





# CERN's storage solution for LHC Run3 and beyond

***G. Amadio, Andreas J. Peters, Elvin Alin Sindrilaru***

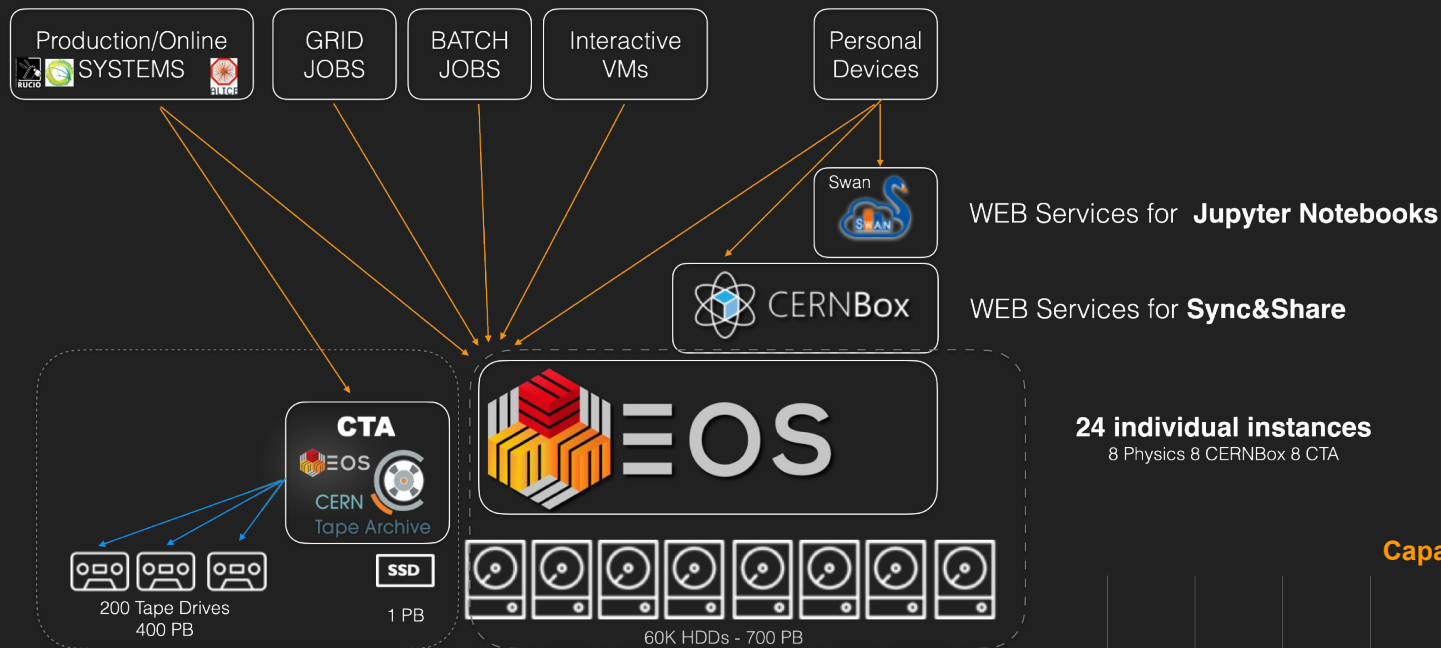
CERN IT Storage Group

*31 March 2023*

# Overview

- EOS Services at CERN
- EOS Usage statistics
- EOS Architecture
- EOS & XRootD
  - Client
  - Server
  - Async Close
  - Token
- Final Remarks

# EOS at CERN



How is EOS used?

## 2023 Targets

Total Space  
**780 PB**

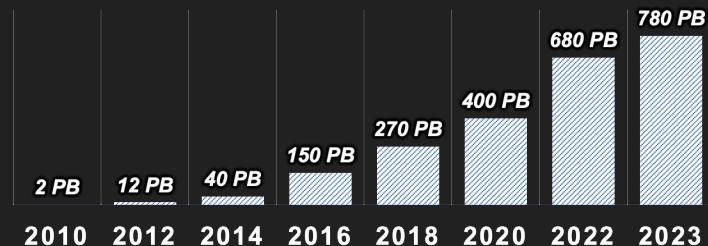
Files Stored  
**~8 Bil**

# Storage Nodes  
**~1300**

# Disks  
**~60000**

**24 individual instances**  
8 Physics 8 CERNBox 8 CTA

## Capacity Evolution



# EOS Deployments at CERN

8 Physics, 8 User/Project (CERNBox), 8 CTA EOS Instances at CERN + various pre-production installations

Number of F...

