TECH WEEK
STORAGE 24

EOS Open Storage

SMR disks in EOS

Dr. **Andreas-Joachim Peters**

for the EOS Project - CERN IT - Storage Group

IT Auditoirium - CERN
15.03.2024

Contents

- Why SMR and what is it?
- Integration into EOS
- Benchmarks & Problems
- Summary & Outlook

SMR
SHINGLED
MAGNETIC
RECORDING

Why do we care about SMR technology?

- **areal density** (1.3 terabits per square inch)
- expectation: **price less $/TB** … to be seen
- also in new **HAMR** technology SMR can be used to increase the areal density

Why do we care about SMR technology?

- **areal density** (1.3 terabits per square inch)
- expectation: **price less $/TB** … to be seen
- also in new **HAMR** technology SMR can be used to increase the areal density

Example WD HC 670 SMR Drive **+18%** capacity compared to HC 570 CMR Drive

Why do we care about SMR technology?

- **areal density** (1.3 terabits per square inch)
- expectation: **price less $/TB** … to be seen
- also in new **HAMR** technology SMR can be used to increase the areal density

Example WD HC 670 SMR Drive **+18%** capacity compared to HC 570 CMR Drive

Why do we care about SMR technology?

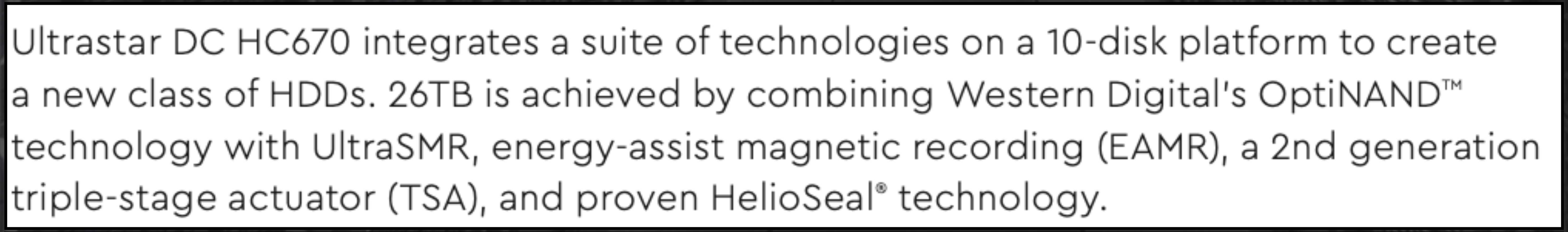
- **areal density** (1.3 terabits per square inch)
- expectation: **price less $/TB** … to be seen
- also in new **HAMR** technology SMR can be used to increase the areal density

Example WD HC 670 SMR Drive **+18%** capacity compared to HC 570 CMR Drive

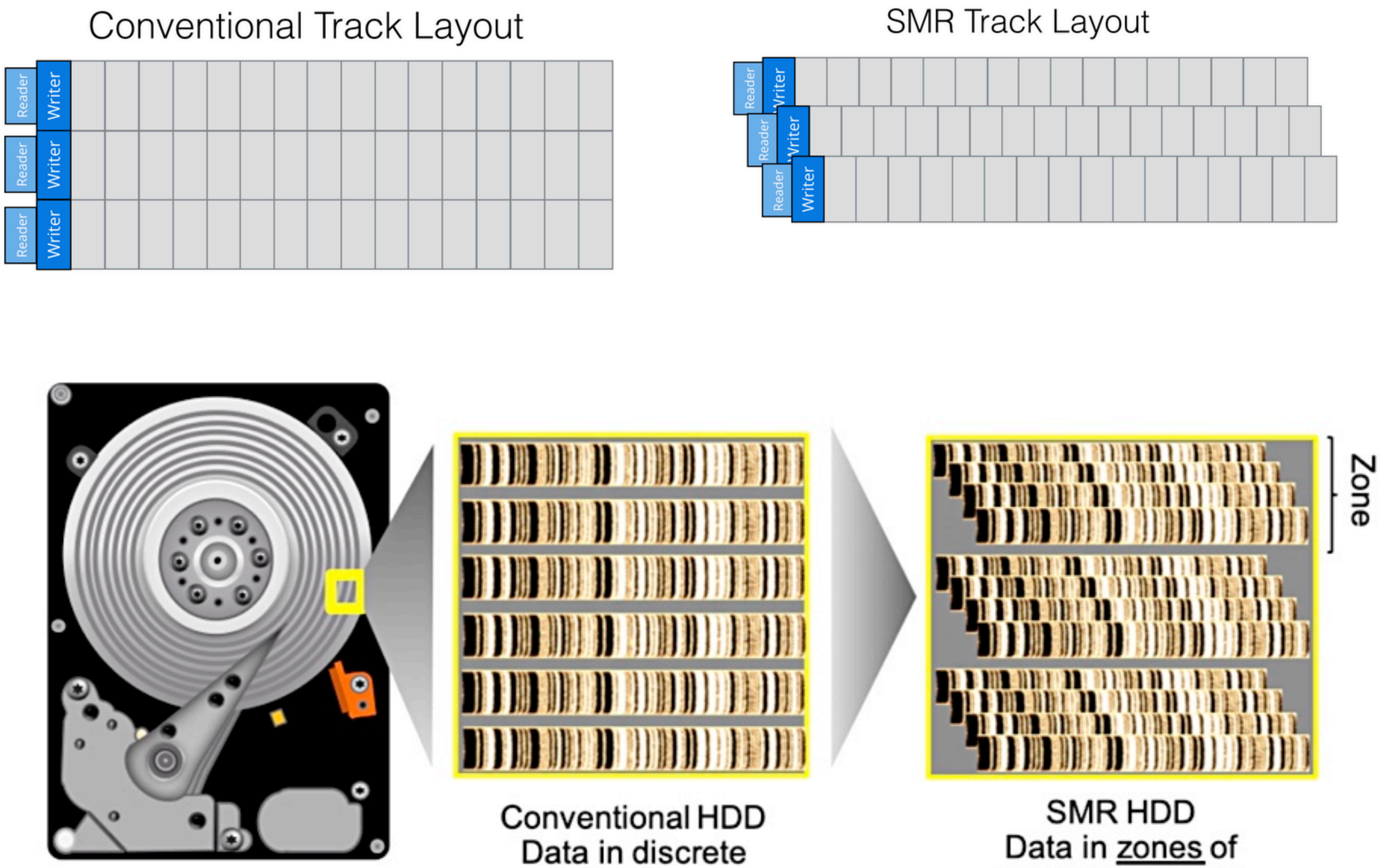
Why do we care about SMR technology?

- **areal density** (1.3 terabits per square inch)
- expectation: **price less $/TB** … to be seen
- also in new **HAMR** technology SMR can be used to increase the areal density

Example WD HC 670 SMR Drive **+18%** capacity compared to HC 570 CMR Drive


Ultrastar DC HC670 integrates a suite of technologies on a 10-disk platform to create a new class of HDDs. 26TB is achieved by combining Western Digital's OptiNAND™ technology with UltraSMR, energy-assist magnetic recording (EAMR), a 2nd generation triple-stage actuator (TSA), and proven HelioSeal® technology.

