SSD Performance

RP1 Sebastian Carlier
& Daan Muller

Research topic

Maximizing Solid State Disk throughput.

Official research question:

How can SARA implement **S**olid **S**tate **D**isks in their setups to improve **sequential read performance** over conventional spinning disks and what parameters should be used to accomplish this?

Testing parameters

- RAID stripe sizes
- File system block sizes
- RAID levels
- Software vs. hardware RAID
- Areca vs Dell RAID controller
- Number of disks (scalability)
- File systems

Server setup

Base:
Intel Xeon CPU X5550 (Quad core + HT)

6 GB DDR3

8 disk SAS backplane

SAS Controllers:
Dell PERC 6/i

Areca ARC1680xi-12

Disks
Spinning:
Dell 7.2k rpm 160 GB (x5)

Solid state:
Intel X25-M G2 160GB (x5)

Intel X25-M G2 160 GB

- 10 x 16 GB Intel NAND chips
- multi level cell
- 512 KB block / min. overwrite size
- 4 KB page size / min. write size
- Manufacturer claims up to:
  - 250 MB/s read speed
  - 0.065 ms read latency

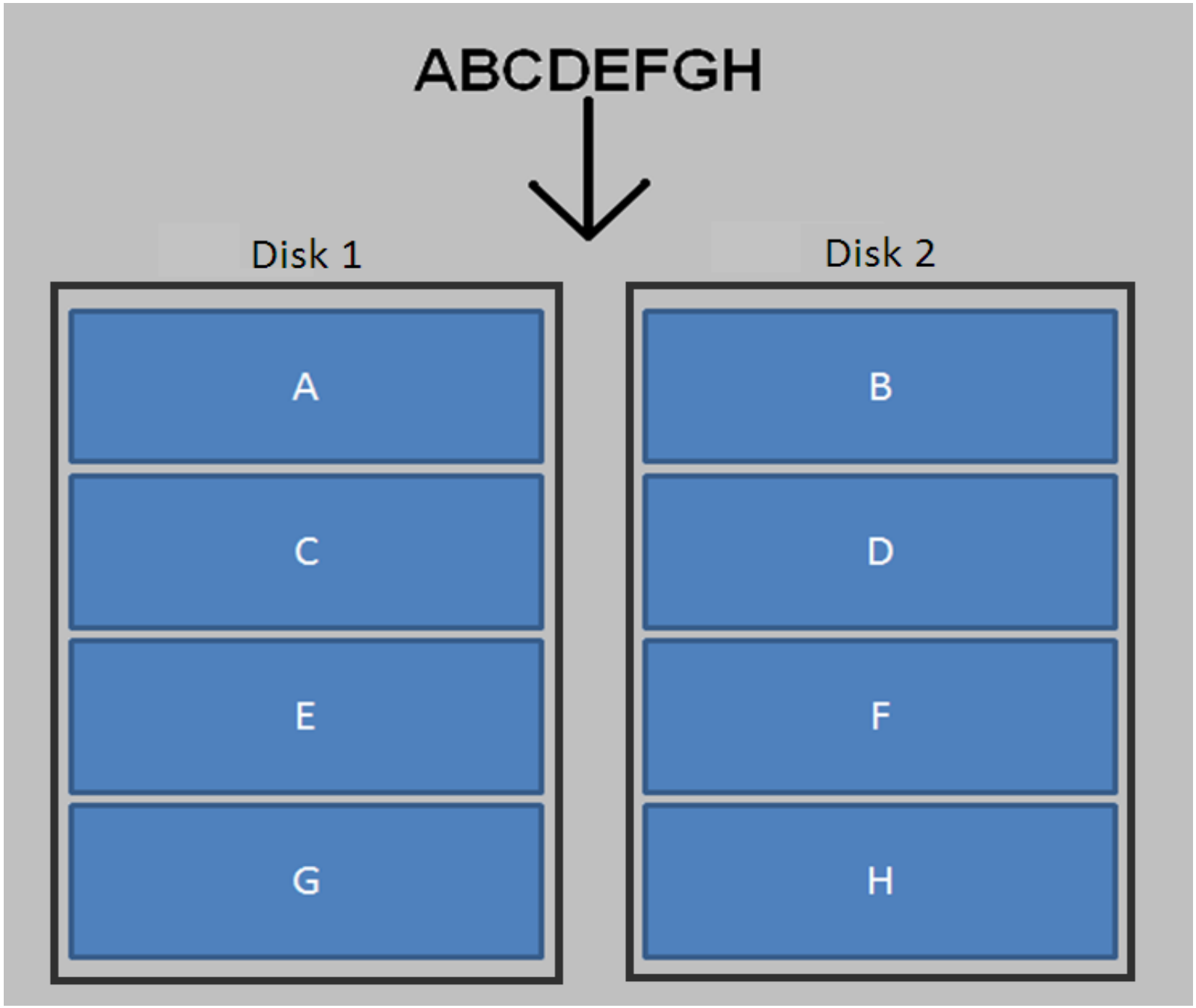
Predictions

- Read performance scaling linearly.
- Achieving higher read speeds with hardware RAID.
- RAID-0 / 1 arrays would be fastest in Linux.
- RAID-5 arrays would be a close second.
- Larger RAID stripe size would increase speed.
- Changing filesystem blocksize would have a significant effect.
- File systems having a noticeable effect.
- Predictions made during preliminary tests:
  - Perc i6 RAID controller would be a bottleneck.
  - Areca ARC 1680xi-12 would answer our prayers (that is reach ~1100MB/s read performance).