**7.50** Bil

Number of ...

**549** Mil

Total Space

**668.93** PB

Used Space

**462.91** PB

Difference

**-3.28** PB

Free Space

**206.0** PB

Current Writ...

**4.62** K

Current Rea...

**68.2** K

IOPS

**2M** io/s

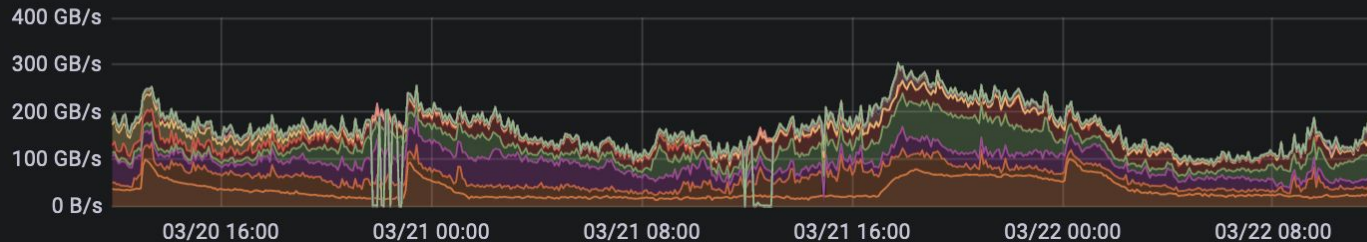
Write Throughput

**14.0** GB/s

Read Throughput

**98.2** GB/s

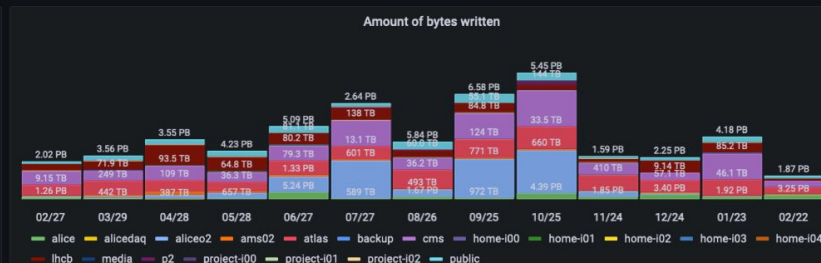
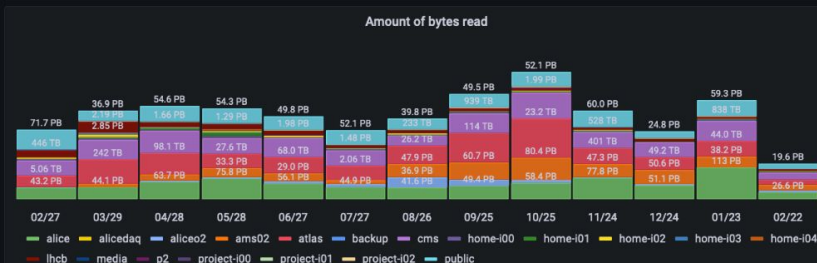
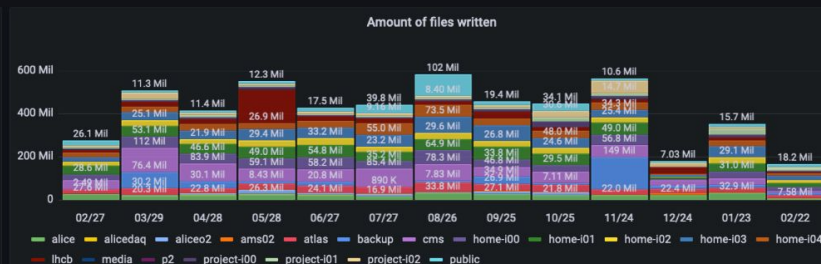
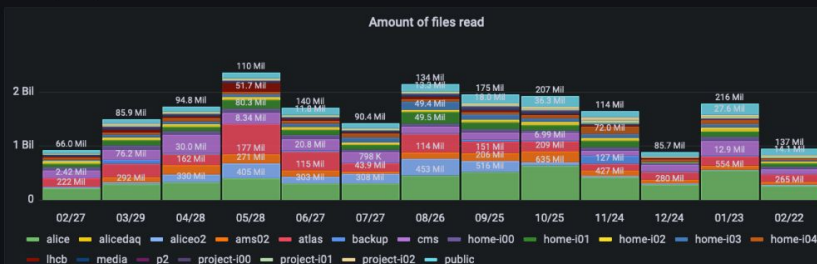
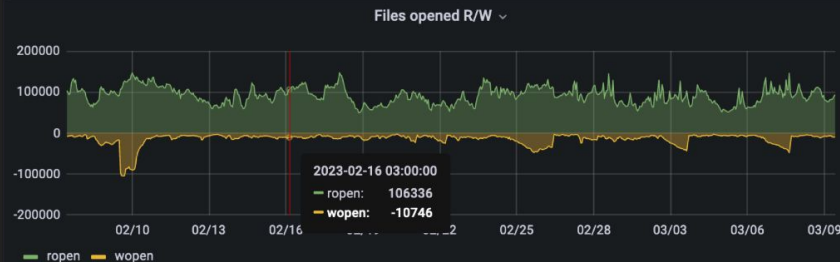
Cluster Network Rates (out)



