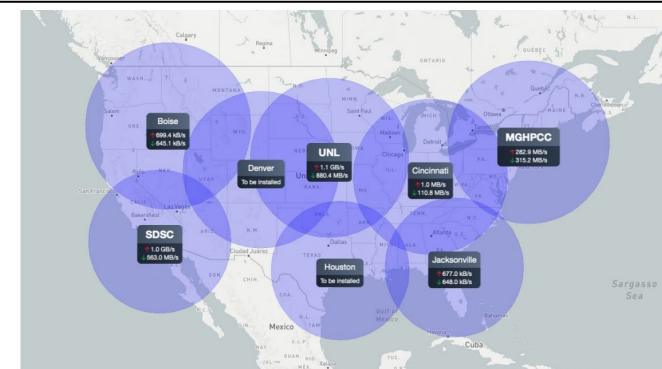
The OSDF provides data for jobs run within OSG, as well as containers used by OSG.

In a separate talk he presented advantages and disadvantages in using containers and Kubernetes.

Monitoring was also discussed.



<https://osdf.osg-htc.org>



<https://nrg-website.vercel.app>

Open Science Data Federation

- The Open Science Data Federation (OSDF) is an OSG service designed to support the sharing of files staged in autonomous "origins", for efficient access to those files from anywhere in the world via a global namespace and network of caches.
- The OSDF may be used either from within OSG or independent of OSG

Example OSDF Use Cases

- A researcher wants to share a dataset with their community such that others may process it.
- A researcher produces data on the OSG that they need to store for future processing or sharing with the community.
- A researcher has a GB to TB-scale dataset that they want to analyze. Their workflow processes the same data many times, thus benefiting greatly from the caching within OSDF.

Open Science Data Federation - OSDF Kubernetes - docker - containers

Fabio Andrijauskas
UCSD

To the OSPool and Beyond: The guts of the OSDF client

Brian Bockelman
XRootD Workshop, March 2023

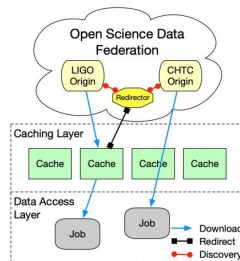
Brian presented the stashcp client used within OSDF to transfer data.

He gave a short live demo of it, discussed the stages needed in the client (invocation, authorization, discovery, file transfers) to execute transfers across sites.

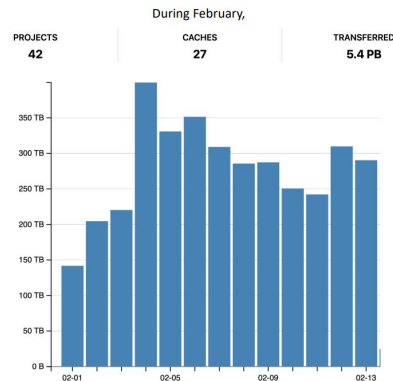
He also discussed the importance of having human readable error messages in XRootD, and the possibility that a browser-based client would be desirable in the future.

The Open Science Data Federation (OSDF)

- The OSDF aims to help researchers move objects to and from computation in support of open science.
- The OSDF provides:
 - The *origin service*, which integrates an existing object store or filesystem into the OSDF.
 - A *redirector*, helping clients to find the objects.
 - A set of distributed *caches* for scalable data distributions.
 - A client for accessing objects from jobs.



OSDF “at a Glance”



The OSDF Client: Invoke

- The client has two binaries,
 - **stashcp**: A ‘cp’ like interface. Intended to be invoked by users at the CLI.
 - **stash_plugin**: A HTCondor file transfer plugin.
- ‘stashcp’ has user-friendly features like progress bar, transfer resumption, recursive downloads.
- ‘stash_plugin’ provides structured output and error messages about transfer results.
- Implementation is in go; everything is in a self-contained, statically-linked binary.

```
bbockelm ~$ stashcp /osgconnect/public/ashish_tripathie/full-o3/clean...
F4HP7QL65F:~$ bbockelm$ stashcp /osgconnect/public/ashish_tripathie/full-o3/clean...
ned_30Hz_150Hz/L1/L1_split_aa_748800.splittft /tmp/
L1_split_aa_748800.splittft 1.24 MiB / 3.02 GiB [-----] 7h45m25s | 127.62 KiB/s
```

Works on Windows, Mac, and Linux!

Active threads & Future Activities

- We want the clients to be usable by anyone. Particularly, this means error messages must be “human optimized” not “developer optimized”.
 - This is not just ‘write better error messages’ in the client but also ‘change XRootD to provide better error messages’.
- Clients discussed have all been command line. To really capture “normal users”, we need **browser-based clients**.
 - Goal is to allow uploads/downloads from laptop through the browser; no standalone software download needed.

XCACHE - Developments & Plans

XRootD Workshop @ JSI Ljubljana

March 30, 2023

Matevž Tadel, UCSD

Matevž discussed XCache in detail, including an overview of what it is, what a minimal configuration to run XRootD as a caching server looks like, recently developed features in XCache, like support for pgread and asynchronous operations; and development plans for the future, including the addition of support for per directory usage, quotas & monitoring, as well as an improved algorithm that may allow prefetching data based on usage patterns for better performance.

Introduction

XCache – brand name for XRoot disk-based file proxy cache

In code referred to as PFC – proxy-file-cache, so you will see:

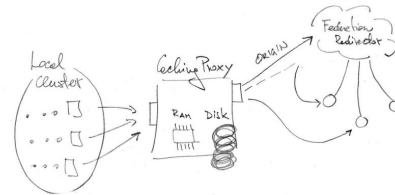
- Classes and file-names prefixed by **XrdPfc**, e.g. XrdPfcFile and XrdPfcFile.hh/cc
- Configuration options prefixed by **pfc**, e.g. pfc.blocksize

M. Tadel, XCache Developments & Plans, XRootD@JSI Ljubljana, March 2023

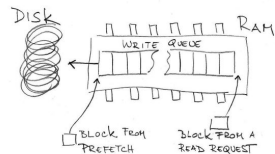
3

XCache in one slide

- Serve data to local clients:
 - Origin - remote data source (usually data federation)
 - Data read in "blocks"
 - Optional prefetching
 - Store data on local disk via write queue
 - Rely on VFS to help
 - Purge old files as disks get full



- XCache server is a "normal" XRootd server:
 - Authentication / authorization controls
 - LVM / multi-disk support
 - Tracing and monitoring
 - Clustering - Caching Cluster
 - Can use http on both ends



M. Tadel, XCache Developments & Plans, XRootD@JSI Ljubljana, March 2023

5

Minimal XCache Server configuration

Mandatory directives

```
all.role      proxy server      # This is a proxy
all.export    /store cache      # Exported namespace

pss.cachelib  libXrdFileCache.so # Request Proxy File Cache / PFC
pss.origin    cmsxrootd.fnal.gov:1094 # Remote data source

oss.localroot /data/xrd-cache    # Where data is stored on local disk
```