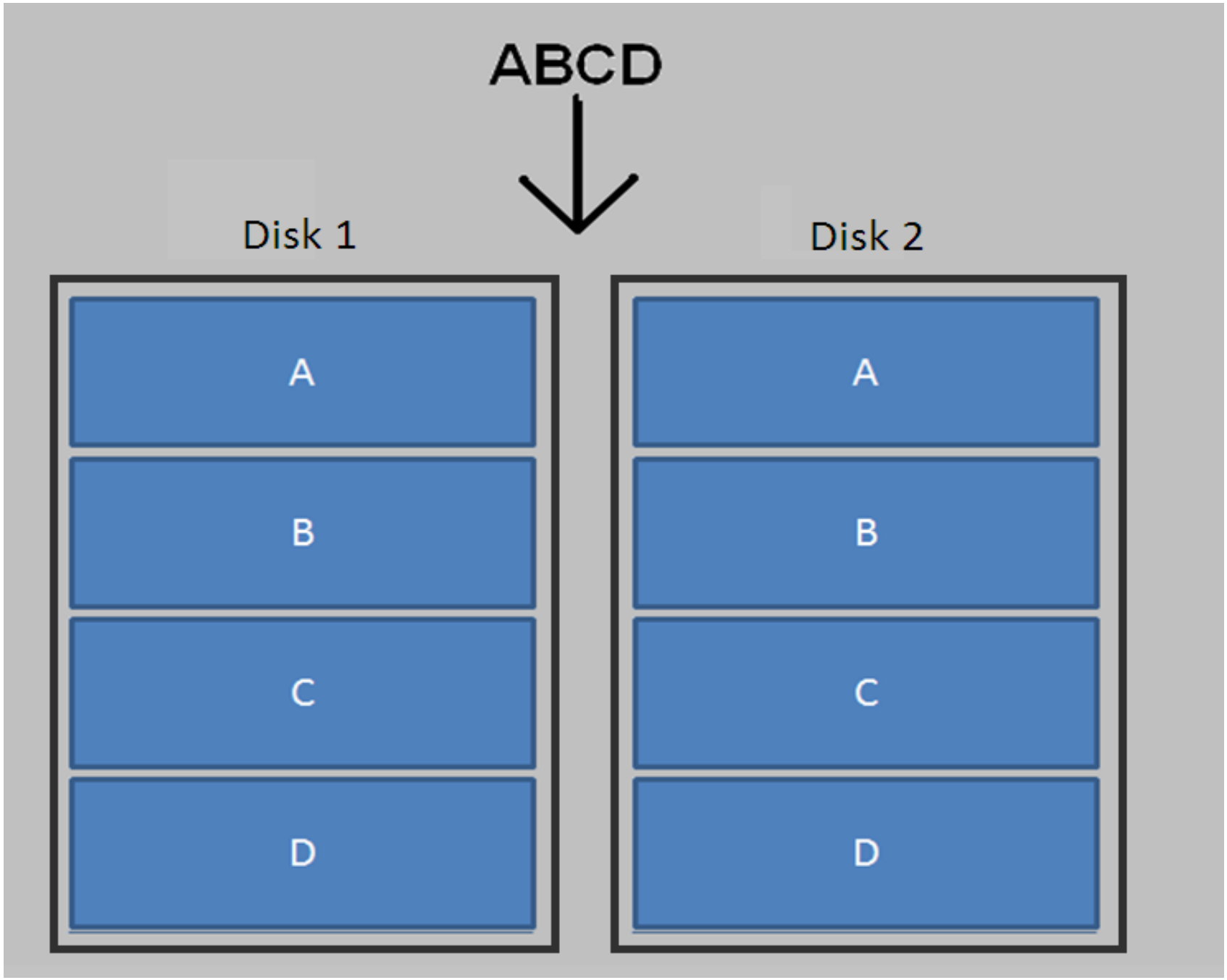
RAID levels

- RAID-0
- RAID-1
- RAID-10
- RAID-5
- RAID-6

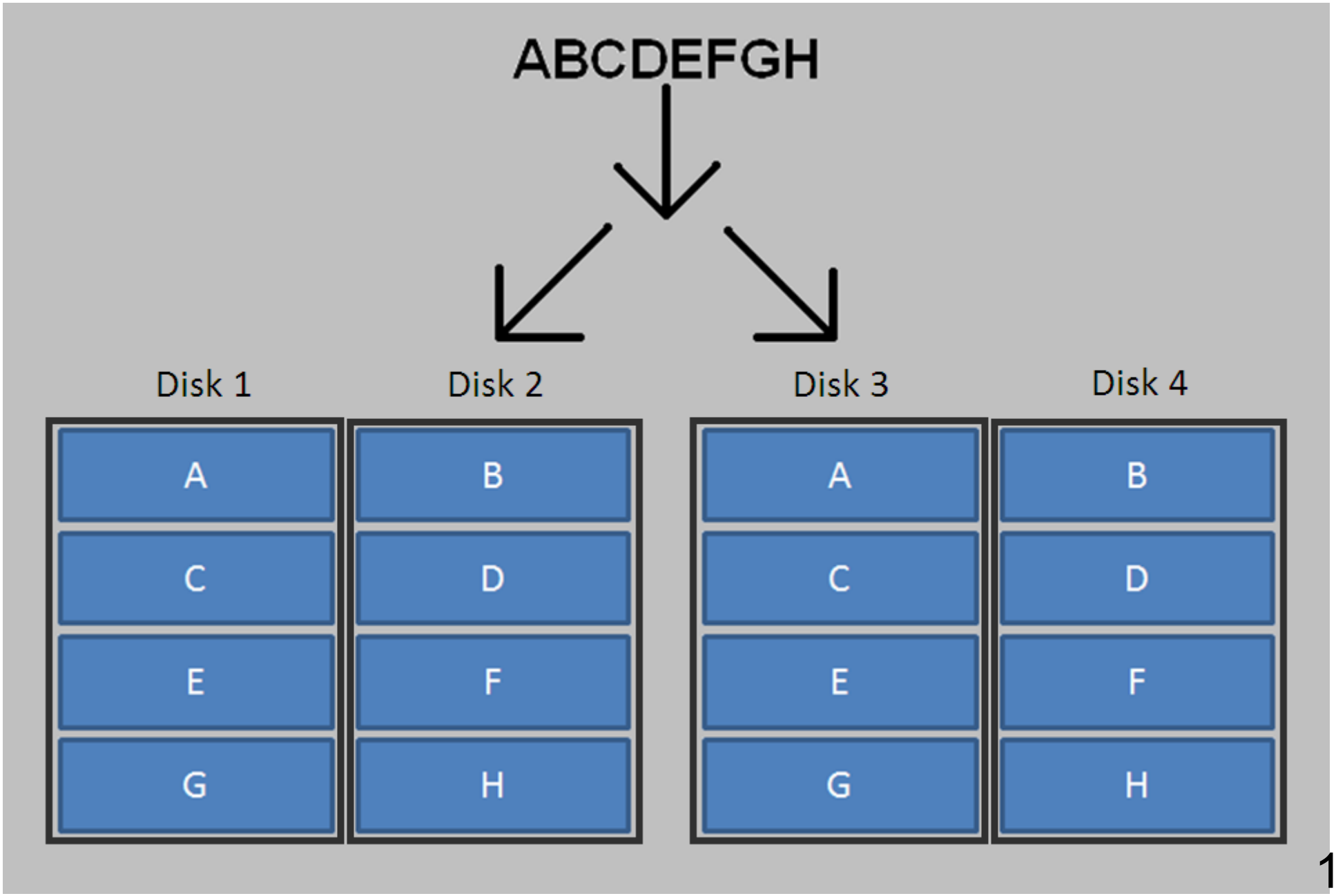
RAID-0



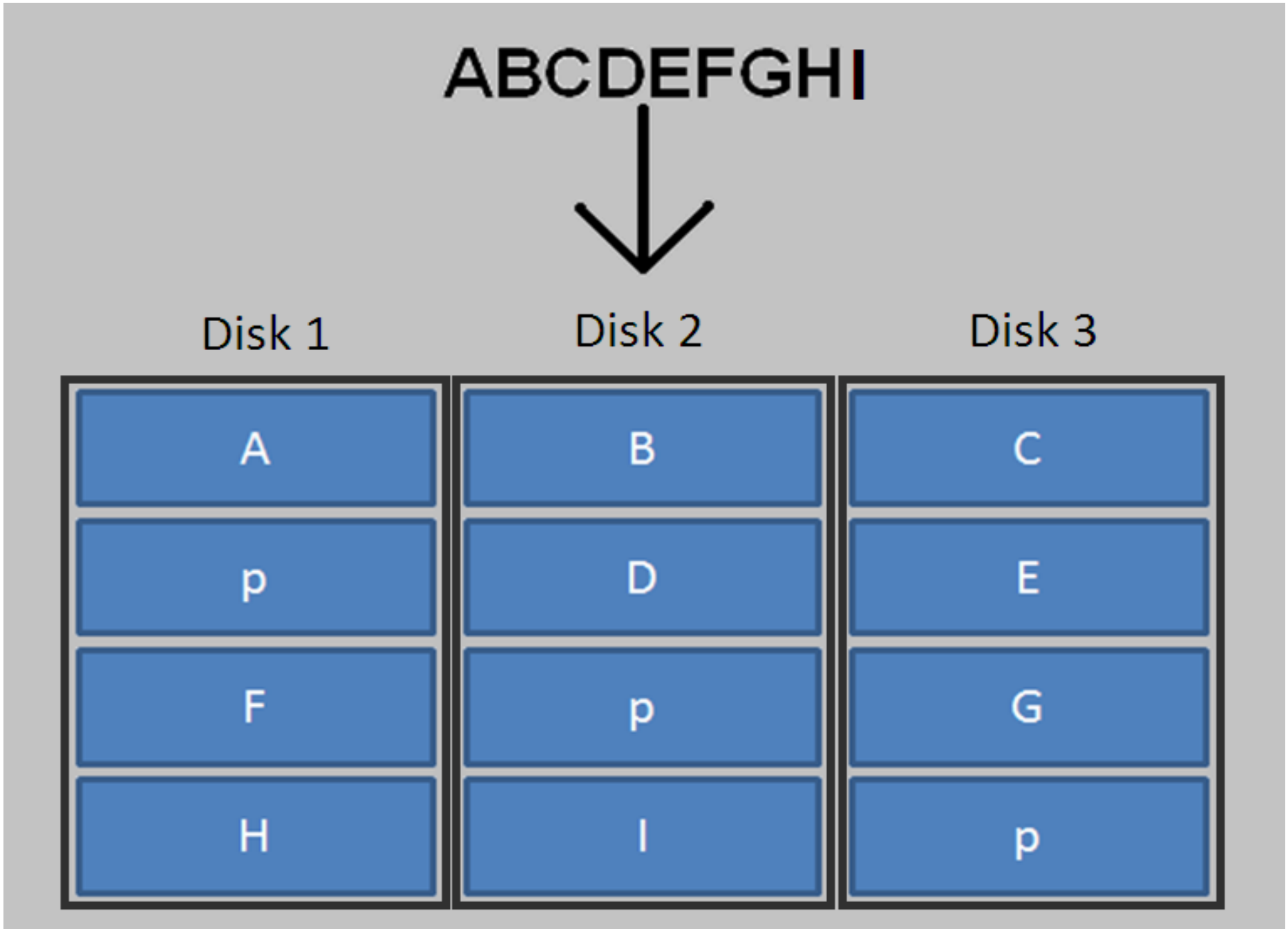
RAID-1



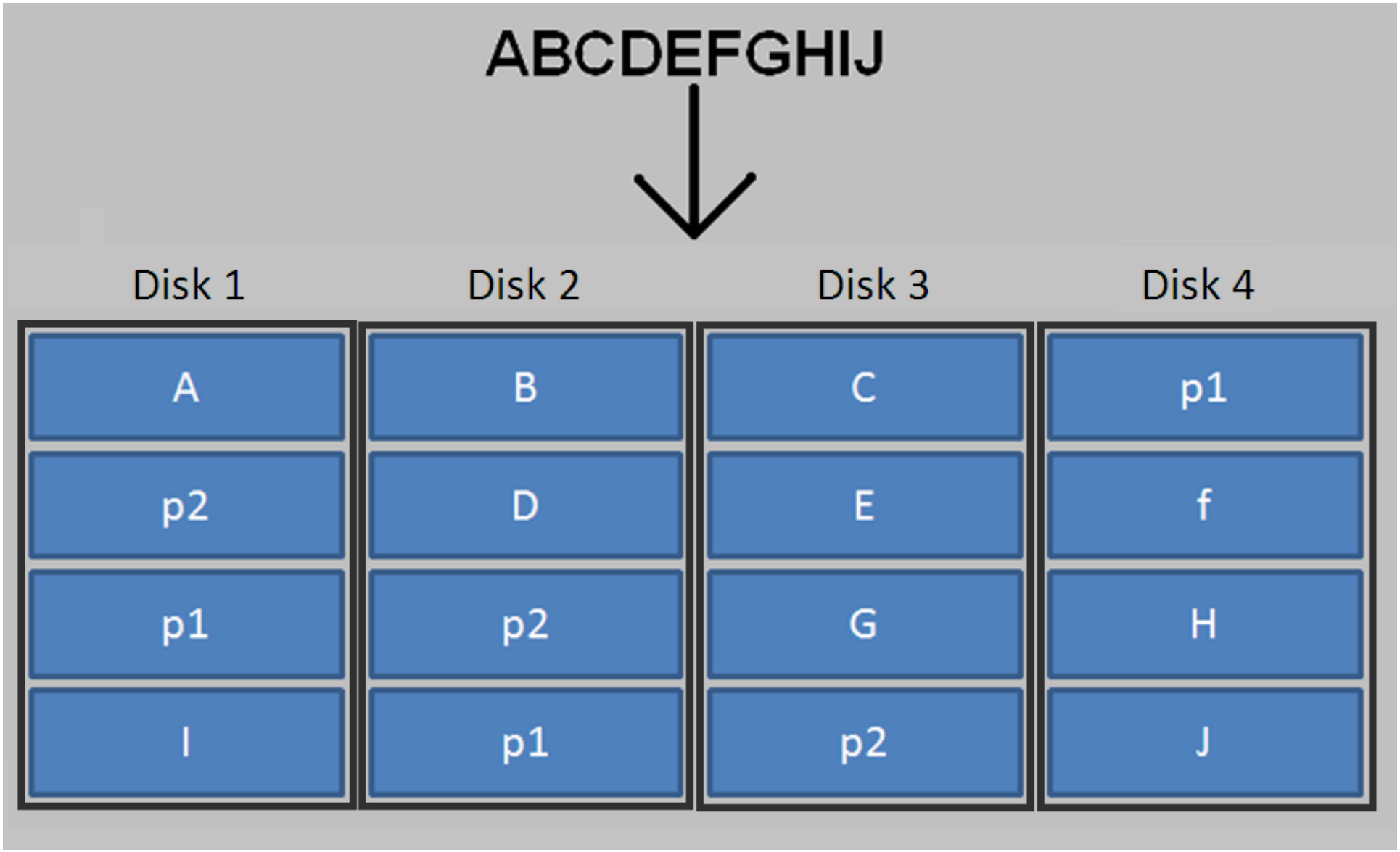
RAID-0+1



RAID-5



RAID-6


ABCDEFGHIJ
Disk 1
Disk 2
Disk 3
Disk 4
A
B
C
p1
p2
D
E
f
p1
p2
G
H
I
p1
p2
J

File systems

- ext4
  - current Linux standard
- nilfs2
  - filesystem focused on recovery
- btrfs - 'butter FS'
  - early version not fit for production use
  - good scalability
  - possible future standard on Linux
- zfs
  - combining file system with disk array
  - good scalability
- logfs/jffs2
  - designed for raw flash memory

ZFS: RAIDZ

- ZFS version of RAID-5
- variable stripe size
  - corresponds to file system block size
  - only possible because of ZFS integration
  - prevents partial-stripe write errors

