EOS workshop 2022

Direct IO, IO Priority and Bandwidth Policies in **EOS**

IT-ST-PDS

Andreas-Joachim Peters
CERN IT-ST for the EOS team

Contents

- **Introduction**
  - why did we implement these?

- **IO Types**
- **IO Priorities**
- **Bandwidth Policies**

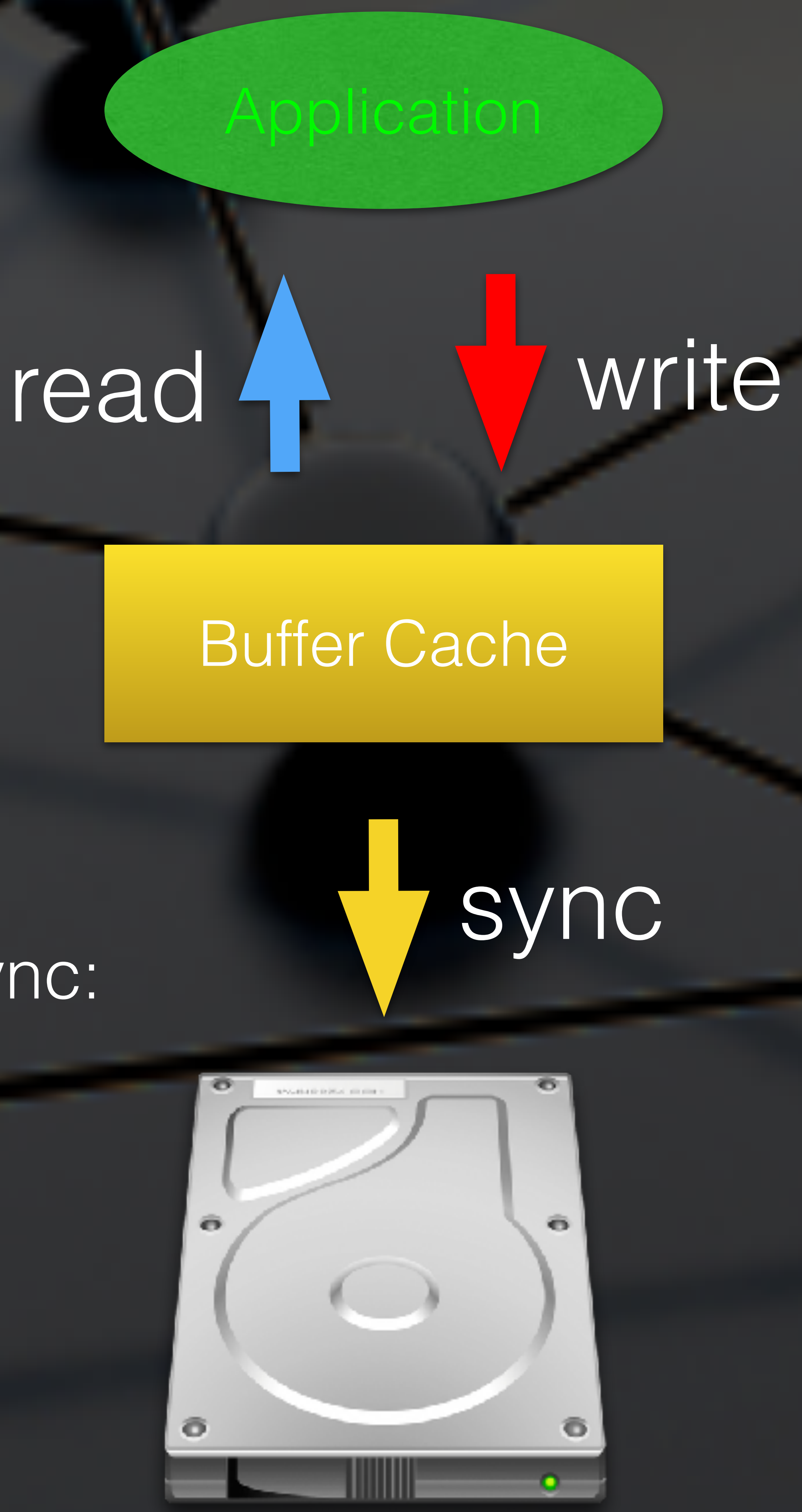
- **Summary**

Introduction

- **EOS** instances are used by a **large user community - EOS is a large shared resource**
  - the **criticality** of individual access types **varies** a lot
    example:
    - data INGRES from online DAQ systems requires highest priority (real-time critical)
    - background scanning to verify file checksum is a low-priority task, which should back off for most other use cases
  - **real-time driven** applications require minimal IO fluctuations
    example:
    - the transfer time for an online system can vary within the given time budget, but large tails in transfer times have to be avoided

