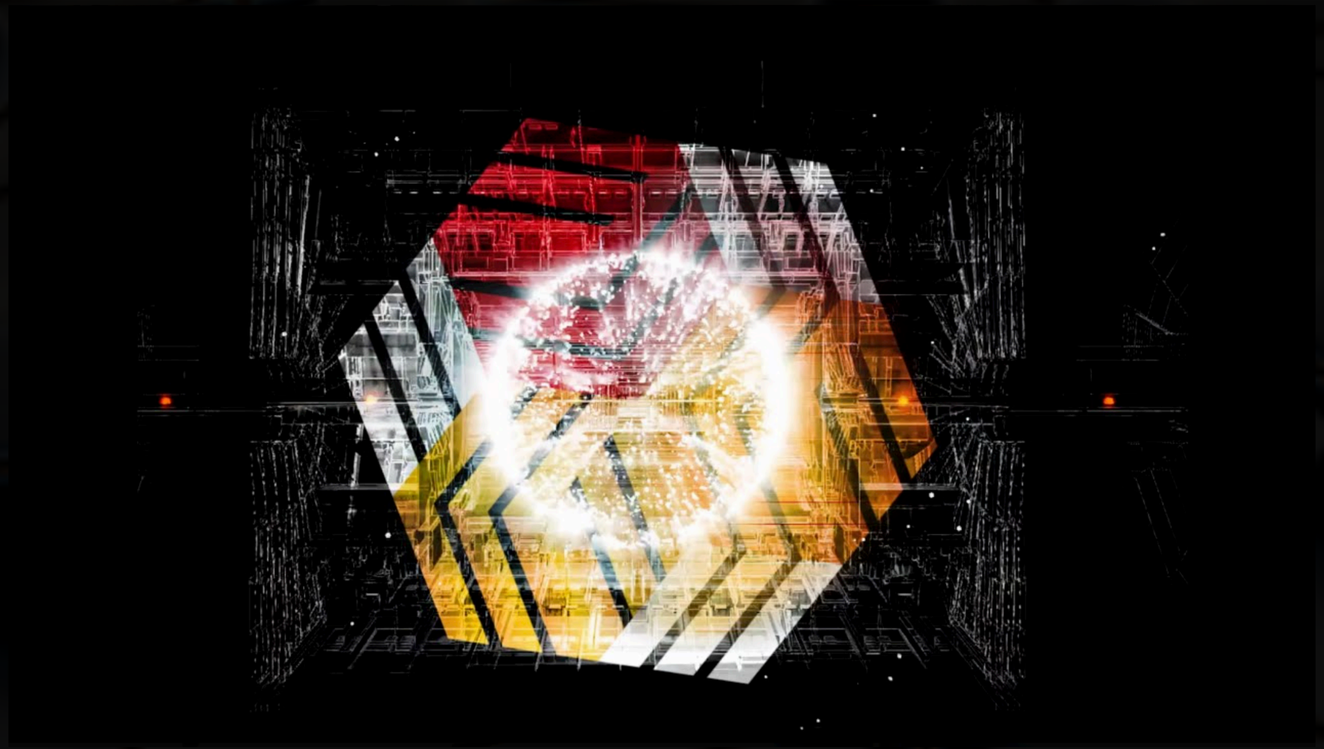
Local vs Remote

A battle with IO insights

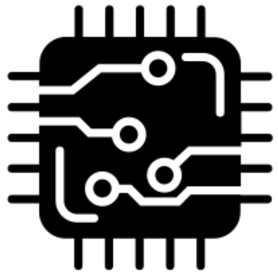


Andreas-Joachim Peters
IT-SD

Outline

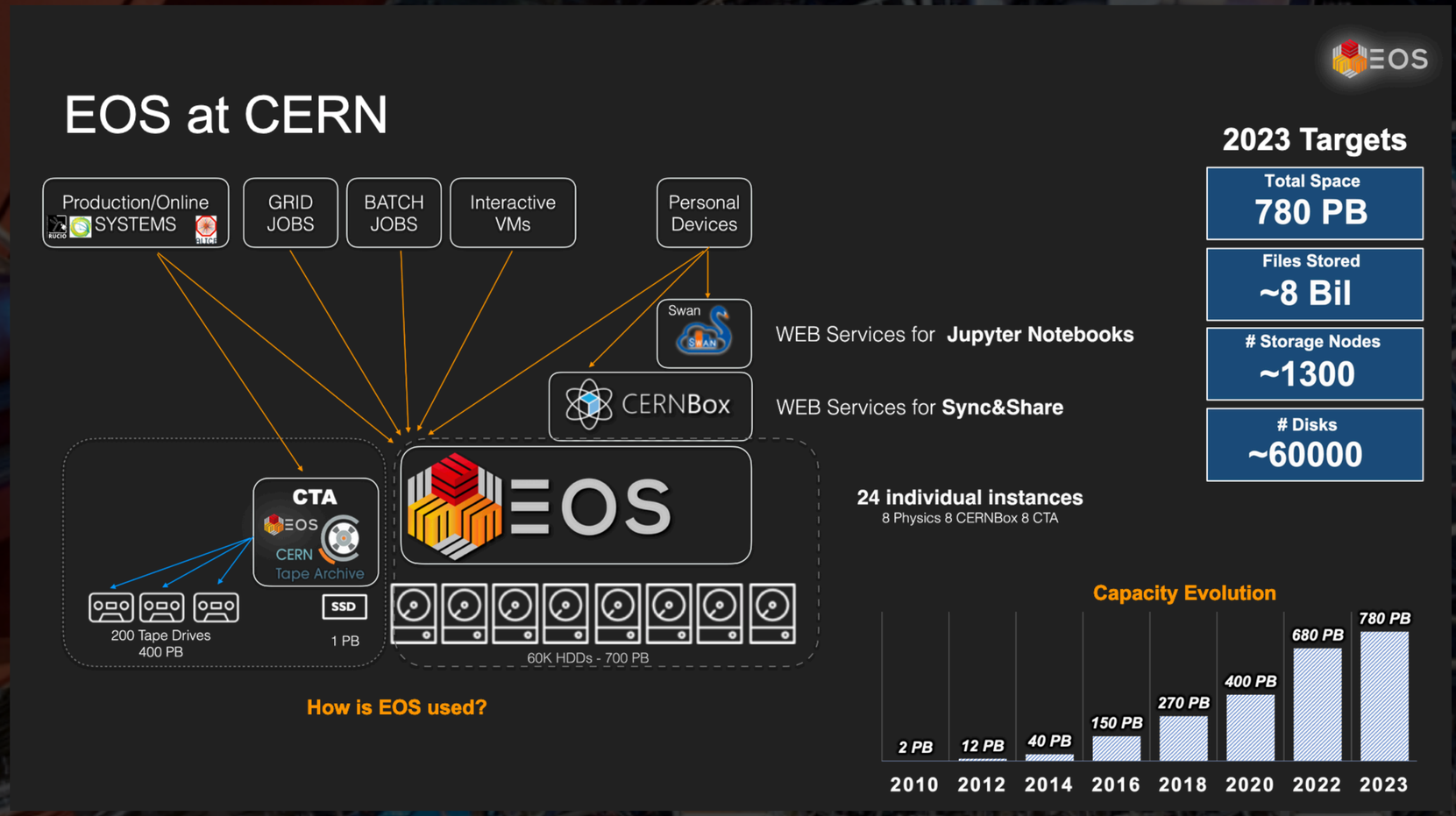
- Quantitive analysis between high-performance local and remote file access
  - where are we today?
  - what can we improve?
  - where are the bottlenecks?

