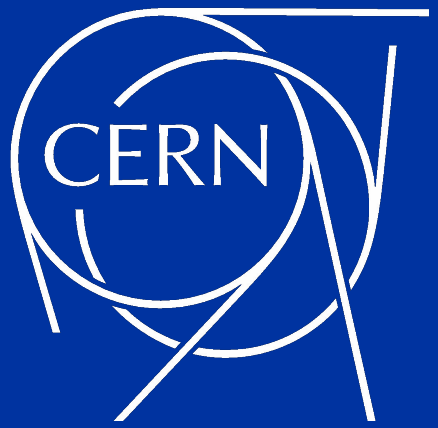
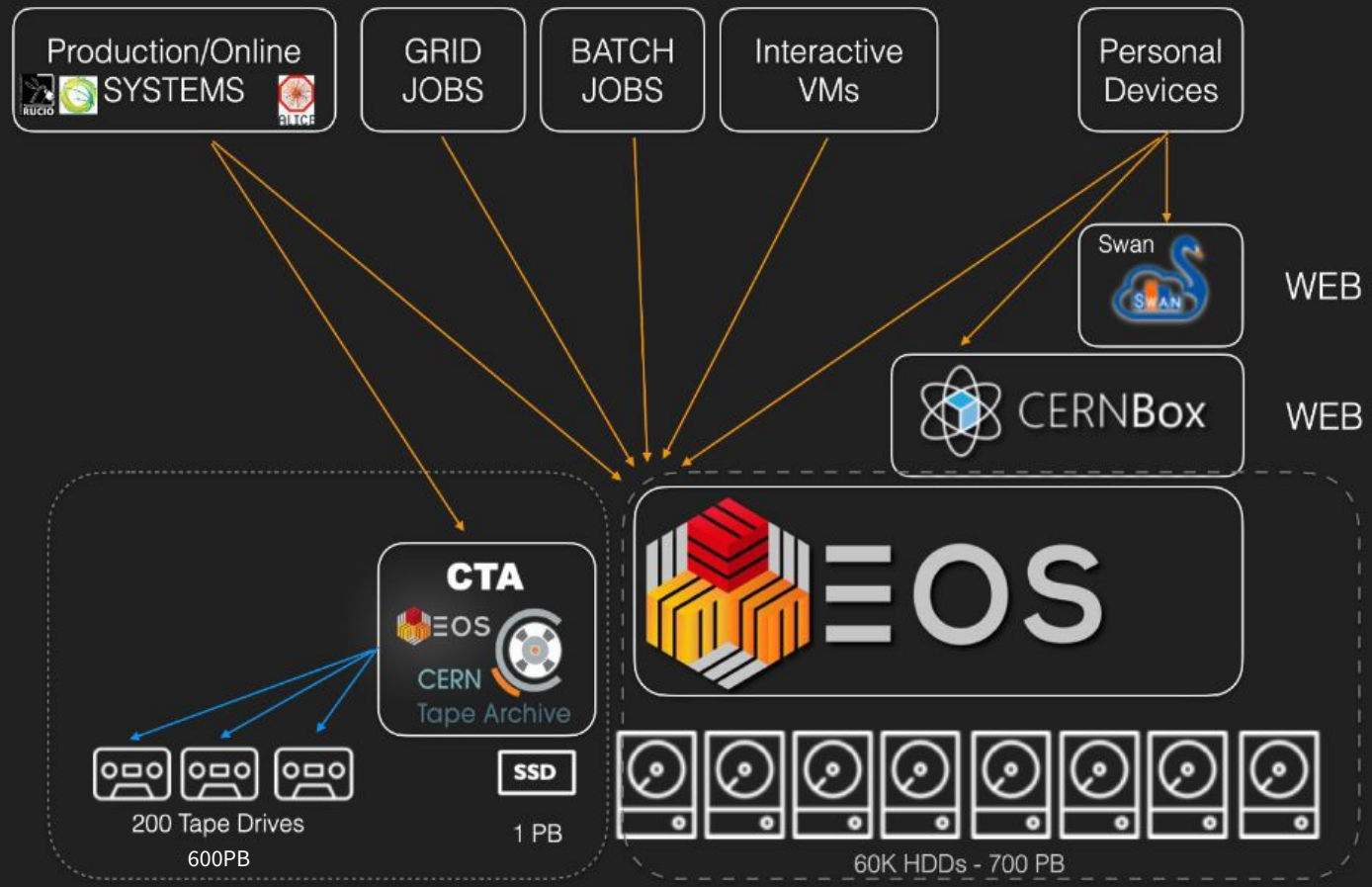


EOS Software evolution enabling LHC Run 3

Presented by Cedric Caffy on behalf of the EOS team

CHEP 2023
08/05/2023

EOS Service in Numbers



WEB Services for **Jupyter Notebooks**

WEB Services for **Sync&Share**

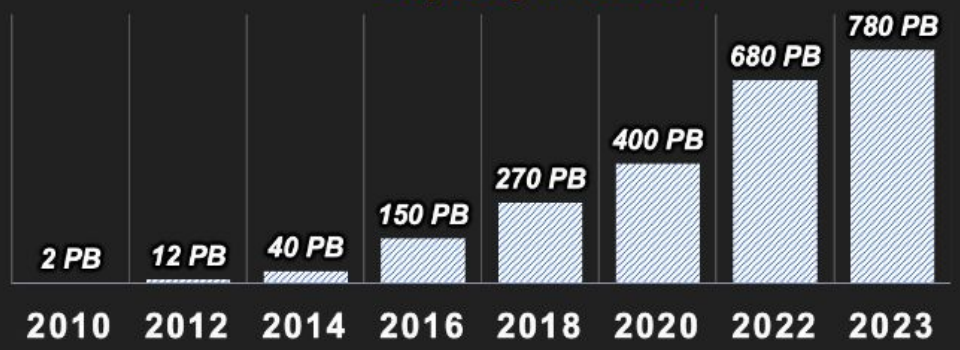
24 individual instances
8 Physics 8 CERNBox 8 CTA

2023 Targets

Total Space (raw)	**780 PB**
Files Stored	**~8 Bil**
# Storage Nodes	**~1300**
# Disks	**~60000**