Frequently used pfc directives (the numbers given are defaults)

```
pfc.blocksize 128k      {4k, 512M}
pfc.prefetch  10        { 0, 128}
pfc.ram        1G        {1G, 256G}
pfc.diskusage 0.9 0.95  {no limits, can also be given in bytes}
```

```
xrootd -c hello-cache.cfg
```

M. Tadel, XCache Developments & Plans, XRootD@JSI Ljubljana, March 2023

10

Per directory usage, quotas & monitor / purge plugin

- This is already partially implemented
 - pfc.dirstats maxdepth 2 dir /foo/bad/users/*
 - dumps report in a file every purge cycle
- There has been a desire for per-directory quotas around for a while
 - OSG with multi-tenant caches; runaway usage by a VO or even a single user
 - but never strong enough push to actually implement it (it is quite hard)
- Alternative or extension, proposed by Brian / Kingfisher project is:
 - LotManager component that takes over cache monitoring & management
 - can be a plugin ... but could also be a service ... or both

M. Tadel, XCache Developments & Plans, XRootD@JSI Ljubljana, March 2023

35

XCACHE experience Virtual Placement (and ServiceX)



Ilija Vukotic
US ATLAS Computing Facilities @SLAC F2F
2022-12-01



Ilija Vukotic presented ServiceX and XCache usage for ATLAS distributed data management. They use virtual placement, which is a way to enable efficient data access over a WAN by using XCache.

He also discussed ServiceX, which is a system to provide transformations of large datasets nearly interactively.

ATLAS Distributed Data Management

Almost all of our data in Rucio (exception are some unregistered datasets in CERN EOS).

Data placement/movement governed by Rucio rules and available disk space.

Jobs go where the data is (except - Panda can move the data by creating temporary replicas).

That all works, specially for large workloads - MC generation, reprocessing campaigns, etc.

2

What is Virtual Placement?

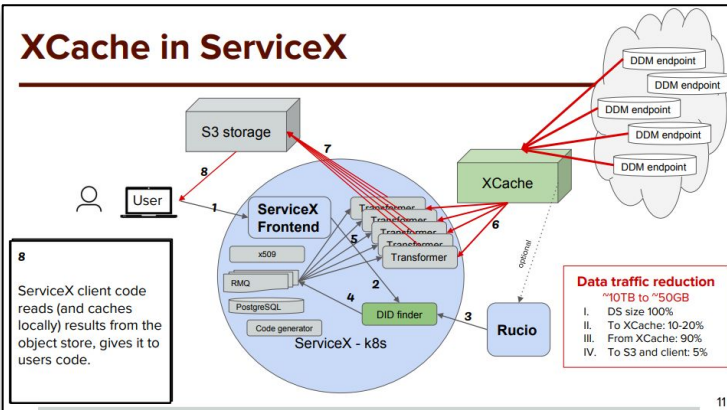
VP is a mechanism that enables efficient and reliable data access over WAN.

Expected benefits:

- Enables storageless sites
- Less WAN bandwidth usage
- Less rescheduling
- Faster task turnaround
- Less replicas

4

XCACHE in ServiceX

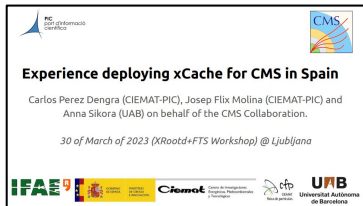


11

What is ServiceX?

[ServiceX](#) aims to provide nearly interactive filtering, enrichment, transformation of very large datasets and result delivery in multiple formats, with emphasis on pythonic style analysis.

10



Carlos Perez Dengra presented a study on the performance implications on CPU efficiency when executing the same job accessing MINIAOD data files from different sites.

CPU efficiency improves when the job being run reads data from a local xcache rather than from remote sites.

Try to save costs by keeping popular analysis datasets in local xcache.

Context

- **CMS jobs** have the **capability to read data remotely** using the **CMS XRootD Federation** (overflow to close sites, files opened in fallback and so on).
- We have been **exploring the xCache** service at PIC Tier-1 and CIEMAT Tier-2 centers to cache data which is read from remote centers
- xCache helps **reducing data access latency, improving CPU efficiency** and potentially **reducing the storage** deployed in the region
- We have deployed xCache services at both sites, and dedicated **studies and performance measurements** have been performed to configure the service

2

Conclusions

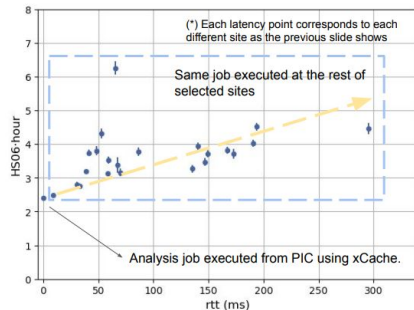
- During the initial phase of our project, we deployed the XCache from both PIC and CIEMAT separately to compare and validate the effects of different configurations on the service.
- Subsequently, we deployed the monitoring service and unified the service in PIC to efficiently and effectively serve data to the Spanish region.
- There is still some room to improve the re-accessibility of cached data → Update the the site configuration with the actual data popularity of the service in production.
- The results of executing controlled analysis jobs remotely accessing MINIAOD at various sites reveal a significant degradation of CPU efficiency with latency due network and distance (as expected)

10

CPU efficiency dependence of remote reads (2)

The cornerstone of the study is to evaluate which is the **impact over CPU efficiency of executing the same job accessing similar MINIAOD files from different sites.**

We have measured how **the CPU efficiency improves by executing the same job reading locally from the xCache** compared to doing it remotely -> **HS06-hours 'loss'** during the execution (see the figure)



9

Outlook

- The ongoing studies over the degradation of CPU efficiency will be complemented by dedicated studies evaluating the potential walltime saved by caching these data.
- Also, bringing the data closer to the cache increase the CPU efficiency of those jobs on our nodes. This effect has also to be well evaluated in jobs in production.
- Since xCache service is expected to alleviate the storage costs in the future, more dedicated studies will tell us how much storage we can save by keeping the most popular data for Analysis jobs within the cache.

11

Getting the most out of XCache



Ilija Vukotic
UChicago

27-31 March 2023, US, Ljubljana



Ilija Vukotic presented feedback on how to get the most out of XCache, comparing it with other solutions from the industry.

He provided a wishlist of which features he'd like to see based on prior experience with XCache and concluded that XCache could become an attractive solution for caching in other domains if some of these features were implemented.

Motivations

Now that XCache network is working and we have experience with VP, we can step back and see what it took and how it could be made better.

