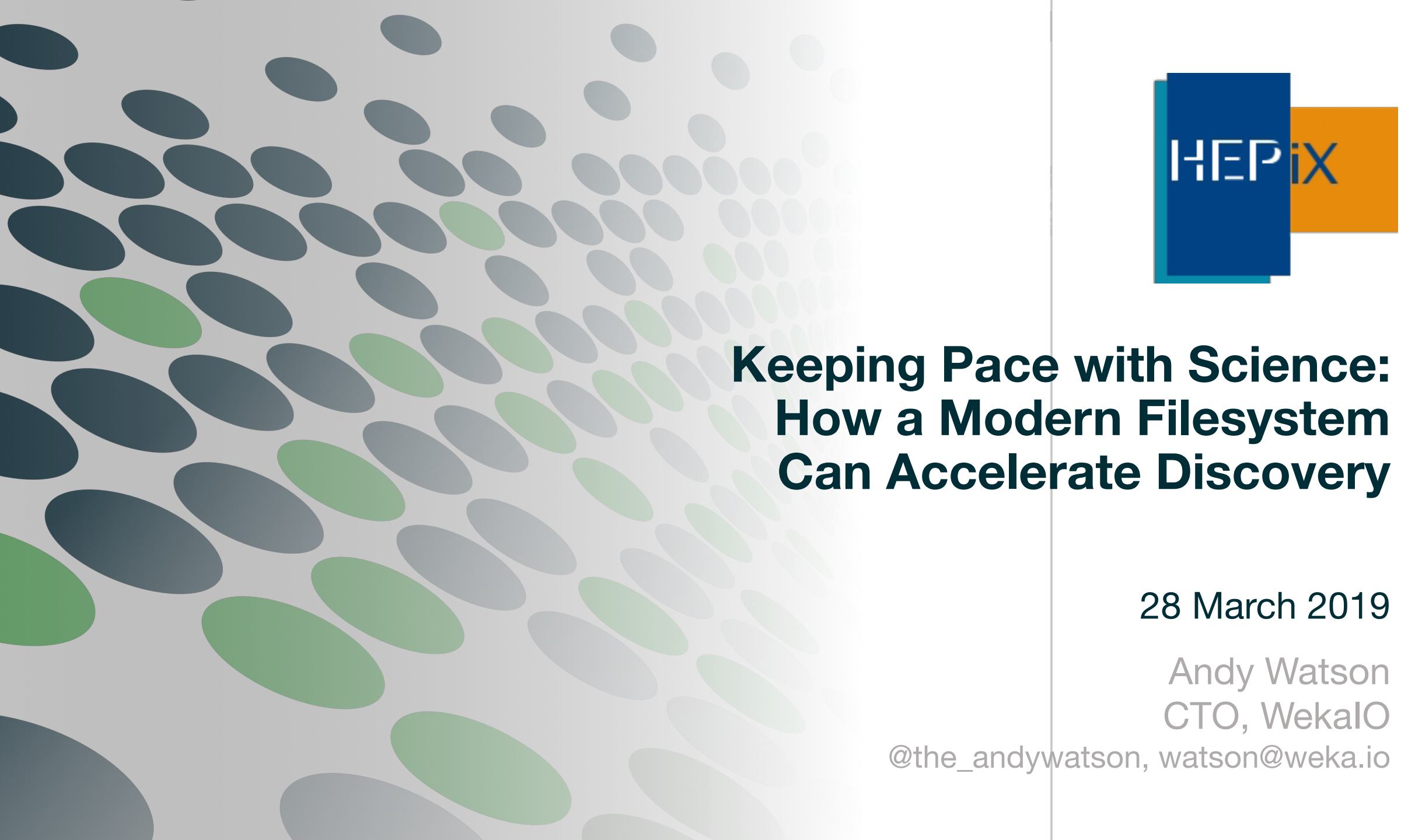


Keeping Pace with Science: How a Modern Filesystem Can Accelerate Discovery

28 March 2019

Andy Watson
CTO, WekaIO
@the_andywatson, watson@weka.io

Comm y Overview

WEKA.IO

Performance

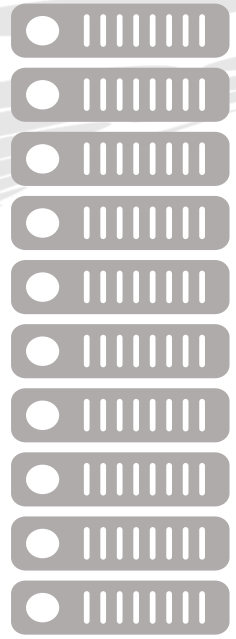
Huge Improvement in AI/ML Workflow Productivity