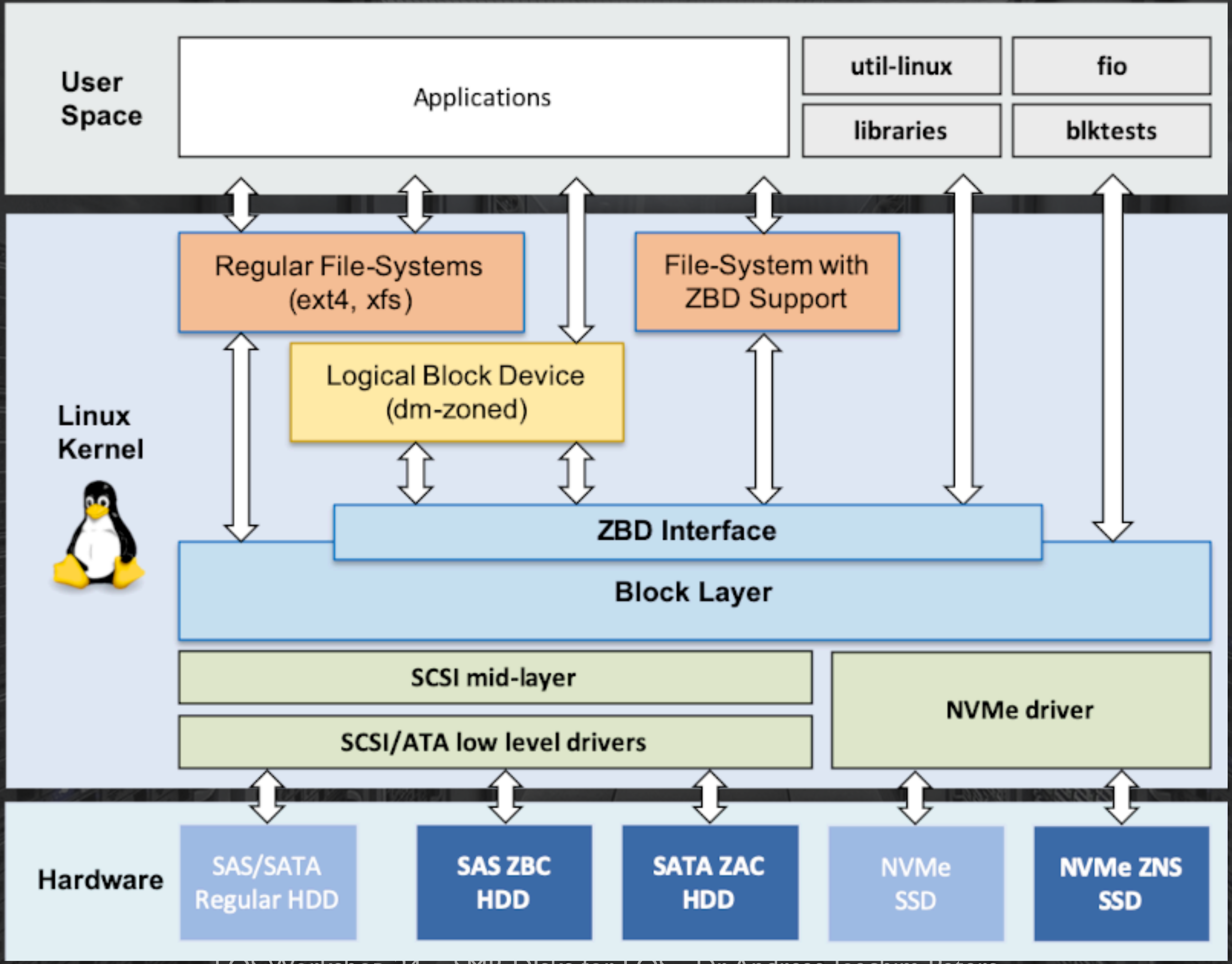
Conventional CMR vs Shingled SMR drive



Conventional HDD
Data in discrete tracks

SMR HDD
Data in zones of overlapped tracks

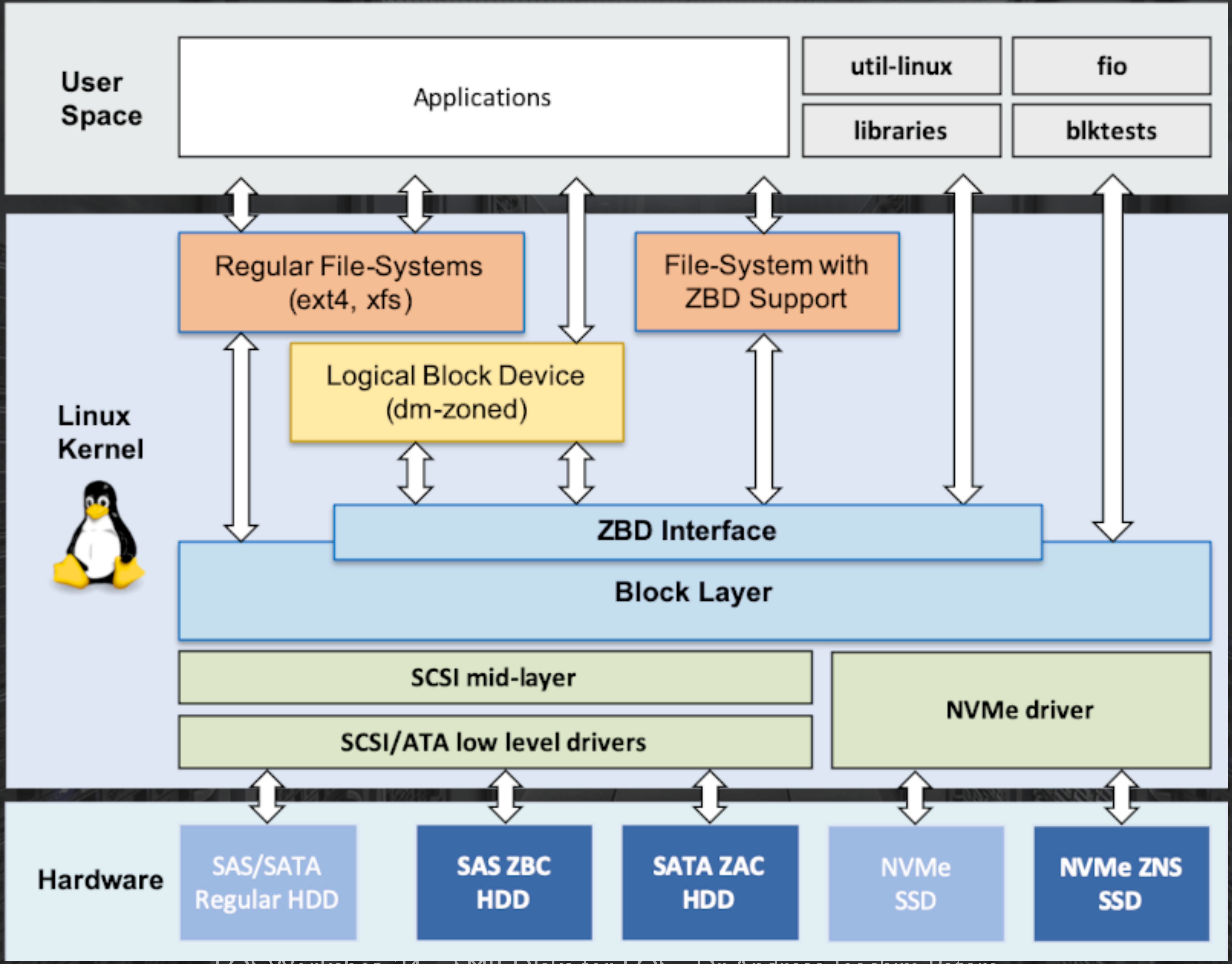
Using SMR in Linux



Using SMR in Linux

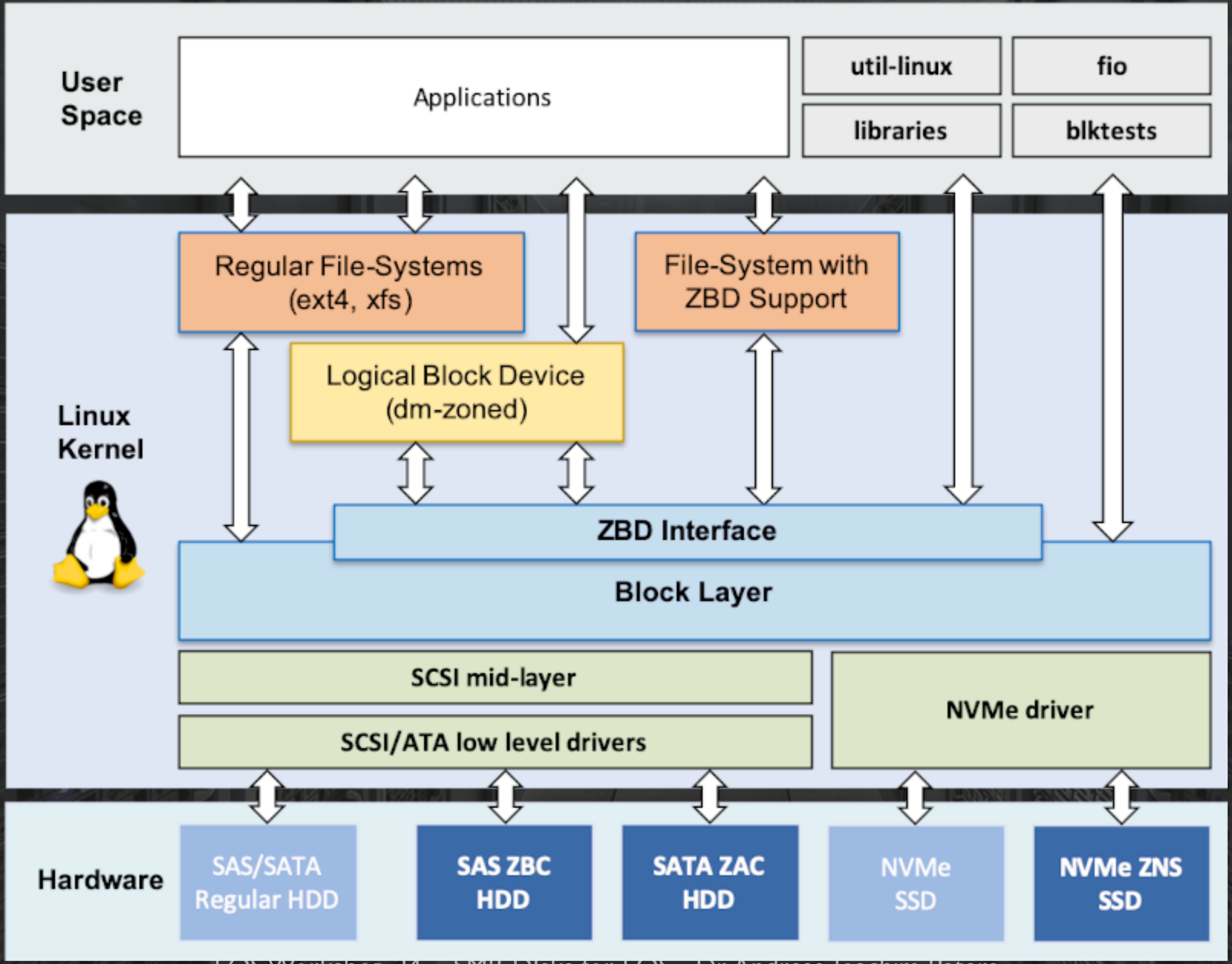
Option 1

conventional filesystem on top of dm-zoned logical block device