The difference between a single node cache and a CDN caching can't be overestimated.

We looked at possibilities for HTTP caching, tested and evaluated three options: Squid, Varnish, Nginx and Apache Traffic Server. This gave us experience of how modern CDNs are made, what features are important.



Fishes

Everyone gives you free caching as long as you are a small fish. Big fish features are paid for.

Squid - no multithreading so no big fish is using it. Also 3-6x less performant.

Varnish - a big fish gets: cache persistence, range requests, ssl, centralized configuration and monitoring tools, Massive Storage Engine...

NGINX - a big fish gets: active health checks, cache purging api, PLUS api for dynamic reconfiguration, JWT, SSO, build in dashboards.



My wishlist

- Disk failure handling
- Docker image, docker compose, k8s deployment, helm chart
- No burn-in test
- Liveness / heartbeats
- Configuration reload at runtime
- Client/origin/path management at config/runtime
- JWT, SSO
- Multiorigin, multiprotocol accesses
- Modern monitoring
- Self configuring
- CDN support (Centralized configuration, monitoring, path handling, SSO).



Conclusion



Some of these suggestions would be great addition even without going all the way.

Most of it was already done by different people but in disparate ways.

We could make all of these as a single separate service (just don't make it cmsd) so core code is not touched.

If you go all the way in, you could even sell it. Make most of it free and make CDN part pay-for but cheap (for the start). And free for academic users.




Data-Aware Scheduling with Hash-Based Data Distribution

M. Giffels, A. Gottmann, **R. Hofsaess**, M. Horzela, G. Quast, M. Schnepf | 30.03.2023


FTS and XRootD Workshop @ JSI (27-31 March 2023)



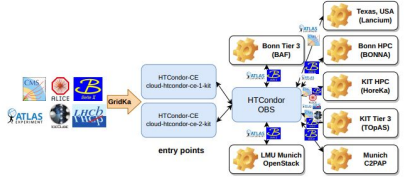
www.kit.edu

Robin Hofsaess discussed the potential benefits of data-aware scheduling of jobs for integrating opportunistic resources.

The idea is still in its conceptual stage. It consists in using hashes to distribute datasets across sites, then use the same hashes to distribute jobs that used the datasets across sites. The benefit is that this avoids the necessity of having a database service running to provide information about where each dataset is, and it allows easier integration of HPC sites, which often do not have a connection to fetch data on demand.




Integration of Opportunistic Resources



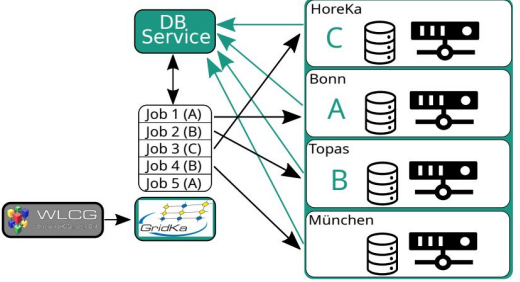
- Opportunistic resources are integrated dynamically with COBaID/TARDIS
- Very heterogeneous infrastructure
- No permanent, managed storage at integrated sites – only caches

Introduction ○○ Data-Aware Scheduling ○○○ Setup ○○○○ Other Projects ○○○○○

2/16 30.03.2023 FTS & XRootD Workshop 2023 @ JSI Robin Hofsaess




Data-Aware Scheduling for ORs

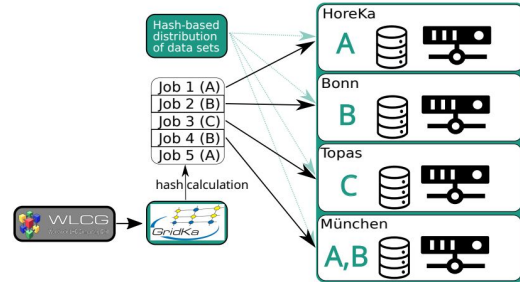


Introduction ○○ Data-Aware Scheduling ●○○ Setup ○○○○ Other Projects ○○○○○

4/16 30.03.2023 FTS & XRootD Workshop 2023 @ JSI Robin Hofsaess




Hash-Based Distribution of Datasets



Introduction ○○ Data-Aware Scheduling ○● Setup ○○○○ Other Projects ○○○○○

6/16 30.03.2023 FTS & XRootD Workshop 2023 @ JSI Robin Hofsaess



Summary

With our hash-based approach:

- We avoid actually keeping track of thousands of files on volatile caches
- The caching decision is determined locally (or centrally) → **horizontally scalable**
- With local on-the-fly caching, no prefetching is necessary (but possible)
- No database service needed (or at least more lightweight, tbd)
- Highly automatizable → **no active data management necessary**
- Allows inclusion of HPC sites without remote connection of WNs

→ reduction of data transfers and more efficient utilization of resources achievable

Introduction ○○ Data-Aware Scheduling ○○○ Setup ○○○○ Other Projects ○○○○○

10/16 30.03.2023 FTS & XRootD Workshop 2023 @ JSI Robin Hofsaess

Kingfisher: Storage Management for Data Federations

This project is supported by National Science Foundation under Grant OAC-2209645. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Brian discussed some ideas for managing data and his new plugin that implements some of these ideas, as well as a few use cases for the new plugin.

An example is using a cache as a temporary buffer to move data across sites, all while enforcing that space usage will remain under an existing quota without causing a failure after lots of data has moved, by preemptively creating a “lot” which is accounted for.

The Small Catechism of Storage Management

- **Allocation:** All usage of storage space throughout a distributed system is explicitly allocated.
- **Requirements:** Workloads explicitly state their storage space requirements.
- **Policy:** Allocated storage space has a size limit (e.g., byte and object limits), reclamation policies, access control rules, and an owner.
- **Recursive:** Storage space allocations are recursive; an owner of a space can partition it into sub-spaces and assign allocations to others.

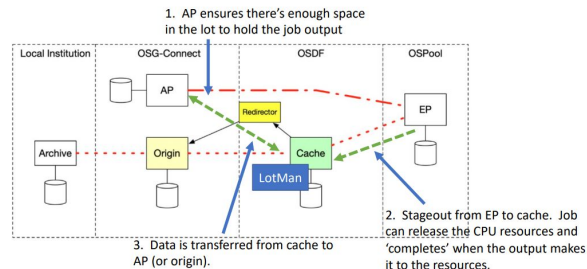
LotMan

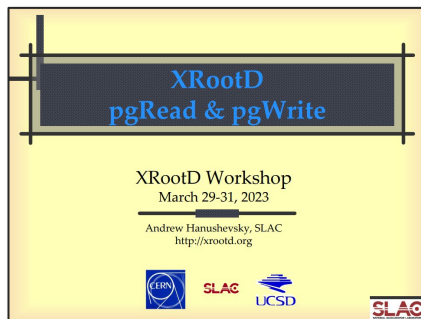
- The LotMan library is the new piece of software:
 - Exposes API to allow creation of “lots”.
 - Each “lot” has an owner, associated storage, reclamation policy, and ACLs.
 - LotMan is only the *accountant*. It does not measure use, move files, or delete files. It is expected to plug in to a larger system (HTCondor, XRootD) that provides the usage information.
- Example reclamation policies:
 - **Classic cache:** Delete any files when needed. (Typical cache setup)
 - **Temporary buffer:** Delete all files after time next Tuesday.
 - **Managed cache:** Delete files, as needed, until the lot is under 1TB of use.