Buffered vs direct IO …

buffered IO

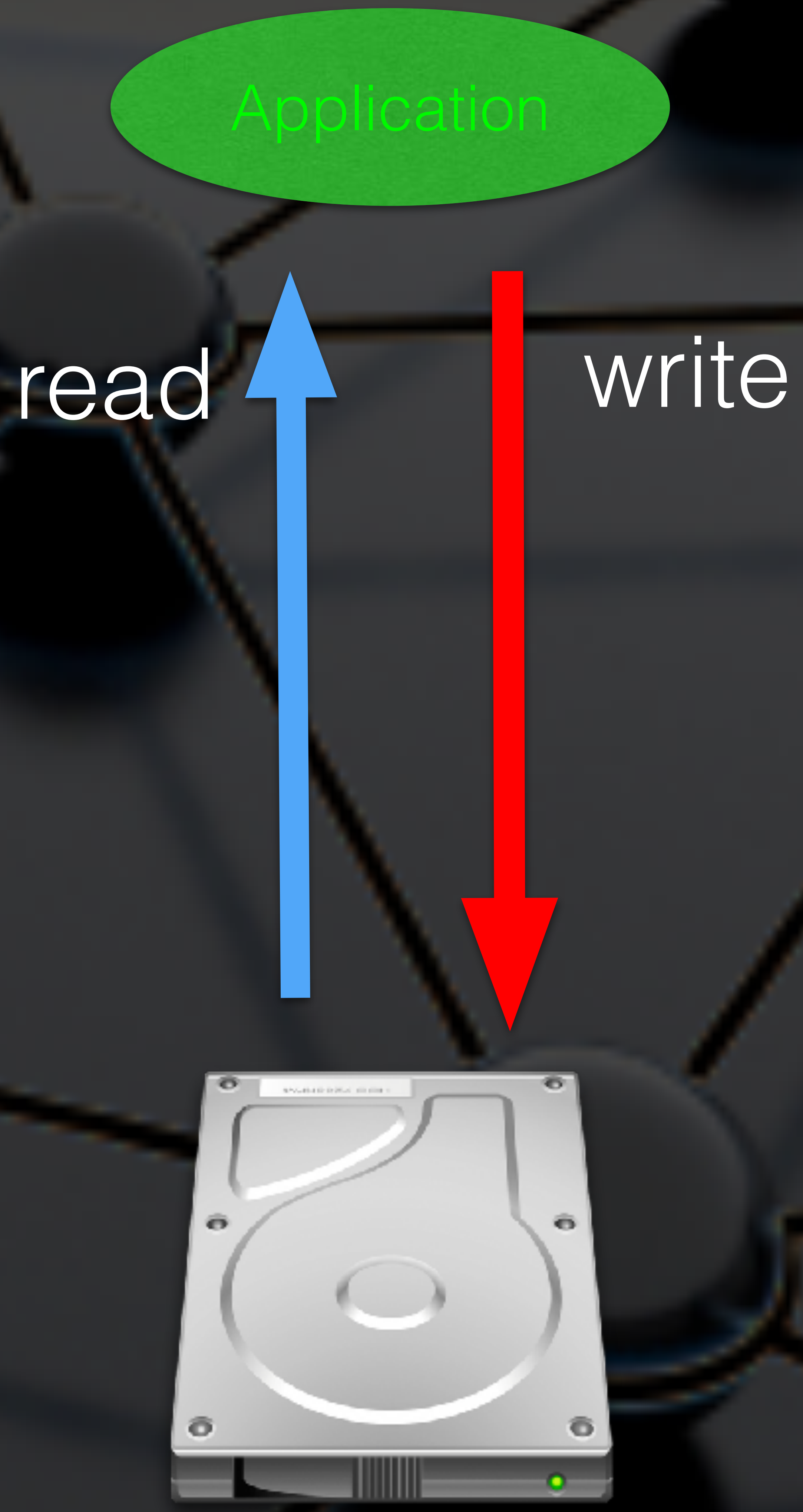


provides:
caching
read-ahead
write-back

requires:
memory
cpu

pitfalls without fsync:
data safety
ENOSPC
IO stalling

direct IO



provides:
less memory
less cpu

requires:
IO + buffer alignments

The three **EOS IO Types**

- **direct IO** - implemented in EOS OSS plug-in
  - uses two file descriptor
    1. **direct IO** for IO which fullfills alignment
    2. **buffered IO + fdatasync + posix_fadvice** for writes which does not fulfil alignment
       [will be changed to O_SYNC]

- **sync** = **dsync** - use synch. IO
  - use file descriptor with **O_SYNC**

- **csync** - sync on close
  - write via buffer cache but **fdatasync** on close without the client calling sync directly

Selecting **IO Types**

Evaluation Order

via CGI: "**root://myeos?eos.iotype=direct**"

instance default:
“**eos config default space.policy.iotype=direct**”

space specific:
“**eos config erasure space.policy.iotype=sync**”

space+application specific default for app ‘foo’:
“**eos config default space.iotype.foo=csync**”

space+application specific for app ‘foo’:
“**eos config erasure space.iotype.foo=csync**”

directory enforced :
“**eos attire set sys.forced.iotype=direct /eos/daq/**”

Impact of direct IO

- measured that direct IO **improves** maximum **WRITE** performance of standalone XRootD server with standard CERN disk server from **7 GB/s to 9 GB/s**
- direct IO **increases** instance performance for **WRITE** workloads
- using direct IO **reduces** performance **tails for WRITE** workloads
- direct IO **reduces** instance performance for **READ** workloads

IO Priorities

IO Priorities

- IO priorities currently available only with **CFQ/BFQ** scheduler on LINUX - support in deadline scheduler coming
- three levels: idle (**idle:0**), best-effort (**be:0-7**) , real-time (**rt:0-7**)

