TECH WEEK
STORAGE 24

EOS Open Storage

CERN Benchmarks with EOS

Dr. **Andreas-Joachim Peters**

for the EOS Project - CERN IT - Storage Group

IT Auditoirium - CERN

15.03.2024

Contents

- **EOS** internal **namespace benchmark tool**
- **EOS read record** – summer 2023
- **EOS O2 write tests** – december 2023
- **EOS DC'24 prepration tests** – january 2024
- Summary & Outlook

EOS Internal Namespace Benchmark

- Since **EOS** 5.1.20 the MGM can run an internal performance benchmark
  - for this purpose <N> threads are spawned running a pure meta-data performance test

- The benchmark is invoked with this syntax

```
ns benchmark <n-threads> <n-subdirs> <n-subfiles> [prefix=/benchmark]
   run's a MD benchmark inside the MGM - results are printed into the MGM logfile and the shell
               n-threads  : number of parallel threads running a benchmark in the MGM
               n-subdirs  : directories created by each threads
               n-subfiles : number of files created in each sub-directory
               prefix     : absolute directory where to write the benchmark files - default is /benchmark
```

EOS Internal Namespace Benchmark

The benchmark reports the following values in the MGM log file and on the console output - be careful not to create too many threads on low-memory nodes

EOS Internal Namespace Benchmark

The benchmark reports the following values in the MGM log file and on the console output - be careful not to create too many threads on low-memory nodes

Benchmark	VM 100,10,10	PHYS 100,1,100	PHYS 1000,1,100
Directory Creation	4 kHz	1.6 kHz	1.3 kHz
File Creation	1.6 kHz	2.8 kHz	2.75 kHz
File Creation EEXIST	1.6 kHz	43 kHz	37 kHz
File Read Open	16 kHz	35 kHz	29 kHz
File Open Update	15 kHz	34 kHz	28 kHz
File Deletion	34 kHz	2.9 kHz	2.6 kHz

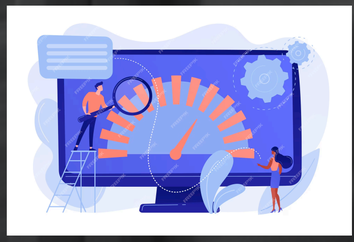
EOS Internal Namespace Benchmark

The benchmark reports the following values in the MGM log file and on the console output - be careful not to create too many threads on low-memory nodes

```
EOS Console [root://localhost] |/eos/ams02/proc/conversion/> ns benchmark 1000 1 100
240310 17:43:16 time=1710088996.586753 func=BenchmarkSubCmd          level= logid=static.............................. unit=mgm@eosar
n.ch:1094 tid=00007fa3cd16f640 source=NsCmd:1166                     tident= sec=(null) uid=99 gid=99 name=- geo="" [   mkdir     ] c
=0.75 dir-rate=1333.33
240310 17:43:53 time=1710089033.060240 func=BenchmarkSubCmd          level= logid=static.............................. unit=mgm@eosar
n.ch:1094 tid=00007fa3cd16f640 source=NsCmd:1208                     tident= sec=(null) uid=99 gid=99 name=- geo="" [   create    ] t
ime=36.47 file-rate=2741.72 Hz
240310 17:43:55 time=1710089035.756839 func=BenchmarkSubCmd          level= logid=static.............................. unit=mgm@eosar
n.ch:1094 tid=00007fa3cd16f640 source=NsCmd:1255                     tident= sec=(null) uid=99 gid=99 name=- geo="" [   exists    ] t
irs=1000 time=2.70 dir-rate=370.84 file-rate=37084.18 Hz
240310 17:43:59 time=1710089039.200416 func=BenchmarkSubCmd          level= logid=static.............................. unit=mgm@eosar
n.ch:1094 tid=00007fa3cd16f640 source=NsCmd:1297                     tident= sec=(null) uid=99 gid=99 name=- geo="" [   read      ] t
ime=3.44 file-rate=29039.86 Hz
240310 17:44:02 time=1710089042.807398 func=BenchmarkSubCmd          level= logid=static.............................. unit=mgm@eosar
n.ch:1094 tid=00007fa3cd16f640 source=NsCmd:1339                     tident= sec=(null) uid=99 gid=99 name=- geo="" [   write     ] t
ime=3.61 file-rate=27724.26 Hz
240310 17:44:40 time=1710089080.533804 func=BenchmarkSubCmd          level= logid=static.............................. unit=mgm@eosar
n.ch:1094 tid=00007fa3cd16f640 source=NsCmd:1381                     tident= sec=(null) uid=99 gid=99 name=- geo="" [   deletion  ] t
```