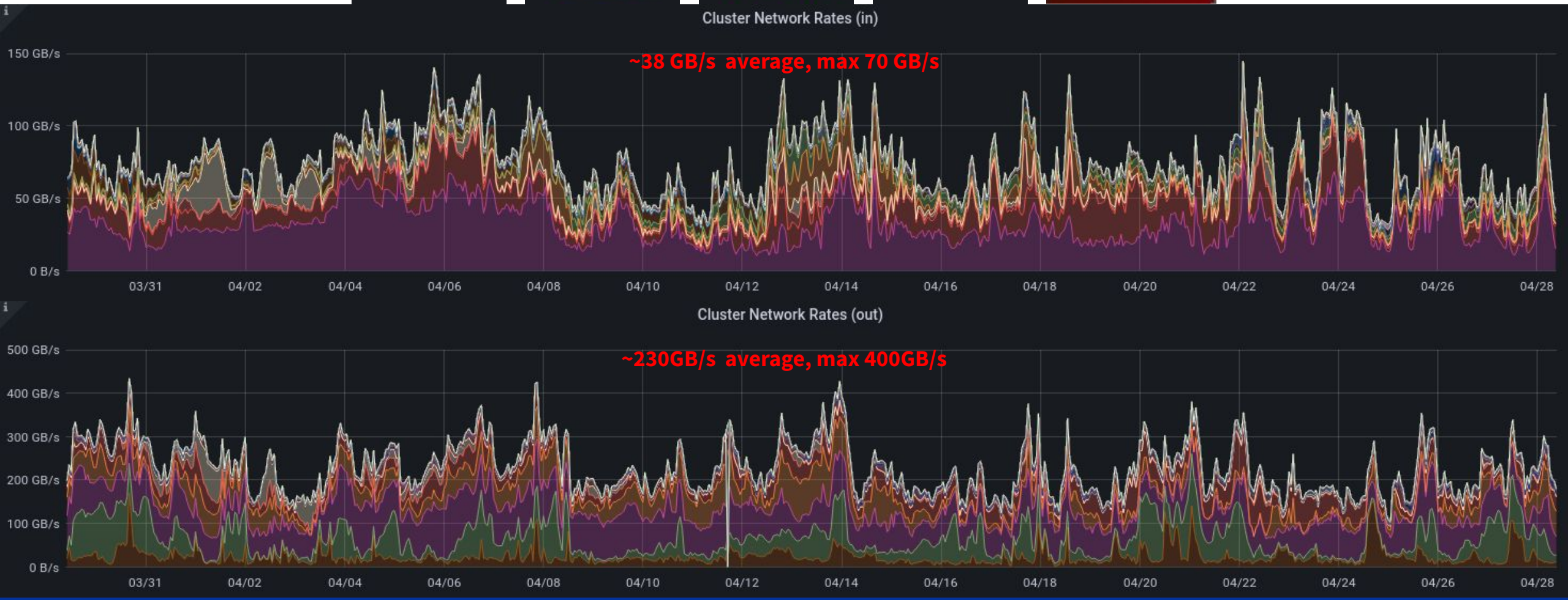
How is EOS used?

Capacity Evolution

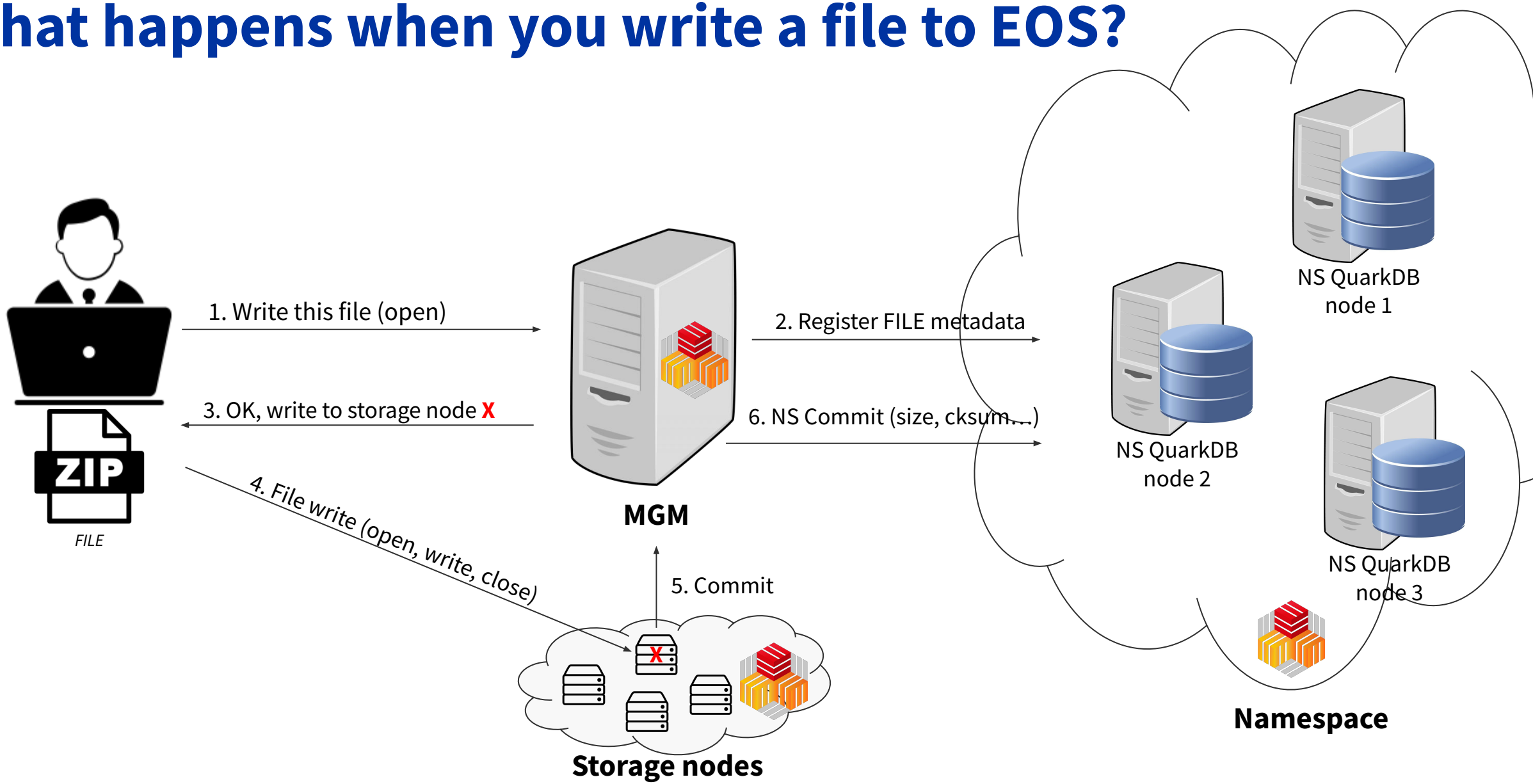


During the last 30 days…

Number of Files	Number of Dire…	Used Space	Total Space	IOPS
7.91 Bil	610 Mil	487.54 PB	729.93 PB	103M io/s



What happens when you write a file to EOS?



EOS Software evolution enabling LHC Run 3

What does that mean?