Hardware



Comparisons

EOS as Remote



EOS services at CERN are provided by 60k HDDs

- in the largest instance a single USER can involve 9600 HDDs in a single workflow
- IOPS & bandwidth is available on large scale
- ‘only’ bottleneck to compensate is latency!

Dataflow & Storage
ALICE LHC Experiment

Worldwide LHC Computing GRID

ALICEO²

Now 135 PB

CERN Computer Center

CERN Experimental Site

10-40 GB/s

shared
1 PB

TAPE
EOS
SDD

10 GB/s

110 PB

EOS

9600
HDDs

10+ GB/s

10-100 GB/s

Processing
Analysis

CERN
Cloud

shared
20 PB

CEPH
HDD/SSD

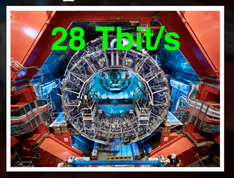
100 GB/s

48h Storage
Fallback Buffer
Disk

EOS

14 PB

Experiment



100-250 GB/s

250 Nodes
+
2k x GPUs

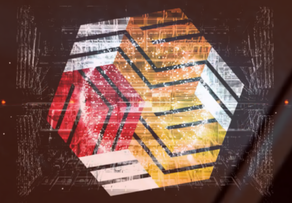
5 Tbit/s

100-250 GB/s

3h Storage
Realtime Buffer
Flash

Posix FS

1 PB



O²

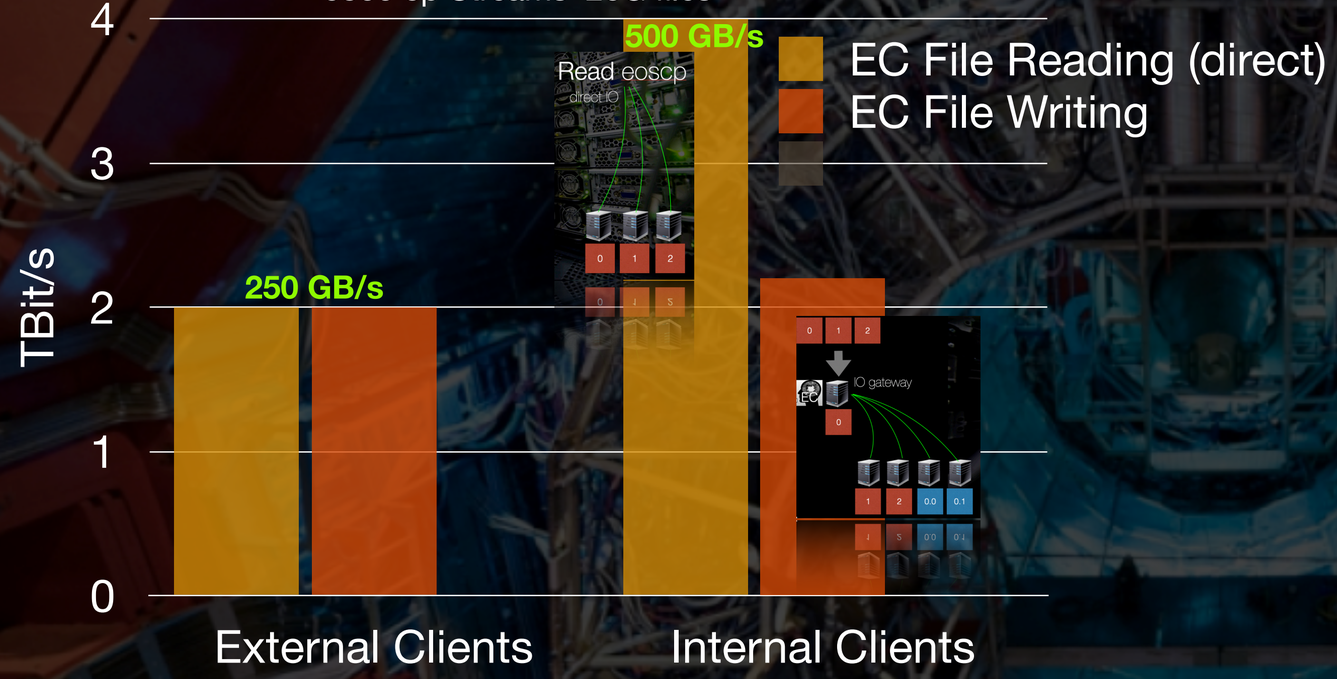
New Standard Model for **EOS** Physics Storage