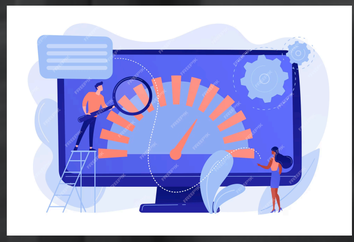
File Open Update	15 kHz	34 kHz	28 kHz
File Deletion	34 kHz	2.9 kHz	2.6 kHz

EOS Reading 1 TB/s

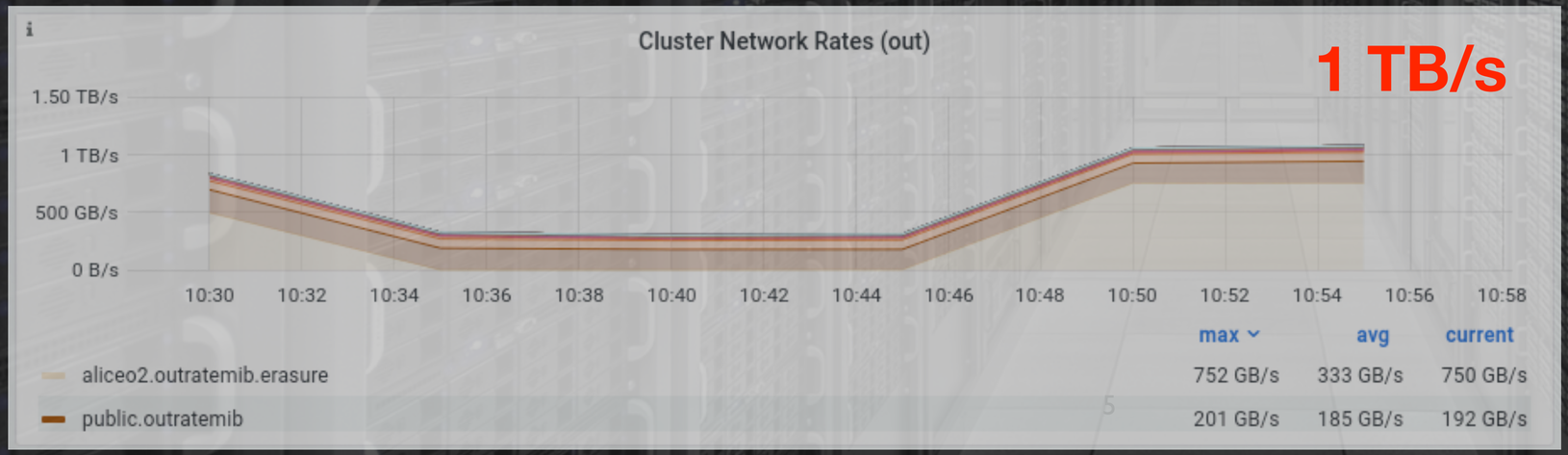


- In Summer 2023 we were running a read test on O2 using eoscp with 2 GB files and PIO mode - client talk directly to FSTs for EC
- We reached **700 GB/s** plus the current read activity in other EOS instances at that moment - the rate was sustained over 5+ minutes

