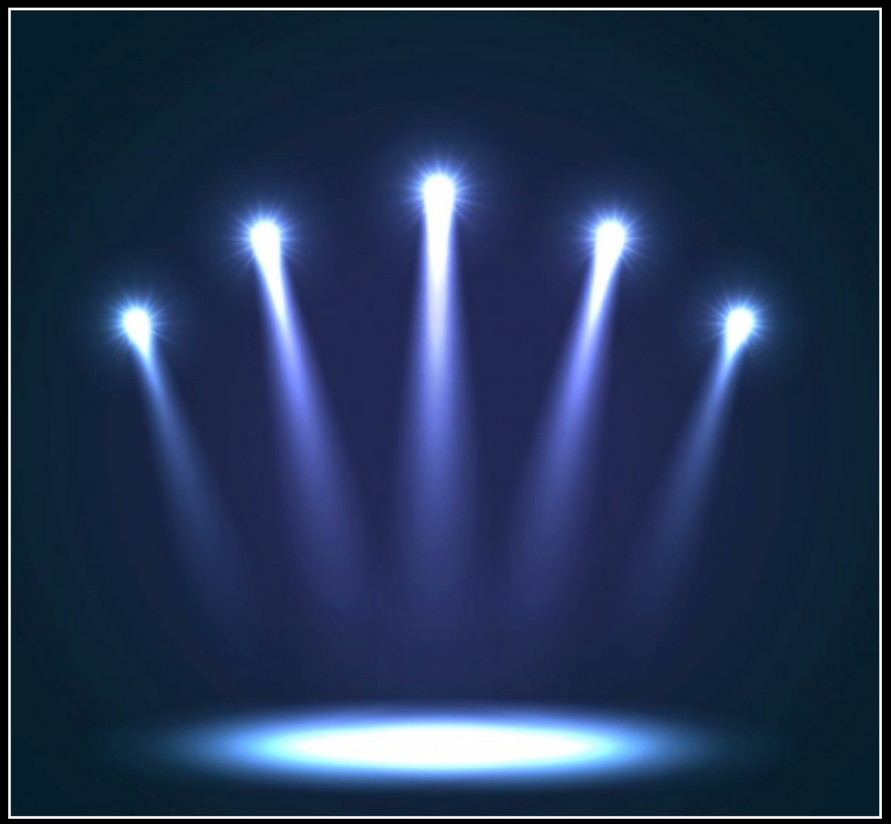
Erasure Coding for Online/Offline in the EOS Open Storage system

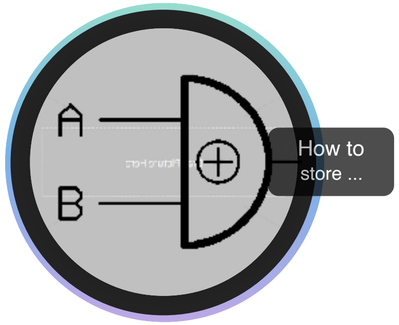
EOS WORKSHOP

Andreas-Joachim Peters
CERN IT Storage Group

Overview

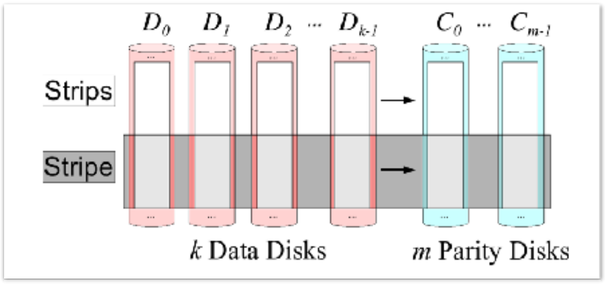
- Introduction to Erasure Encoding
  - Principles of EC
- Benchmarks
  - Single Box Tests
  - O2 Scenario: o2test space
- Outlook

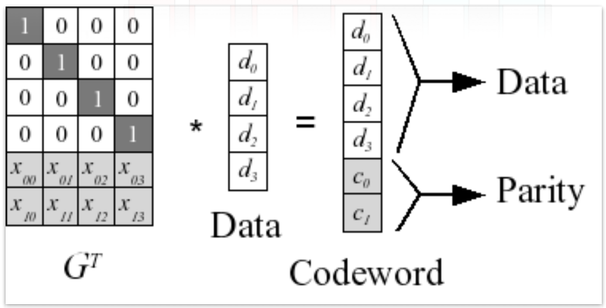




What is Erasure Encoding?