- **O²** 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 2.5 GB/s - >800 IOPS

O² Benchmarks 07/03/23

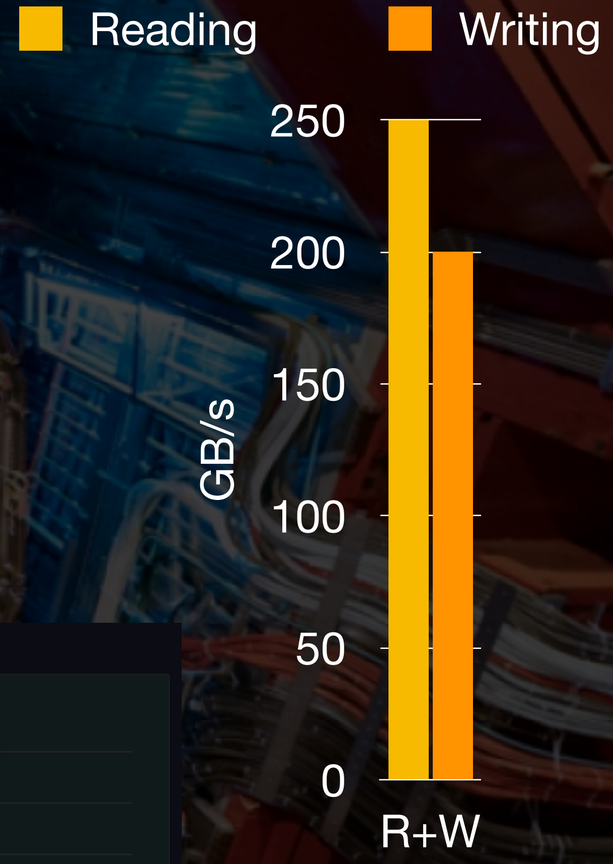
READ or WRITE

6800 cp Streams 20G files

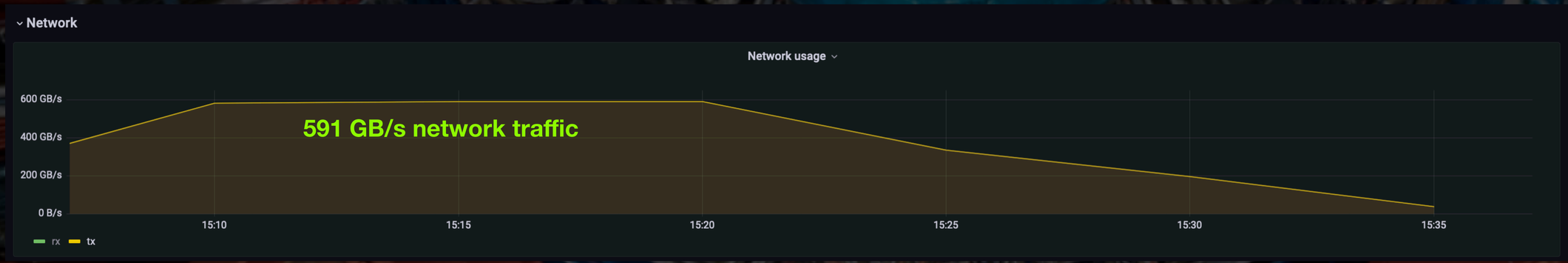


READ and WRITE

~10 streams per HDD
6.8k cp Streams in total



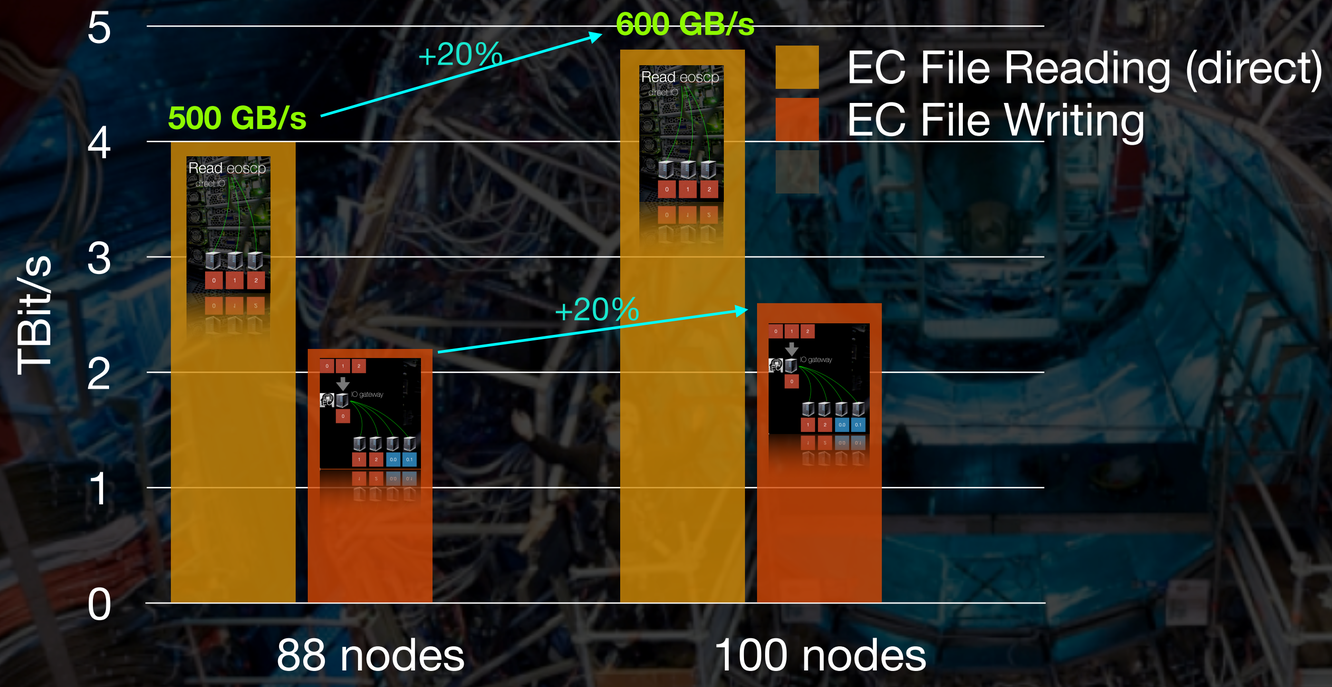
Expect Instance Capacity/Perf
+50% extension soon…