Benchmark setup

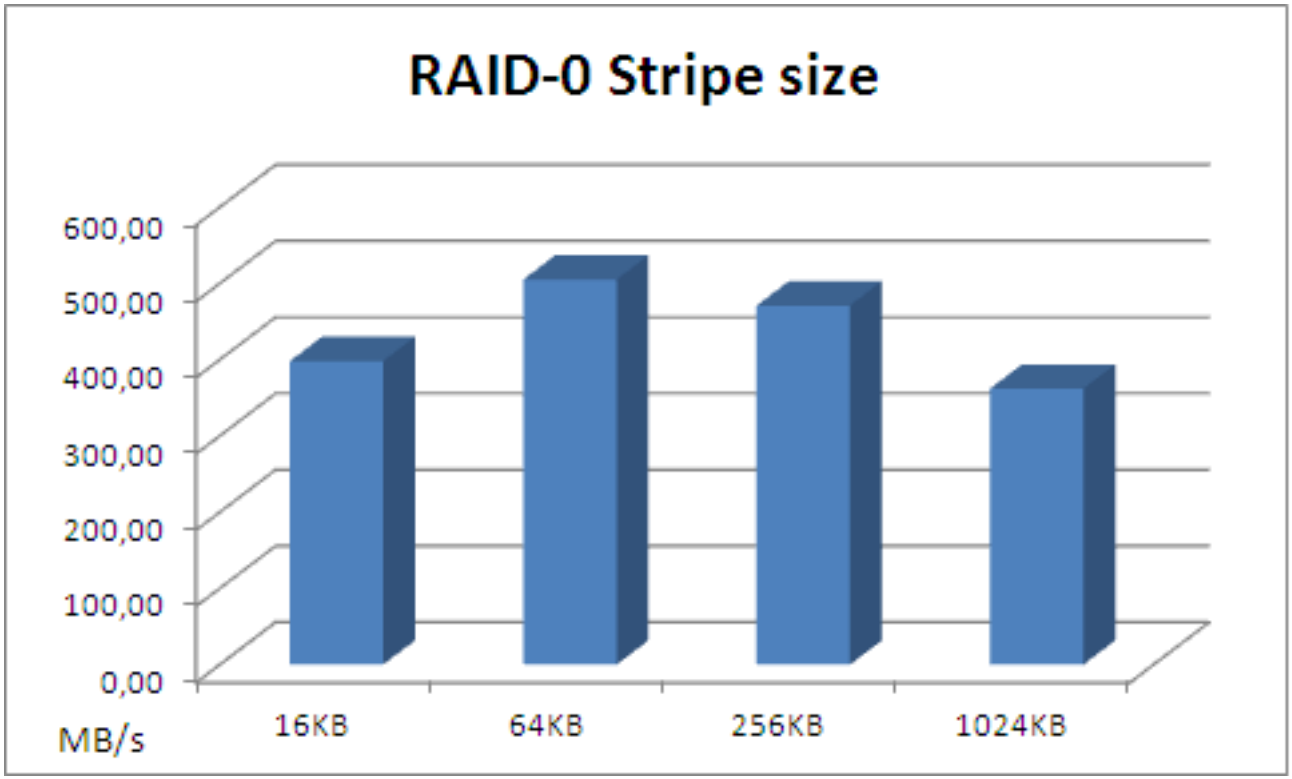
- IOZone.
- Phoronix-test-suite used to automate testing.
- Sequential reads on an 8GB file.
- Every setup tested 3 times.
- Results rounded to MB/s