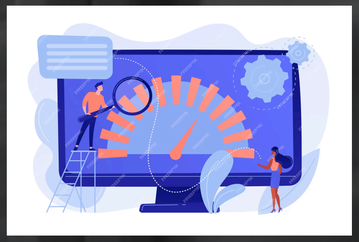
EOS Reading 1 TB/s



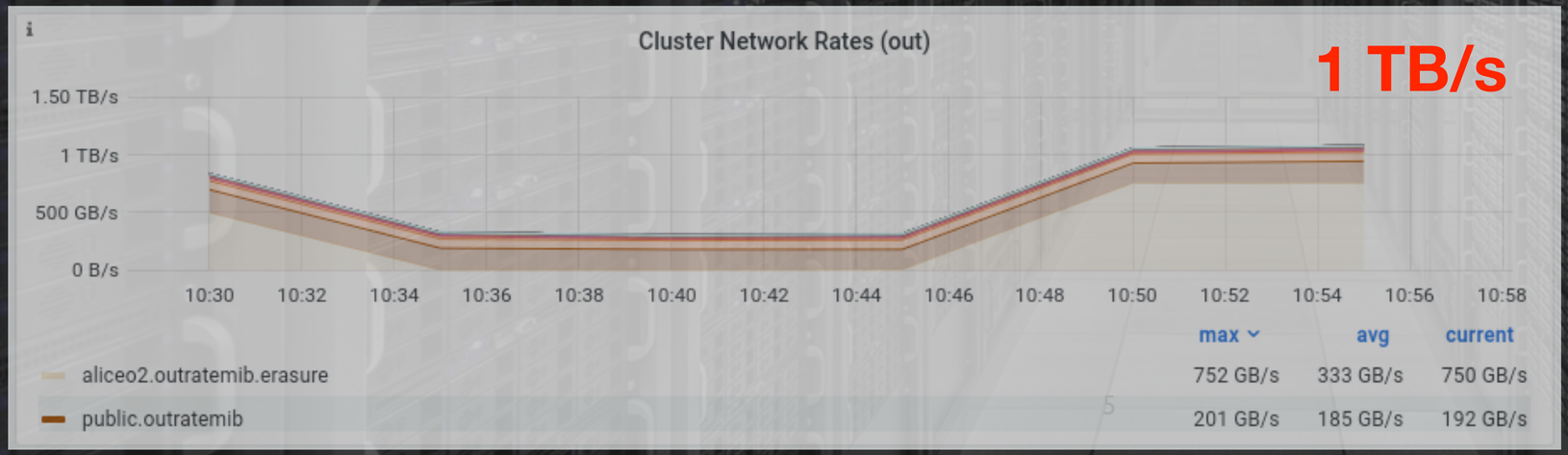
- In Summer 2023 we were running a read test on O2 using eoscp with 2 GB files and PIO mode - client talk directly to FSTs for EC
- We reached **700 GB/s** plus the current read activity in other EOS instances at that moment - the rate was sustained over 5+ minutes

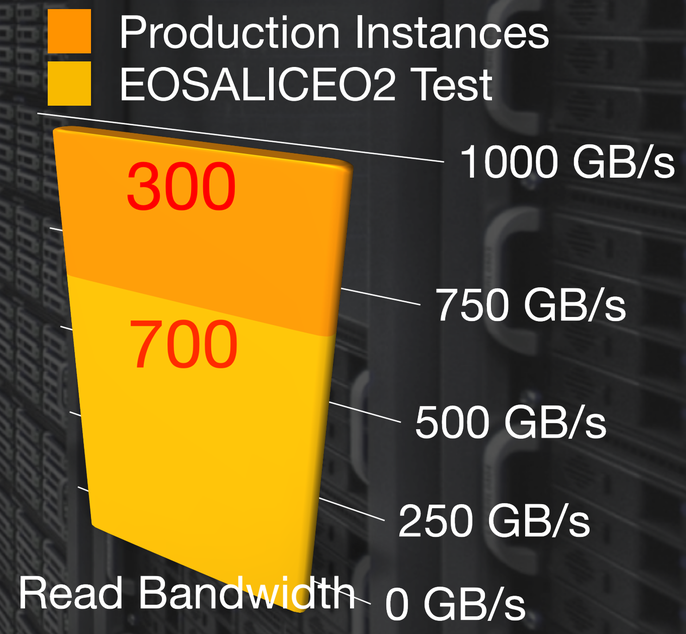


EOS Reading 1 TB/s



- In Summer 2023 we were running a read test on O2 using eoscp with 2 GB files and PIO mode - client talk directly to FSTs for EC
- We reached **700 GB/s** plus the current read activity in other EOS instances at that moment - the rate was sustained over 5+ minutes