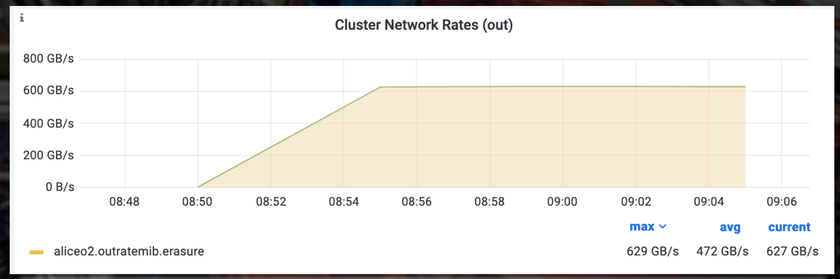
O² Benchmarks 23/03/23

READ or WRITE

6800/7400 cp Streams 20G files



600 GB/s network traffic

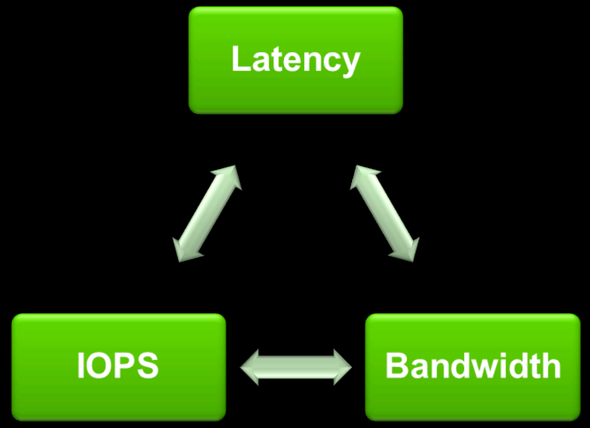


Two extremes in analysis use-cases

MD(**IOPS)** dominated vs Data(**Bandwidth)** dominated

HPC Local NVMe Storage

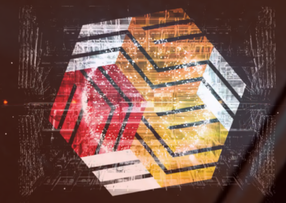
HPC Remote Volume Storage



	read 128kb from 20.000 files in parallel 2.5 GB in total	read 8 GB from 180 files in parallel 1.5 TB in total	
Capacity 56 TB	few seconds first run local NVME RAID0 **0.3s** second run local NVME RAID0	**210s** first run local NVME RAID0 **210s** second run local NVME RAID0	**16xNVMe RAID0**
Capacity 100 PB	**200s** @ 100Hz on EOS p=200	**210s** first run on EOS O2 **190s** second run on EOS 02	**O2 EC12+2**
	1 MB/s [remote] - few GB/s [local]	7-8 GB/s [local/remote]	

in this mode the whole 100 PB EOS O2 instance delivers **500 MB/s**
Unfortunate mode of operation!