# EOS Usage at CERN

## IO Statistics of the last 12 month

▼ KPIs per experiment @ 🗑



### Total data read per protocol: All and instance: All

- None
- fuse::bi
- fuse::lxlplus
- root.exe
- http

Value: 1.60 EB Percent: 40%

Value: 1.12 EB Percent: 28%

Value: 215 PB Percent: 5%

Value: 193 PB Percent: 5%

Value: 169 PB Percent: 4%

21.1 Bil

3.95 EB

5.39 Bil

590 PB

Reading 2020: 2.5 EB => [last 12mo]: 3.95 EB

+58%

# EOS O<sup>2</sup>

88 Node Disk pool with Erasure Coding 10+2  
2022 Capacity 110 PB

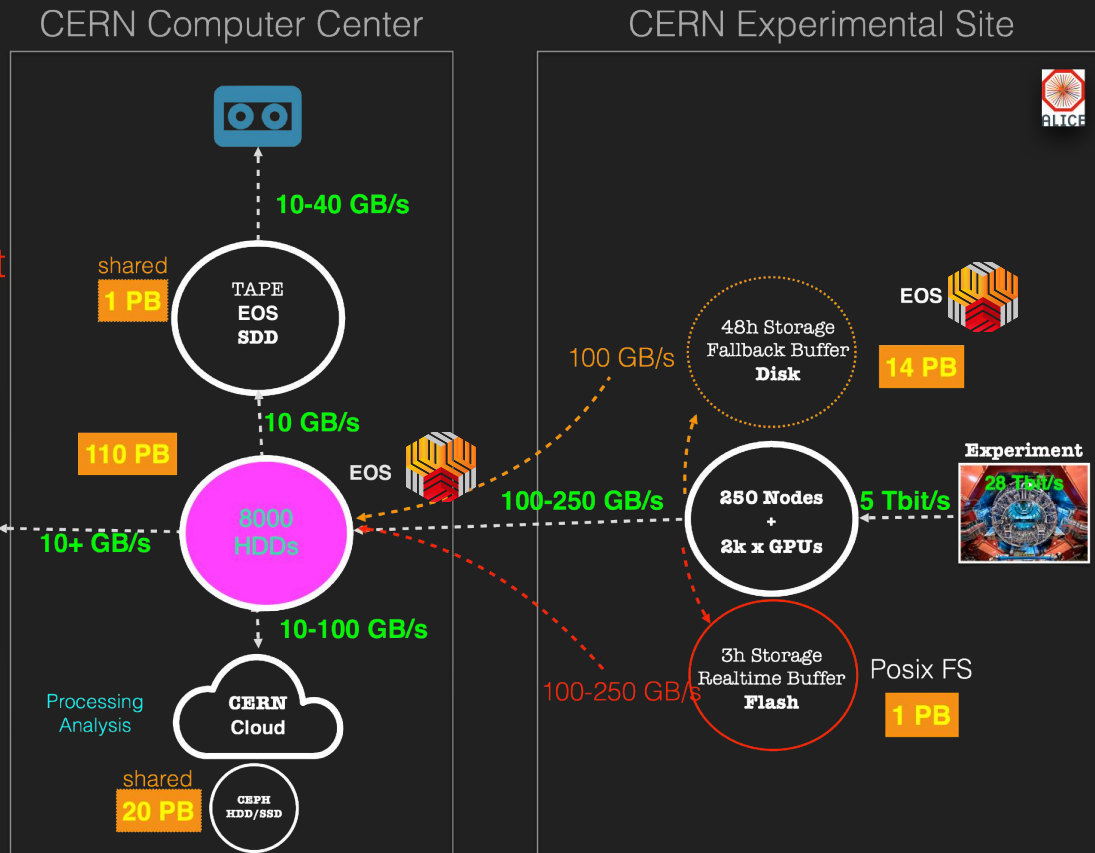
## Dataflow & Storage ALICE LHC Experiment