Reed & Solomon





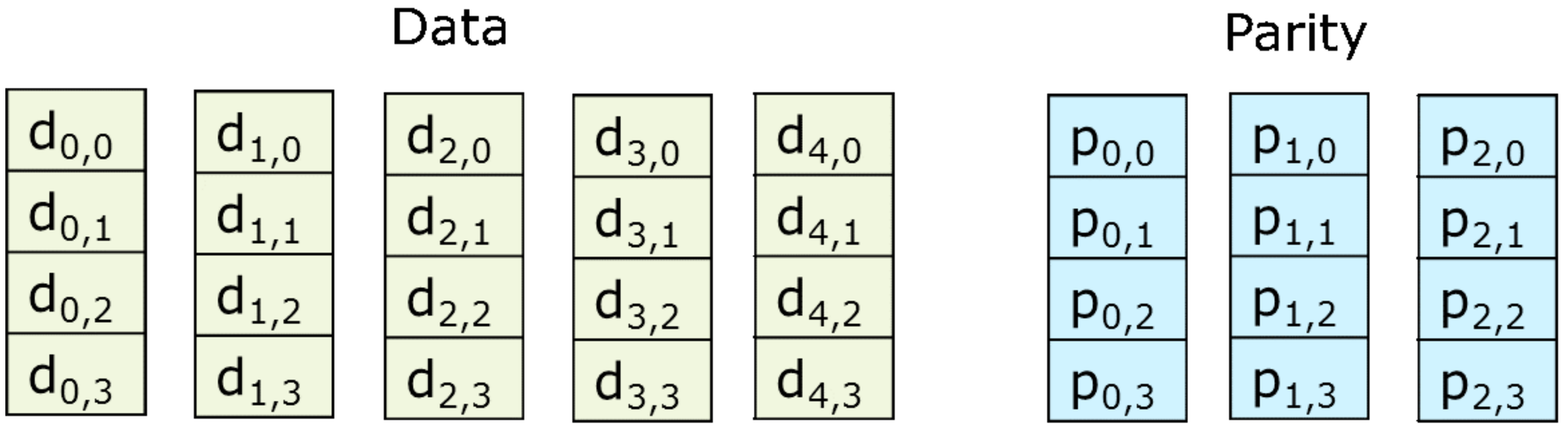
- **Erasure Coding** (EC) is a method of data protection in which data is broken into **fragments**, expanded and **encoded** with redundant data pieces and stored across a set of different locations or storage media
  - example of a widely used algorithm
    Reed Solomon Forward Error Correction **RS FEC**
  - Xeon CPU >5 GB/s/core (AVX)
  - open source libraries: **JERASURE** + Intel **ISA-L**

Data loss probability $p = e^{-\lambda}\frac{\lambda^k}{k!}$

where $\lambda = \frac{AFR \times (Number\ of\ Disks)}{365 \times 24 \div MTTR}$

MTTR = Mean Time To Repair
AFR = Annualized Failure Rate

Blockbased Erasure Encoding



Key Parameters: EC algorithm, chunk/block size,
number of data blocks (k), number of parity blocks (m)

Algo(m+k,k) e.g.

RS(14,10) [Reed Solomon with 10 data and 4 parity blocks]

Why EC now?

- **EC** was **incompatible** with CERN/Wigner **CC split** - breaks data locality concept
  - but Wigner almost decommissioned now (90 PB)
- **EC offers**
  - **resilience** and high **availability**
  - more **IOPS** and **bandwidth** per file
  - high **saving potential** for CERN IT - several million of CHF

EOS native EC implementation

Example

RS(5,3) [Reed Solomon with 3 data and 2 parity blocks]