EOS O^2 writing 380 GB/s

EOS O^2 writing 380 GB/s

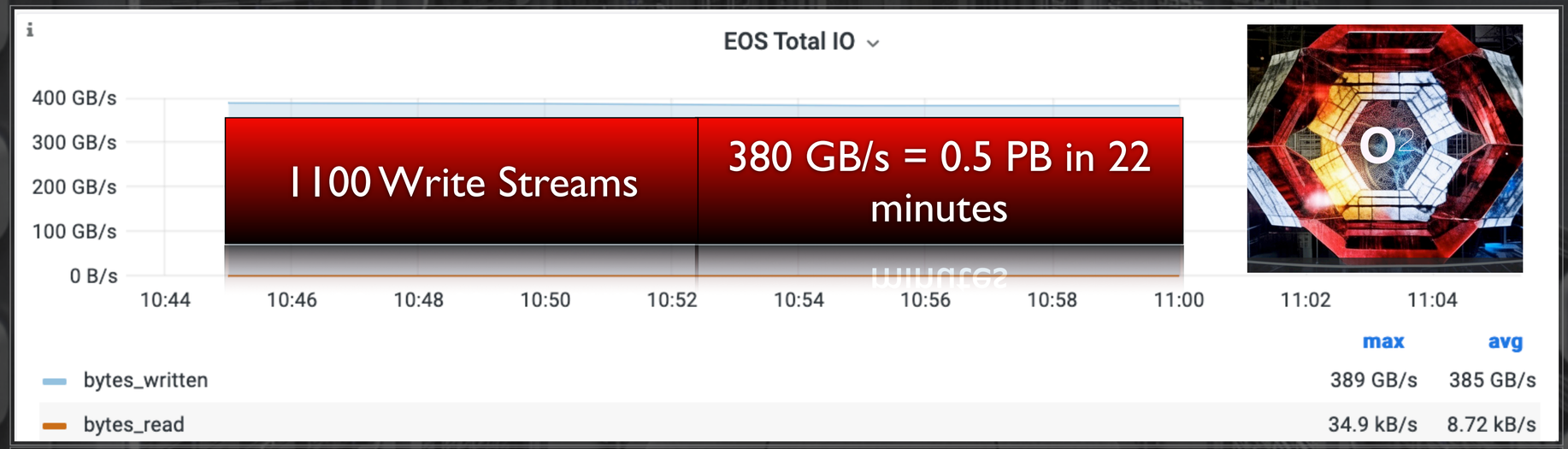
- In December 2023 we did a test to evaluate if it is possible to run O^2 with a reduced capacity writing at 170 GB/s - result was with 50% of the disks this was still no problem [~6k HDDs]

EOS O^2 writing 380 GB/s

- In December 2023 we did a test to evaluate if it is possible to run O^2 with a reduced capacity writing at 170 GB/s - result was with 50% of the disks this was still no problem [~6k HDDs]
- During the test learned lessons about importance of balancing within groups and between groups to guarantee high bandwidth even if the instance is > **90 %** filled

EOS O^2 writing 380 GB/s

- In December 2023 we did a test to evaluate if it is possible to run O^2 with a reduced capacity writing at 170 GB/s - result was with 50% of the disks this was still no problem [~6k HDDs]
- During the test learned lessons about importance of balancing within groups and between groups to guarantee high bandwidth even if the instance is > **90 %** filled