Testing file system blocksize

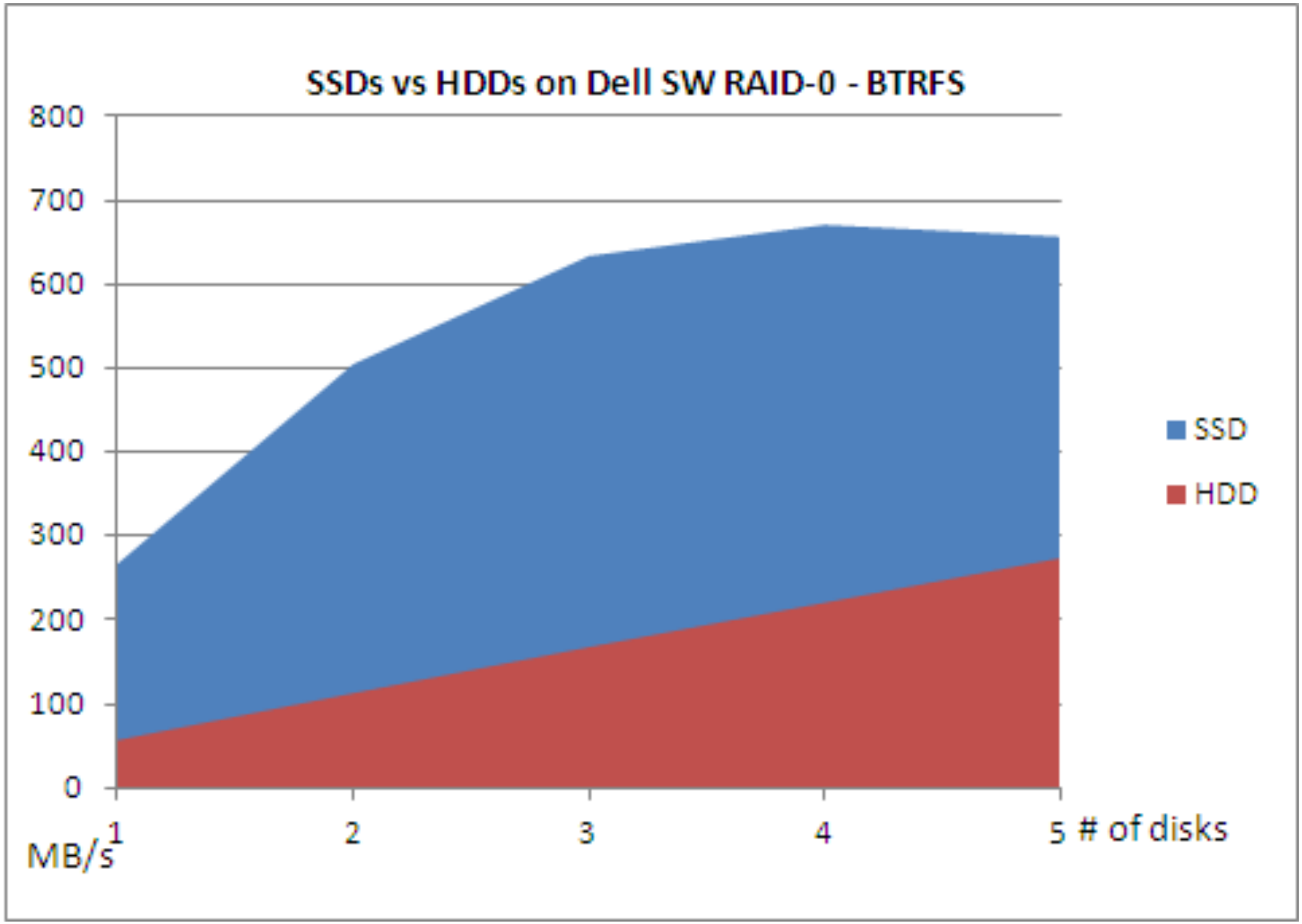
- single SSD
- file system ext4
- file system block size:
  - 1KB - 208 MB/s
  - 4KB - 213 MB/s (default setting)

- Conclusion: file system block size is not a significant factor at these high read speeds.

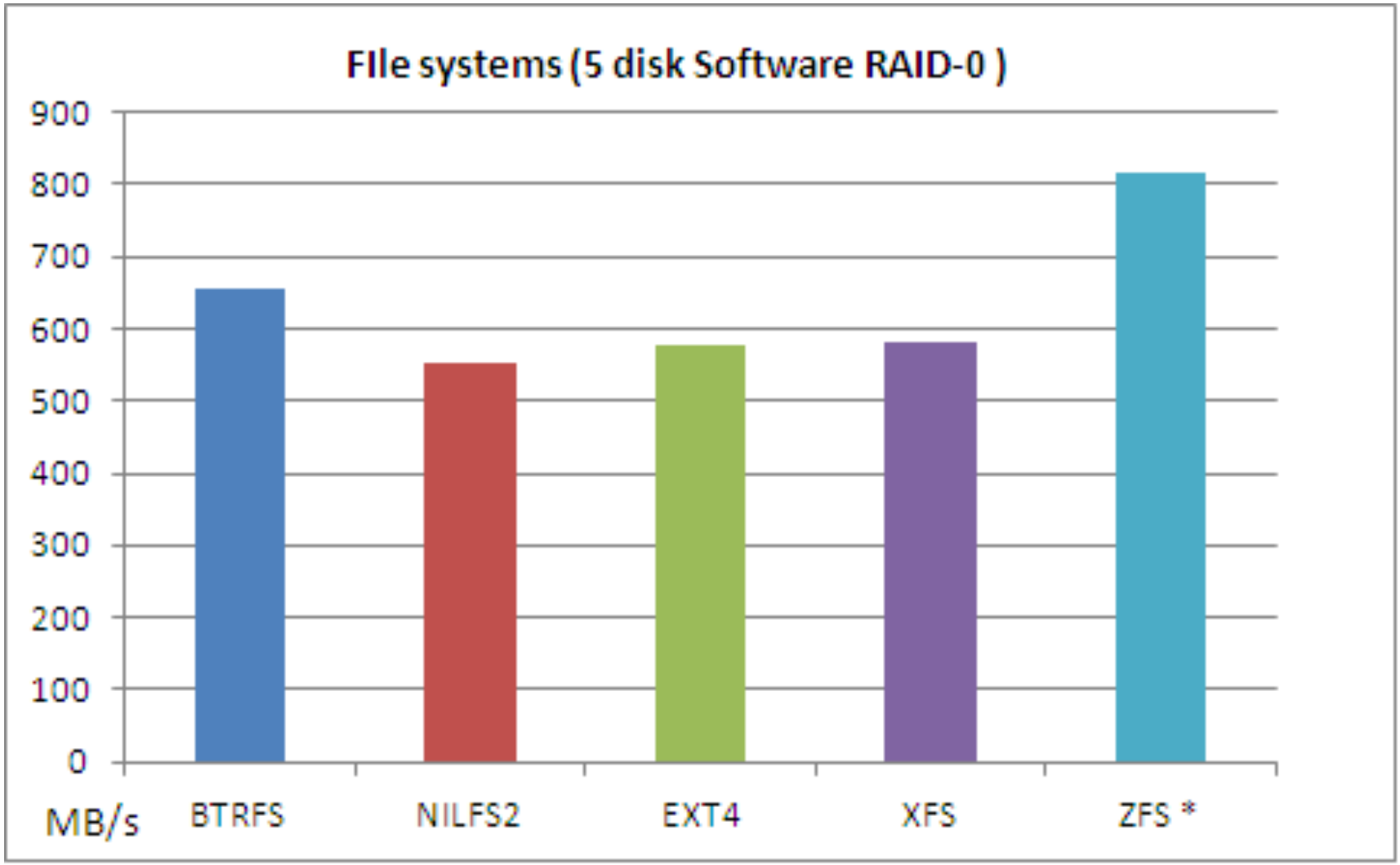
5 SSD Hardware RAID-0.