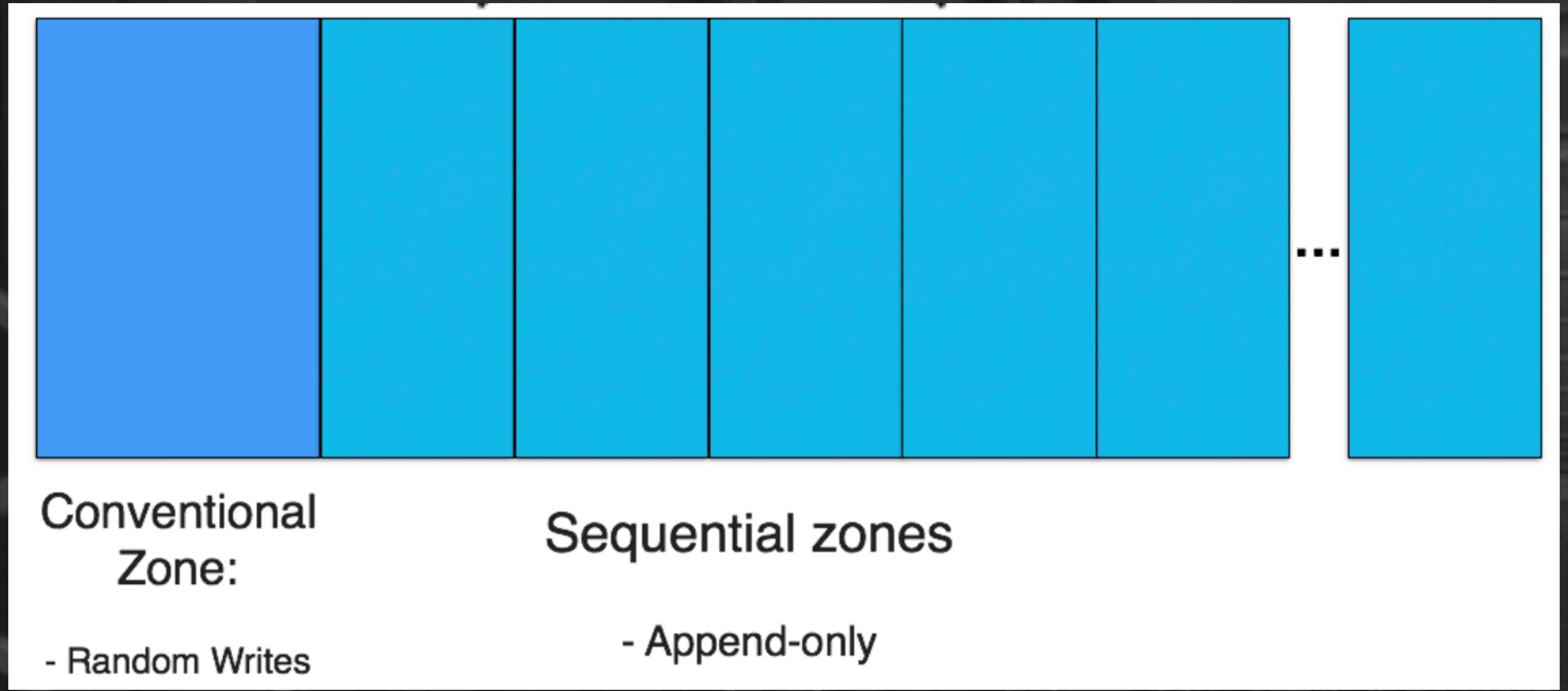
Using SMR in Linux

Option 1
conventional filesystem on top of dm-zoned logical block device



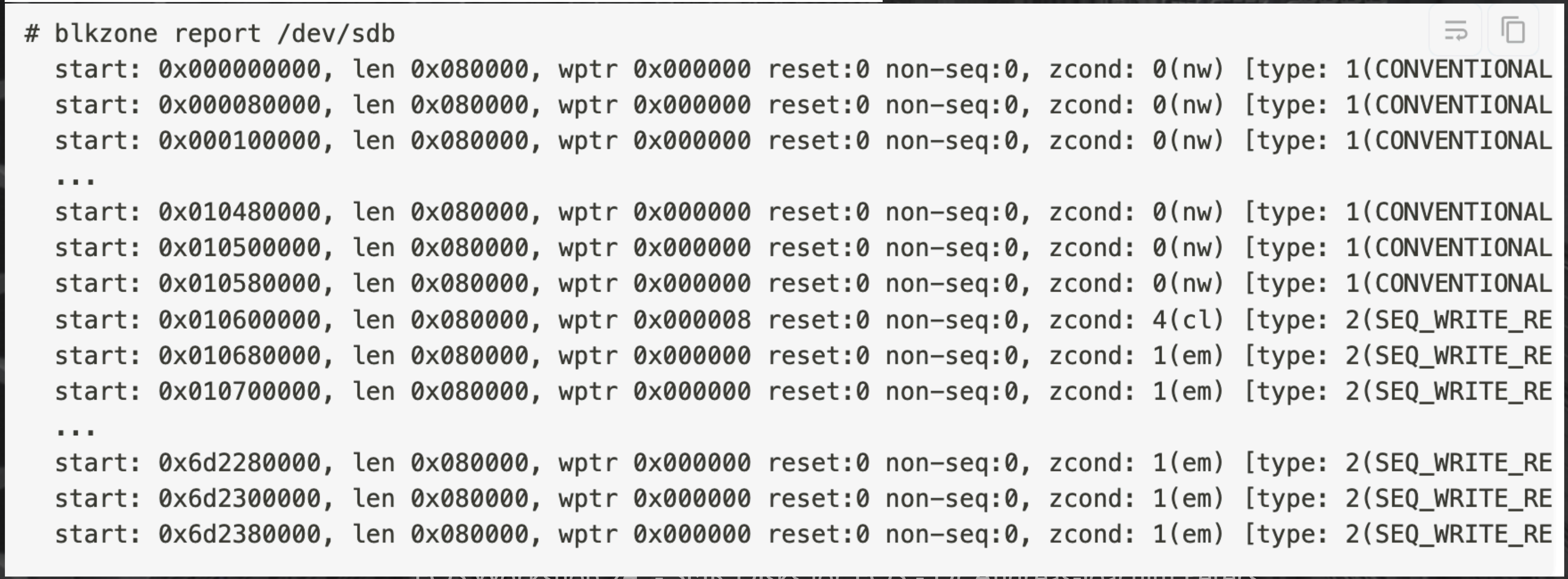
Option 2
modern filesystem with zoned block device support

SMR Disk Structure



Filesystem can profit from a conventional zone for superblocks, random writes, or meta-data.