- New authentication / authorization mechanism
  - Token support
- Store more data with less hardware
  - Erasure coding
- I/O optimizations
  - I/O types, I/O priorities, bandwidth shaping
- Improving reliability
  - FSCK

Authn / Authz - Token support

Authentication / authorization - token support

EOS token - provided by the EOS MGM

```
$ eos root://eos.cern.ch// token --path /path/to/file.txt --expires 1681807613 --permission rwx
zteos64:MDAwMDAyMmN4nOP6z8jFXFReIfB348[....]>-4PQ%3d%3d
```


```
{
 "token": {
  "permission": "rwx",
  "expires": "1681807613",
  "owner": "ccaffy",
  "group": "it",
  "generation": "1",
  "path": "/path/to/file.txt",
  "allowtree": false,
  "vtoken": "",
  "voucher": "208f1f84-ddc6-11ed-84f6-fa163e6ca3c9",
  "requester": "[Tue Apr 18 10:50:53 2023] uid:112019[ccaffy] gid:2763[it]
tident:eosdev.25745:427@localhost name:ccaffy dn: prot:krb5 app: host:localhost
domain:localdomain geo: sudo:1",
  "origins": []
 },
 "signature": "[...]",
 "serialized": "[...]",
 "seed": 878494853
}
```

Authentication / authorization - token support

Macaroons - provided by the EOS MGM

- Request a macaroon using your X509 certificate

```
curl --cert [...] --key [...] --cacert [...] --capath [...] -X POST -H 'Content-Type: application/macaroon-request' -d '{"caveats":
["activity:UPLOAD,DELETE,LIST"], "validity": "PT3000M"}' https://eos.cern.ch//path/to/file.txt | jq -r '.macaroon'
```

```
location eosdev
identifier bc8bedfd-072c-4fea-b3bc-042cf73d8bb3
cid name:ccaffy
cid activity:READ_METADATA
cid activity:DOWNLOAD,UPLOAD,MANAGE
cid path:/path/to/file.txt
cid before:2020-01-29T15:13:35Z
signature
b8d9b5e4d09badbeb628222fc710e54a0af080c64a8c63eb3bb370c454302327
```

Token authorization (activity) = file permission (ACL) set on the file on EOS

Authentication / authorization - token support

sci-token - Provided by an IAM (Identity and Access Management) provider

- Using oidc-token tool


```
{
  "wlcg.ver": "1.0",
  "sub": "4d863cdd-5736-44a0-a03b-81ce144b5fe3",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "nbf": 1681818273,
  "scope": "openid profile storage.read:/ eduperson_entitlement wlcg
storage.create:/ offline_access eduperson_scoped_affiliation
storage.modify:/ email wlcg.groups",
  "iss": "https://wlcg.cloud.cnaf.infn.it/",
  "exp": 1681821873,
  "iat": 1681818273,
  "jti": "acd4f929-3294-4383-ba7a-3fadb30e8321",
  "client_id": "2002057c-bacc-4d5c-b79d-52d42a7a3596",
  "wlcg.groups": [
    "/wlcg",
    "/wlcg/xfers"
  ]
}
```

Authentication / authorization - token support

Usage - All tokens

HTTP

```
curl -x GET -H "Authorization: Bearer $TOKEN" https://eos.cern.ch//path/to/file.txt
```

XRootD

```
xrdcp ./file.txt root://eos.cern.ch//path/to/file.txt?authz=$TOKEN
```

Authentication / authorization - token support

Summary

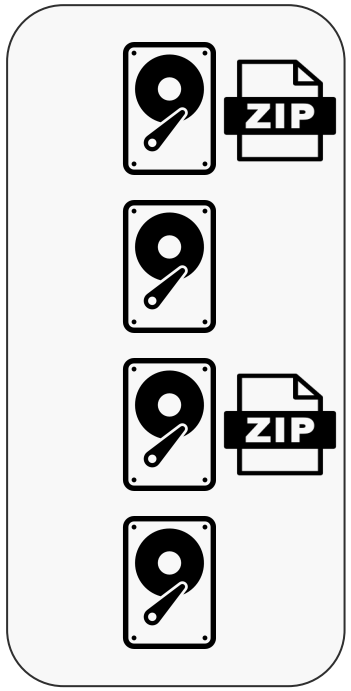
Token type	Issuer	Permissions
EOS token	EOS MGM	Whatever is set by the token creator
Macaroon	EOS MGM (authentication with X509)	File/Parent directory permissions
WLCG scitoken	IAM Provider	Maps to a user - scopes limit permissions

Store more data with less hardware

How do we ensure files availability in EOS?

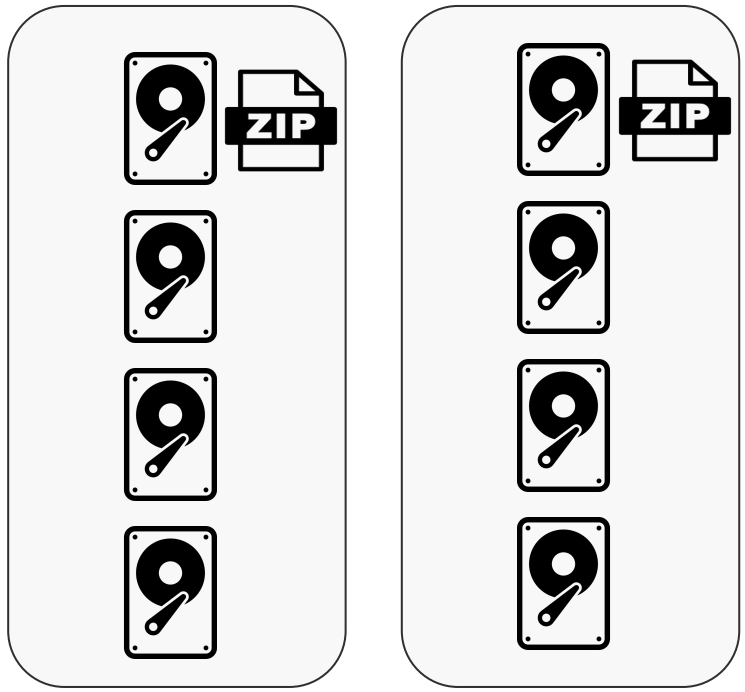
- **RAID vs RAIN**

RAID (Redundant Array of Inexpensive Disks)



ServerDisk1

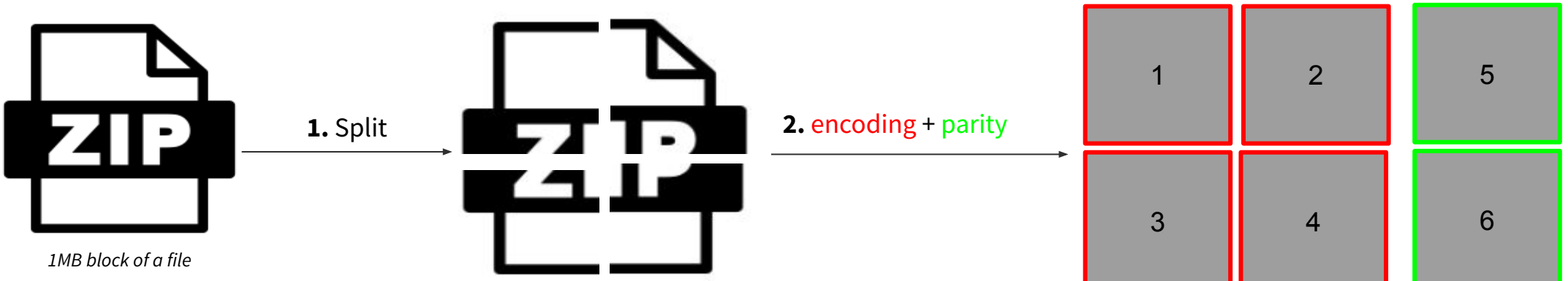
RAIN (Redundant Array of Inexpensive Nodes)