0	1	2	0.0	0.1
3	4	5	3.0	3.1
6	7	8	6.0	6.1
9	10	11	9.0	9.1

0 — data blocks block offset in MB

0.0 — parity blocks block offset in MB / parity block

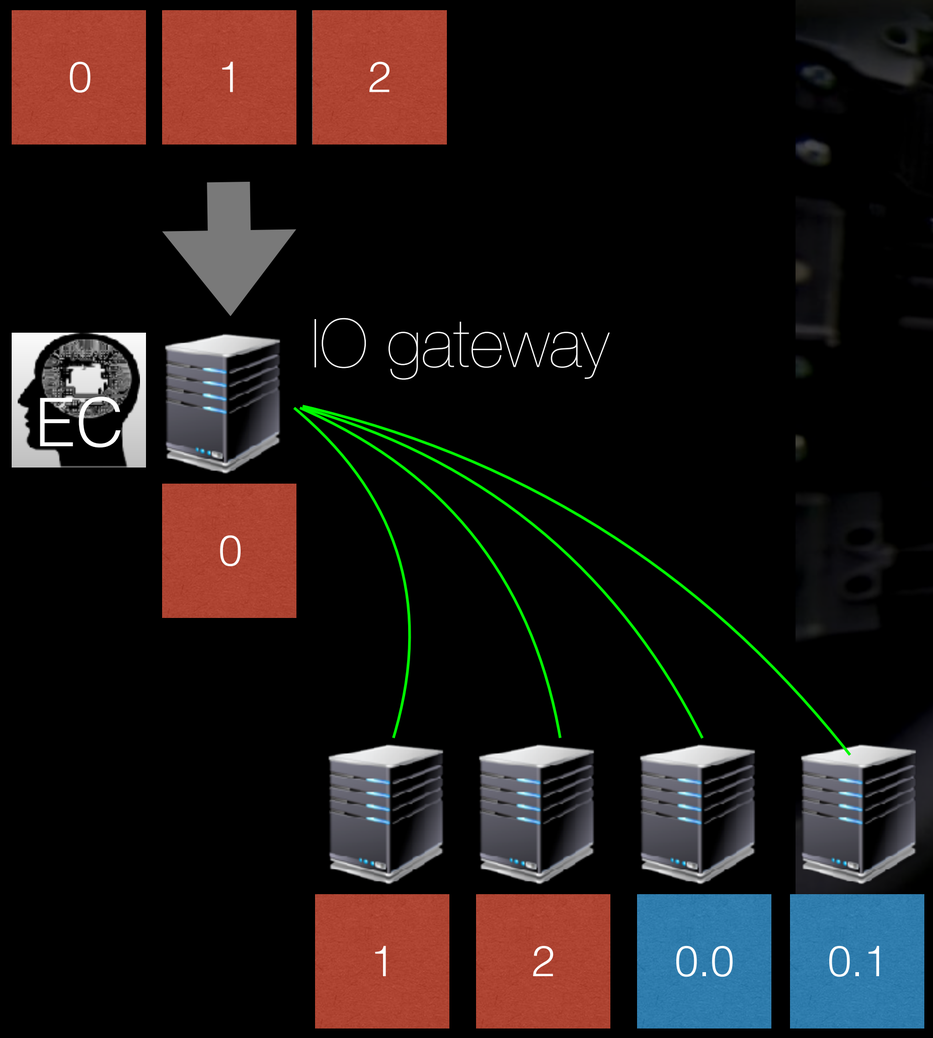
- each file stored with block and file checksums

- each file is placed on 5 nodes within one placement group
- IO bandwidth & IOPS equivalent to max. speed of 3 disks - non-degraded
- IO couples to 3-5 nodes
- IO is forward sequential on each disk

- each data & parity block is made out of 4k slices each with a CRC32C checksum
- each slice is checksum is checked on read

EOS EC IO Path

Write



Read XrdCl

Read eoscp XrdCl plug-in

direct IO

Network traffic amplification for EOS EC

amplification factor	RS(2,1)	RS(3,2)	RS(6,4)	RS(9,6)	RS(18,14)
Write gateway	2	2.33	2.25	2.33	2.12
Read gateway	1	1.66	1.75	1.83	1.92
Read direct	1	1	1	1	1
Drain*	**x1**	**x2**	**x4**	**x6**	**x14**

* 340 PB - 100 % EC - 60k harddisks - 2% failure rate

19 TB/day to be drained for RS(10,8) - **220 MB/s repair streams - dT<24h**

O2 TEST BENCHMARKS

from Olga Chuchuk

Single Machine Tests

st-096-dd9061d2.cern.ch

latest hardware

2 Xeon Gold 5218 2.3GHz

192 GB DDR4@ 2.6 GHz

2x Intel SSDC2KB01 **2T**

96x HGST HUH721212AL **12 TB**

2 x 25 Gbit (one connected)