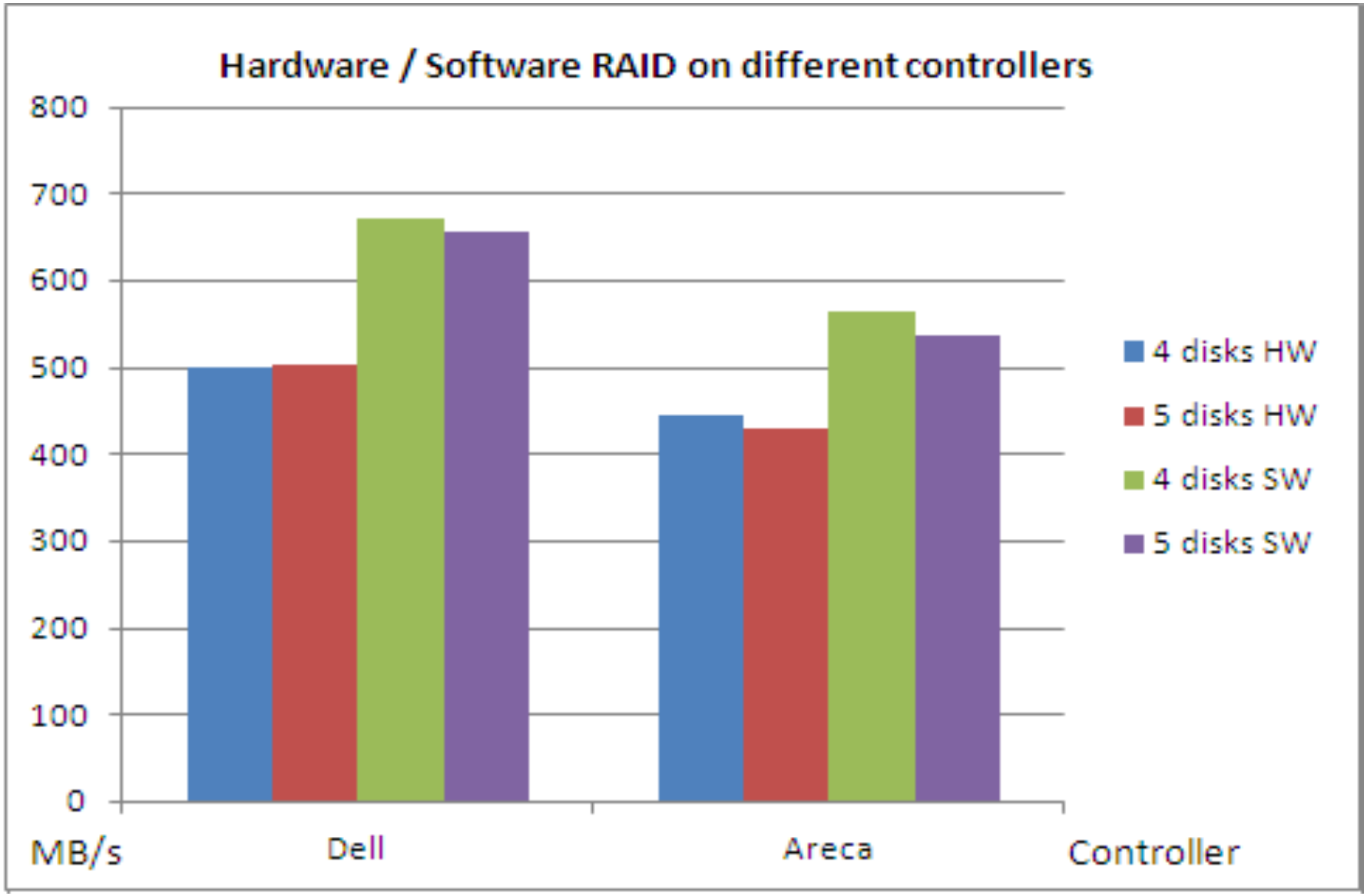
- Controller does not handle a non-default stripe size efficiently.
- The 16KB stripe size generates too many operations.