EOS O^2 writing 380 GB/s

EOS O^2 writing 380 GB/s

- Read the full journey here

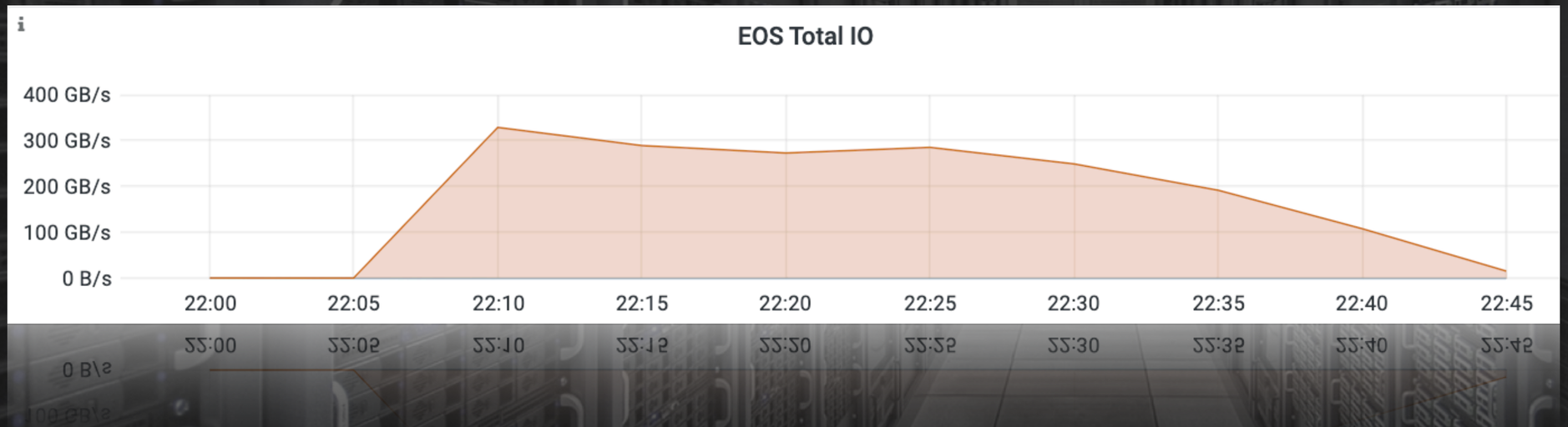
EOS O^2 writing 380 GB/s

- Read the full journey here

EOS O² writing 380 GB/s

- Read the full journey here

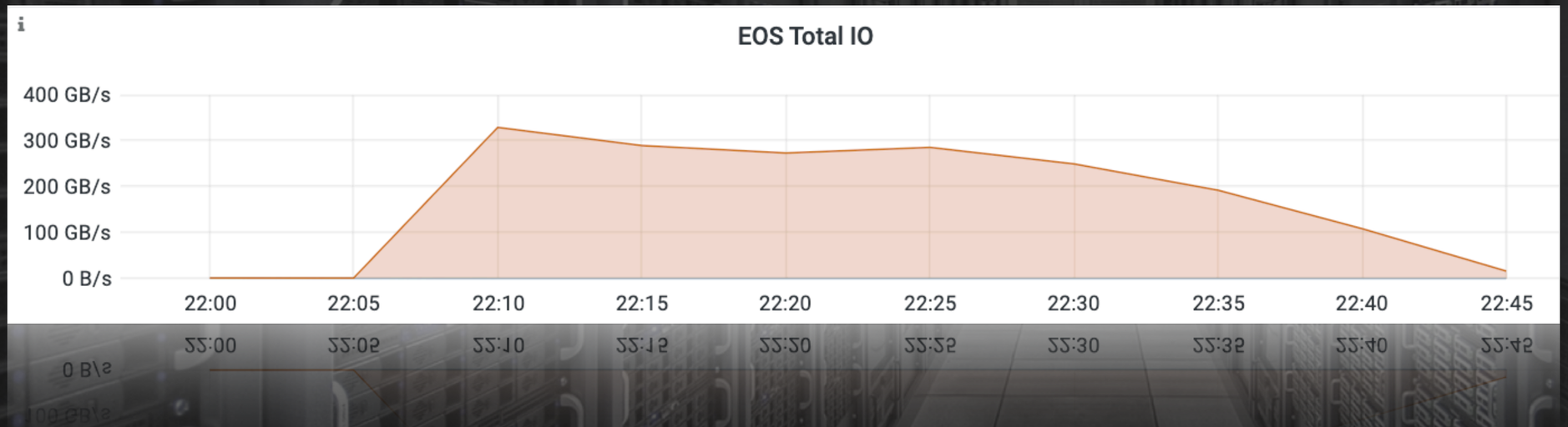
- This is how a the benchmark looks when you have packet loss on one out of 125 nodes (dropped packets)