2 x 10 Gbit

Various test setups

96 FS XFS

96 FS ZFS

EOS EC (m,k)

1 DataRS(16,2) CephFS

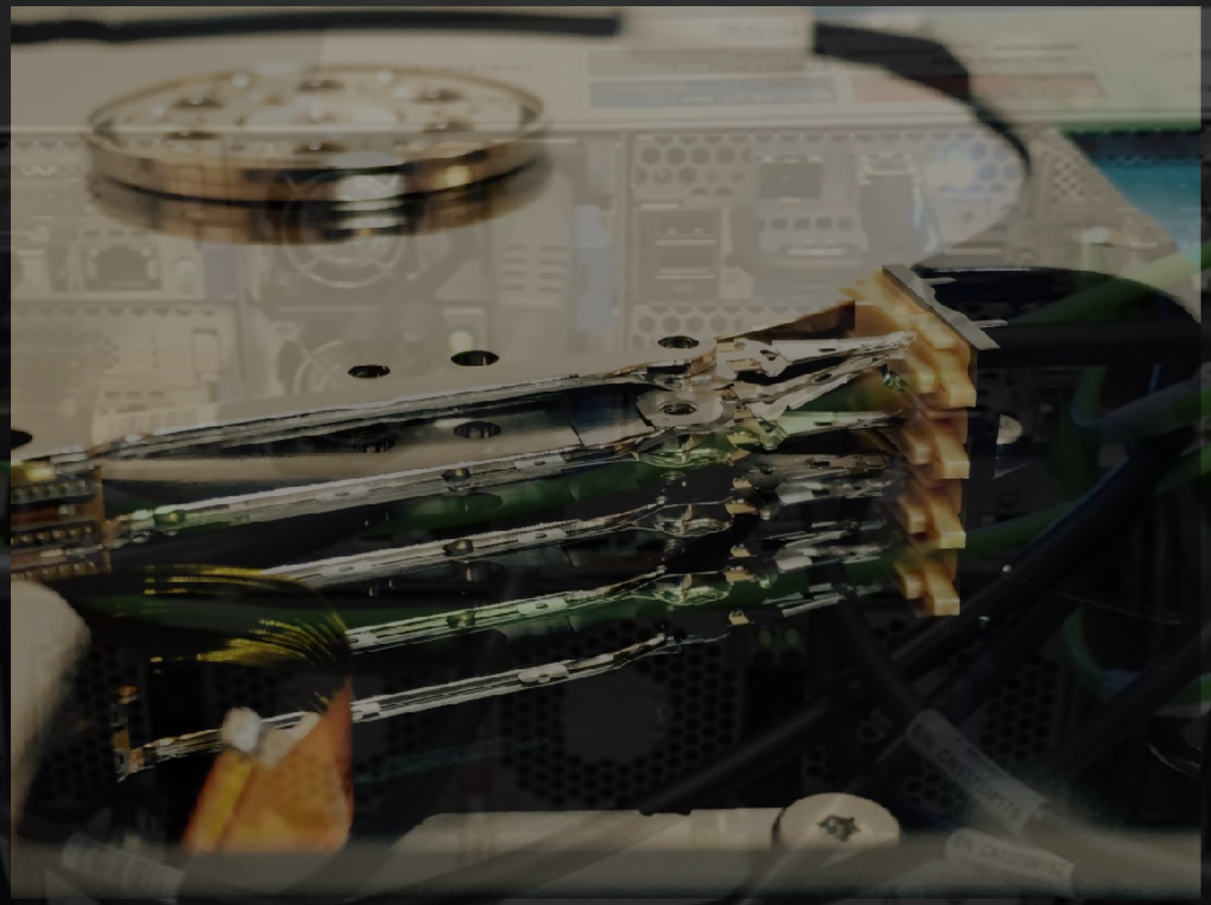
Perfect & simple playground to validate EC models, investigate fluctuations, performance figures!