Worldwide LHC  
Computing GRID



## ALICEO<sup>2</sup>

Update to 170 PB in 2023





# EOS O<sup>2</sup>

New Standard Hardware for EOS Physics Storage

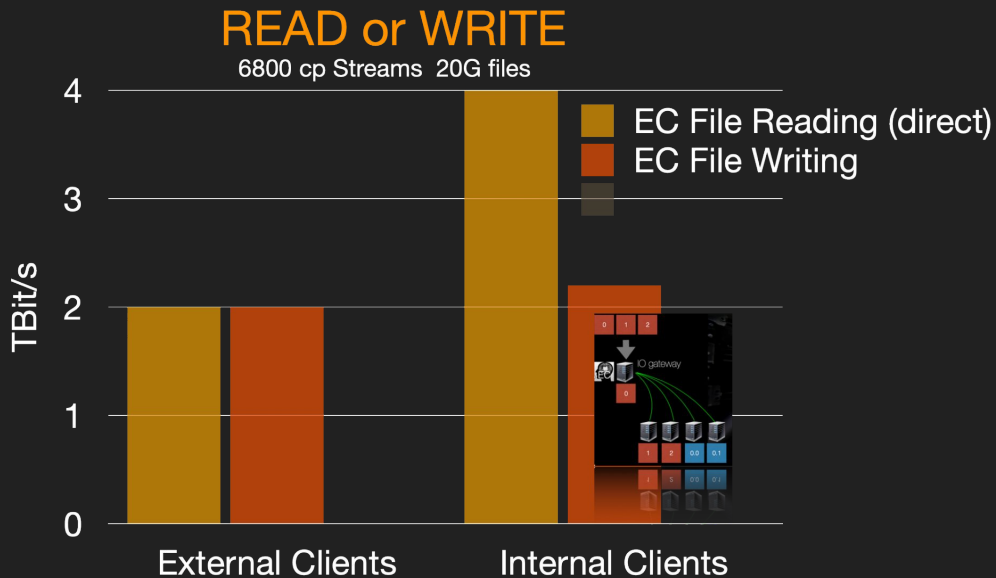
O<sup>2</sup> disk server have 96 HDDs with 100GE ethernet connectivity

- This type of hardware is the new standard getting installed also in other LHC experiment EOS instances [HDD sizes 14++ TB]
- Performance baseline is around 6 GB/s streaming reads and 3.5 GB/s streaming reconstruction/writes with erasure coding per disk server
- Excellent Run-3 operation experience for ALICE with erasure coding RS 10+2
  - like 3 replicas but only 20% volume overhead
  - bandwidth per file up to 2.5 GB/s - >800 IOPS



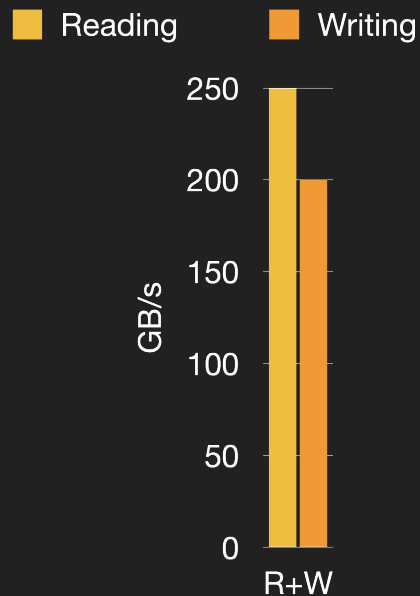
# EOS O<sup>2</sup> Benchmark<sup>07/03/2023</sup>

Uses EOS EC implementation (not XrdEc) with 2022 capacity (88 nodes)