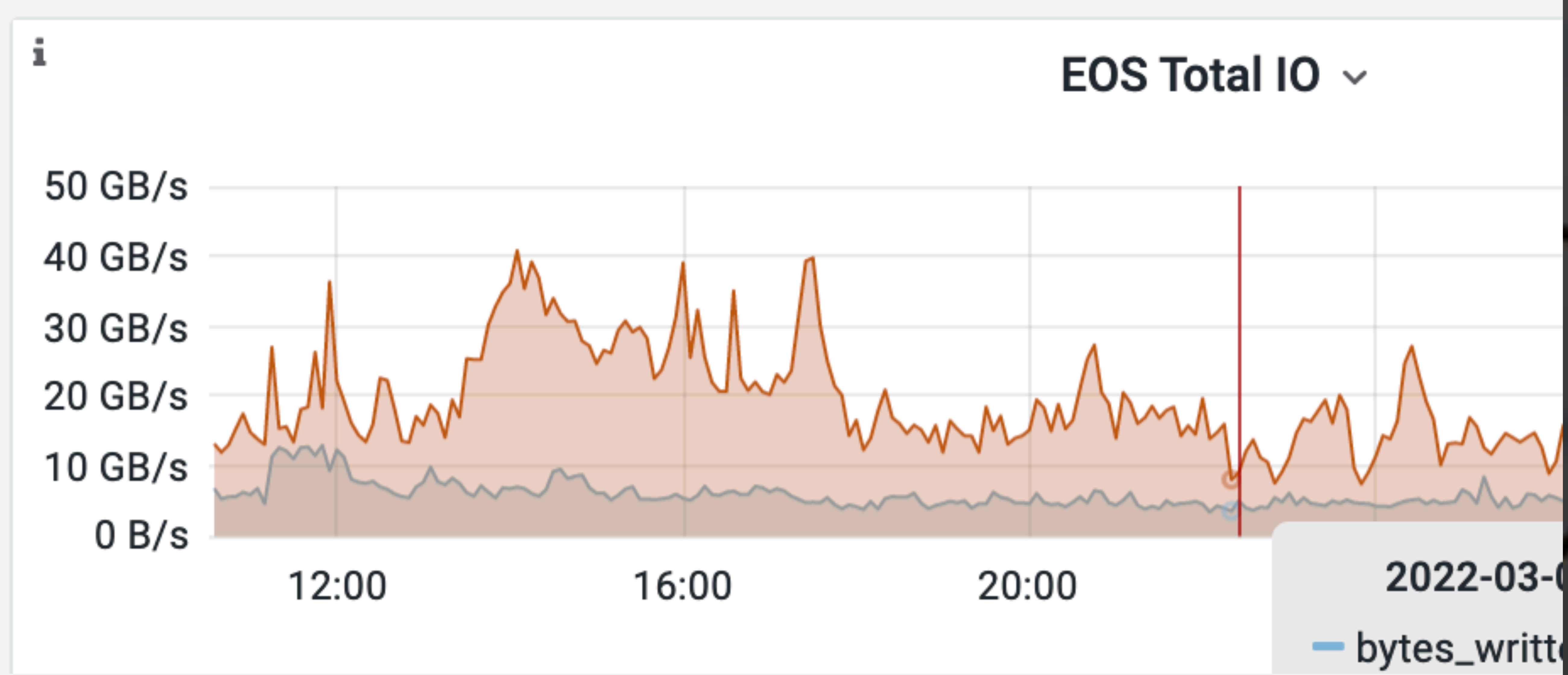
real-time >> best-effort 1 >> best-effort 7 >> idle