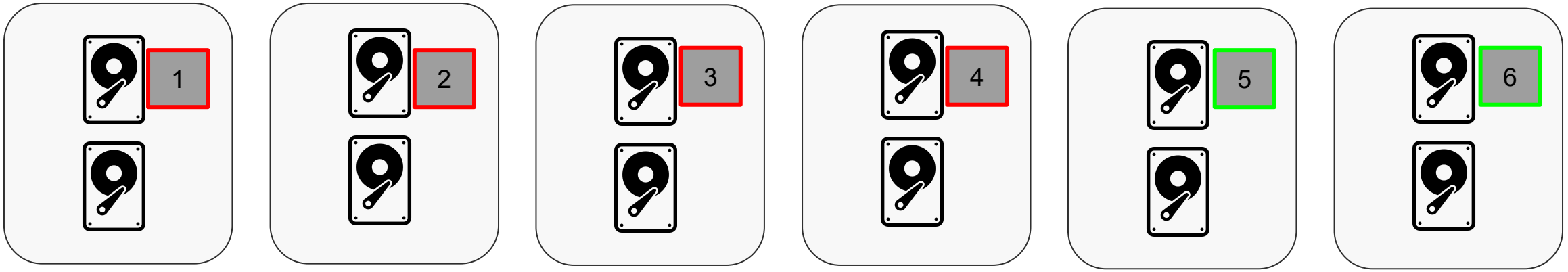
ServerDisk1 ServerDisk2

Storing 2 replicas is a good solution, but expensive!