## READ and WRITE

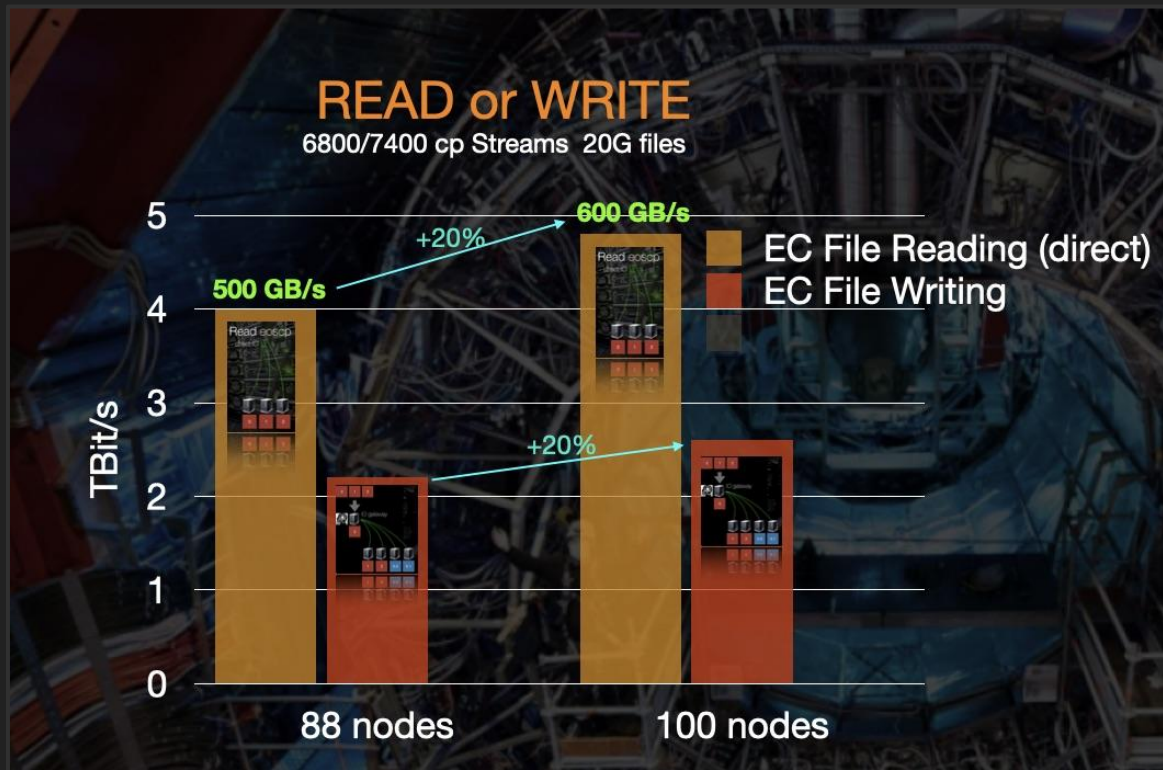
~10 streams per HDD  
6.8k cp Streams in total



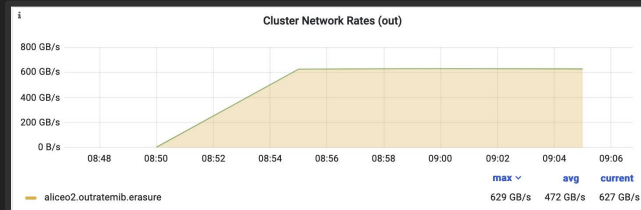
Expect Instance Capacity/Perf  
+50% extension soon...

# EOS O<sup>2</sup> Benchmark<sup>23/03/2023</sup>

Uses EOS EC implementation (not XrdEc) with extended capacity (100 nodes / +20 % space/hdds)