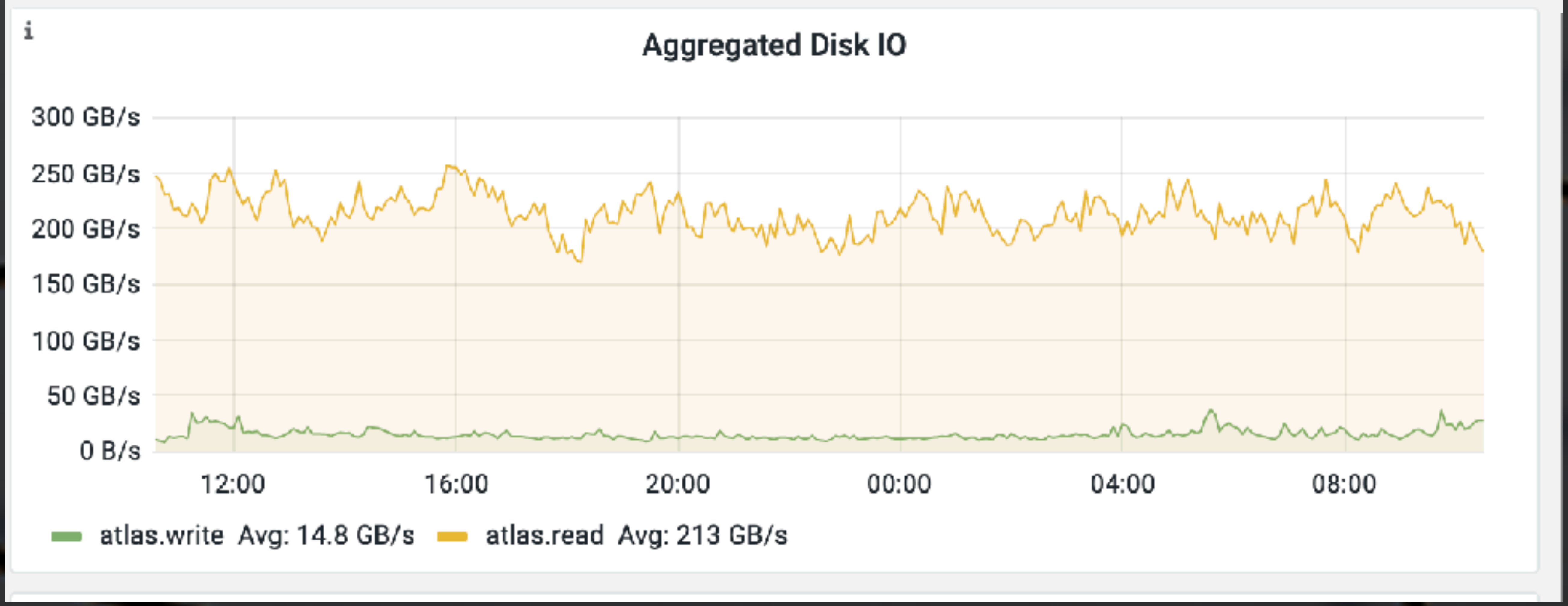
default IO priority is **be:4**

- the EOS background scan for checksum verification is runnning with best-effort **be:7**

- **realtime** supported from 4.8.79 on (+ 5.0.15)
- **IO** priority works only for **read + direct write**

IO Priorities





IO originating from applications in EOSATLAS

Disk IO measured during the same period mainly background scanning

Selecting **IO Priority**

a

via CGI: **"root://myeos?eos.iopriority=be:1"** if user has the 'operator' role (member of operator in VID interface)

Evaluation Order

instance default:
"**eos config default space.policy.iopriority=rt:0**"

b

space specific:
"**eos config erasure space.policy.iopriority=be:2**"

space+application specific default for app 'foo':