Kingfisher

- The idea of the project is to demonstrate the impact of our small catechism of storage management.
 - The project aims to implement this in a reference platform and showing the value for a few science drivers.
- The reference platform is:
 - Using **XCache** (particularly in OSDF) as the space which will be managed.
 - **HTCondor** to communicate workload needs.
 - A new library, **LotMan**, to track allocations and policies.

Goal for “tomorrow”





Andy discussed in this presentation the features introduced recently in XRootD for pgread and pgwrite, which ensure data integrity while in motion, with low cost recoverability.

Reduced latency over the network. Not used locally.

Avoids retransmission of large files.

Client reverts to TLS when available if server does not support this feature.

Page read/write (pgRead/pgWrite)

- ✦ These are page aligned reads/writes
 - 4K pages on 4K boundaries
 - Does allow misalignment for 1st page (later)
 - Each page is check summed using crc32c
 - Follows IETF RFC 7143 standard
 - Client/server perform on-the-fly correction
 - Reads: client rereads pages in error
 - Writes: server supplies pages in error to rewrite

XRootD Workshop @ JSI

March 27-31, 2023

2



Why page read/write

- ✦ Transmission errors do occur
 - Some not caught by the TCP 16 bit checksum
 - Reports of errors on some international links
 - Typically during high usage periods
 - Avoids retransmission of large files (> 10GB)
 - When only a few bits are corrupted
 - Avoids having sticky errors in Xcache
 - A serious concern in a long-lived page cache

XRootD Workshop @ JSI

March 27-31, 2023

3



Final Notes on Async I/O

- ✦ Async only enabled for networked devices
 - Linux async I/O useless for locally attached disk
 - Implemented at user level via threads
- ✦ May change with new io_uring interface
 - Available since Linux Kernel version 5.1
 - RH 8.7 uses 4.18
 - RH 9.1 uses 5.14 (yay!)
- ✦ Adoption rates push this 1 to 2 years hence

XRootD Workshop @ JSI

March 27-31, 2023

7



Conclusion

- ✦ **XRootD** pgRead/Write is a game changer
 - Provides integrity for data in motion
 - Low cost for computation & recoverability
 - Integrated with integrity for data at rest (XrdOssCsi)
- ✦ Our core partners
 -
- ✦ Community & funding partners *(not a complete list)*
 -

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

March 27-31, 2023

14





Michal presented about the recently added erasure coding feature of XRootD.

It allows to save space by allowing data recovery from the parity blocks that are erasure coded, which depending on the configuration requires only 20% overhead instead of 100% overhead of a full replica.

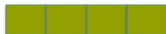
Performance is also better as each file is spread over several remote disks and access performance is the aggregate performance of all disks.

Writing

- Client buffers the data until it has a full block



- The block is divided into chunks



- The chunks are erasure coded (Intel ISAL Reed-Solomon)



- All chunks (data/parity) are checksummed (h/w assisted CRC32C)



30/03/2023

Michal Simon

3

Integrating XrdCl+EC with the xrootd storage

- Mode 1. Use xrootd storage directly as an EC store
 - Xroot protocol and xrootd client (with EC support) only
- Mode 2. Use XRootD Proxy as gateway to backend storage
 - Enable EC in the proxy's xrootd client component.
 - EC is invisible to the users
 - They use existing xrdcp/xrdfs, gfal, curl
 - Support all WLCG security, protocols, TPC, etc.
 - The backend xrootd storage is plain and simple

This mode is good for local administration

This mode is better for user access

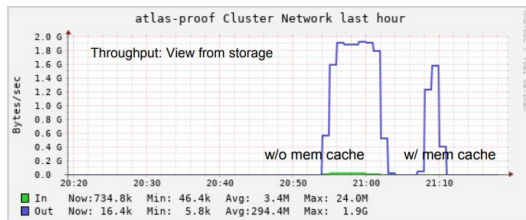
- The rest of the slides are about this mode

30/03/2023

Michal Simon

12

Aggregate read performance by many clients



- Read the pre-placed 312 data files, repeat 5 times
- Spread the read to 150 concurrent clients
- Memory cache clearly helped, it both
 - cache (reduce read from storage)
 - enable large block read (align with EC blocks)

30/03/2023

Michal Simon

16

Network upper limit

- 19 Gbit/s or
- 2.375GB/s

Summary

- XrdEc is a very performant implementation, we run almost at h/w speed (Intel ISAL, h/w assisted CRC32C)
- The tests at AliceO2 and SLAC yield very good results
- We started work on ops tools this summer (using student workforce), which resulted in a xrdrepair tool

30/03/2023





Michal Simon

18

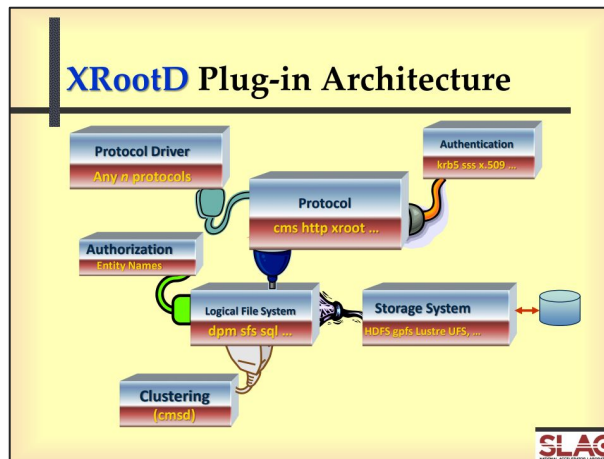
XRootD Server Plug-ins

XRootD Workshop
March 29-31, 2023

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

Andy gave an overview of all plugins available in XRootD.



Why Plug-ins?

- ✦ Makes it much easier to
 - Adapt, customize, add new features
- ✦ Any cons?
 - Need to know available plug-in points
 - These are documented but not in one spot
 - Described under the relevant directive
 - Usually xxxlib (e.g. xrootd.fslib)
 - However, we did make it a bit easier....