in this mode the whole EOS O2 instance delivers **250 GB/s**



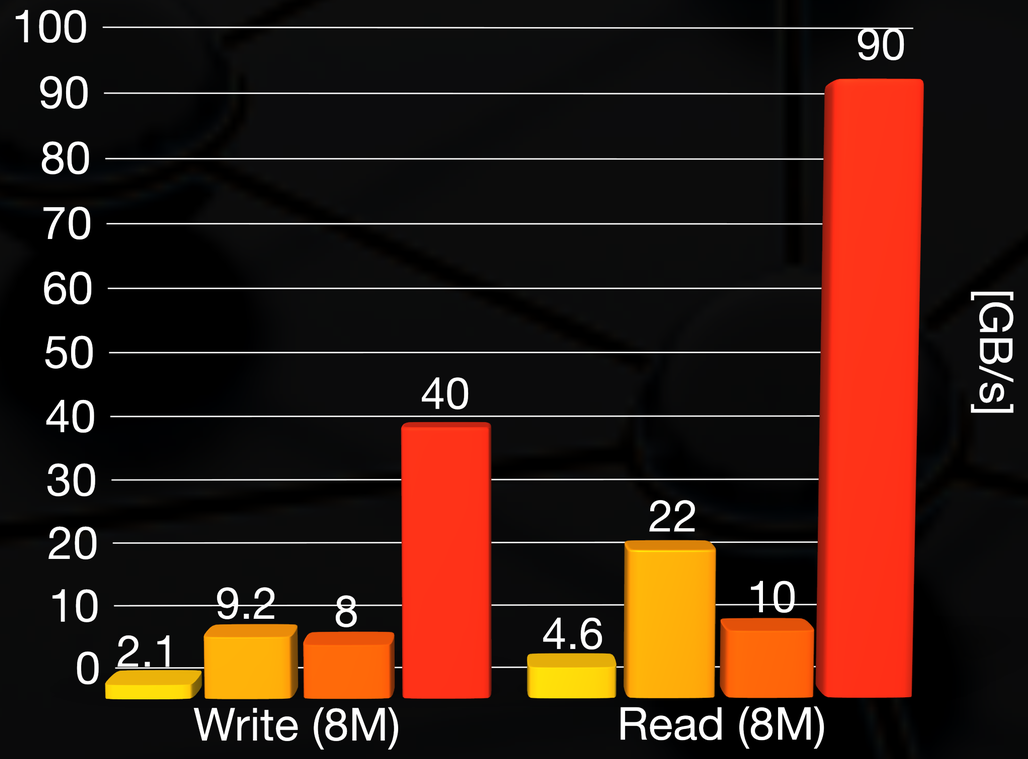
analysis8 lxplus nodes

CERN standard node for local analysis

Memory	**1 TB** **3200 MT/s**	
CPU	**AMD EPYC** **7702 64-Core**	
Cores	128/256	
Network	100GE	
Storage	RAID1 2xNVME **/home**	RAID0 16xNVME = 56 TB **/scratch**

High-Core Analysis Nodes

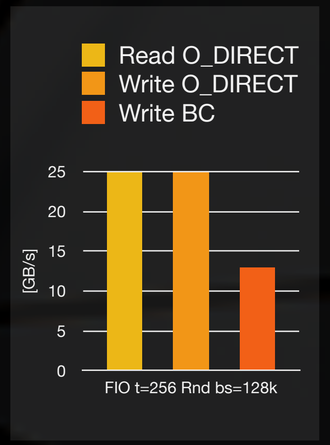
local NVME FS



- you cannot exploit NVME RAID performance when reading through the buffer cache (default for almost everything)
  - ROOT IO, XCache, XRootD no O_DIRECT
    - EOS O2 configured with O_DIRECT for writing but with HDDs !

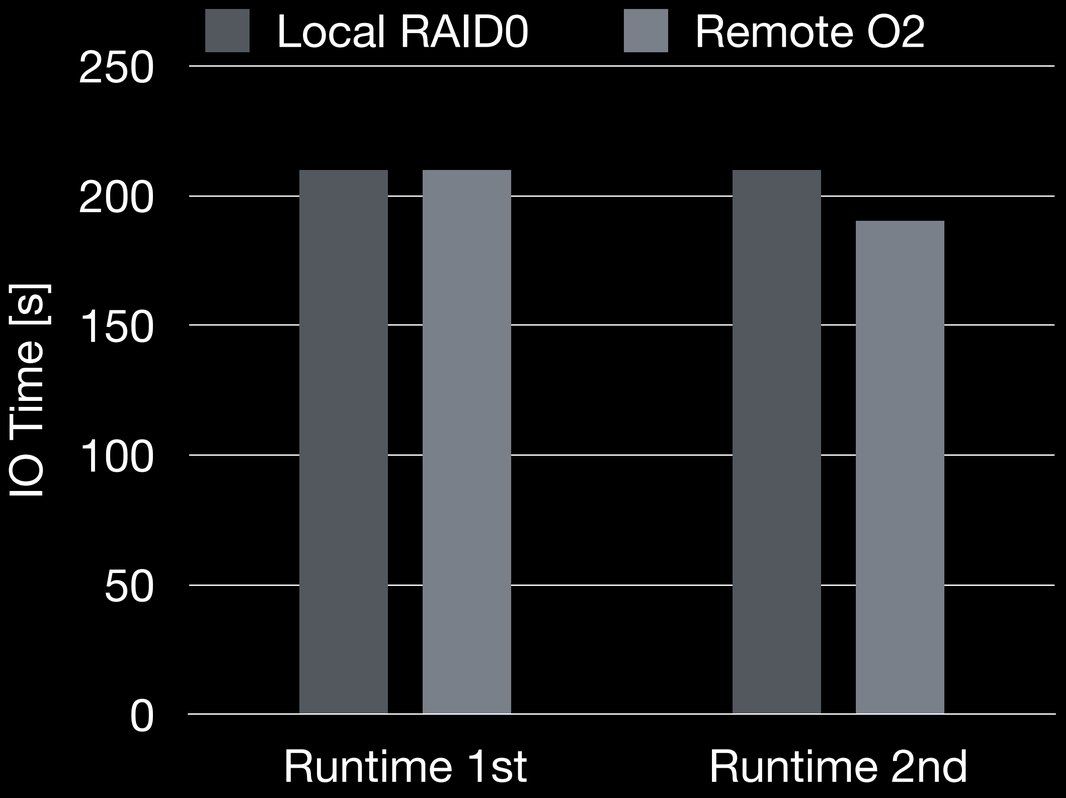
- NVME Software RAID performance also suffers with small block sizes (chunk size 512kb x 16) and shows large fluctuations varying number of streams …