EOS O² writing 380 GB/s

- Read the full journey here

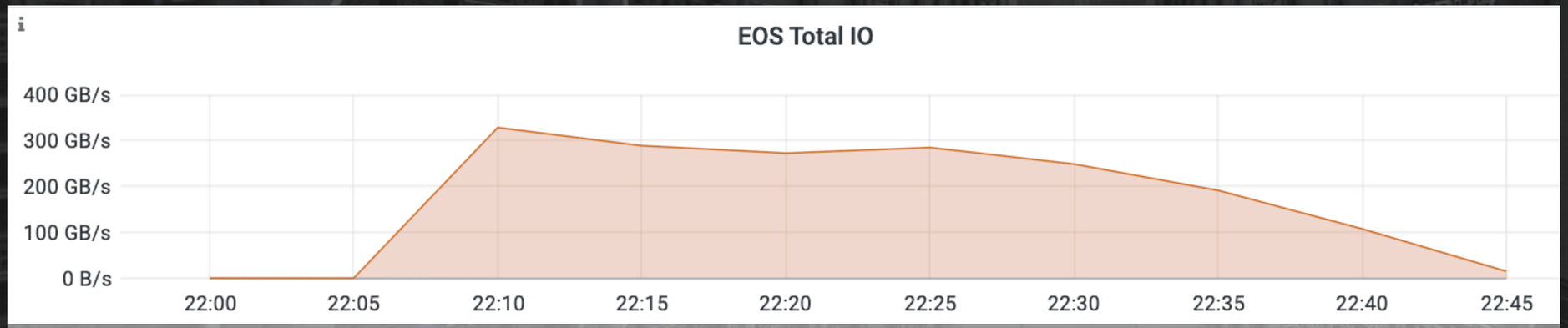
- This is how a the benchmark looks when you have packet loss on one out of 125 nodes (dropped packets)



EOS O² writing 380 GB/s

- Read the full journey here

- This is how a the benchmark looks when you have packet loss on one out of 125 nodes (dropped packets)