XRootD Workshop @ JSI

March 27-31 2023

3



The plug-in points

- ✦ A lot and more plug-ins than points!
- ✦ Get a list using **xrdpinfo** command

```
>xrdpinfo
Required >= 5.0 @logging
Optional >= 5.0 bwm.policy
Required >= 5.0 cms.perf
Optional >= 5.0 cms.unid
Optional >= 5.0 gsi-authfun
Optional >= 5.0 gsi-gmapfun
Optional >= 5.0 gsi-vomfun
Required >= 4.8 http.exchanger
Required >= 4.0 http.sectractor
Required >= 5.0 ofs.authlib
Required >= 5.0 ofs.csklib
Required >= 5.0 ofs.cmlib
Required >= 5.0 ofs.clib
Required >= 5.0 ofs.prelib
Required >= 5.0 ofs.xattrlib
```

```
Optional >= 5.0 oss.nameib
Required >= 5.0 oss.statlib
Optional >= 5.0 pfc.decisionlib
Required >= 5.0 pss.cachelib
Required >= 5.0 pss.cmlib
Required >= 5.0 sec.protocol
Required >= 5.0 sec.protocol-gsi
Required >= 5.0 sec.protocol-krb5
Required >= 5.0 sec.protocol-pwd
Required >= 5.0 sec.protocol-ss
Required >= 5.0 sec.protocol-unix
Untested >= 5.0 xrd.protocol
Required >= 5.0 xrd.monitor
Required >= 5.0 xrd.plugin
Required >= 5.0 xrootd.fslib
Required >= 5.0 xrootd.seclib
```

32 but actual 27

We are missing some plug-ins of plug-ins. Need a generalized way to capture the hierarchy. (future work)

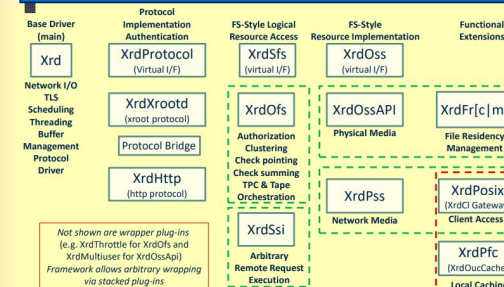
XRootD Workshop @ JSI

March 27-31 2023

4



Architectural Plug-in Interplay

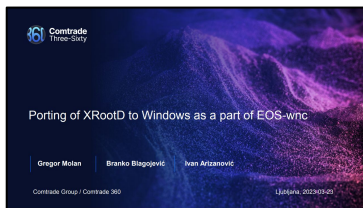


XRootD Workshop @ JSI

March 27-31 2023

8

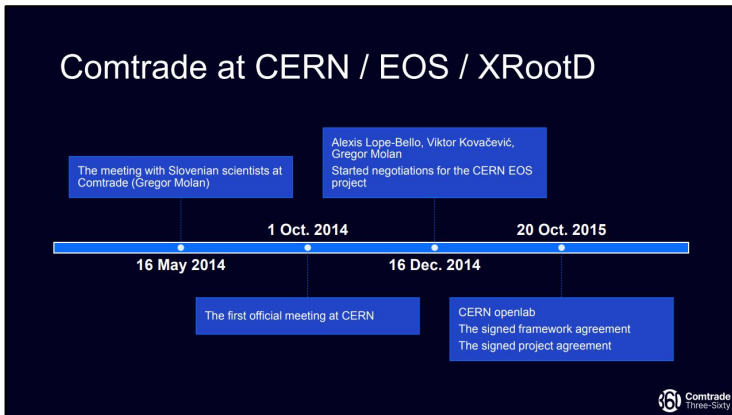




Comtrade 360, a company providing software and AI services, has developed a Windows client for EOS.

The Windows client is closed source and based on the old XrdClient code.

Communication happens with EOS servers via https.

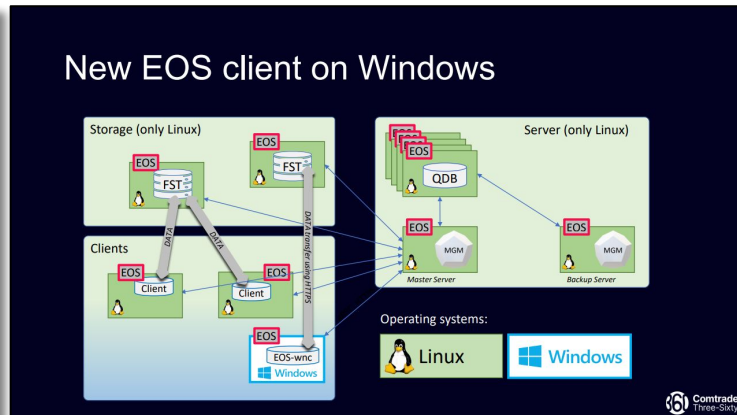


Old XRootD client XrdClient on Windows

Minimal requirement of libraries to build copy binary *xrdcp-old*:

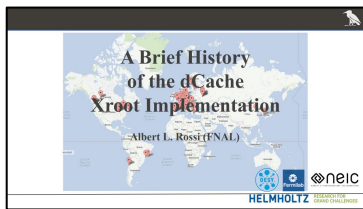
- libXrdAppUtils.so
- libXrdUtils.so
- libXrdClient.so

Successful porting!