"**eos config default space.policy.iopriority.foo=be:6**"

space+application specific for app 'foo':
"**eos config erasure space.policy.iopriority.foo=idle:0**"

directory enforced :
"**eos attire set sys.forced.iopriority=be:1 /eos/daq/**"

Bandwidth Regulation

Bandwidth Regulation

- during various benchmarks we have verified that IO tails are reduced when clients run with limited bandwidth

- an upper bandwidth limit can be set for **xrdcp** and **eoscp** (see help of these commands)

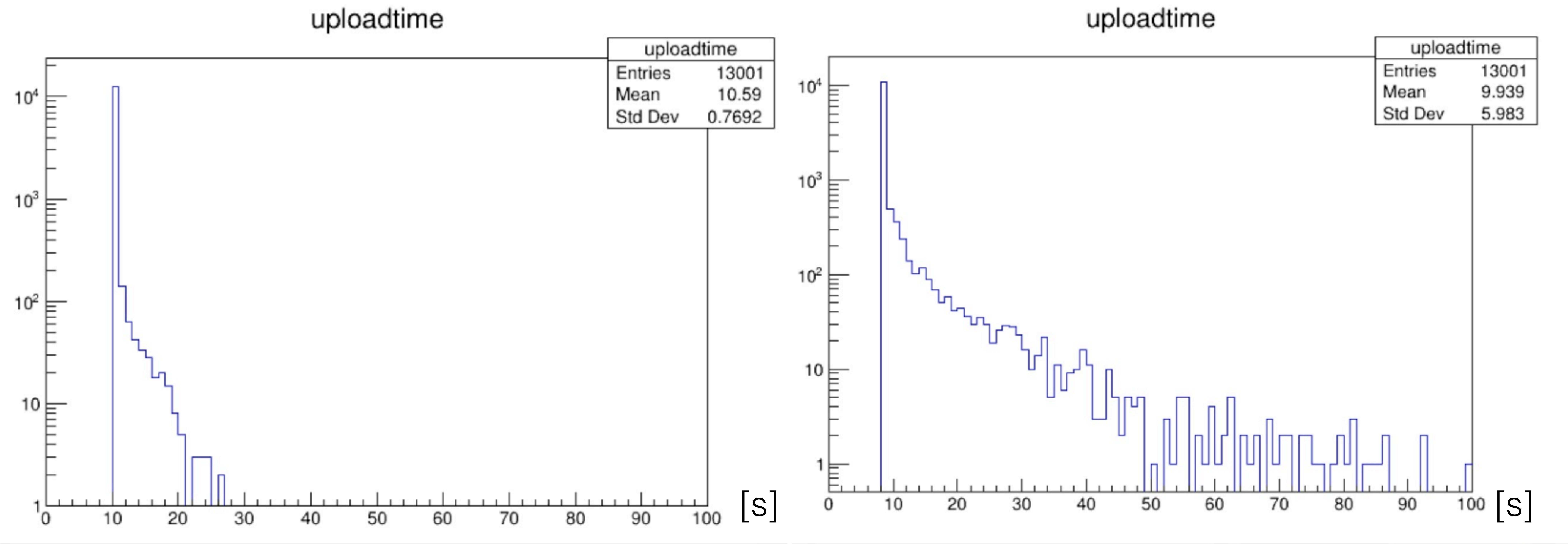
- we have added **bandwidth regulation** to EOS **server-side** to be able to set this limit on the instance itself such that we don't have to tell applications/people to use a reasonable setting and to be able to change this settings on the fly