Raw Hardware

Single Machine Tests
st-096-dd9061d2.cern.ch

SSD
430 MB/s seq write
400 MB/s seq read
IOPS 33k

useful for ZFS, CephFS, not for XFS



HDD
250 MB/s seq write
250 MB/s seq read
IOPS 80

Raw Hardware

Single Machine Tests
st-096-dd9061d2.cern.ch

all disks one stream
1 dev seq read 6.3 GB/s
2 **xfs** seq read 6.3 GB/s
2 xfs seq write 6.4 GB/s

3 **zfs** jbod performance
degrading to 4.0 GB/s
when disks fill up
(zfs not a good choice)

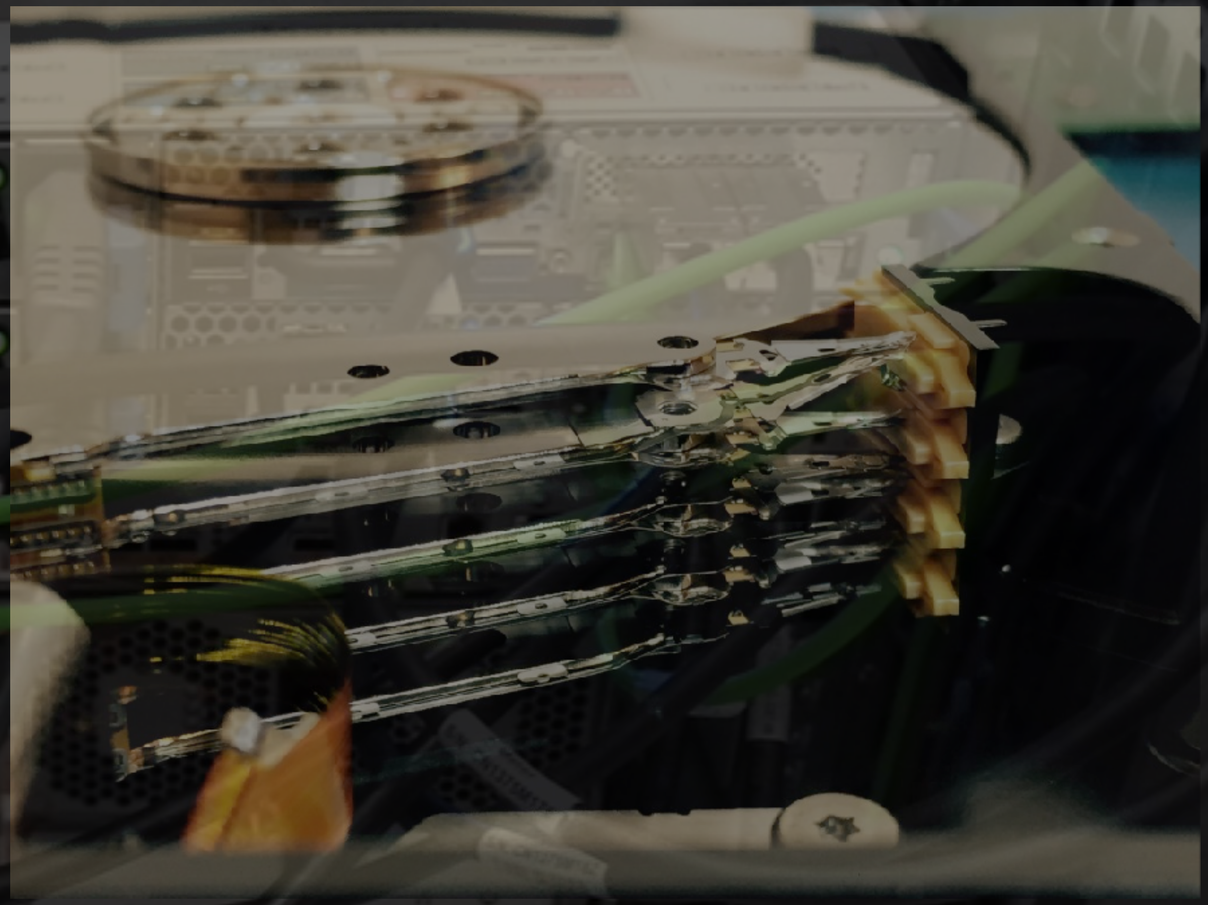
4 **cephfs** performance
bluefs looks really god,
however SSD size and
memory are very close to/over
required limit for 96 disks
saturate disk read/write with
3 streams on single fs
kernel mount on OSDs dead-lock risk