Size depends on the vendor (0.1-1%)



```
# blkzone report /dev/sdb
  start: 0x000000000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 0(nw) [type: 1(CONVENTIONAL
  start: 0x000080000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 0(nw) [type: 1(CONVENTIONAL
  start: 0x000100000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 0(nw) [type: 1(CONVENTIONAL
  ...
  start: 0x010480000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 0(nw) [type: 1(CONVENTIONAL
  start: 0x010500000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 0(nw) [type: 1(CONVENTIONAL
  start: 0x010580000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 0(nw) [type: 1(CONVENTIONAL
  start: 0x010600000, len 0x080000, wptr 0x000008 reset:0 non-seq:0, zcond: 4(cl) [type: 2(SEQ_WRITE_RE
  start: 0x010680000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 1(em) [type: 2(SEQ_WRITE_RE
  start: 0x010700000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 1(em) [type: 2(SEQ_WRITE_RE
  ...
  start: 0x6d2280000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 1(em) [type: 2(SEQ_WRITE_RE
  start: 0x6d2300000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 1(em) [type: 2(SEQ_WRITE_RE
  start: 0x6d2380000, len 0x080000, wptr 0x000000 reset:0 non-seq:0, zcond: 1(em) [type: 2(SEQ_WRITE_RE
```

SMR Evalution

- we had two SMR drives from two vendors for evaluation and compared these to equivalent CMR drives from the same vendor

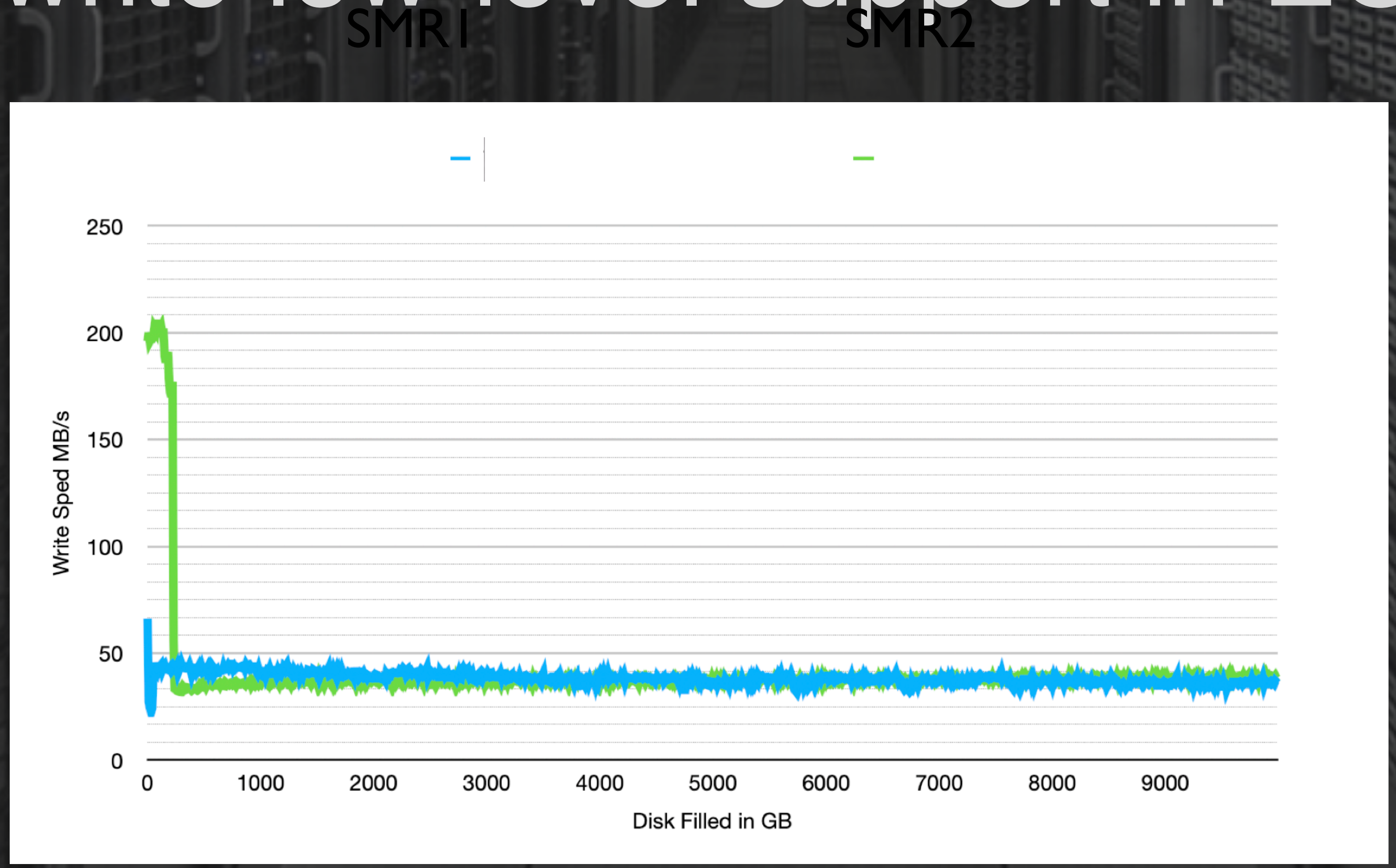
- initially drives didn’t work at all due to an unsupported SAS expander

SMR Option 1 - dm-zoned

- we tried using XFS with dm-zoned

- miserable sequential write performance around 40 MB/s on both SMR drives
  - but good sequential read performance > 200 MB/s