Erasure Coding

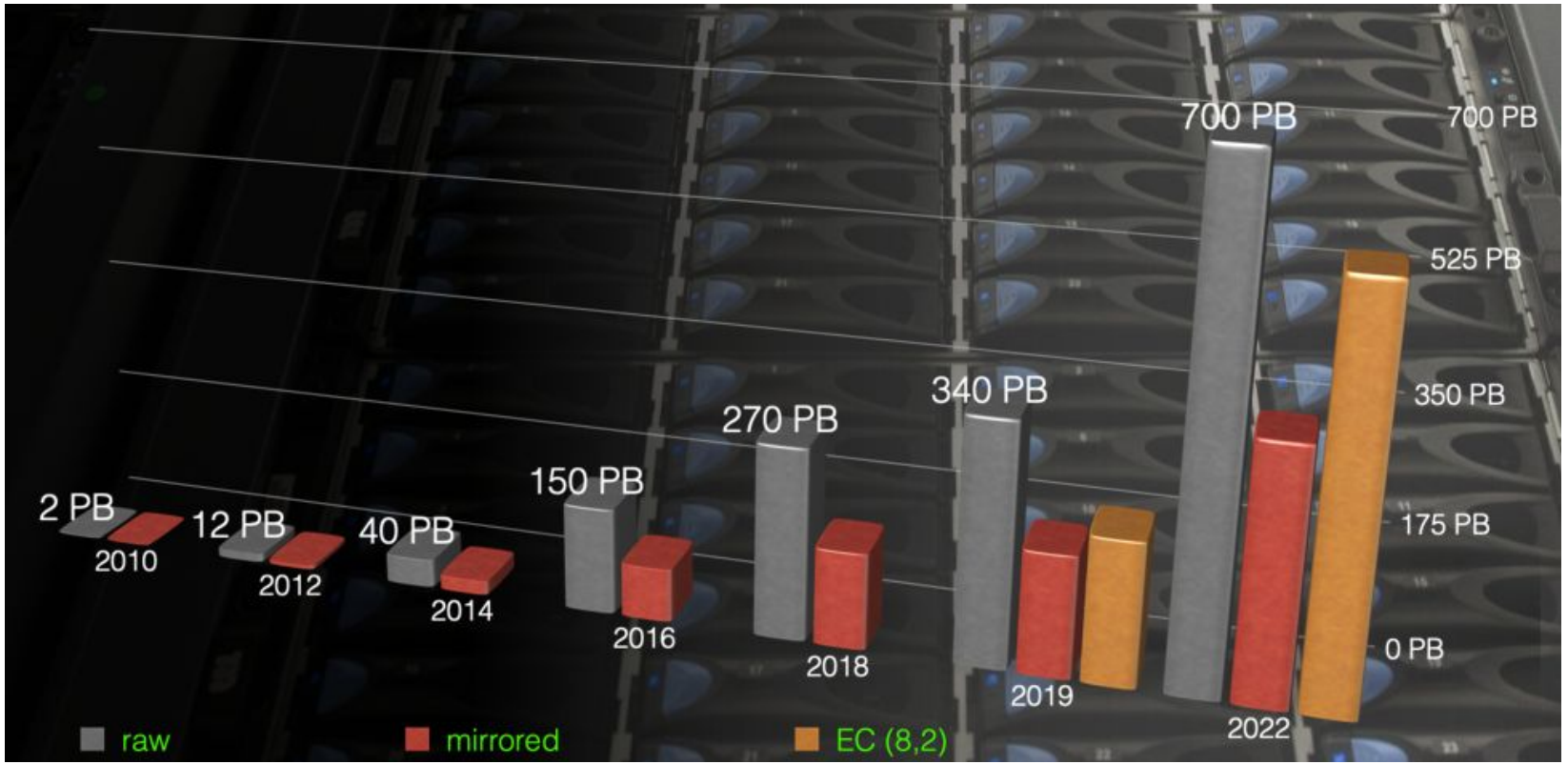


3. Storage

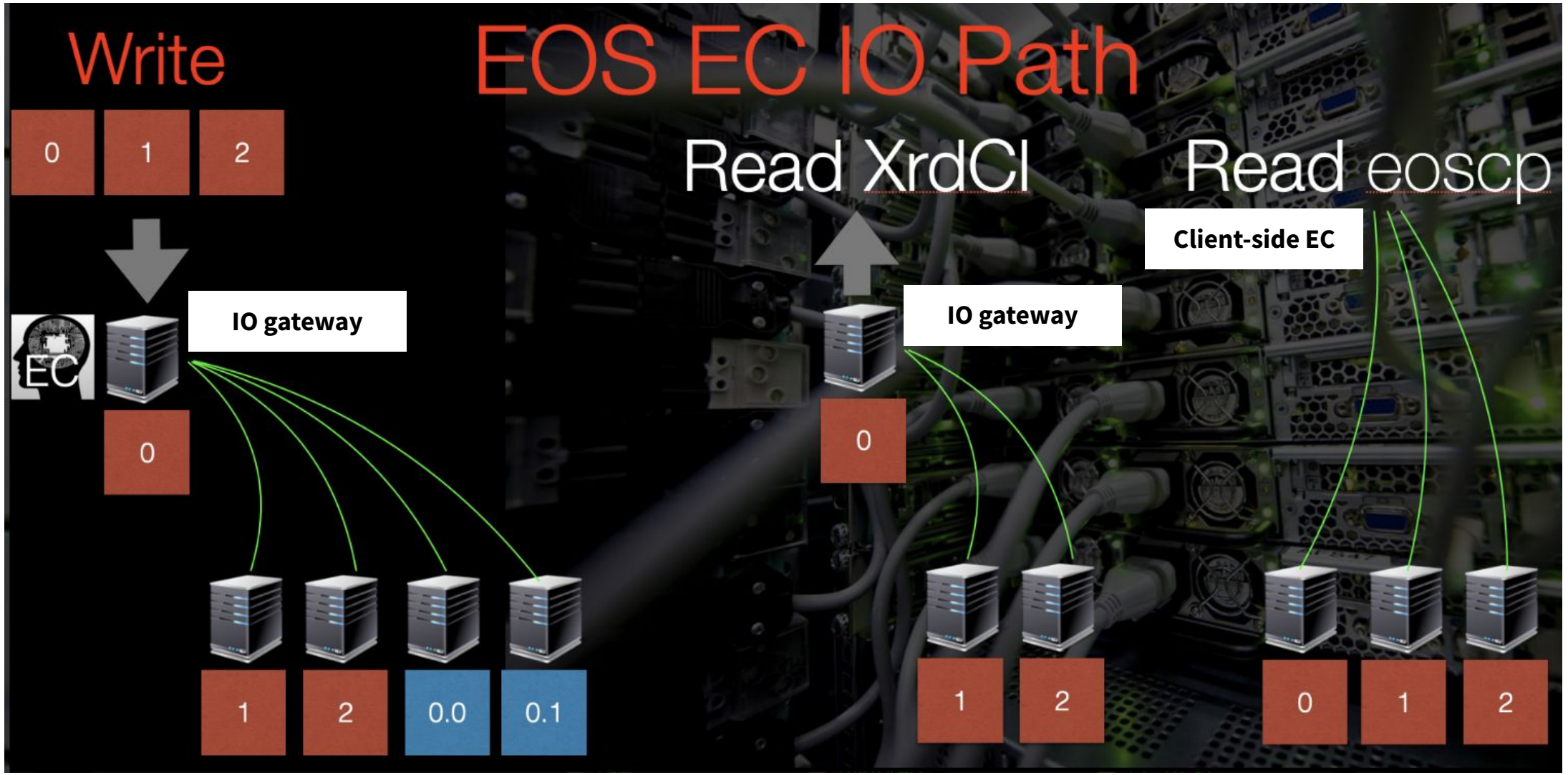


Storing more data with less hardware

Erasure coding



Storing more data with less hardware



Storing more data with less hardware

Performances of EC

- 2x traffic amplification for read using the gateway model compared to replication
- You benefit from the parallel transfers of the different disks involved in Erasure coding
  - Ex: Client-side EC read with buffered I/O is extremely performant!

Drawbacks

- CPU and network intensive
- Do not use for small files (< 100MB)

Storing more data with less hardware

Erasure coding @ CERN

Erasure Coding vs Replica percentage

	Value	Percent
Replica	11.5 PB	54%
Erasure Coding	9.83 PB	46%

Erasure Coding vs Replica percentage

	Value	Percent
Erasure Coding	69.1 PB	100%
Replica	114 GB	0%

AliceO2

Erasure Coding vs Replica percentage

	Value	Percent
Replica	31.6 PB	96%
Erasure Coding	1.39 PB	4%

Space recuperated

	Value
Aliceo2 Savings with Erasure Coding 10+2	55.3 PB
AMS Savings with Erasure Coding 8+2	7.37 PB
CMS Savings with Erasure Coding 10+2	1.12 PB

Total: 63.79PB

Let's also improve I/O performance

Improving performance

I/O shaping

- Disk optimizations
  - I/O Types
  - I/O Priorities
- Bandwidth shaping

Improving performance

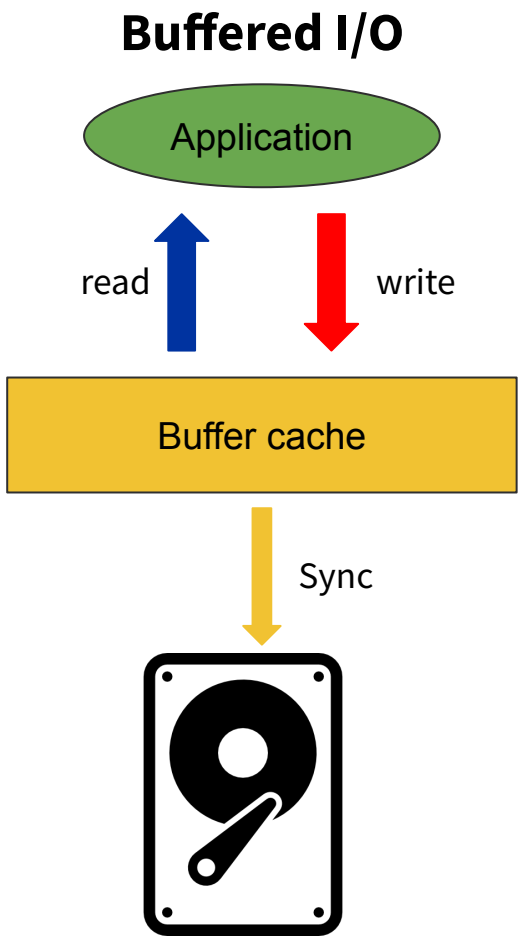
I/O types - Buffered I/O VS Direct I/O

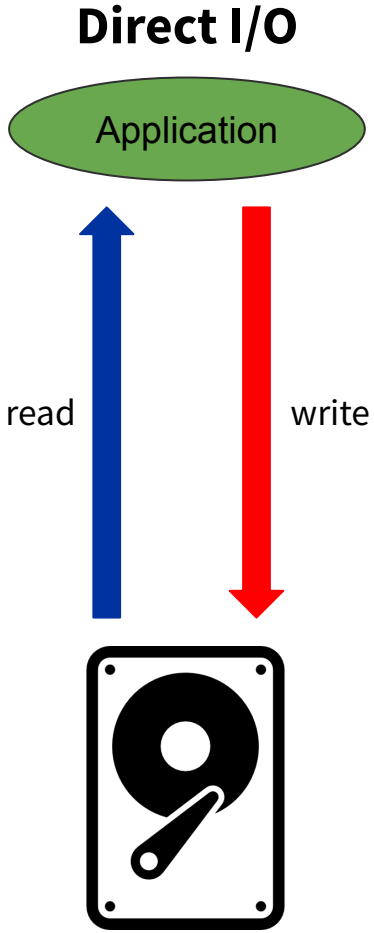
read-ahead: anticipate file read

write-back: file is flushed from the cache on close (sync)

Allows asynchronous writes

If many files are competing for the cache, it may reduce its efficiency…





Reduces CPU/memory usage

Writes are more efficient

Read penalty as they don't benefit from the caching

Every op is synchronous

Improving performance

Direct I/O

Very good for **writes**

- Max perf of a standalone XRootD disk server increased from 7GB/s to 9GB/s
- Reduces perf tails
- Increases overall instance performance for write workloads