96 file systems

96 file systems

1 filesystem

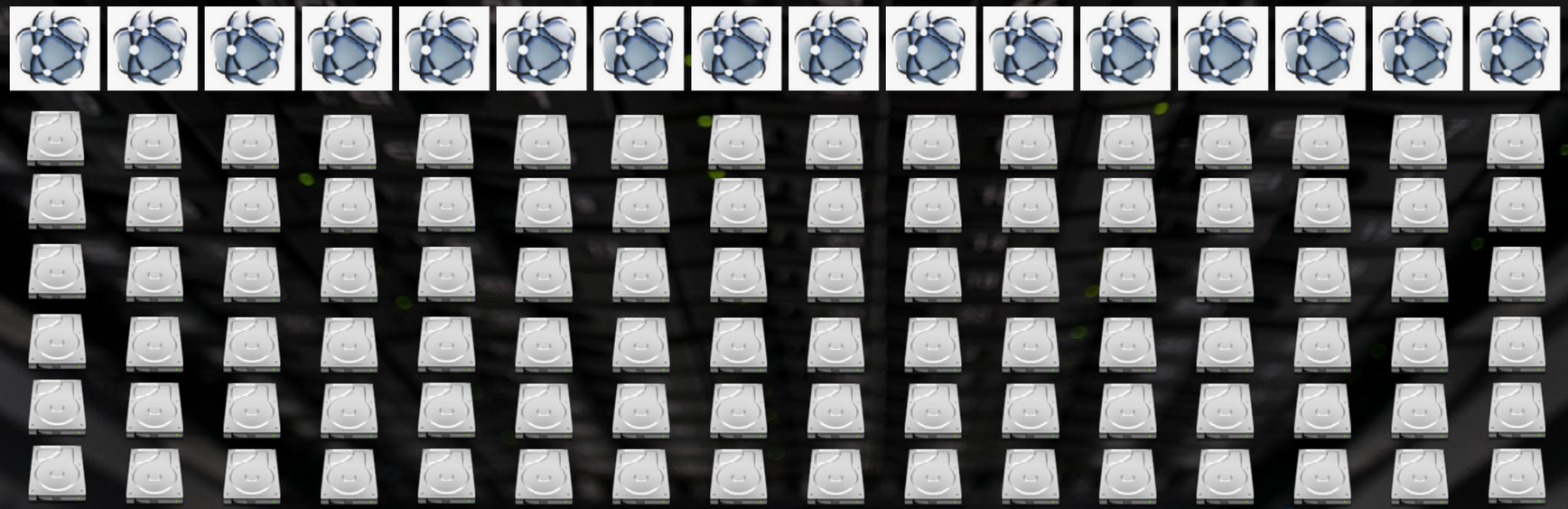
read amplification via gw: 3
write amplification via gw: 3