Speed test results of EOS-wnc

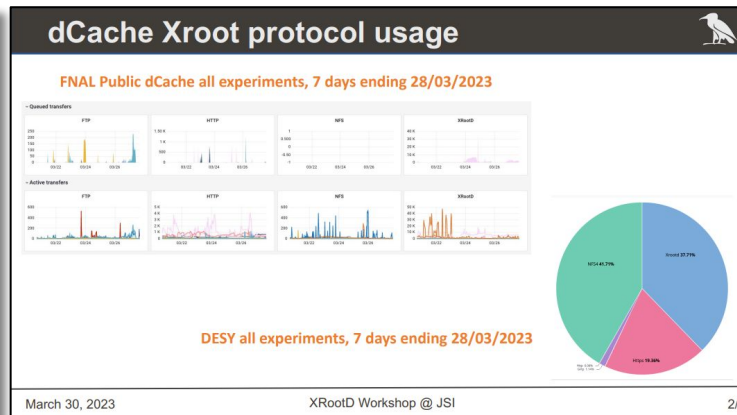
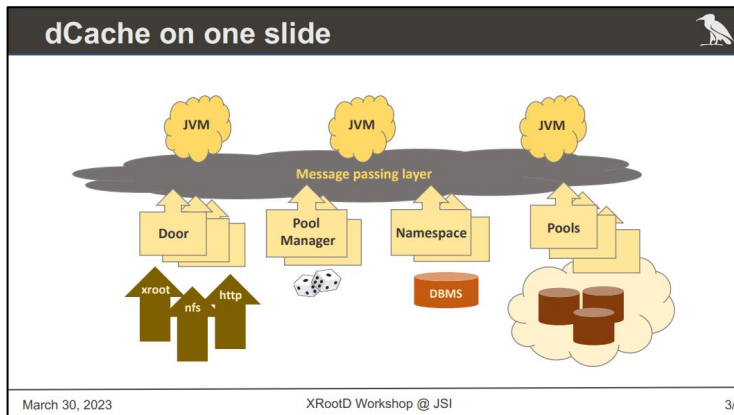
| | | 27 (checksums OK) | 28 (checksums OK) | 52 (checksums OK) | 11 (checksums OK) | Number of files | 2 |
|----------|--------------------|-------------------|-------------------|-------------------|-------------------|-----------------|------|
| | | | | | | File size [MB] | 2000 |
| | | Test (M/R/s) | | | | Avg time [s] | |
| | | min | max | avg | trim35% | | |
| Upload | Linux | | | | | | |
| | EOS: xrdcp command | 329.49 | 405.27 | 371.03 | 371.17 | 5,39 | |
| | EOS: Fusex | 157.82 | 222.63 | 210.26 | 210.43 | 9,49 | |
| | IBM Spectrum Scale | 283.91 | 318.22 | 294.47 | 289.28 | 6,79 | |
| | Ceph on Linux | 143.50 | 193.37 | 157.26 | 158.17 | 12,69 | |
| Win | EOS-wnc | 158.40 | 331.09 | 231.25 | 227.75 | 8,65 | |
| | EOS-drive ST | 212.22 | 294.47 | 237.44 | 234.72 | 8,42 | |
| | Ceph on Win | 128.18 | 158.19 | 153.32 | 154.04 | 11,03 | |
| | Hadoop on Win | 4.61 | 4.72 | 4.66 | 4.66 | 428,85 | |
| | EOS: xrdcp command | 328.68 | 365.97 | 353.00 | 354.47 | 5,67 | |
| Download | Linux | | | | | | |
| | EOS Fusex | 218.66 | 233.36 | 227.13 | 227.15 | 8,81 | |
| | IBM Spectrum Scale | 328.95 | 364.96 | 342.54 | 341.85 | 5,84 | |
| | Ceph on Linux | 188.93 | 355.49 | 265.08 | 264.04 | 7,54 | |
| | Hadoop on Linux | 5.28 | 10.63 | 10.12 | 10.15 | 197,66 | |
| wnc | EOS-wnc | 119.92 | 213.86 | 170.17 | 169.49 | 11,75 | |
| | EOS-drive ST | 179.86 | 210.49 | 190.24 | 189.72 | 10,51 | |



Albert Rossi presented dCache, which is a similar project to XRootD that implements the XRoot protocol in Java.

The XRoot protocol accounts for about 1/3 of dCache usage.

Albert presented the many features added by dCache recently and discussed also some problems encountered during the implementation, especially related to token support. Some features, like pgread/pgwrite are still not implemented.



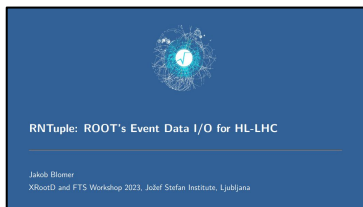
- ### dCache Xroot development 2018-2023
- | Interoperability | dCache Enhancements |
|--|--|
| 1. Added Third-Party Copy (TPC) | 1. Allowed client to reattempt open on pool when I/O stalls |
| 2. Added GSI TPC proxy management | 2. Fixed <code>-P</code> handling: moved upload commit for persist-on-successful-close to the pool |
| 3. Expanded Signed Hash Verification | 3. Added <code>authn</code> protocol chaining and multiple (security) protocol door |
| 4. Implemented <code>unix</code> authentication | 4. Changed chunk size to conform with standard (8 MiB) for both server and TPC client |
| 5. Regularized error codes/messages | 5. Broke up write into max frame-size chunks |
| 6. Added checksum CGI handling to door | 6. Added ability to proxy transfers through door |
| 7. Added support for <code>tried/zo</code> CGI | 7. Added support for relative paths in URL |
| 8. Added TLS support to door and pool | |
| 9. Added token authorization support | |
| 10. Added token authentication support (ZTN) | |
| 11. Added query support for TPC on pools | |
| 12. Added support for <code>kxr_fattr</code> , <code>kxr_prefname</code> | |
- March 30, 2023 XRootD Workshop @ JSI 6/

Currently not supported by dCache

| CLIENT REQUEST TYPES | QUERY REQUEST TYPES | OPEN REQUEST OPTIONS |
|---------------------------|--------------------------|--|
| <code>kxr_gpfile</code> | <code>kXR_QStats</code> | most of them are ignored and fall over to default behavior |
| <code>kXR_prepare</code> | <code>kXR_QPrep</code> | |
| <code>kXR_bind</code> | <code>kXR_Qspace</code> | |
| <code>kXR_pgwrite</code> | <code>kXR_Qckscan</code> | |
| <code>kXR_truncate</code> | <code>kXR_Qvisa</code> | |
| <code>kXR_pgread</code> | <code>kXR_Qopaque</code> | |
| <code>kXR_writev</code> | <code>kXR_Qopaquf</code> | |
| | <code>kXR_Qopaqug</code> | |

March 30, 2023 XRootD Workshop @ JSI 21/

Day 3 (Friday)



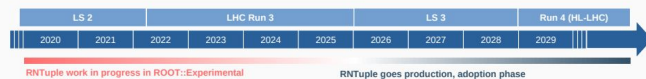
Jakob presented RNTuple, which is the successor in ROOT to the TTree, planned for use in production in Run 4 and beyond.

The new format still uses ROOT's TFile as its container format, but restructures what are now TBaskets in TTree for better performance, both in terms of data size, but more importantly, for reading data while running analysis jobs.

What is RNTuple?