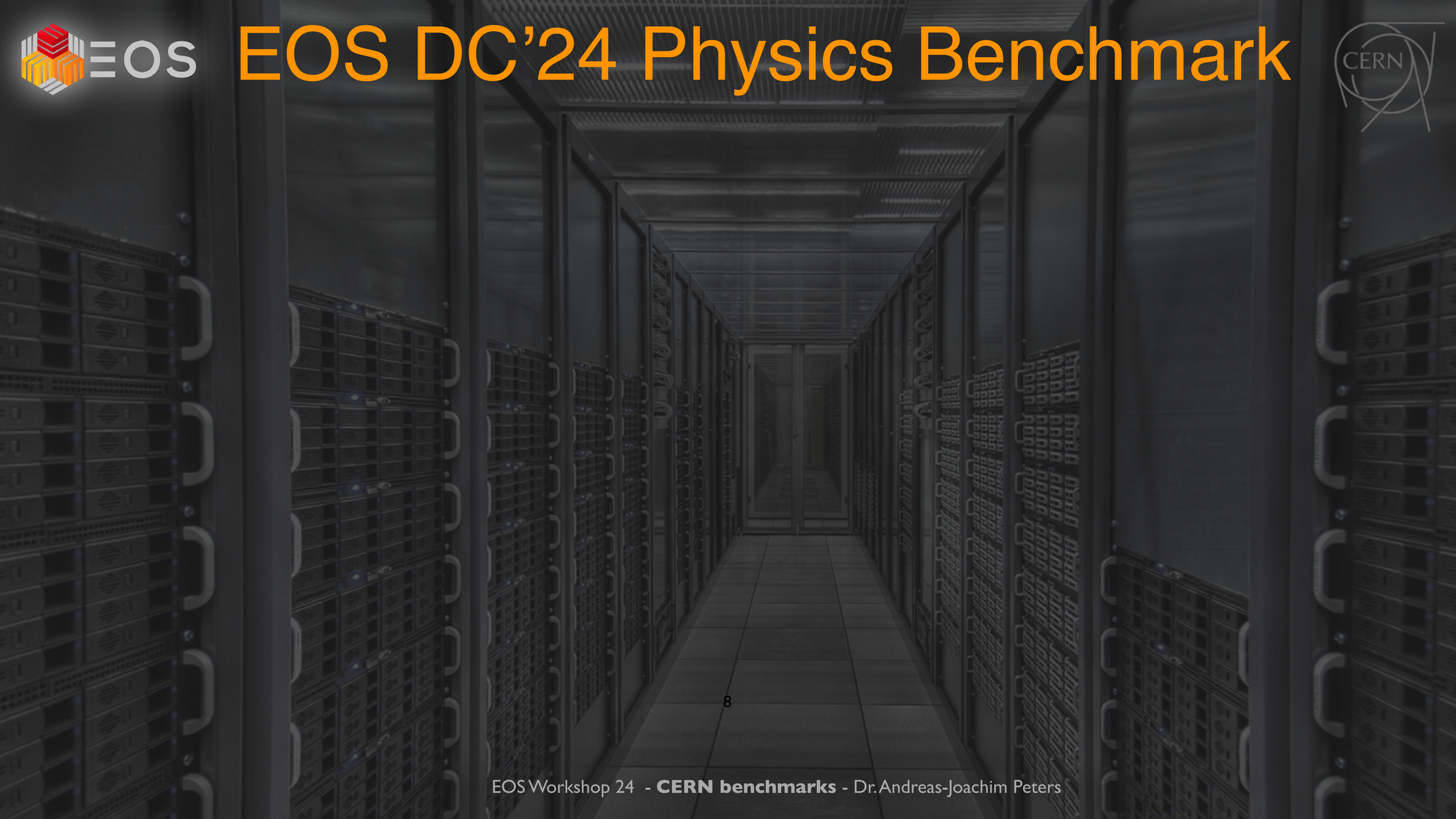



```
commit c77554a52fc6b2b7b5a85c0f30dc8007a4e45cb6
Author: Elvin Alin Sindrilaru <elvin.alin.sindrilaru@cern.ch>
Date:   Mon Jan 22 16:51:49 2024 +0100

    FST: Publish network RX/TX errors and dropped packet counters. Fixes EOS-5971
    sudo eos node status elvin-dev01.cern.ch:2001 | grep net
    stat.net.ethratemib               := 119
    stat.net.inratemib                := 0.00197497
    stat.net.outratemib               := 0.000249533
    stat.net.rx_dropped               := 5
    stat.net.rx_errors                := 0
    stat.net.tx_dropped               := 0
    stat.net.tx_errors                := 0
```

- One +outcome was that you can now see packet loss using eos node status … | grep net