Single Box EOS/EC

RS(12,10)
RS(13,10)
RS(14,10

16 xrootd FST processes

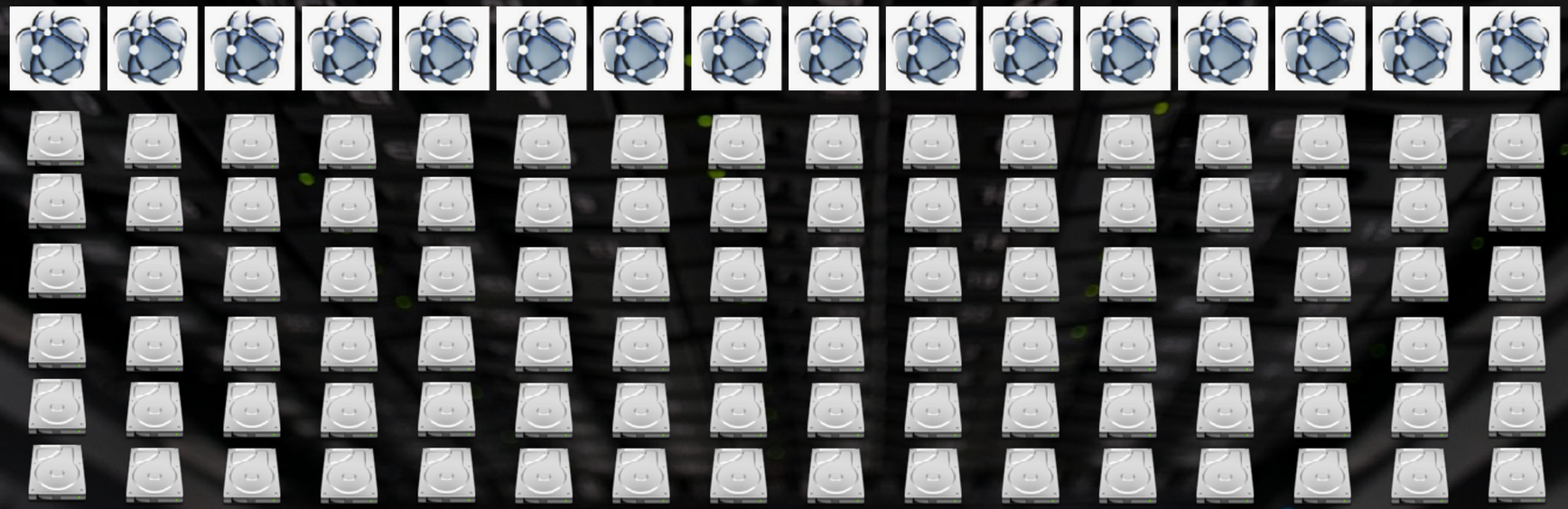


group.0
group.1
group.2
group.3
group.4
group.5

read 12 streams/2G files GW mode - 6.0 GB/s (non degraded) - 78% CPU idle
read 12 streams/2G files direct mode - 5.8 GB/s (non degraded) - 80% CPU idle
read 12 streams/2G files GW mode - 2.8 GB/s (degraded) - 80% CPU idle
read 12 streams/2G files direct mode - 3.2 GB/s (degraded) - 80% CPU idle

Single Box EOS/EC

16 xrootd FST processes



group.0
group.1
group.2
group.3
group.4
group.5

single stream write 500 MB/s (+2,+3,+4)
single stream read direct 1 GB/s - degraded 350 MB/s
single stream read GW 750 MB/s - degraded 350 MB/s

EOSALICEDAQ o2test space

created perfectly balanced space with 12 disks per group

12 xrootd FSTs

o2test.1

o2test..24

single 25GBit client box - 48 stream upload @ 50MB/s (no fluctuation)

single 25GBit client box - 48 stream download @ 50MB/s (no fluctuation)

Configuration/Usage hints

- To get optimal performance, you have to create as many groups as possible for the target layout
  - groups are scheduled round-robin (equal load)
  - within a group the placement is randomised e.g. number of streams per disk is fluctuating with sqrt(N)
- XRootD async IO is ‘bad’ for fair-share streaming
  - lower fluctuation for direct IO mode
  - slowest fluctuations when using a bandwidth limit per client so instances do not saturate
    - single box test (cached files):
      - eoscp rate = 925 +- 20 MB/s
      - eoscp -0 rate = 685 += 35 MB/s
      - eoscp -0 -t 650 rate = 650 +- 0.02 MB/s
    - single box test (uncached files):
      - eoscp rate = 540 +300 - 50 MB/s
      - eoscp -t 500 rate = 490 +- 10

If you want to avoid fluctuations, you should use synchronous IO limiting the maximum single stream performance (if not bottlenecked already by writer)