Before WekaIO

~80x SpeedUp with WekaIO

4 hours

GPU Acceleration — Intensifying IO

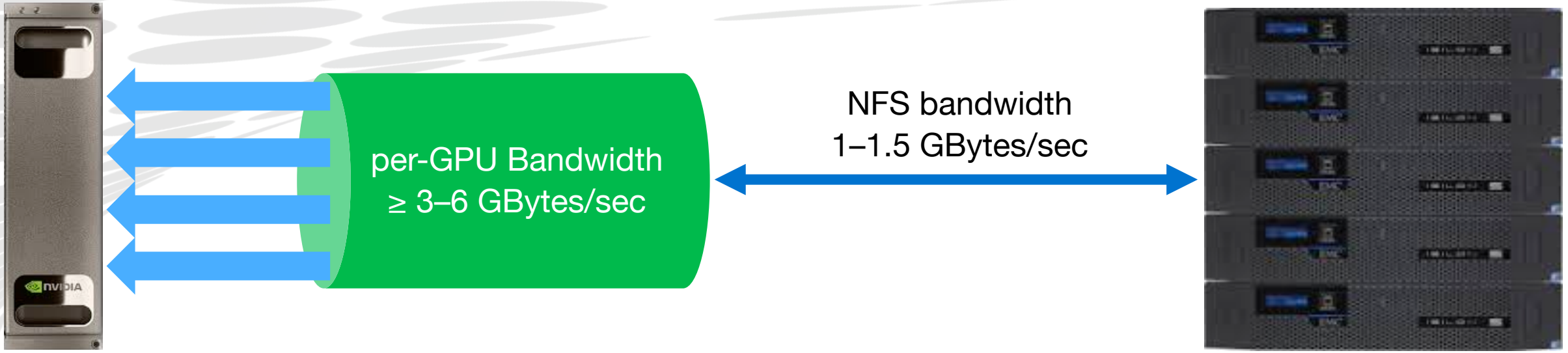


10
CPU-only
Servers

1
GPU-accelerated
Server

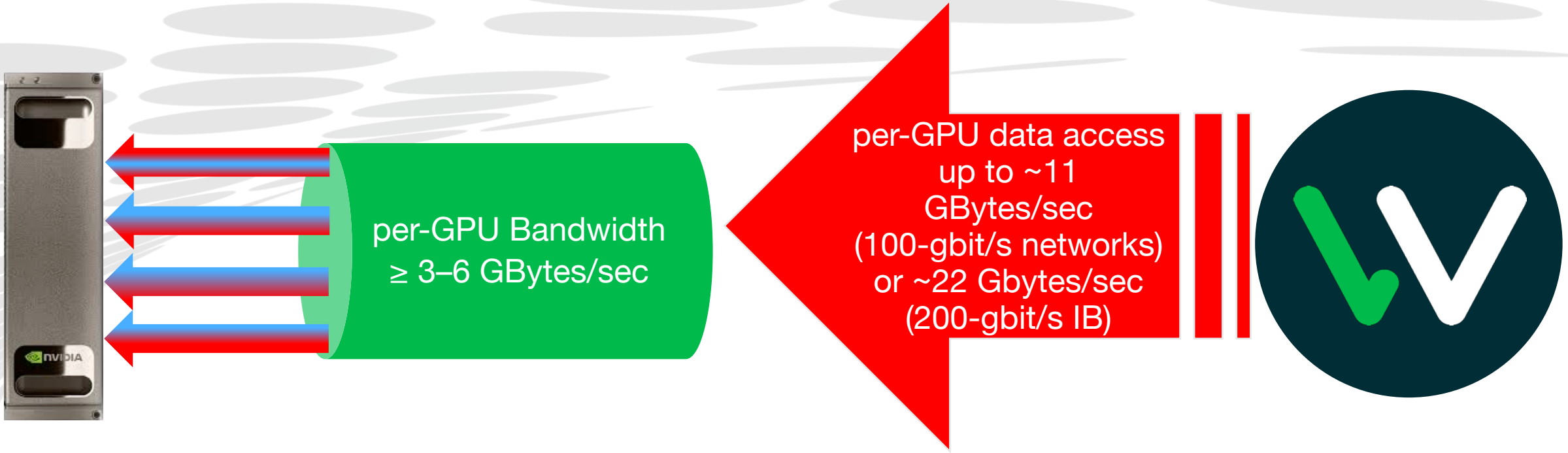
adding GPUs *concentrates* compute infrastructure's associated IO density by ~10x in this example