Not as good for reads…

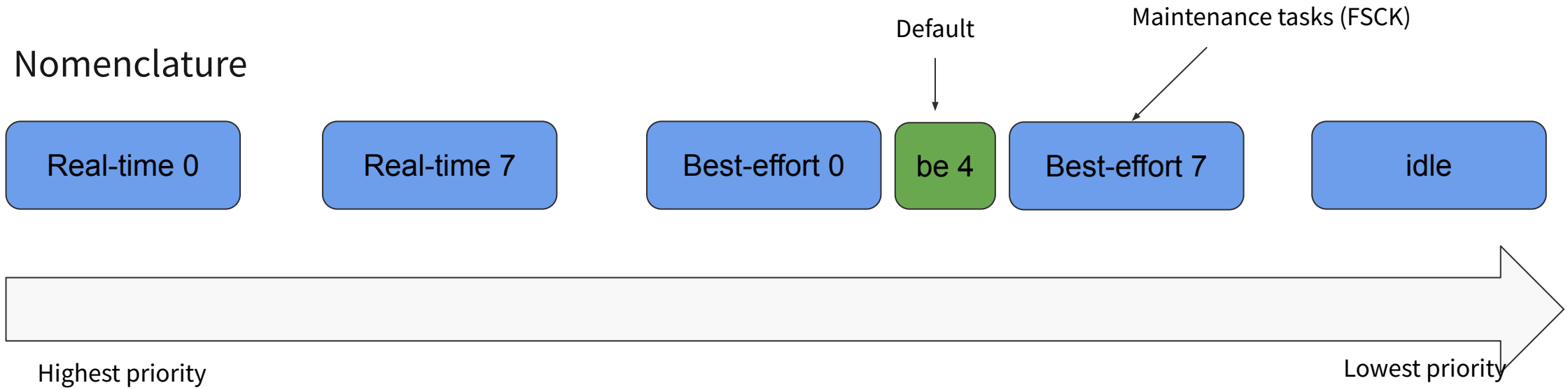
- You don't benefit from the cache

Documentation and configuration:
https://eos-docs.web.cern.ch/using/policies.html?highlight=iotype#setting-user-group-and-application-policies

Improving performance

I/O priorities

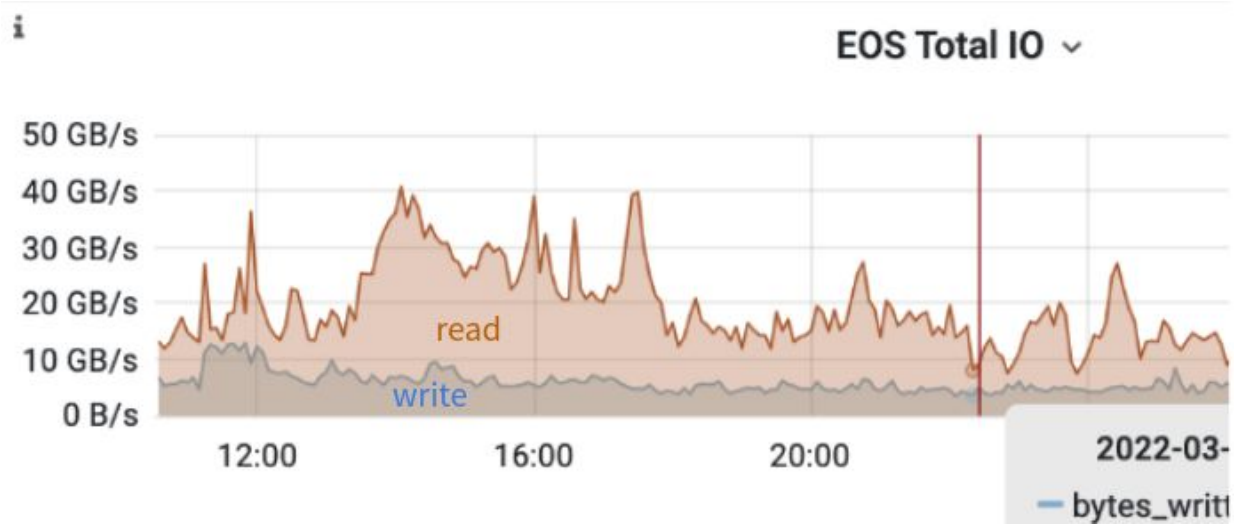
- Balance the needs for high-throughput by fairly sharing I/O requests among processes
  - Maintenance tasks can have lower priority than experiments transfers
- Works with **read** + **direct I/O writes** on devices using BFQ and CFQ scheduler
- Nomenclature



Configuration: https://eos-docs.web.cern.ch/using/priorities.html?highlight=iopriority