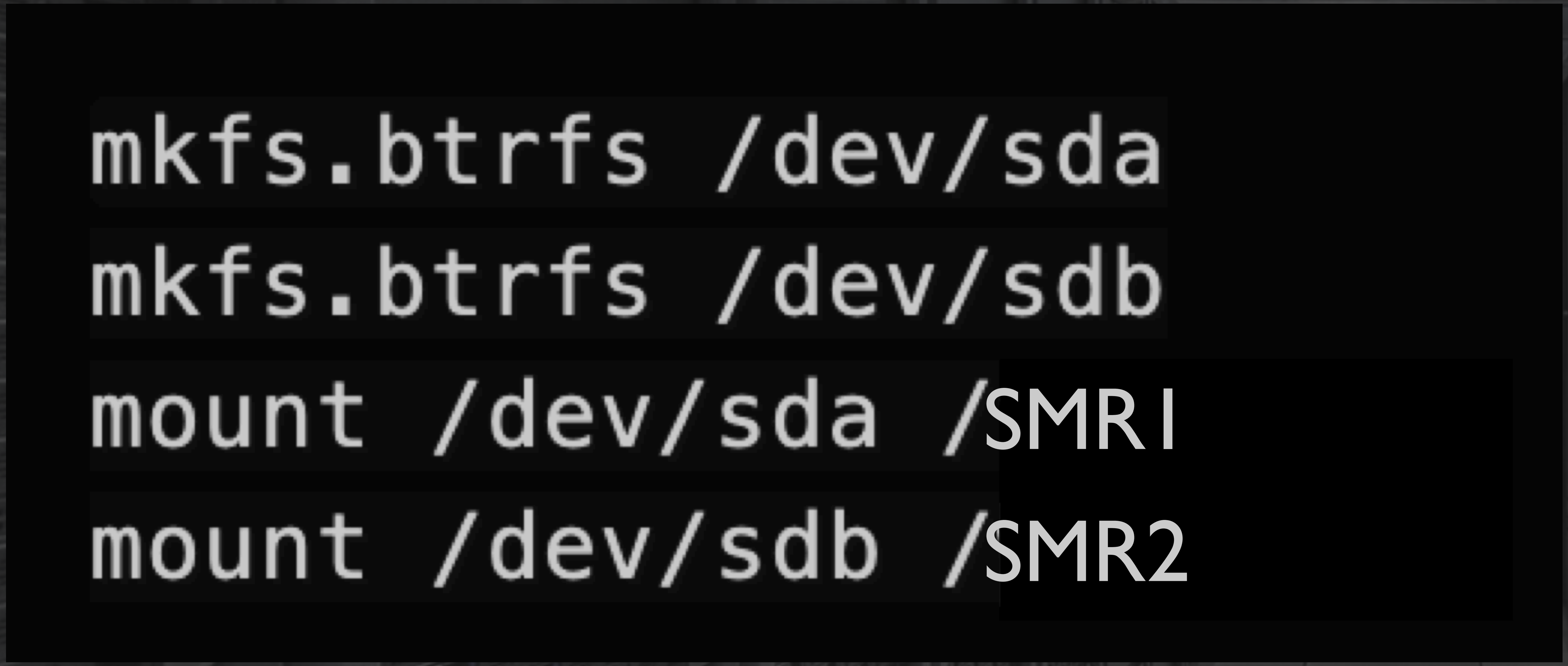
- after iteration with vendor BTRFS was recommended as a better approach - we don’t want to write low-level support in EOS



The green drive has a larger conventional zone, which provides good write performance, while the shingled zone suffers from XFS implementation

SMR Option 2 - BTRFS

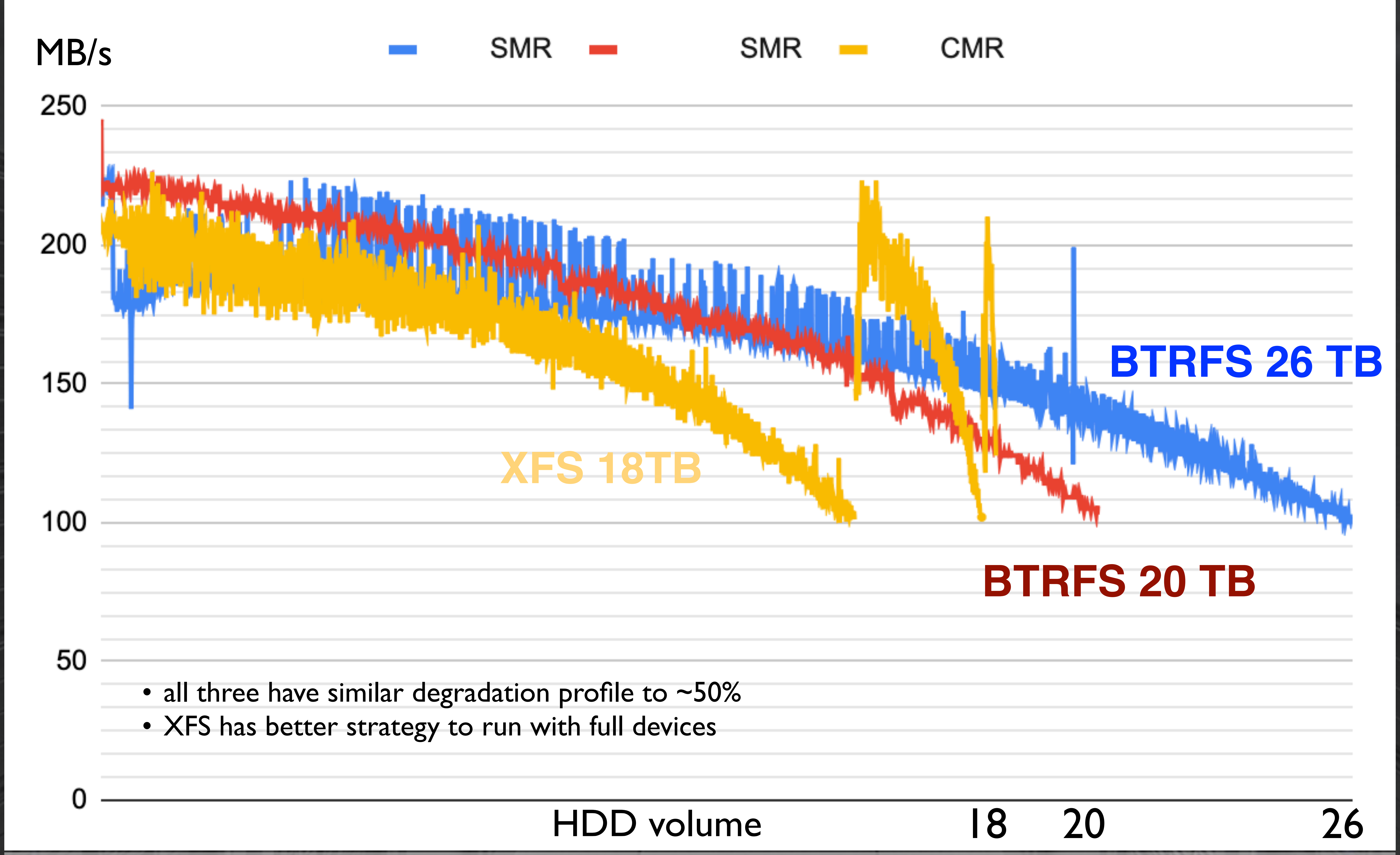
Very simple setup



```
mkfs.btrfs /dev/sda
mkfs.btrfs /dev/sdb
mount /dev/sda /SMR1
mount /dev/sdb /SMR2
```

Performance:utilization

“Measure write performance as function of disk fullness ”