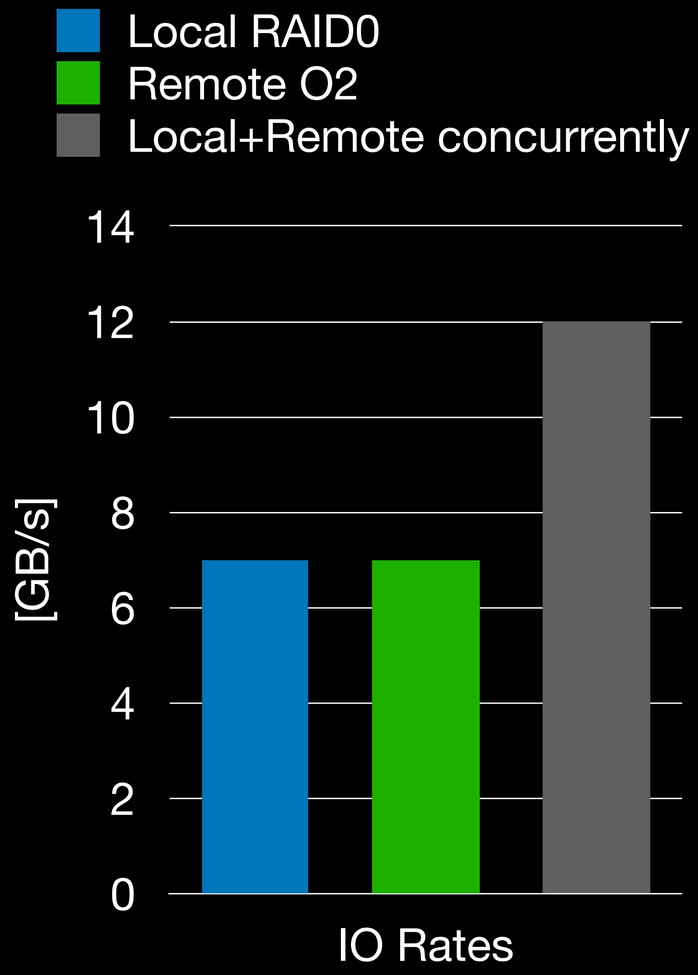
Single Stream (BC)
Single Stream (O_DIRECT)
Multiple Streams (BC)
Multiple Streams (O_DIRECT)

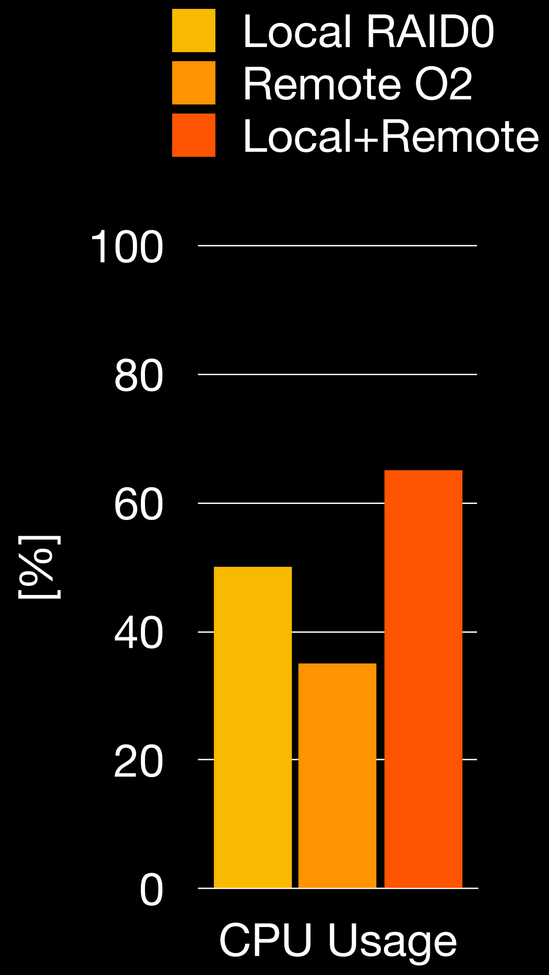


High-Core Analysis Nodes

local fs vs. remote reading (80GB atomic matrix computation J. Bendavid)