- 20% more performance



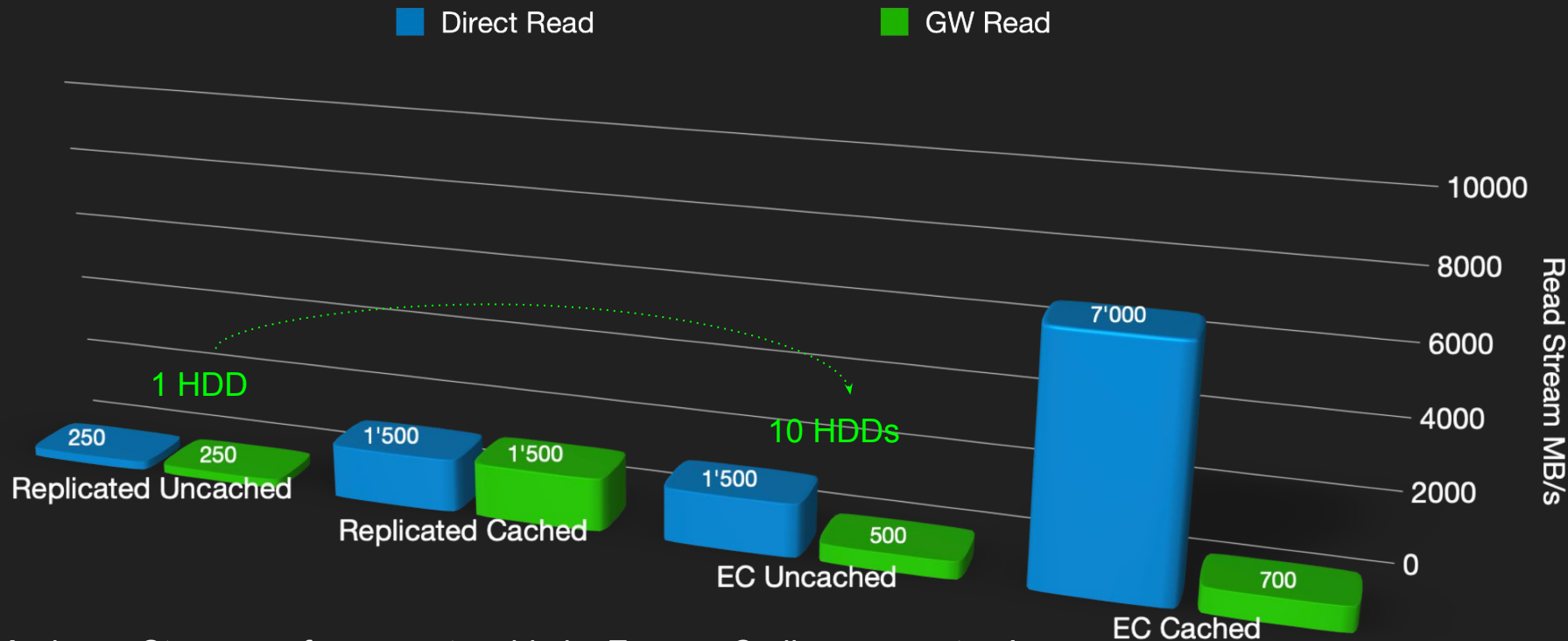
Network Output Rate

External Nodes	Internal Nodes	Filesize	Parallism	Total Clients	Streams	Server	HDDs
48	100	20 GiB	50	7400	88800	100	9598

Test Setup

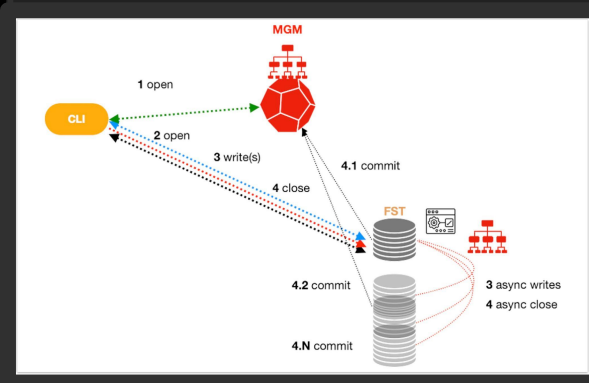
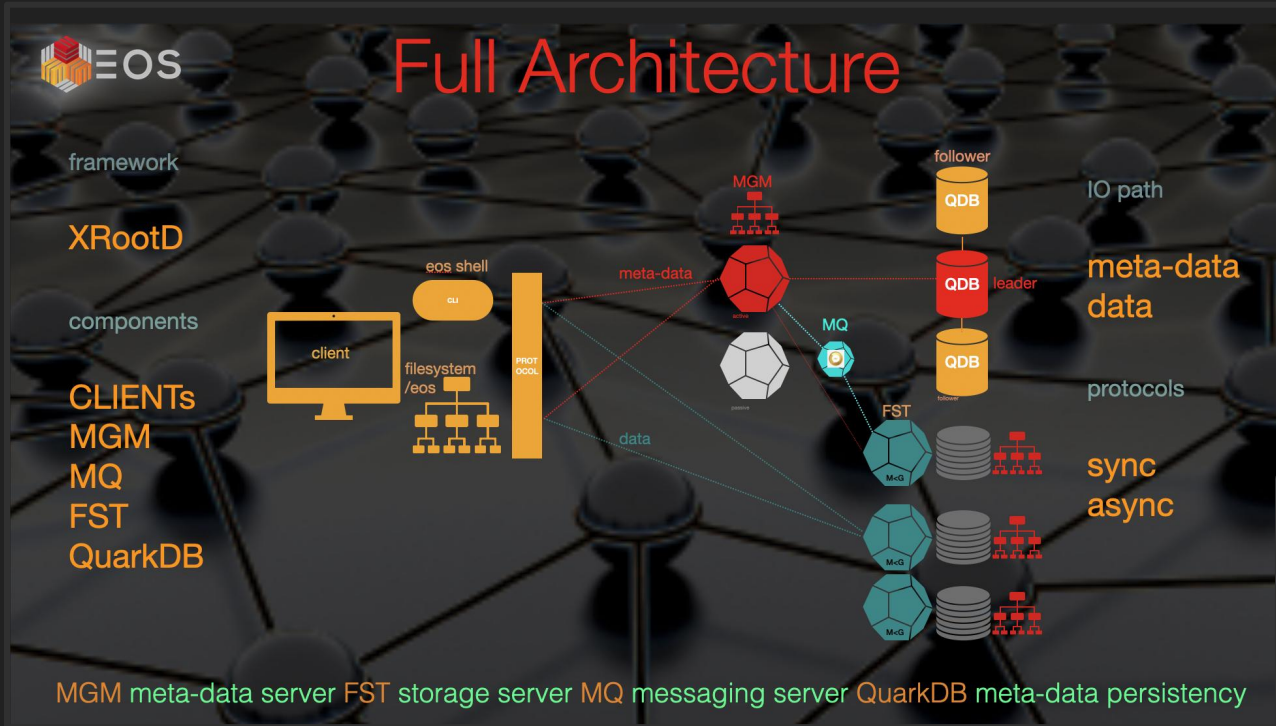
# EOS O<sup>2</sup> Single Stream Read Performance Evolution