Based on 25+ years of TTree experience, RNTuple is a redesigned I/O subsystem aiming at

- Less disk and CPU usage for typical analysis files and tasks
 - 10% to 20% smaller files, $\times 2$ –5 better single-core performance
 - 10 GB/s per box and 500 MB/s per core sustained end-to-end throughput (compressed data to histograms)
- Systematic use of exceptions to prevent silent I/O errors
- Efficient support of modern hardware (async, parallel I/O, many-core friendly, GPU data transfer)
- Native support for object stores (besides local and remote ROOT files)

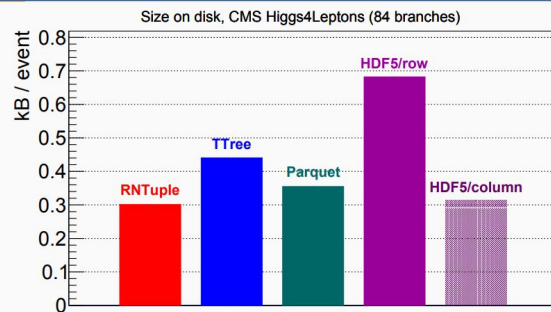


March 31, 2023

RNTuple – XRootD Workshop 2023

1 / 16

Performance plots (I)

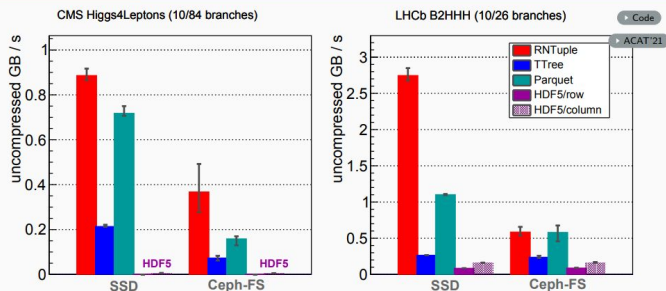


March 31, 2023

RNTuple – XRootD Workshop 2023

2 / 16

Performance plots (III)



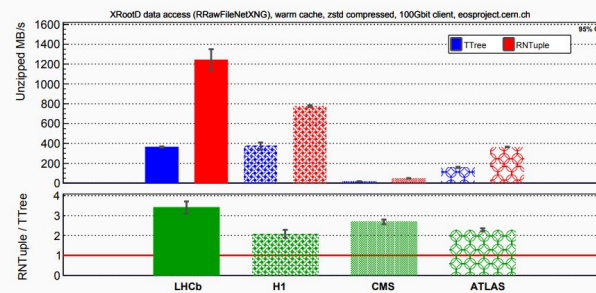
March 31, 2023

RNTuple – XRootD Workshop 2023

4 / 16

Performance plots (V)

Single-core end-to-end throughput (compressed data to histogram), eosproject.cern.ch



March 31, 2023

RNTuple – XRootD Workshop 2023

6 / 16

LHCOPN/LHCONE Update

Marian Babik / CERN, Edoardo Martelli / CERN
XRootD & FTS Workshop @JSI



Marin Babik discussed LHCONE infrastructure upgrades and the SciTags initiative, which is a project for tagging network packets to enable higher level studies of how data is used worldwide. SciTags can also be used in traffic shaping to avoid bursts that can cause network congestion.

FTS & XRootD

FTS and XRootD are key to reaching full potential in programmable networks

XRootD already provides [SciTags implementation](#) (from 5.0+)

- Enables using SciTags by R&E networks analytics (ESnet6 High-Touch)
- Currently looking for sites that would configure/test this in production

FTS/gfal2 needed to propagate SciTags to storages

- Extensions proposed for XRootD and HTTP-TPC

FTS as a transfer broker is key component for NOTED

- Understanding where/when on-demand network provisioning is needed
- Combined with analytics to determine duration, capacity, etc.

Programmable networks can be beneficial for FTS and XRootD to get better network performance, flexibility and monitoring

XRootD & FTS Workshop @JSI 20

The SciTags Initiative

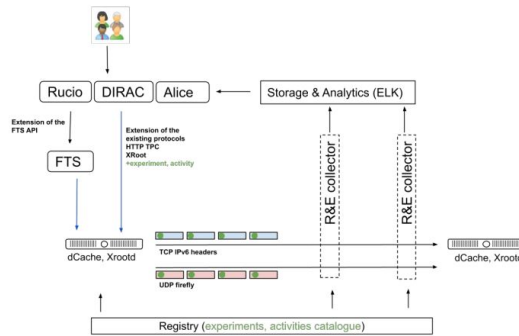
To manage our packet marking and flow labeling efforts, we started the **Scientific Network Tags (scitags)**: an initiative promoting identification of the science domains and their high-level activities at the network level.

The initiative is managed by the RNTWG and is working to:

- Enable tracking and correlation of network transfers with Research and Education Network Providers (R&Es) network flow monitoring.
- Supporting collaborations to better understand network use and impact
 - Improve visibility into how network flows perform (per activity) within R&E segments
 - Get insights into how experiment is using the networks, get additional data from R&Es on behaviour of our transfers (traffic, paths, etc.)
- Allow sites and end users to get detailed visibility into how different network flows perform
 - Network monitoring per flow (with experiment/activity information)
 - E.g. RTT, retransmits, segment size, congestion window, etc. all per flow

XRootD & FTS Workshop @JSI 14

How scitags work



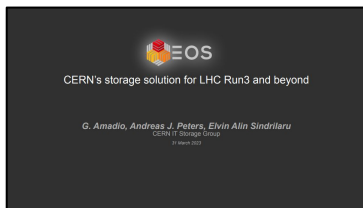
HEPIX IPv6 Workshop

16

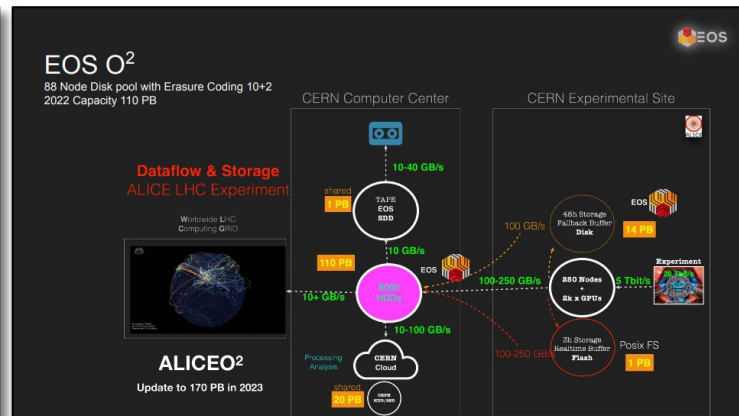
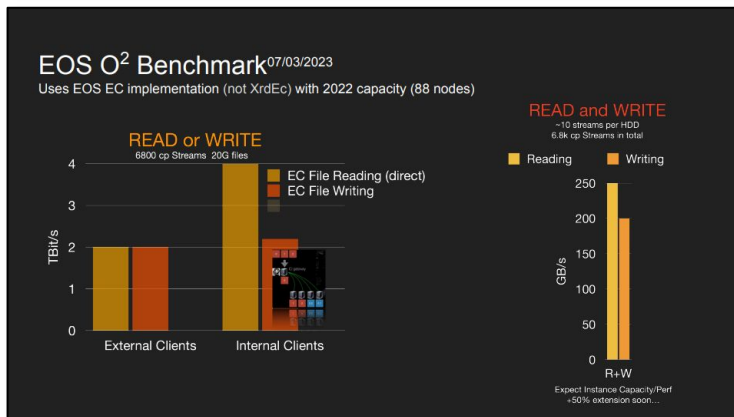
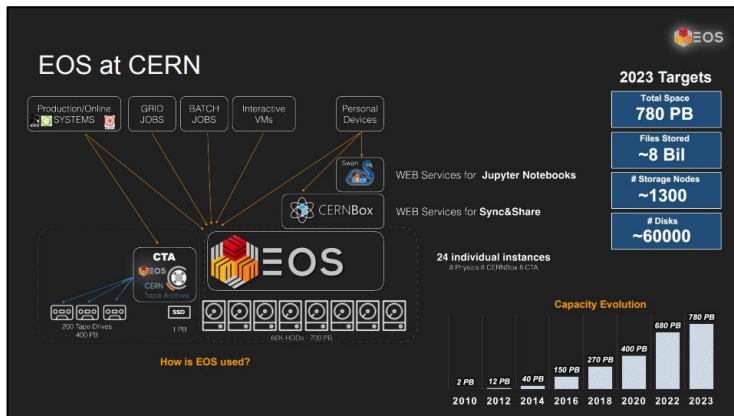
Summary