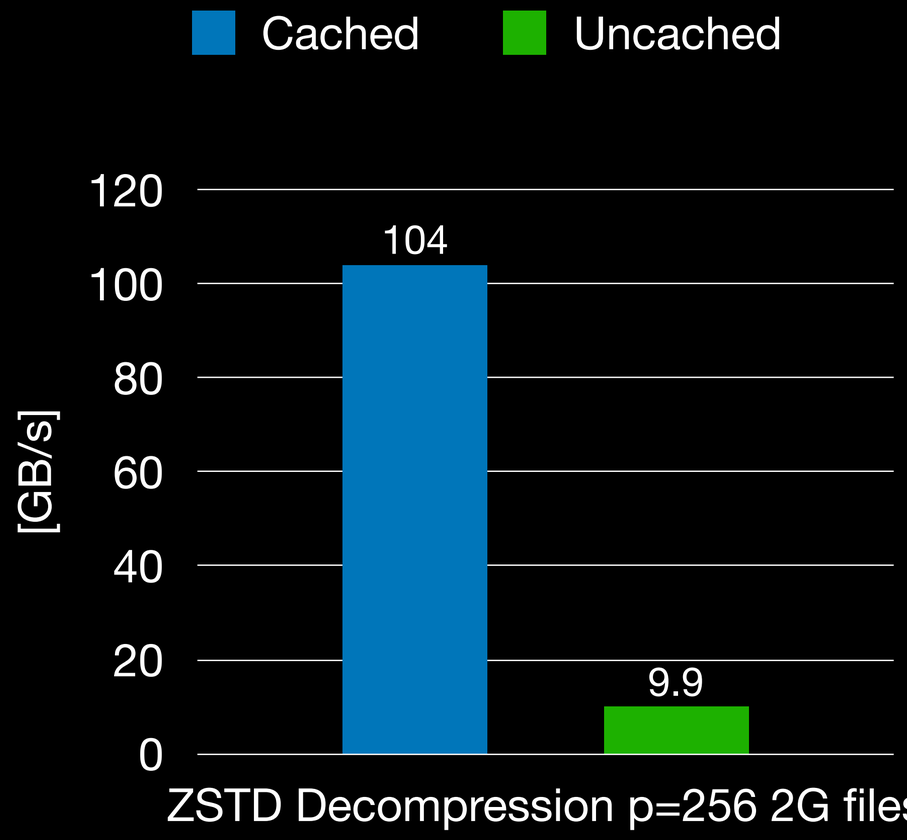


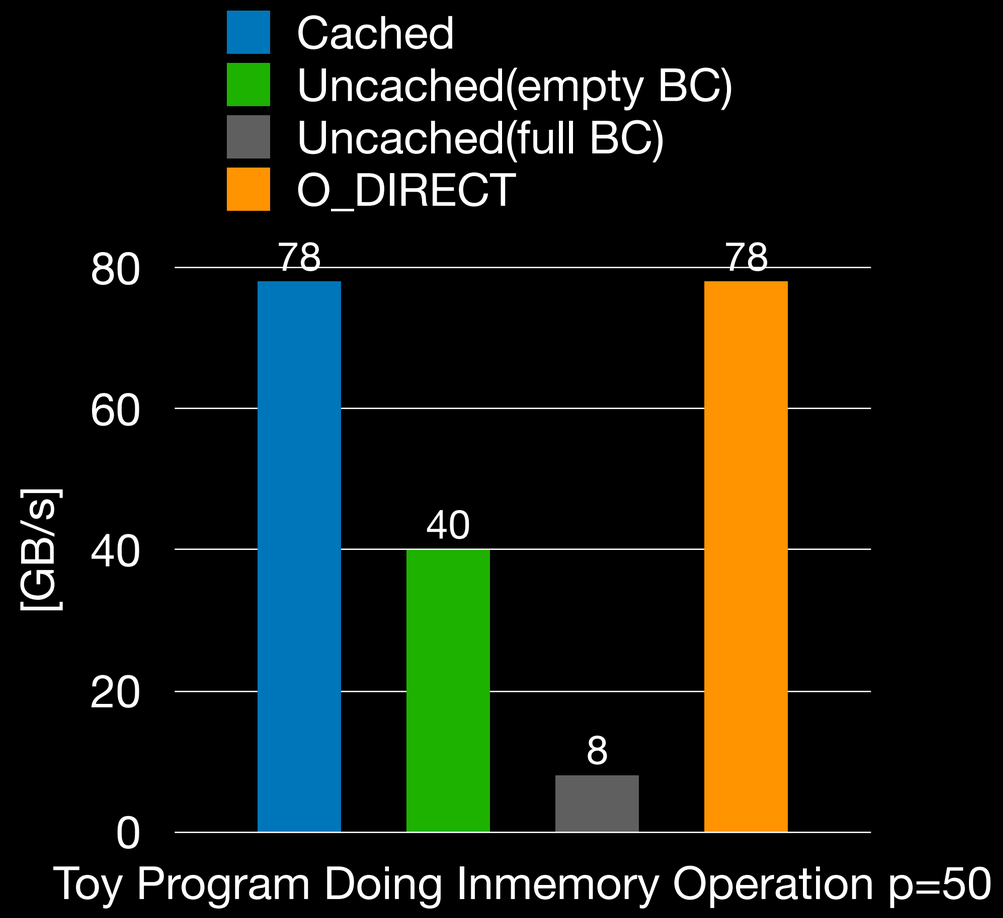




High-Core Analysis

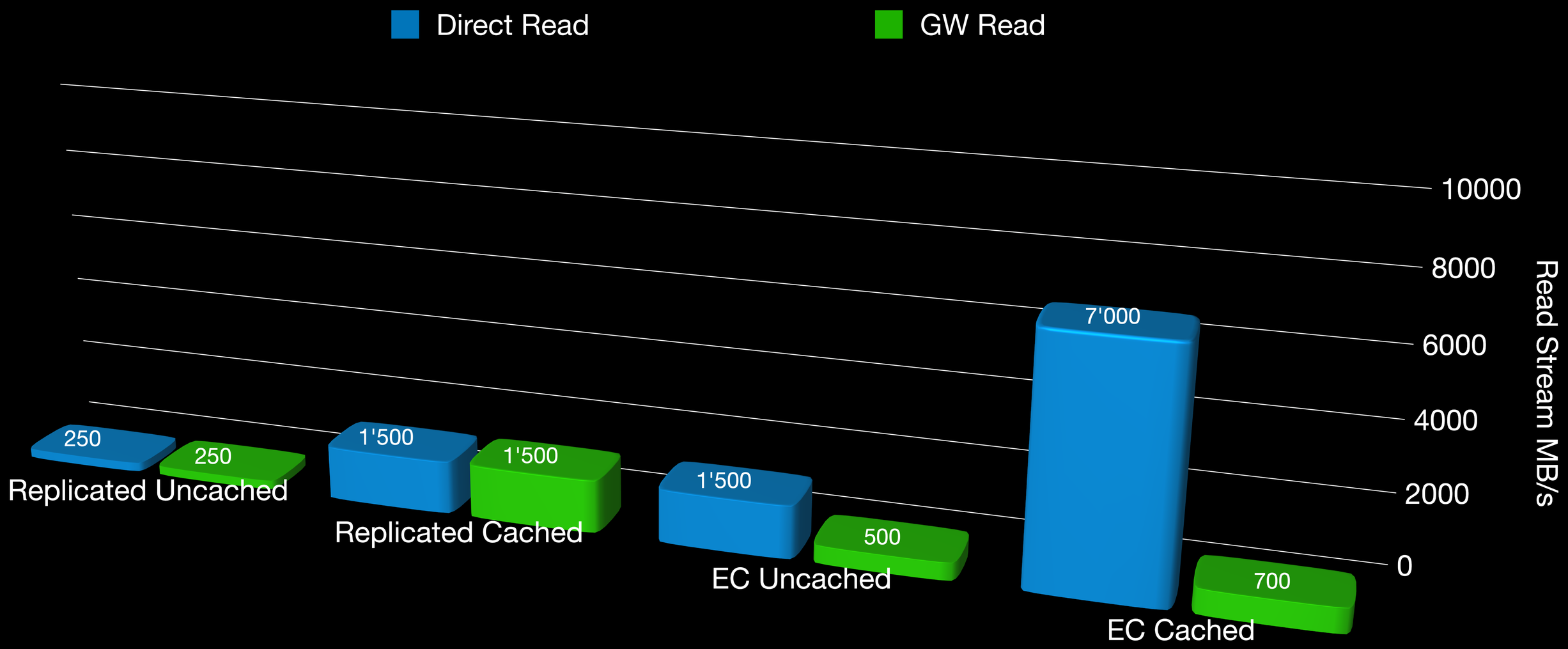
multi-core zstd decompression and toy analysis program