Performance: NFS Leads to IO Starvation



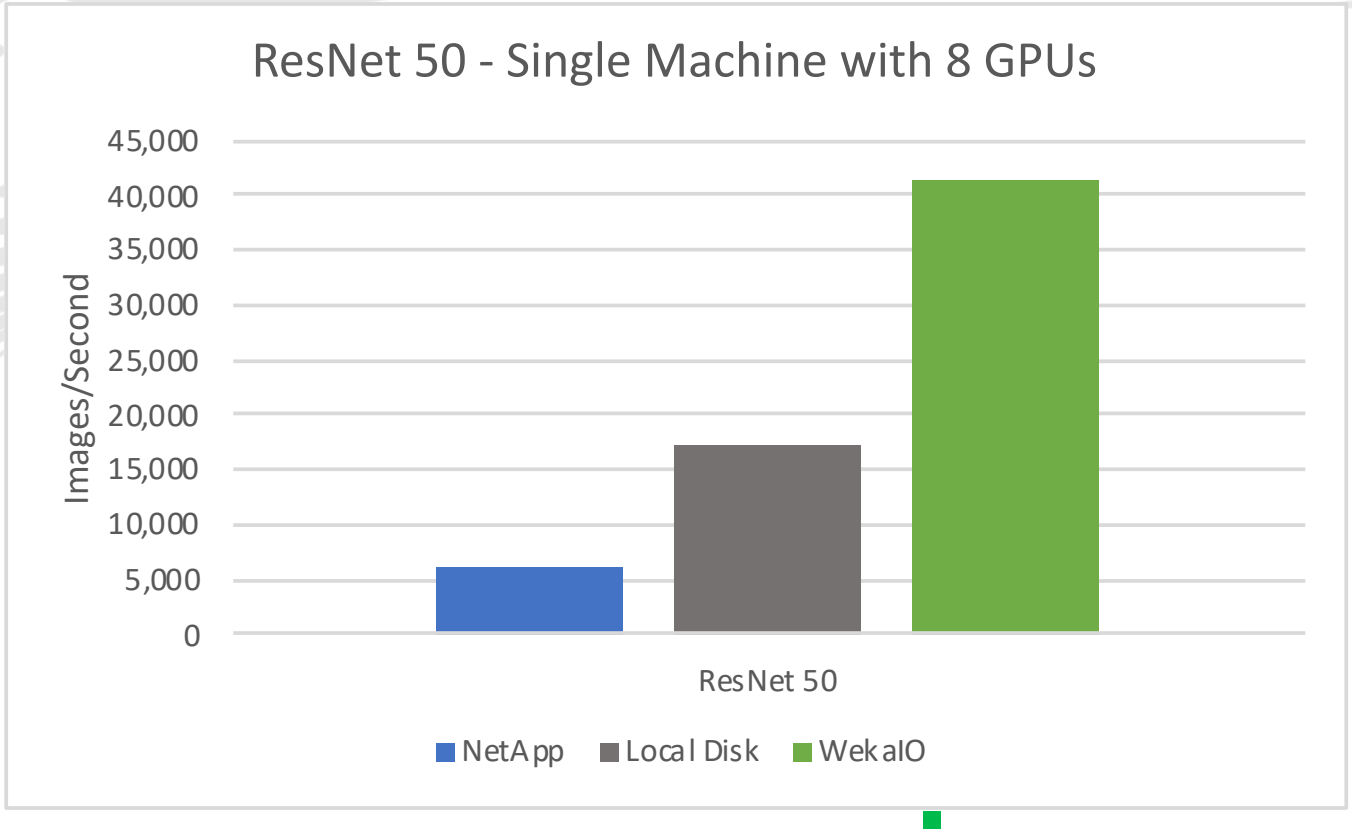
- NFS since 1984… NFSv2, NFSv3, NFSv4: but *still* only 1–1.5 GB/s per client
  - Parallelized NFSv4.1 requires client-side changes not yet accepted for production deployment
  - N.B., we're aware of some use of NFSv4.1 with dCache
- Other alternatives also fall short (e.g., Lustre, GPFS, IBM Spectrum Scale…)
  - Significant complexity associated with configuration and tuning required for workload variations
  - Ongoing issues with large directories (especially huge numbers of small files)