Replication vs EOS Erasure Coding in O<sup>2</sup> using RS 10+2



Maximum Stream performance tunable by Erasure Coding parameters!

# EOS - behind the curtain ...



# EOS & XRootD

## XRootD Framework

xrootd.org

meta-data

messaging

data

meta-data persistency

MGM

MQ

FST

QuarkDB

threadpools

framework

process  
xrootd



XRootD



XRootD



XRootD



XRootD



XRootD

library implementing  
the service

libXrdEosMgm

libXrdMqOfs

libXrdEosFst

libXrdQuarkDB

s.c. OFS plug-ins

protocol plug-in

library providing  
protocols on additinal  
port

XrdHttp

XrdHttp

grpc

grpc



external protocol implementation  
with independent thread pool

# EOS & XRootD

## XrdCl Client Usage

- Most demanding use case is **eosxd(3)** [ FUSE mount ]
  - Highly multi-threaded
    - Uses **XrdCl::File** for data operations
      - good parallelism
    - Uses **XrdCl::FileSystem::Query** for namespace operations
      - bottleneck because server side processing serializes all requests
        - limits create/s for a single client - maybe wrong type of plug-in call?
  - Extremely sensitive to XrdCl bugs
    - clients hanging, idle batch jobs, even complete node lookups ( coupling on a node via **df** etc. )
    - at any moment we have 20k-30k clients - very good QA platform 😊
    - a lot of bugs have been found ( and fixed ) in V5
      - we have reached a similar stability now with V5 as V4 ( EOS client 5.1.14 vs 4.8.51 )

### FUSE performance

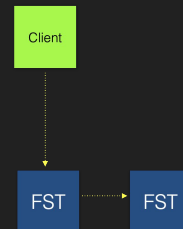
- 100 GE network
- Creations
- Single Stream Performance

Single Client	<b>eosxd3</b>
Seq. Creation	700 Hz
Par. Creation	1000 Hz
Seq. Read	1.6 GB/s
Seq. Write	900 MB/s

# EOS & XRootD

## XrdCl Client Usage

- Communication between storage daemon (FST) uses client for
  - Writing for replicated and erasure coded files
  - Reading erasure coded files
  - TPC
- again very sensitive to client bugs, even more concurrency
  - Several client bugs have been identified and fixed also here!