High-Core Analysis

O2 Single Stream Read Rate Erasure Coded Files 10+2 (using eoscp/xroot protocol)



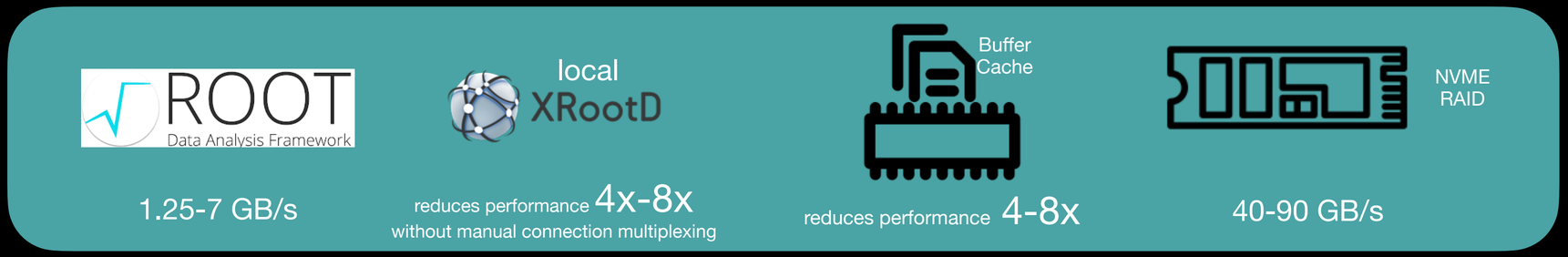
High-Core Analysis

Some conclusions from previous slides

- massive parallel streaming analysis use cases work with EOS …
  - `if (avg. payload is > 100 MB per file) { can exploit instance performance today }`
    - 5000 Hz x 0.1GB = 500 GB/s
- what about meta-data/IOPS limited used cases (end-user analysis …) ?
  - outlook Run-4
    - scale-out namespace
      a) `only local locks in the namespace - better parallelism`
      b) `split EOS namespace into many branches (multiple MGMs)`
    - vector open (bulk) interface for analysis with meta-data limited use-cases
      - `open(1) open(2) … open(N) => open(1,2…N)`

- local IO performance of analysis-type application run into various bottlenecks, which are not *cured* by adding hardware

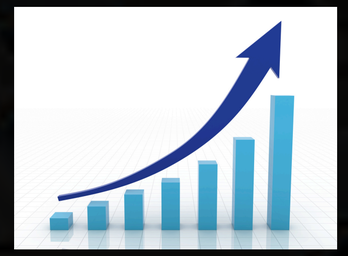
- NVMe with software **RAID0** good until 10GB/s - possibly better with hardware card like GRAID SupremeRaid
  - however **it is hard to imagine to have a mainstream analysis now or in the future which can process data faster than 10GB/s** (100GE) even with 256 cores?



Software

CERN Storage **Scalability**

Example of CMS instance



File Access Limits in EOS

Limitation	Current Usage	Current Limit	Future Limit	Increase by
#Files	200M	1 Billion	10 Billion	Change NVMe's
Total BW	34 GB/s	130 GB/s	600 GB/s -1,2TB/s	10/40GE-> 100GE++
Per File BW		250 MB/s	0.5-2 GB/s	2rep->EC
#open/s	40/100 Hz (2rep w/r)	500/6000 HZ (EC w/r)	5-10x ?	Software Development
Volume	31 PB	40 PB	-	Money

EOS as Remote for Analysis

What users and applications need to take into account now and in the future …

- more and more data will be written with **erasure coding for good reasons**
  - on physics instances
- the speed of individual read streams if faster with EC
  - **~6x** to what you get today
- the time to open a file might be slightly slower
  - but the available IOPS per file is **10x** more

- use latency compensation techniques offered by frameworks …
  - e.g. parallel processing, async open, pre-fetching, vector reads
  - avoid HTTP for analysis

- avoid opening too many files when doing analysis over and over again
  - try to consume at least 100 MB per file open
  - rewrite data for repetitive analysis more efficiently e.g. merge 20k into 200 files
    - stage your data locally if that is an option
    - erasure coding can deal well with large files - **10 -100 GB** is ok
  - use analysis trains!

Conclusions

- It is important to understand and eliminate bottlenecks in application, OS and hardware to provide an efficient platform
  - **XrdCl** needs to be smarter to select multiplexing/multiconnections and adjust thread-pool/event-loop sizes automatically
    - to make the benchmarks fast now, some manual tuning is required!
  - Frameworks (ROOT) and Caches (XCache) might aim to support direct IO
- **EOS** is optimised for streaming use cases
  - improvements for meta-data heavy workloads are possible !
  - an adaption of workflows and the way data is stored to this characteristics is desirable
- **Today** there is no framework which benefits from having local 16xNVMEs. Might be better to invest into more storage-less analysis front-end nodes with remote access than to have less high-performance nodes with local NVME arrays
  - unless you have high-latency to the back-end storage (e.g. remote site!)

The End!

Questions / Comments ?

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