- Significant capacity increases on LHCOPN/LHCONE
 - Important to ensure this capacity becomes available end-to-end (WAN/LAN)
- Number of ongoing R&D projects
 - Network Orchestration
 - NOTED and SENSE are leading projects in programmable networks
 - Network Visibility and Pacing
 - Technology available to make all our flows visible to R&E networks
 - Plan to start deployment and increase adoption as we approach DC24
 - Network Pacing activity will start, aiming to improve performance of the end-to-end transfers
 - Network Routing and Forwarding
 - Investigating technologies that can provide networking beyond LHCONE
- Still a lot of work ahead in many areas, important to continue the efforts
- **We welcome additional contributions/project ideas/experiments; contact us if you are interested!**
 - [LHCOPN/LHCONE Workshop 18-19 April in Prague](#)

XRootD & FTS Workshop @JSI 21



Presented an overview of EOS at CERN, and used the Alice O² site to showcase the benefits of using erasure coding. A benchmark performed by Andreas showed that we can reach about 500GB/s read speed using the whole cluster, and simultaneously 250GB/s read + 200GB/s write speeds are also possible.



- ### EOS & XRootD
- #### Final remarks
- For EOS releases still building own internal XRootD package due to critical/cutting edge bug fixing for production - **hopefully soon not necessary anymore for V5**
 - Since many years excellent support and teamwork within the XRootD collaboration
 - XRootD provides an excellent client-server framework for physics data storage
 - Core framework for EOS moving exabytes reliably each year (almost)

A Prometheus XRootD exporter based on mpxstats

XRootD and FTS Workshop @ JSI, Ljubljana 2023

Jan Knedlik, GSI

GSI Helmholtzzentrum für Schwerionenforschung

31.03.2023

Jan Knedlik, GSI
A Prometheus XRootD exporter based on mpxstats

GSI Helmholtzzentrum für Schwerionenforschung
1 / 12

Jan Knedlik discussed an exporter of monitoring data for Prometheus based on mpxstats.

A discussion followed about adding native support for Prometheus monitoring to XRootD. Brian opened a pull request over the weekend with a draft implementation for such a plugin.

Grafana & Prometheus

- Used as monitoring/visualization tool
- Easy to use/implement
- Prometheus as database for time series data (in simple cases)
- Part of monitoring tooling consolidation @ GSI data group

-> get XRootD metrics via an exporter in some way

Jan Knedlik, GSI
A Prometheus XRootD exporter based on mpxstats

GSI Helmholtzzentrum für Schwerionenforschung
2 / 12

mpxstats

- XRootD binary for **aggregated statistics reports** (xrd.report)
- Listens on configured UDP port, prints to stdout
- Thanks to whoever implemented it in a unix-like way!
- A lot of metrics (open connections, data rate in/out, files open, inodes, ...)

Jan Knedlik, GSI
A Prometheus XRootD exporter based on mpxstats

GSI Helmholtzzentrum für Schwerionenforschung
3 / 12

XRootD Prometheus exporter

- Gets service metrics of the XRootD service
- Listens via mpxstats
- Python 3.X Prometheus exporter (~100LOC)
- XRootD's statistics reports via mpxstats -> prometheus metrics

Jan Knedlik, GSI
A Prometheus XRootD exporter based on mpxstats

GSI Helmholtzzentrum für Schwerionenforschung
5 / 12

Workflow

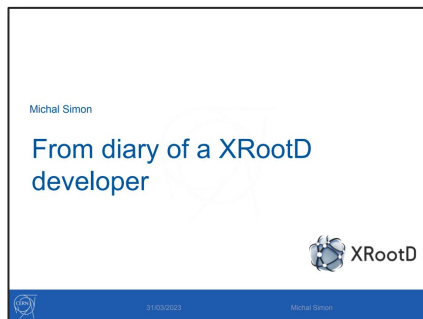
```

graph LR
    subgraph "Grafana & Prometheus Server"
        GS[grafana service] -- "<promql>" --> PS[prometheus service]
    end
    subgraph "XRootD Server"
        XS[xrootd service] -- "<mpxstats>" --> XE[xrootd exporter]
    end
    PS -- "<http>" --> XE
  
```

Figure 1: Exporter Workflow

Jan Knedlik, GSI
A Prometheus XRootD exporter based on mpxstats

GSI Helmholtzzentrum für Schwerionenforschung
6 / 12



Michał had the final presentation, where he told us the stories about interesting situations that happened during XRootD development.

Few facts ;-)

- First commit (29/05/2015): *RPM: Remove xrootd-libs dependency for xrootd-client-compat*
- First PR (#247): *Release Stream lock before invoking callbacks. #216*
- First release (29/05/2015): 4.2.1
 - Single commit: *[XrdCl] Make sure kXR_mkdir is set for classic copy jobs when the destination is xrootd (backward compatibility fix).*
- Last release (18/08/2022): 5.5.1
- Commits: 1'819
 - Which makes me the second most active contributor after Andy ;-)
- Releases: 55 (1 major, 16 feature, 38 bugfix)



31/03/2023

Michal Simon

Year 2021

- **PgRead / PgWrite** (xrdcp); **XrdEc**
- **GPFS vs sendfile** syscall
 - Recalling missing pages **amplifies CPU** usage
 - Instead of fetching data using 8MB block we effectively used 4KB



31/03/2023

Michal Simon

Year 2022

- **Record / replay** plug-in
- **SSS authentication vs OpenSSL 3**
 - Fixed in: *[XrdCrypto] bf32: respect the key length when encrypting/decrypting.*



31/03/2023

Michal Simon

Thank you!

- Big thank you to the whole XRootD community! :-)
- ... and a **special, big thank you to Andy!** :-)



31/03/2023

Michal Simon



XRootD