EOS workshop 2022

Bandwidth Regulation

Impact on performance tails

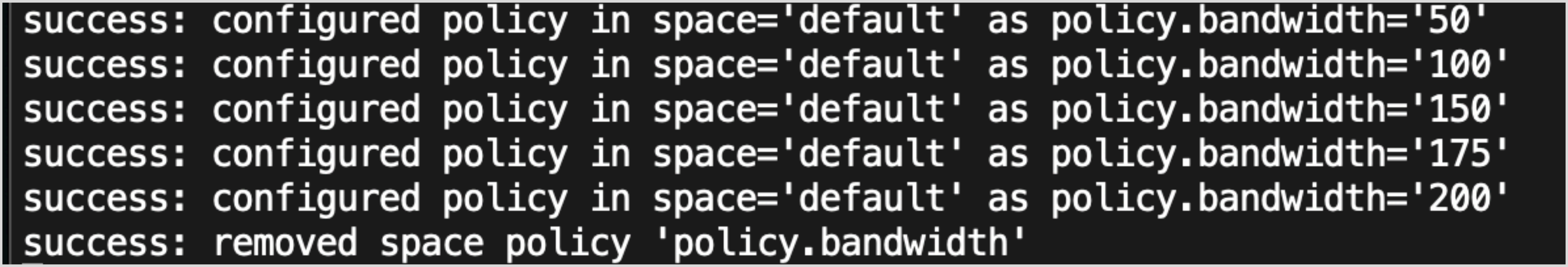
left with bandwidth limitation - **right** without



EOS workshop 2022

Bandwidth Regulation

Test changing bandwidth limit on the fly and effect on instance IO performance and tails



```
success: configured policy in space='default' as policy.bandwidth='50'
success: configured policy in space='default' as policy.bandwidth='100'
success: configured policy in space='default' as policy.bandwidth='150'
success: configured policy in space='default' as policy.bandwidth='175'
success: configured policy in space='default' as policy.bandwidth='200'
success: removed space policy 'policy.bandwidth'
```

bw policy	50	100	150	175	200	unlimited
avg [s]	42	22	15	13.4	12.3	11.9
95 perc	43.7	23.3	16.6	15.6	15.7	40.7
99 perc	44.2	24.5	18.1	20.2	25.4	104
Total BW GB/s	8	16	23	26	28	30

Network usage
40 GB/s
30 GB/s
20 GB/s
10 GB/s
0 B/s
20:30
20:40
20:50
21:00
21:10
21:20
21:30
21:40
21:50
22:00
22:10
22:20
22:30
22:40
22:50
23:00
23:10
23:20
23:30
160 x 50 MB/s
160 x 100 MB/s
160 x 150 MB/s
160 x 175 MB/s
160 x 200 MB/s
unlimited
2021-07-25 23:20:00
rx: 37.1 MB/s
tx: 31.1 GB/s
rx
tx