Improving performance

I/O priorities

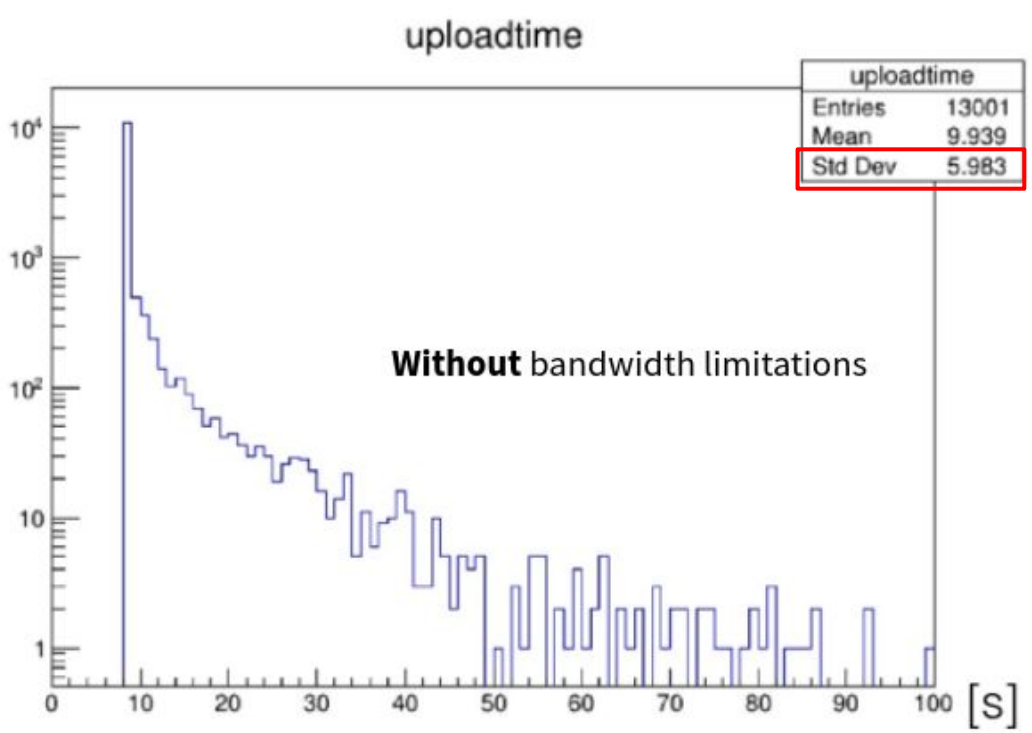


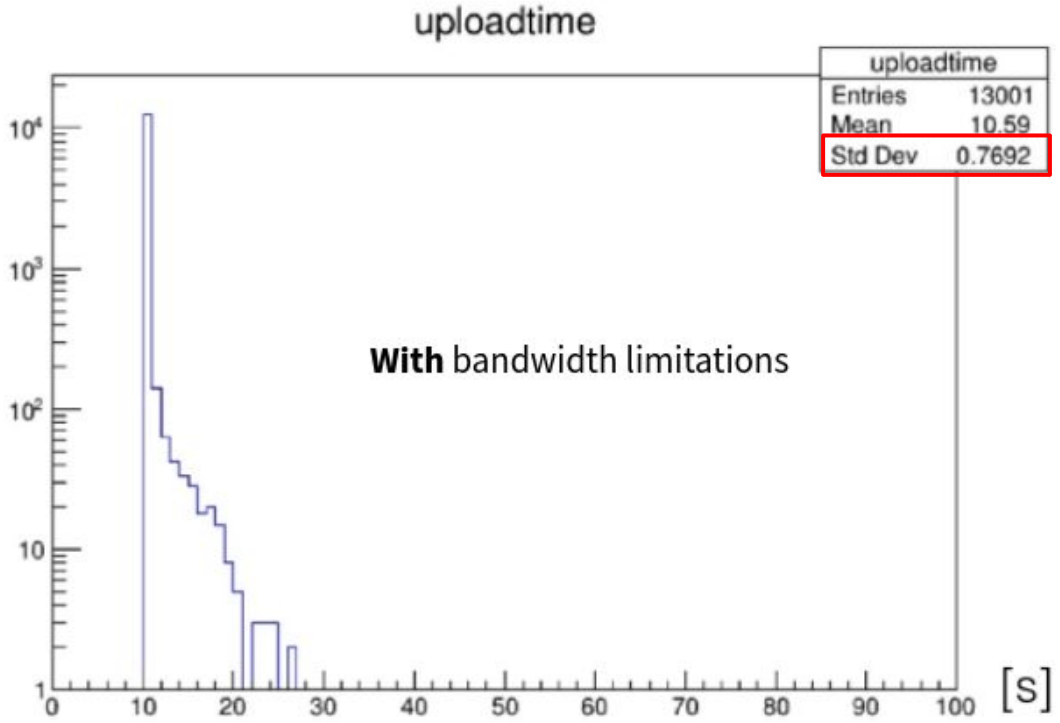
ATLAS prod workflows

Improving performance

Bandwidth regulation

- Benchmarks did show that I/O performance tails are reduced by limiting the bandwidth of clients





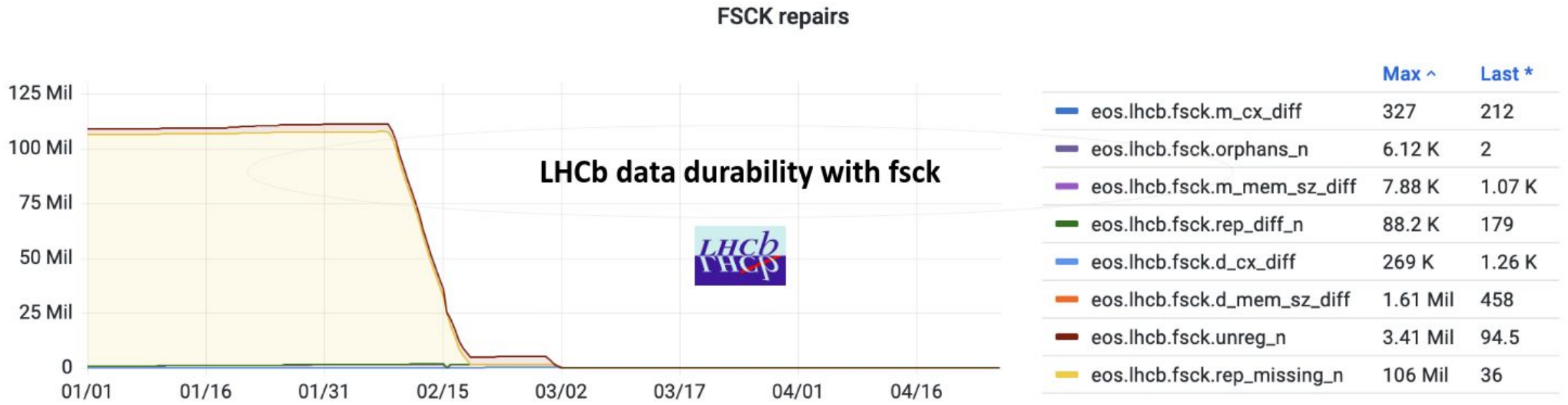
Configuration: https://eos-docs.web.cern.ch/using/policies.html?highlight=bandwidth

Very good for data taking use case!

Improving the reliability

FSCK

- Automatic detection and repair of different type of recoverable errors
  - Background thread by filesystem, scanning all the files in the disk (*IO priority: be:7*)

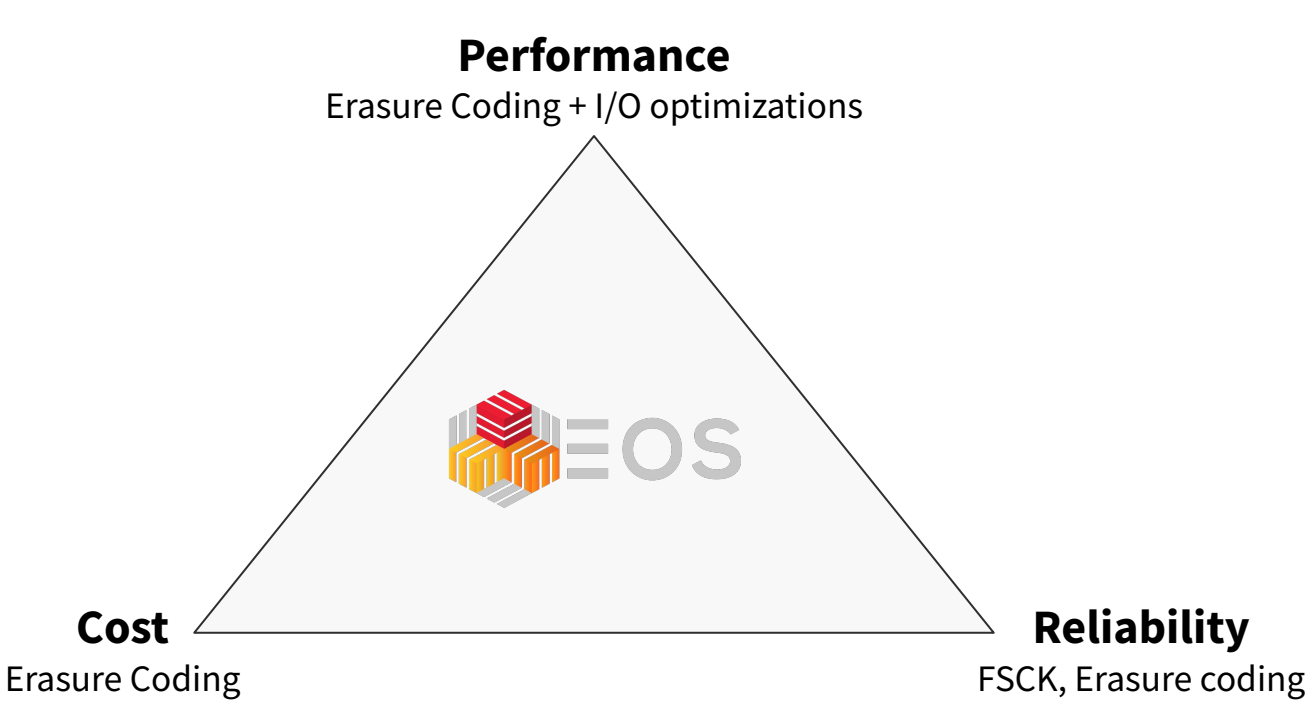


Outlook

- **Maintain Run 3 success**
- **EOS operations improvements**
  - High availability (EOS MGM failover)
    - Automatic recovery
  - Monitoring improvements
- **EOS developments - start to think about Run 4!**
  - MGM performance improvements (finer-grained namespace locking)
  - API evolution (bulk API for opening files)
  - Data format evaluation (RN Tuple with erasure coding)