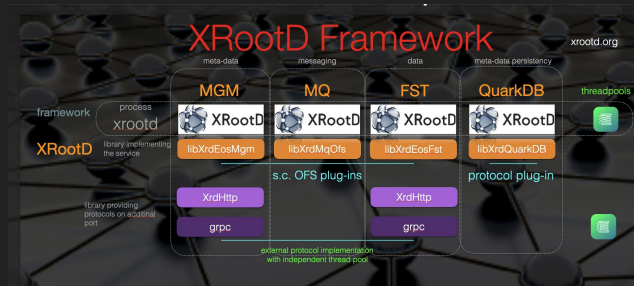




# EOS & XRootD

## XRootD Server Usage

- Namespaces (MGM) implemented as OFS plug-in
  - single thread-pool in XRootD server for everything
    - we have **no way to prevent DOS** in the authentication process
      - EOS has mechanisms to restrict number of active threads per users and request rates, but they only apply after authentication has been done
- Storage Server (FST) implemented as OFS+OSS plug-in
  - we have implemented direct IO in our OSS with very good results, could be useful to have this in the default OSS (can be useful for XCache, NVMEs etc.)
- In general
  - core server is very stable
  - HTTP** is still **moving target**
    - streaming performance very good
    - requests/s scaling/latency worse (saturates around 100kHz)
    - TPC implementation for WAN suboptimal (libcurl **pipelining vs multiplexing vs chunking** )
  - Token** authentication, authorization still **moving target** and often confusing
  - parallel socket** implementation in XRootD with **low performance** for single file transfers in LAN (not better than single socket!)
    - but managed single file transfer of 10 GB/s using extreme copy with **manual connection multiplexing** ( why not an option for xrdcp and xrdcp/tpc ? )

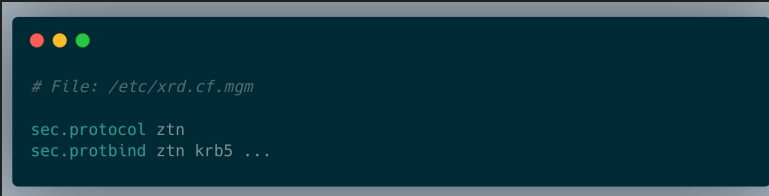


# EOS & XRootD - Use of Async Close

- **EOS file checksum mechanism**
  - Best-effort: for streaming files check is computed "in-flight"
  - For non-streaming cases the file is re-read during the close operation
- **Problem**
  - For large files (>10GB) can take more than the default XRD\_STREAMTIMEOUT
- **Side-effects**
  - Client sees a timeout error and a failed close operation
  - The server happily re-computes the checksum and closes the file successfully
- **Mitigation**
  - Use the async close functionality (SFS\_STARTED / kXR\_waitresp)
  - The client will receive a notification from the server then the operation is done
  - The client will wait for a certain amount of time for the response
- **Outcome**
  - Deployed in production instances and no more complaints from the users
  - Hit a few nasty bugs along the way but now running stable

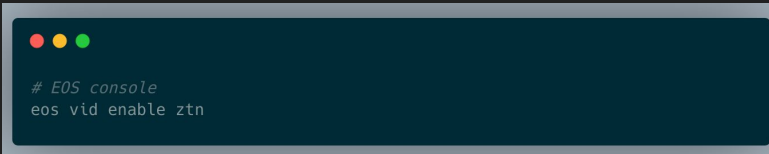
# EOS & XRootD - Integration of XRootD token support

- EOS has now full support for **tokens over xroot** protocol (since eos-5.1.15)
  - Configuration wise identical to XRootD ztn

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays configuration lines for XRootD.

```
# File: /etc/xrd.cf.mgm  
  
sec.protocol ztn  
sec.protbinding ztn krb5 ...
```

- **ztn** support needs to be enabled explicitly in EOS

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays a command to enable ztn support in EOS.

```
# EOS console  
eos vid enable ztn
```

- **EOS HTTP (TPC)** already has support for different types of tokens (macaroons/scitokens)
  - EOS relies on XrdHttpTPC plug-in
- EOS also supports **SE-tokens** called “EOS tokens” - to use these over ztn, token validation has to be disabled using **ztn -tokenlib none** ( the scitokens library cannot validate EOS tokens )

# 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)

Thank you! Questions? Comments?



# EOS 2023 Workshop

24–27 Apr 2023  
CERN  
Europe/Zurich timezone



Overview

Scientific Programme

Call for Abstracts

Timetable

Contribution List

My Conference

My Contributions

Book of Abstracts

Registration

Participant List

Privacy Information

Videoconference



You are invited to join the 7th EOS workshop end of April at CERN

<https://indico.cern.ch/event/1227241/>