Selecting **IO Bandwidth**

The bandwidth parameter unit is MB/s

Evaluation Order

via CGI: “root://myeos?eos.iobw=100”

instance default:
“**eos config default space.policy.bandwidth=250**”

space specific:
“**eos config erasure space.policy.bandwidth=150**”

space+application specific for app ‘foo’:
“**eos config erasure space.bw.foo=50**”

Limitations:

- currently IO bandwidth can **not** be **configured on** the **directory level**
- IO bandwidth does **not distinguish reading** and **writing** !

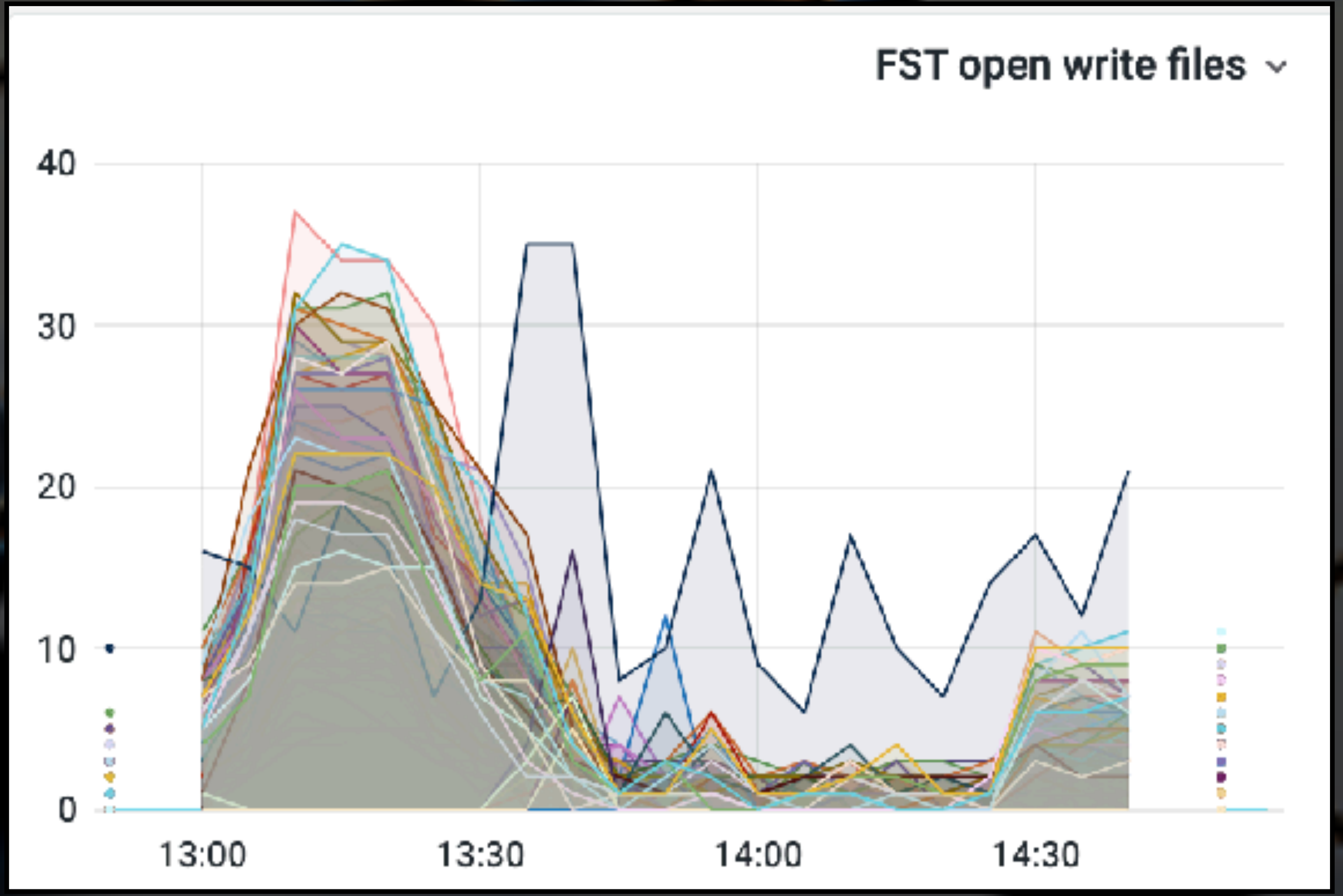
Filesystem Scheduling Overload

Filesystem Overload

- during data challenges and benchmarking we have observed **black-hole effects on filesystem**
  - certain filesystem aggregate streams over time and the overall instance performance is defined/degraded by few overloaded filesystems
- to avoid this we have added an **overload status to filesystem**, which can be triggered when a threshold of max. readers or writers is reached on a filesystem

EOS workshop 2022

Filesystem Overload



# define when a filesystem is marked as overload

eos space config default **space.max.ropen=200**

eos space config default **space.max.ropen=50**

when the max. number of streams is reached,
the scheduler stops scheduling on this filesystems!

```
EOS Console [root://localhost] |/eos/ajp/> fs ls

┌──────────────────────┬────┬───┬────────────────────────────────────────────────────────┬──────────┬────────┬────────┬────────────┬───────┬────────┬───────────┐
│host                  │port│ id│                                                    path│schedgroup│  geotag│    boot│configstatus│  drain│  active│     health│
└──────────────────────┴────┴───┴────────────────────────────────────────────────────────┴──────────┴────────┴────────┴────────────┴───────┴────────┴───────────┘
 ajp.cern.ch            1095  17 /ceph/edeeecc1-6aaf-4672-a657-ff8910ca9ed3/fst.00/         cephfs.0      ajp  opserror        drain  failed   online  no smartctl
 ajp.cern.ch            1095  18  /ceph/edeeecc1-6aaf-4672-a657-ff8910ca9ed3/fst.01         cephfs.0      ajp  opserror        drain  failed   online  no smartctl
 ──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
 ajp.cern.ch            1095   1                                                 /data/01  default.0      ajp    booted           rw nodrain overload  no smartctl