O2 Scenario

1600 GPU @ avg. 62 MB/s each

O2 100 GB/s writing

50 disk server @ 12 TB @ 96 Disks <= 5 streams/disk

< 30 MB/s per disk

need to add expected drain rates

O2 Scenario

3200 reader @ 62 MB/s each

O2 200 GB/s reading

50 disk server @ 12 TB @ 96 Disks <= 10 streams/disk

< 60 MB/s per disk

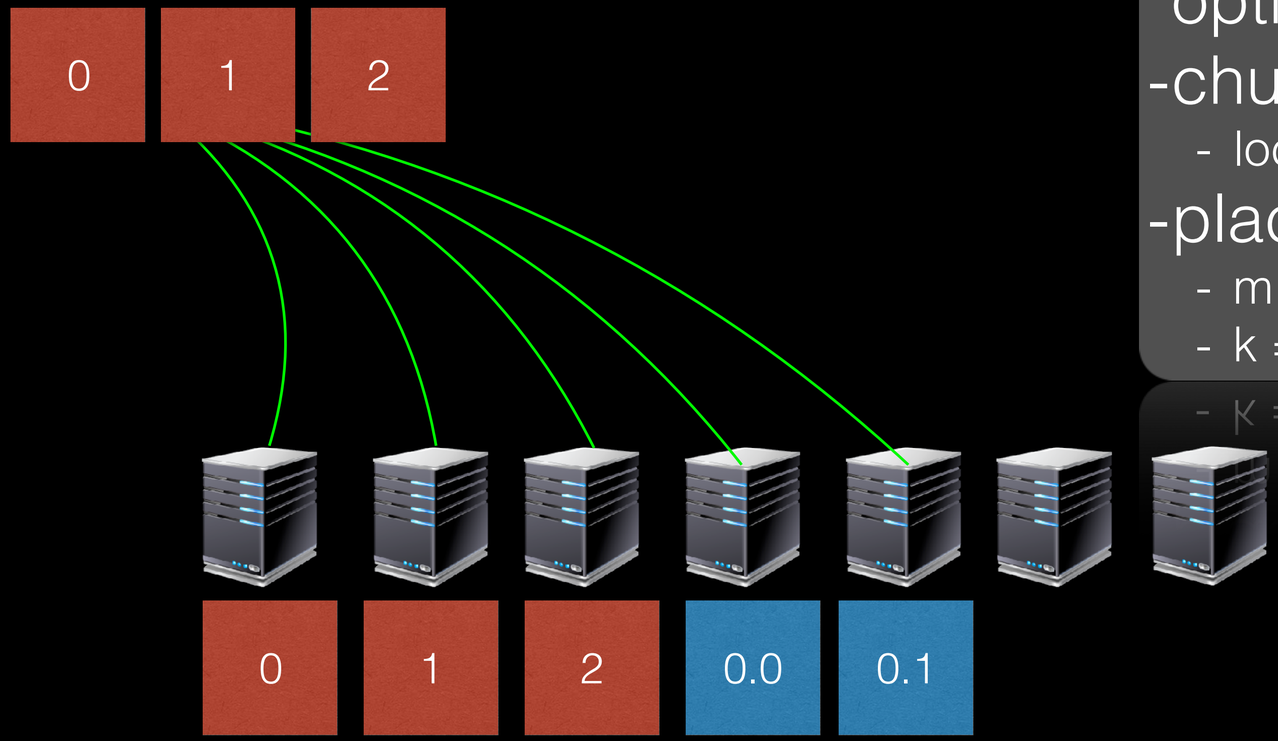
need to add expected drain rates

Outlook

- XRootD TPC does not support bandwidth limitation
  - consider adding the BW regulation server side(?)
  - consider adding the BW regulation client side(?)
  - there are fluctuations, but the lowest speed is good enough (?)
- assuming 50 nodes: O2 hardware needs ~**35** GBit/s per storage node to deliver 100GB/s write and 200 GB/s read for RS(12,10
- don’t count on **10 GB/s** IO per node - to be measured with hardware

Evolution EOS **D**irect**O**bject**S**torage
EC Implementation

Write



- client-side EC using XRootD plug-in
  optimised for seq. IO & simplicity
- chunk placement is random
  - locations are not hashed
- placement groups are sized as n=m+k+k
  - m = data stripes
  - k = parity stripes

Direct EC Tests

Michal K. Simon

XrdCl plug-in talking to 12 native XRootD Server on EOSALICEDAQ

- ISA-L EC throughput on the client machine: **~3GB/s**
- CRC32C throughput on the client machine: **~21GB/s**
- Single disk throughput: ~200MB/s
- NIC: 25Gbit/s

- Aggregated write throughput per input stream (2GB file):
  - Async writes of 40MB blocks (10x 4MB data chunks + 2x 4MB parity chunks)
  - open + writes + close = ~1.8 GB/s
  - open + writes + sync + close = ~750 MB/s

Thanks for the attention!
Questions?