- SSDs no longer scale after 4 disks
- HDDs continue to scale after 5 disks



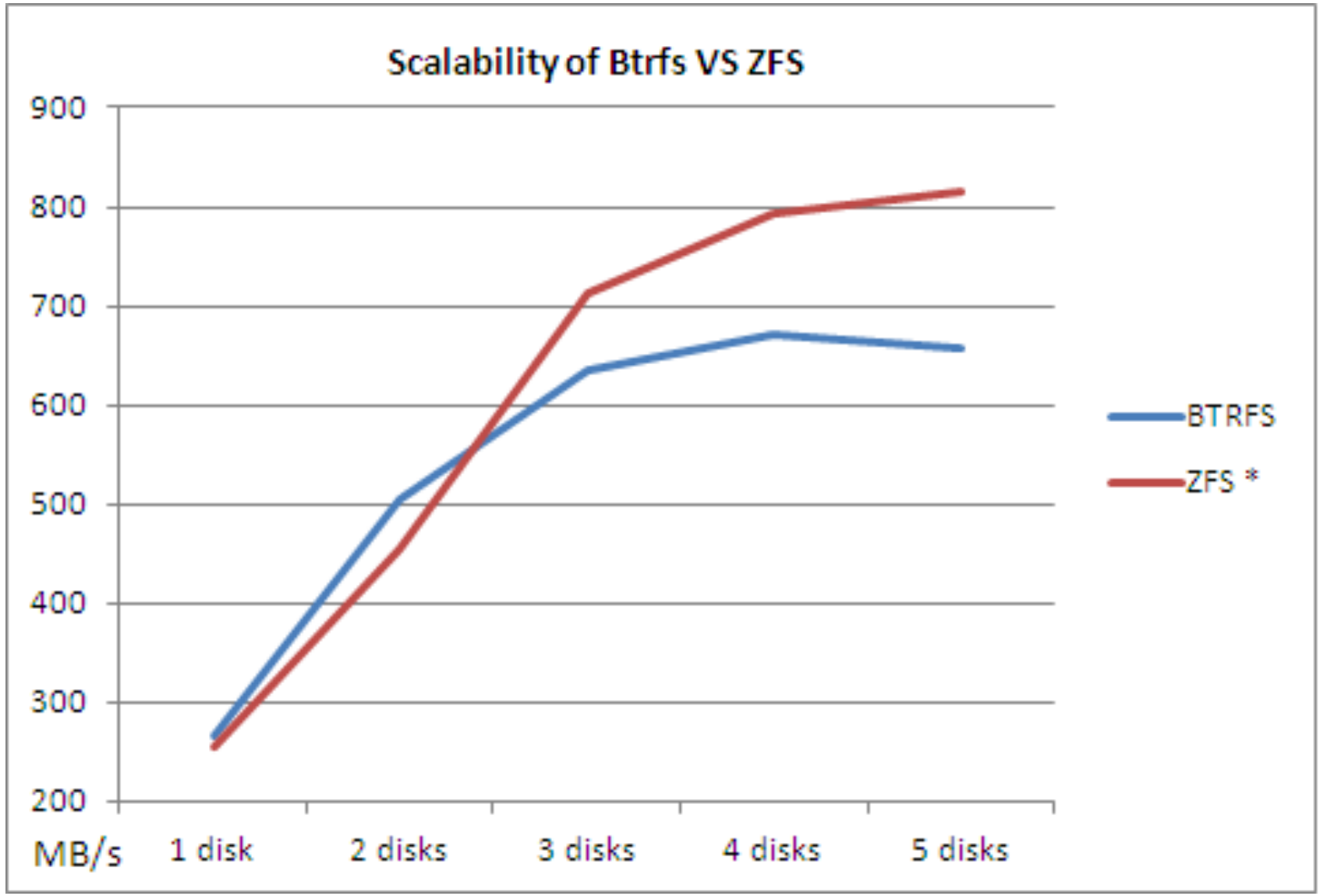
File system has a noticeable influence.



- Dell controller handles SSDs better than Areca controller.
- Software RAID beats hardware RAID.

Areca ARC 1680xi compatibility problems

- Intel IOP 348 CPU.
- Intel refuses to release source code.
- Areca cannot fix this problem themselves
- Many PCI-E SAS RAID cards use this CPU.
- Next Areca1880 series use a Marvell CPU
  - Not available yet.
- Conclusion:
  - Money well spent :)
  - Wait for the ARC 1880 series or find a card with a different CPU.



- RAIDz is faster while handling more disks than software RAID-0.

Proving the theory

- Where is the bottleneck for Linux ?
  - Btrfs ?
  - PCI-E 2.0 8x ?
  - RAID Controller ?

- Connected 2 disks to Areca, 3 disks to PERC
  - Software RAID-0 over 5 disks, btrfs
  - 815 MB/s

Predictions revisited

- Read performance scaling linearly.
- Achieving higher read speeds with hardware RAID.
- RAID-0 / 1 arrays would be fastest on Linux.
- RAID-5 arrays would be a close second.
- Larger RAID stripe size would increase speed.
- Changing filesystem blocksize would have a significant effect.
- Filesystems having a noticeable effect.
- Predictions made during preliminary tests:
  - Perc i6 RAID controller would be a bottleneck.
  - Areca ARC 1680xi-12 would answer our prayers (that is reach ~1100MB/s read performance).

Conclusions

- Btrfs seems a valid solution for the future for Linux OS.

  "Ext4 is simply a stop-gap, Btrfs is the way forward." - Theodore Ts'o, ext4 developer

- RAID controller development is lagging behind.

- ZFS is the fastest solution available today.

- SSDs leave spinning disks behind.
  - even when it comes to sequential reads.
  - comparing them to consumer grade hardware.
    - (FC / SAS)

Further research

- The PCI-E v2.0 x8 bus has a 2GB/s throughput limit.
  - Test a setup with a different RAID controller and 8 SSDs.
  - Test a setup with multiple RAID controllers and 16 - 32 SSDs.
- PCI-E x16 RAID Controllers should hit the market in the near future.
  - Test if 16 SSDs reach the theoretical 4 GB/s throughput.
- Test if a single system can stream enough data for an entire video wall.

Thanks !

For their expert opinions:
Freek Dijkstra
Ronald van der Pol
Mark van de Sanden

For lending us a RAID controller:
www.WebConneXXion.com

Questions?