 ajp.cern.ch            1095   2                                                 /data/02  default.0      ajp    booted           rw nodrain   online  no smartctl
 ajp.cern.ch            1095   3                                                 /data/03  default.0      ajp    booted           ro nodrain   online  no smartctl
 ajp.cern.ch            1095   4                                                 /data/04  default.0      ajp    booted           ro nodrain overload  no smartctl
 ajp.cern.ch            1095  11                                                 /data/05  default.0      ajp    booted           ro nodrain   online  no smartctl
 ajp.cern.ch            1095  12                                                 /data/06  default.0      ajp    booted           ro nodrain   online  no smartctl
 ajp.cern.ch            1095  13                                                 /data/07  default.0      ajp    booted           ro nodrain   online  no smartctl
 ajp.cern.ch            1095  14                                                 /data/08  default.0      ajp    booted           ro nodrain   online  no smartctl
 ──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
 ajp.cern.ch            4001   5                                                 /rain/1/     rain.0                              rw nodrain  offline
 ajp.cern.ch            4002   6                                                 /rain/2/     rain.0                              rw nodrain  offline
 ajp.cern.ch            4003   7                                                 /rain/3/     rain.0                              rw nodrain  offline
 ajp.cern.ch            4004   8                                                 /rain/4/     rain.0                              rw nodrain  offline
 ajp.cern.ch            4005   9                                                 /rain/5/     rain.0                              rw nodrain  offline
```

Summary

- The presented features allow to **improve** overall and individual **performance** experience

- **Use** these features **with caution** - only apply them to well defined workloads - they can bit you back …

- Still the **best** is **not to have to use them** because your resources are overcommitted

https://eos-docs.web.cern.ch/using/policies.html

CERN storage technology
used at the Large Hadron Collider (LHC)

EOS Open Storage

Thank you!

Question or Comments?

eos.web.cern.ch