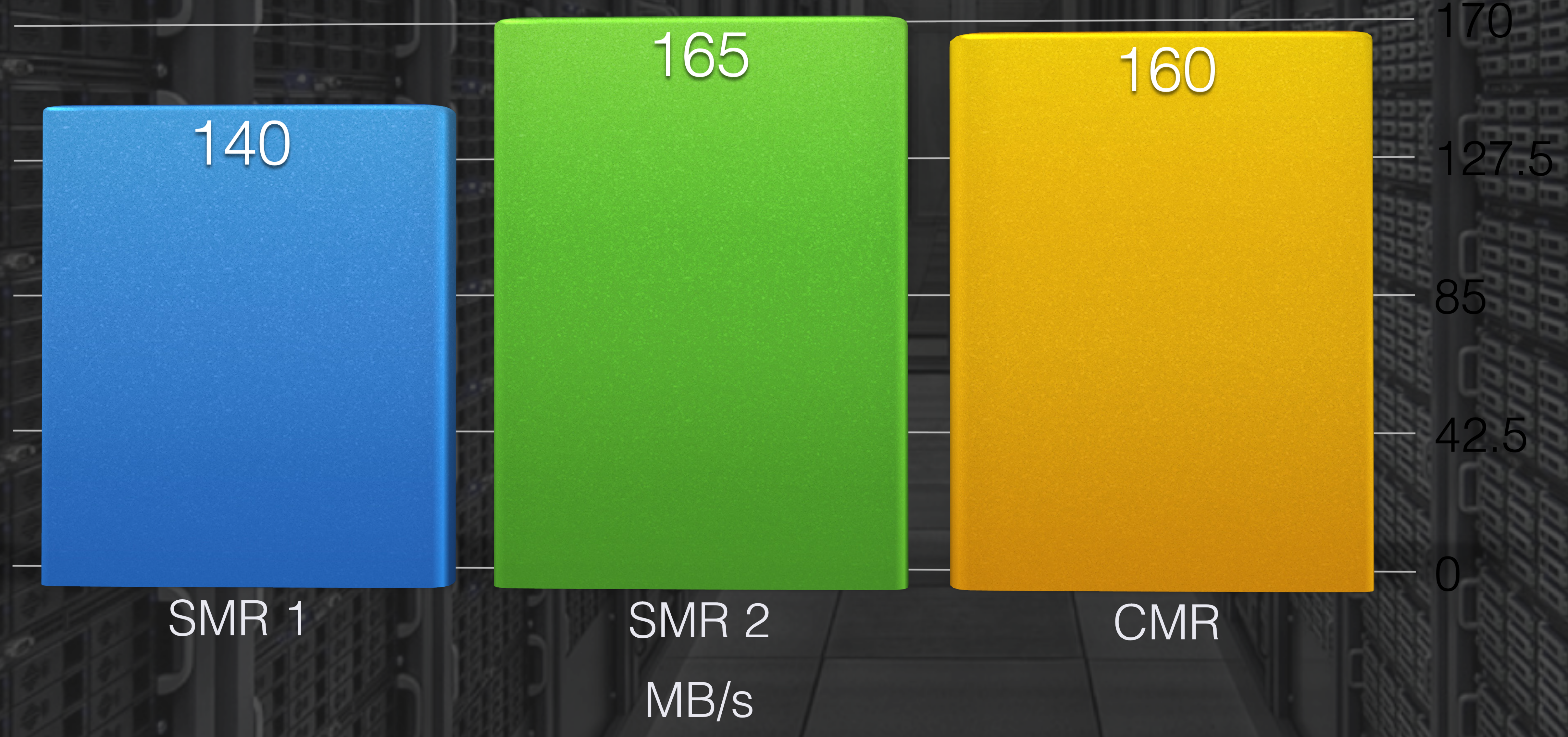


- all three have similar degradation profile to ~50%
- XFS has better strategy to run with full devices

Random Read Performance

"Measure large random read performance on full disks "

- 10 parallel readers reading randomly 100 MB blocks from full disks



EOS

EOS

100%

100%

00%

[redacted]

[redacted]

EOS with SMR & BTRFS

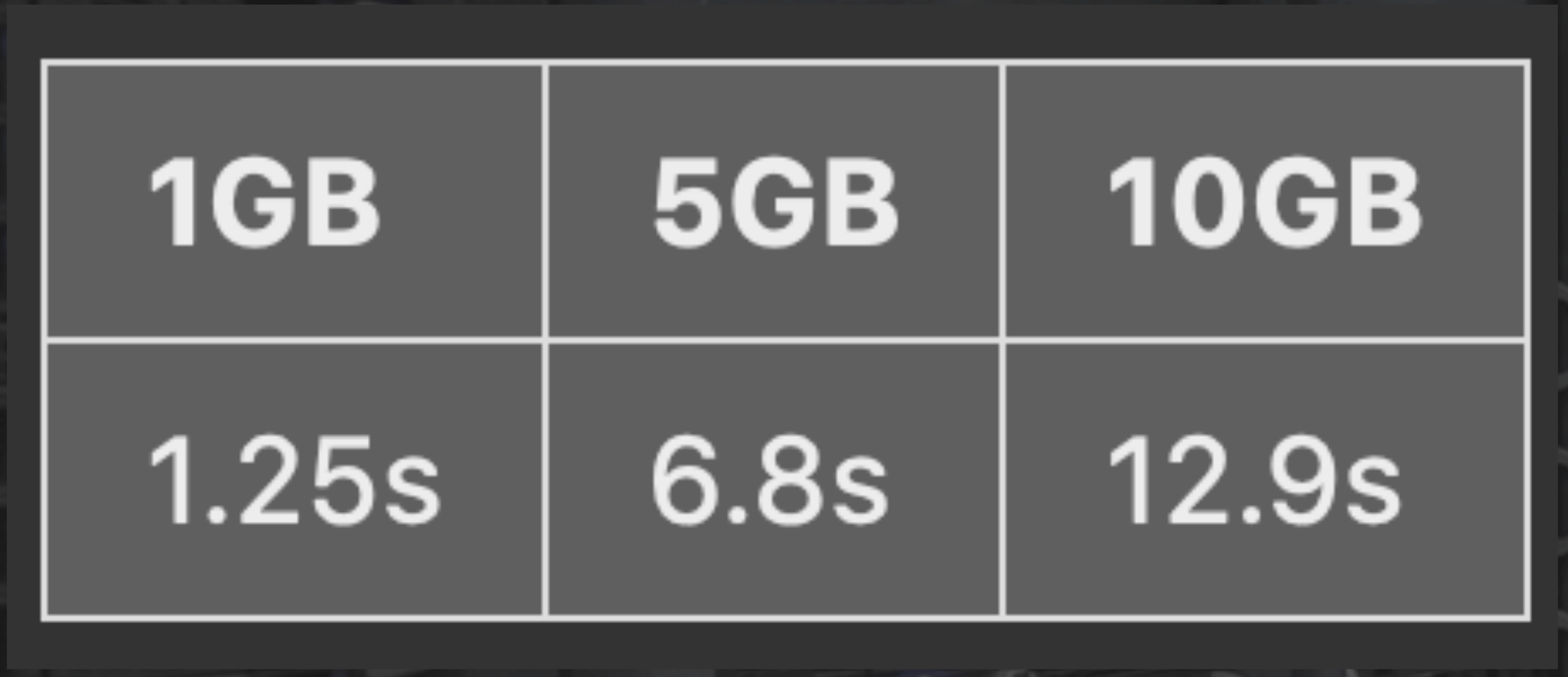
- BTRFS with SMR support is **not part of RedHat distributions**
- initially we used Fedora Core 38, later Alma9 with self-compiled kernel
  - as a side product we have now FC38 packages for server deployment
  - BTRFS tools had to be compiled manually for ALMA9 - no package available
- finally we managed to have a single box EOS setup with two SMR disks
  - simple small file creation tests (xrdstress) had equivalent rates on CMR and SMR disks
  - large file uploads (eoscp) were very slow (?)

Dos and Don’ts

- EOS uses for XFS a fast pre-allocation function for the size of a complete file upload - for non XFS filesystems it uses posix_fallocate

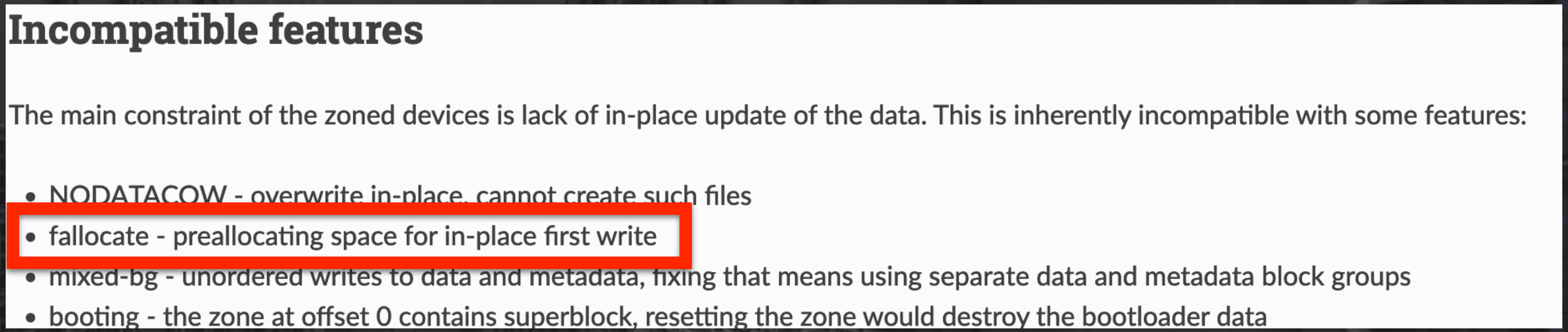
Dos and Don’ts

- EOS uses for XFS a fast pre-allocation function for the size of a complete file upload - for non XFS filesystems it uses posix_fallocate