EOS DC’24 Physics Benchmark



EOS DC’24 Physics Benchmark

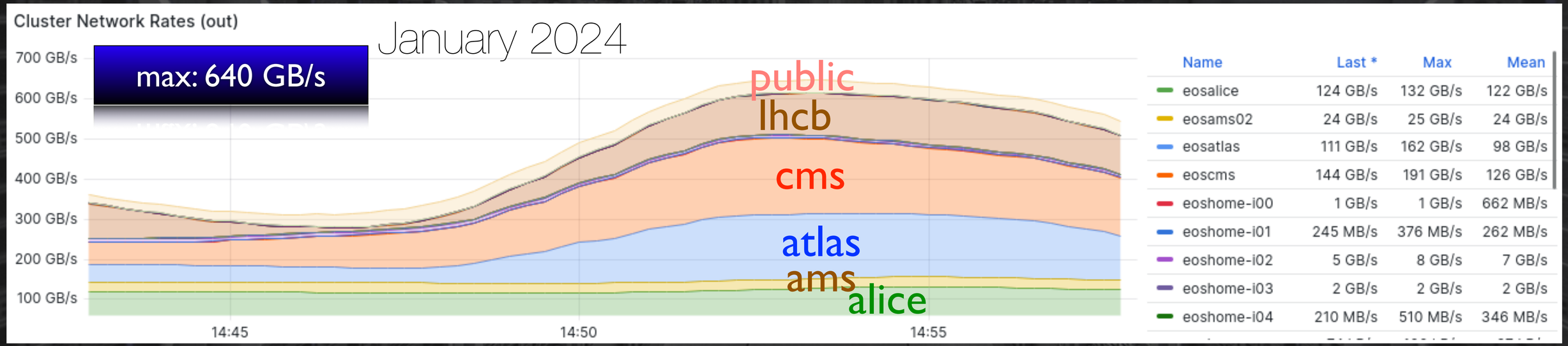
As preparation to the **DC’24** we wanted to see, if there is an interference if we read a lot of data out of 4 LHC instances and EOSPUBLIC at the same time - we used 75x 100 GE clients as readers

Result: there was no visible interference between instances - all instances can easily go over 100 GB/s

EOS DC'24 Physics Benchmark

As preparation to the **DC'24** we wanted to see, if there is an interference if we read a lot of data out of 4 LHC instances and EOSPUBLIC at the same time - we used 75x 100 GE clients as readers

Result: there was no visible interference between instances - all instances can easily go over 100 GB/s



Summary & Outlook

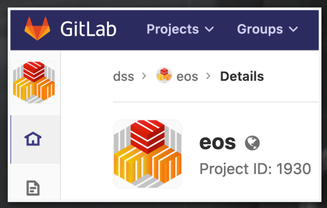
- You can now benchmark your MGM hardware using eos ns benchmark
- EOS instances provide very good streaming performance and there is no visible network interference between physics instances
- When reading from **O²** we are close to the client bandwidth limit (75x10GE)
  - to increase write performance we might add client-side erasure coding as a plug-in avoiding doubling of the network bandwidth - an alternative would be to upgrade the FST ethernet to 200GE - if required
  - however in production usage we are still far away from hitting the FST based EC writing limit measured 380 GB/s
- Benchmarks are helpful to improve error monitoring!

Useful Links

Web Page https://eos.cern.ch

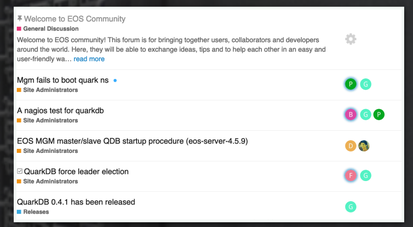
GITLAB Repository https://gitlab.cern.ch/dss/eos

GITHUB Mirror https://github.com/cern-eos/eos



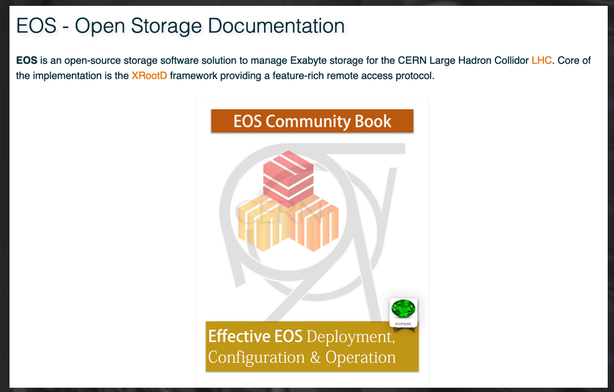
Community Forum https://eos-community.web.cern.ch/

email: eos-community@cern.ch



Documentation http://eos-docs.web.cern.ch/eos-docs/

Support email: eos-support@cern.ch




Thank you for your attention!
Questions?
SOCAR
VESTIAR
CHANGE