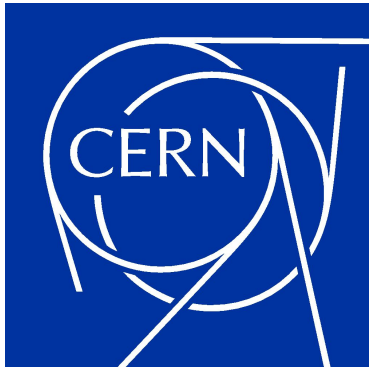
Conclusion

- **Authentication / Authorization**
  - EOS supports different tokens (EOS token, Macaroons, scitokens)
- **Improvements**



Performance, reliability and cost can be optimized…

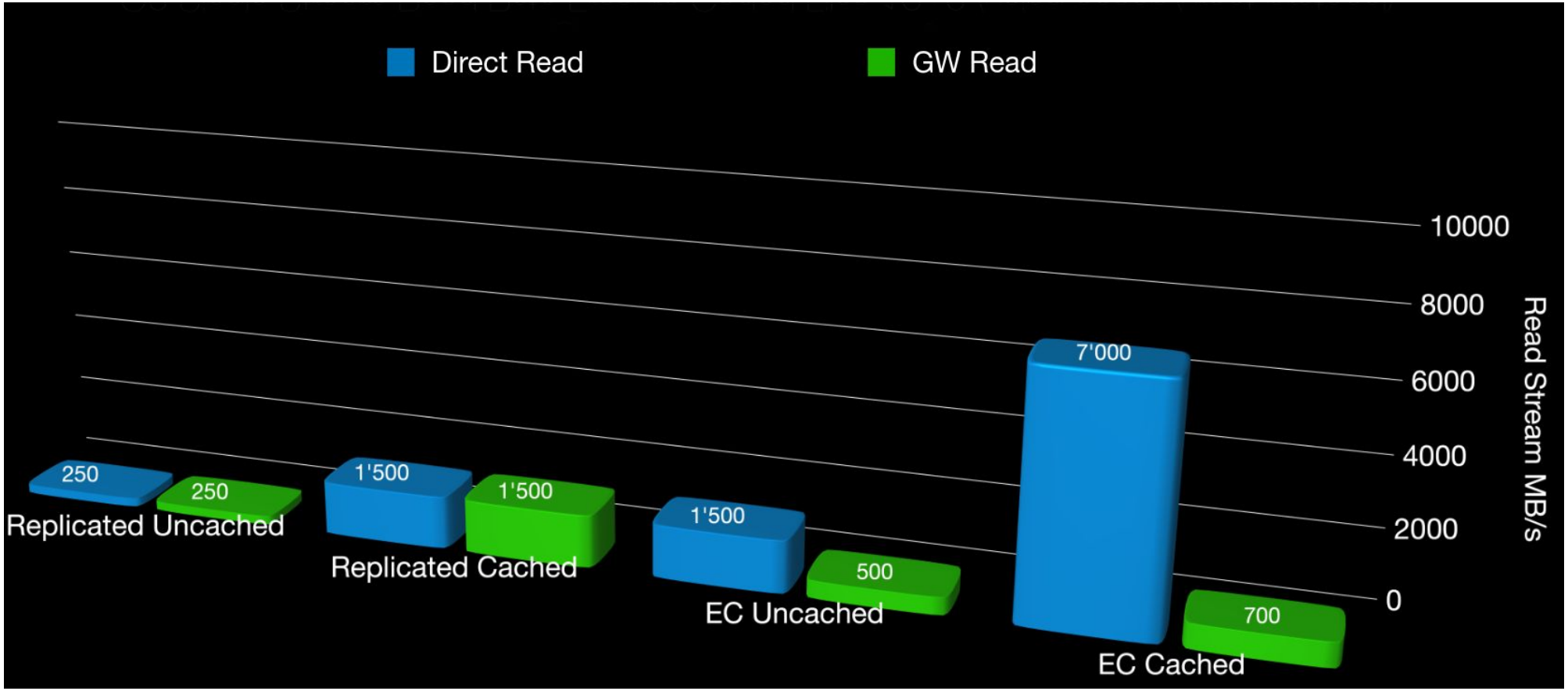
Depending on the use-case



home.cern

Backup slides

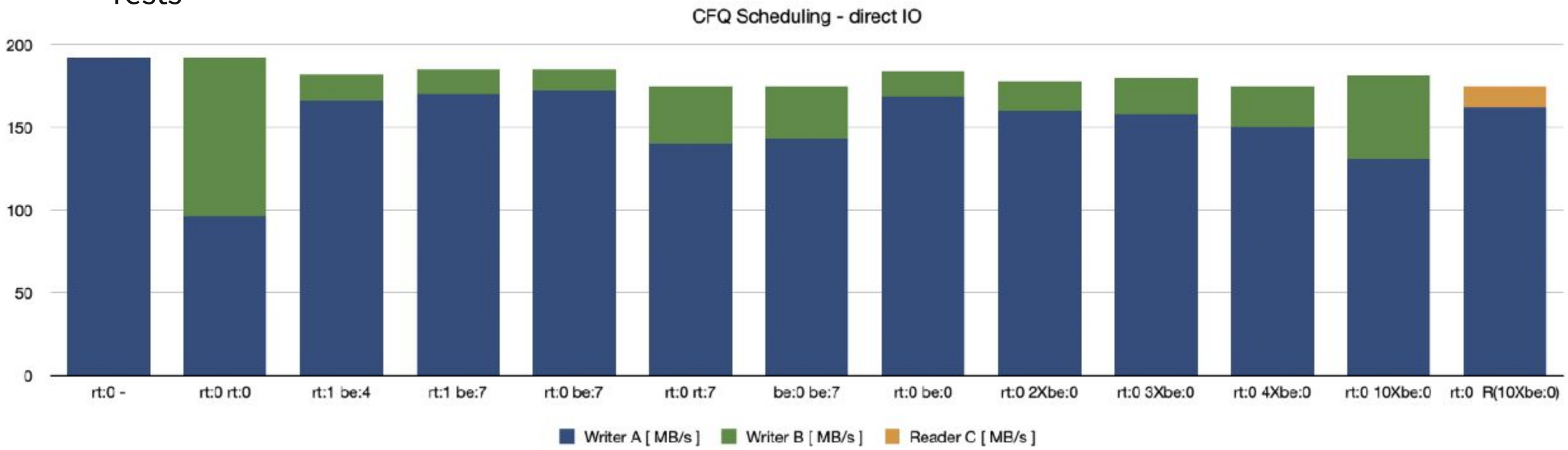
Performances of EC - example: ALICEO2 EC read 1 file



Backup slides

I/O priorities

- Tests



Configuration: https://eos-docs.web.cern.ch/using/priorities.html?highlight=iopriority