1GB	5GB	10GB
1.25s	6.8s	12.9s

posix_fallocate time on BTRFS/SMR vs. size : ~linear



Incompatible features

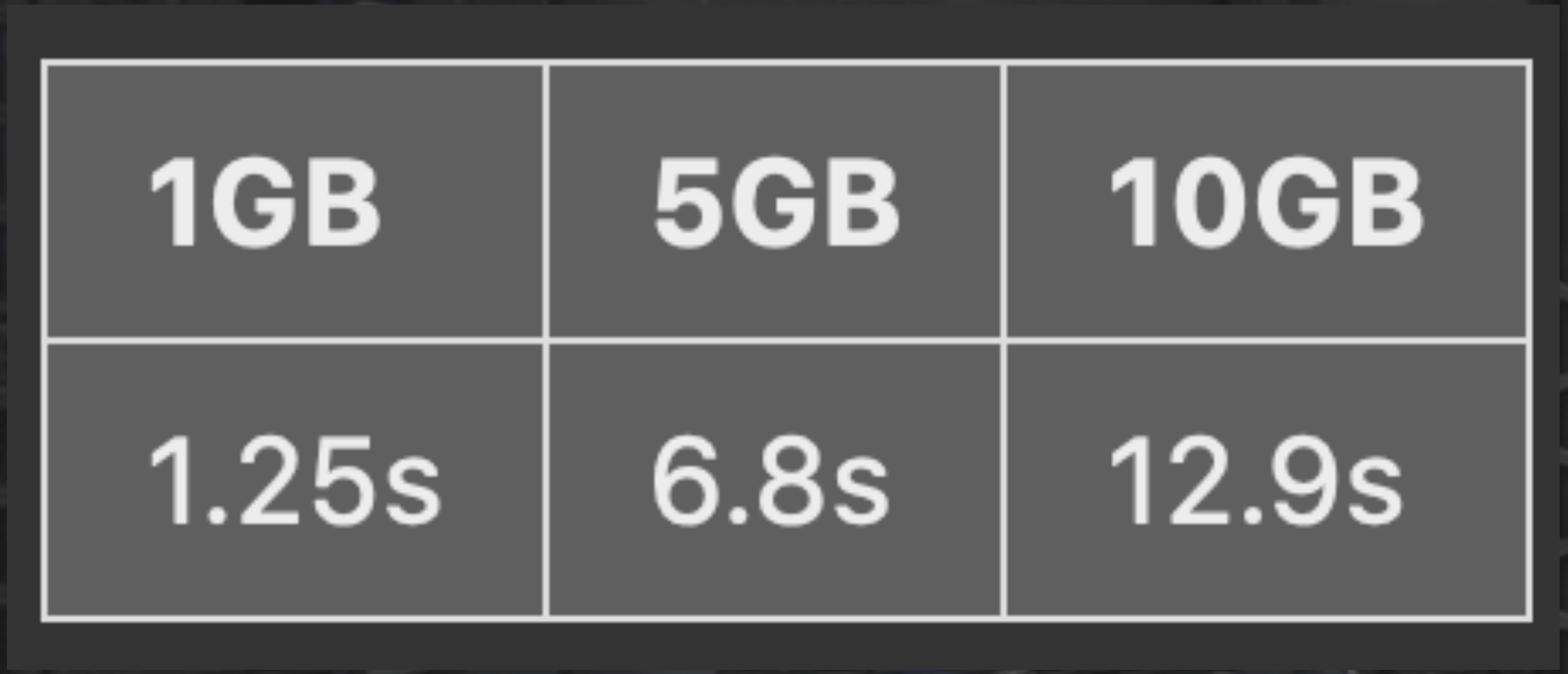
The main constraint of the zoned devices is lack of in-place update of the data. This is inherently incompatible with some features:

- NODATACOW - overwrite in-place, cannot create such files
- fallocate - preallocating space for in-place first write
- mixed-bg - unordered writes to data and metadata, fixing that means using separate data and metadata block groups
- booting - the zone at offset 0 contains superblock, resetting the zone would destroy the bootloader data

BTRFS Docs

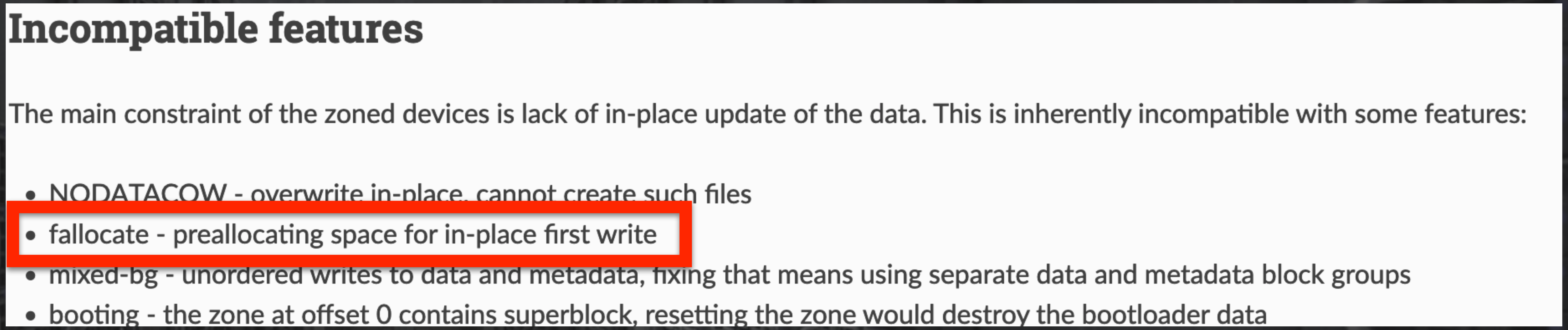
Dos and Don’ts

- EOS uses for XFS a fast pre-allocation function for the size of a complete file upload - for non XFS filesystems it uses posix_fallocate



1GB	5GB	10GB
1.25s	6.8s	12.9s

posix_fallocate time on BTRFS/SMR vs. size : ~linear



Incompatible features

The main constraint of the zoned devices is lack of in-place update of the data. This is inherently incompatible with some features:

- NODATACOW - overwrite in-place, cannot create such files
- fallocate - preallocating space for in-place first write
- mixed-bg - unordered writes to data and metadata, fixing that means using separate data and metadata block groups
- booting - the zone at offset 0 contains superblock, resetting the zone would destroy the bootloader data

BTRFS Docs

`Since EOS 5.2.2 no posix_fallocate by default anymore for ~~XFS~~`

Full Disk Scenarios

- when SMR disks are written up to 100% they are switched into read-only state - it was impossible to make the device writable again

- we observed that even if you deleted a significant amount of data during the filling, you can end up with no space left on device and read-only

- SMR disk behave **similar to a tape** when you delete data
  - it requires a similar operation like 'repack on tape' to rewrite half empty zones to recover the space [defragmentation operation]
  - the output of statvfs or df is misleading, you can see 50% free space, but the device is not writable
    - we need to use btrfs filesystem df /disk

- the space reclaim process did not work as intended in our tested kernel version 6.6.11 (ALMA9)

SMR Summary

- We managed to have a working integration to run SMR disk in EOS

- SMR disk performance with BTRFS is very similar to CMR disk for large file streaming workloads
  - would work very well in workloads like large erasure coded files used in O2
  - SMR spaces could be filled by policy from CMR spaces

- The full disk scenario / fragmentation is still of concern
  - will be hopefully fixed in newer kernel releases

- The missing support of BTRFS in RH flavored OS is of concern
  - running a tainted kernel might be problematic for production/security

SRM Outlook

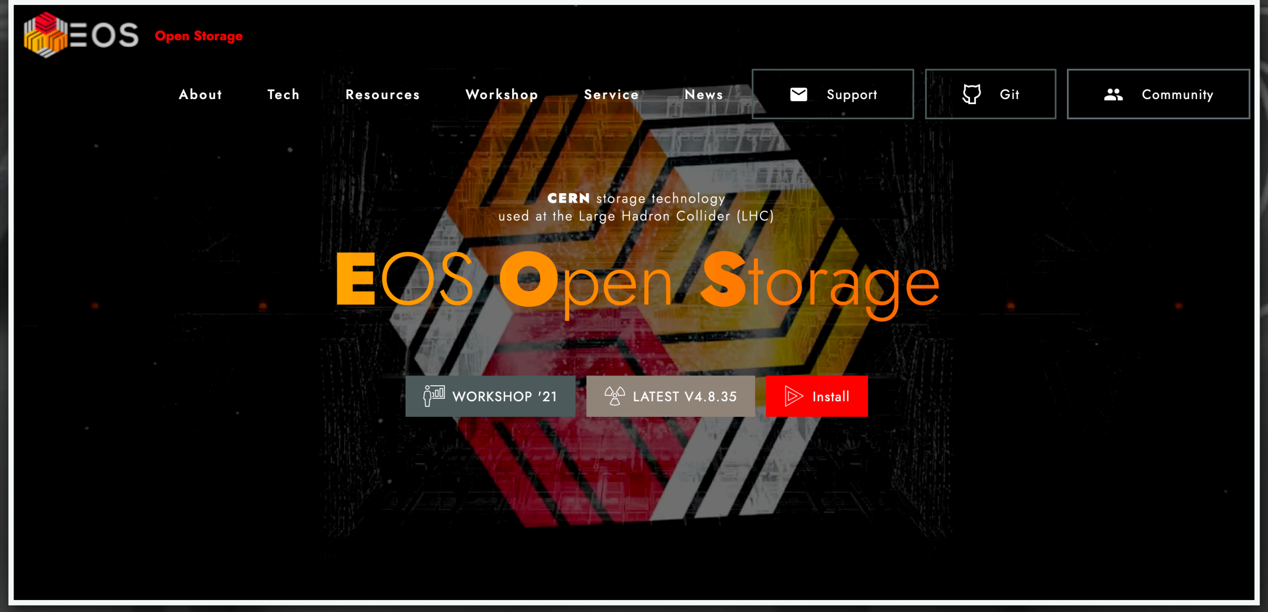
- We need to use BTRFS /proc information for disk statistics EOS-6086 to stop writing on non-writable disks

- We intend to run few large servers with SMR disks in production instances for comparison under realistic workloads once hardware is available

- We need to find an acceptable solution to run in production with ALMA9 + BTRFS tools and kernel support

Useful Links

Web Page https://eos.cern.ch



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

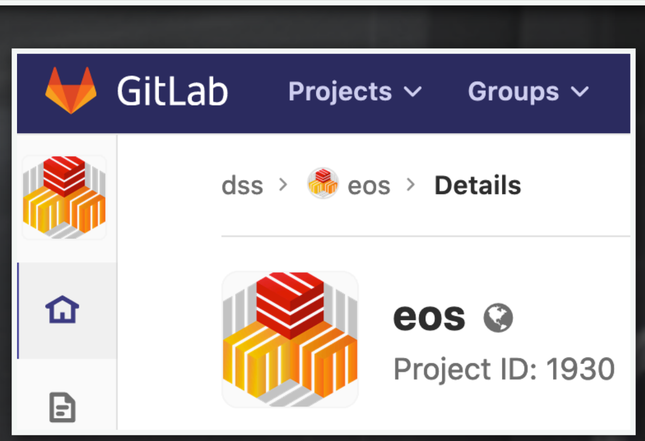
EOS Open Storage

19

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

19

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

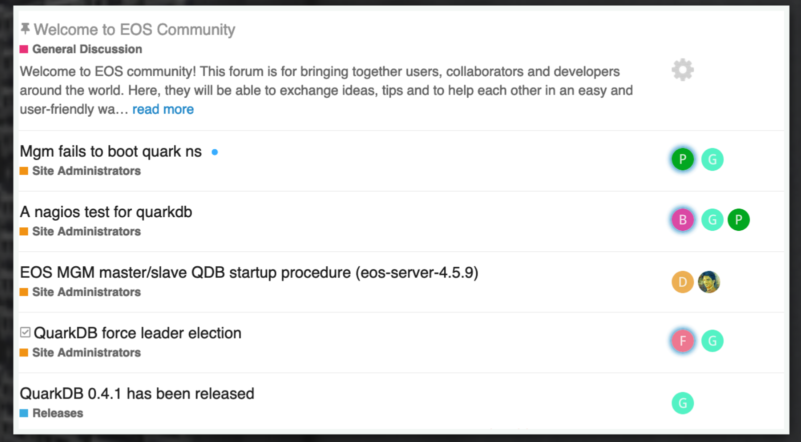


eos

Project ID: 1930

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

email: eos-community@cern.ch



Welcome to EOS Community
General Discussion
Welcome to EOS community! This forum is for bringing together users, collaborators and developers around the world. Here, they will be able to exchange ideas, tips and to help each other in an easy and user-friendly wa… read more

Mgm fails to boot quark ns
Site Administrators

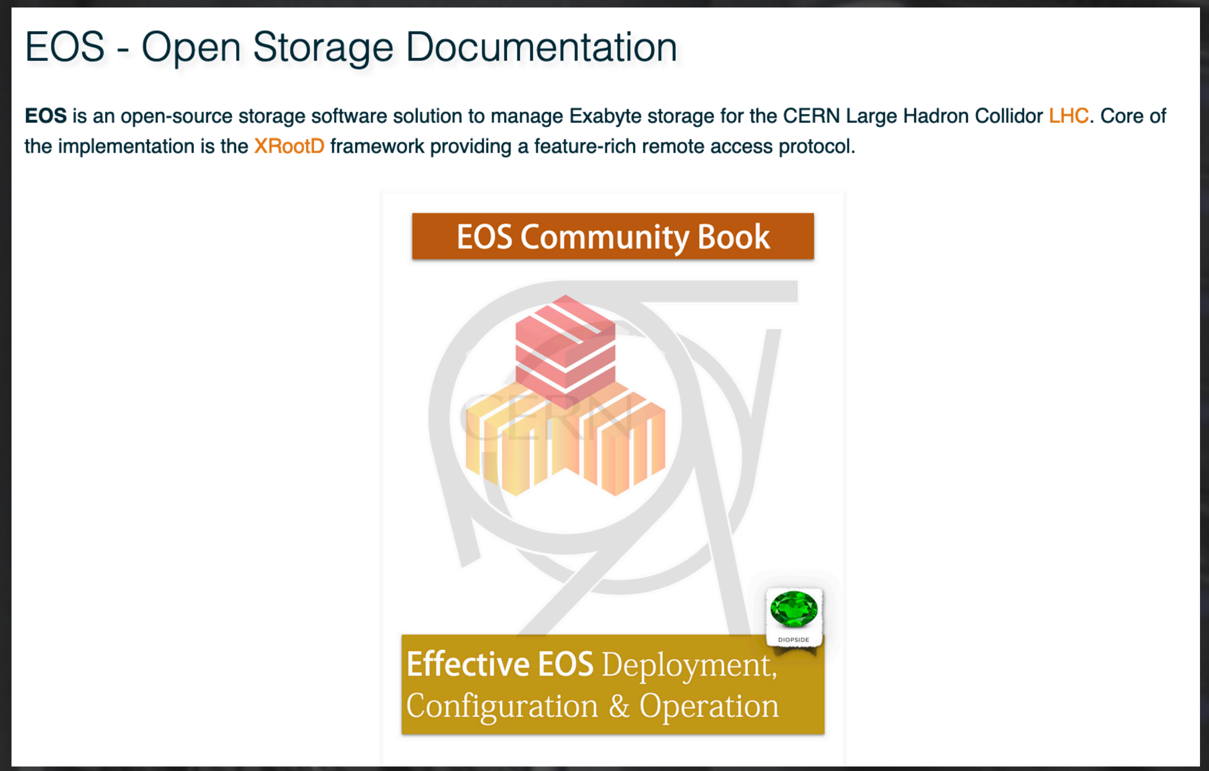
A nagios test for quarkdb
Site Administrators

EOS MGM master/slave QDB startup procedure (eos-server-4.5.9)
Site Administrators

☑ QuarkDB force leader election
Site Administrators

QuarkDB 0.4.1 has been released
Releases

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



EOS - Open Storage Documentation

EOS is an open-source storage software solution to manage Exabyte storage for the CERN Large Hadron Collidor LHC. Core of the implementation is the XRootD framework providing a feature-rich remote access protocol.

Support email: eos-support@cern.ch

19

Thank you for your attention!
Questions?