WekaIO Ends IO-Starvation



- WekaIO's Matrix™ is a shared parallel filesystem written for flash (not HDD)
  - Optimized for NVMe flash storage, including clustering using WekaIO's own NVMe Fabric
  - Low-Latency networking via InfiniBand or Ethernet (minimum 10-GbE; preferably 100-GbE)
  - Global Namespace includes transparent tiering to S3-API object storage on low-cost HDD
- WekaIO's Client is a local-mount POSIX filesystem in user-space on GPU Servers

ResNet 50 Inference Benchmark Performance



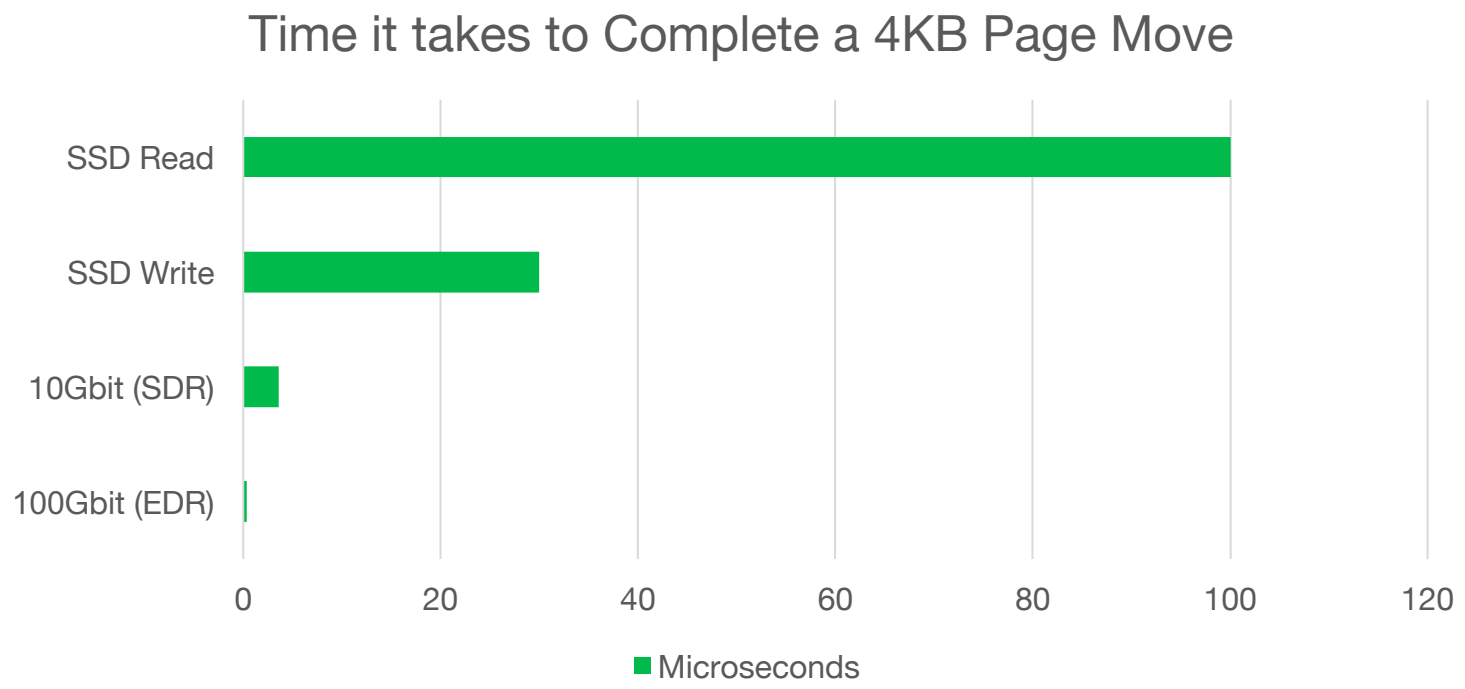
WekaIO is:-

- 140% Faster than local disk
- 7x faster than NetApp

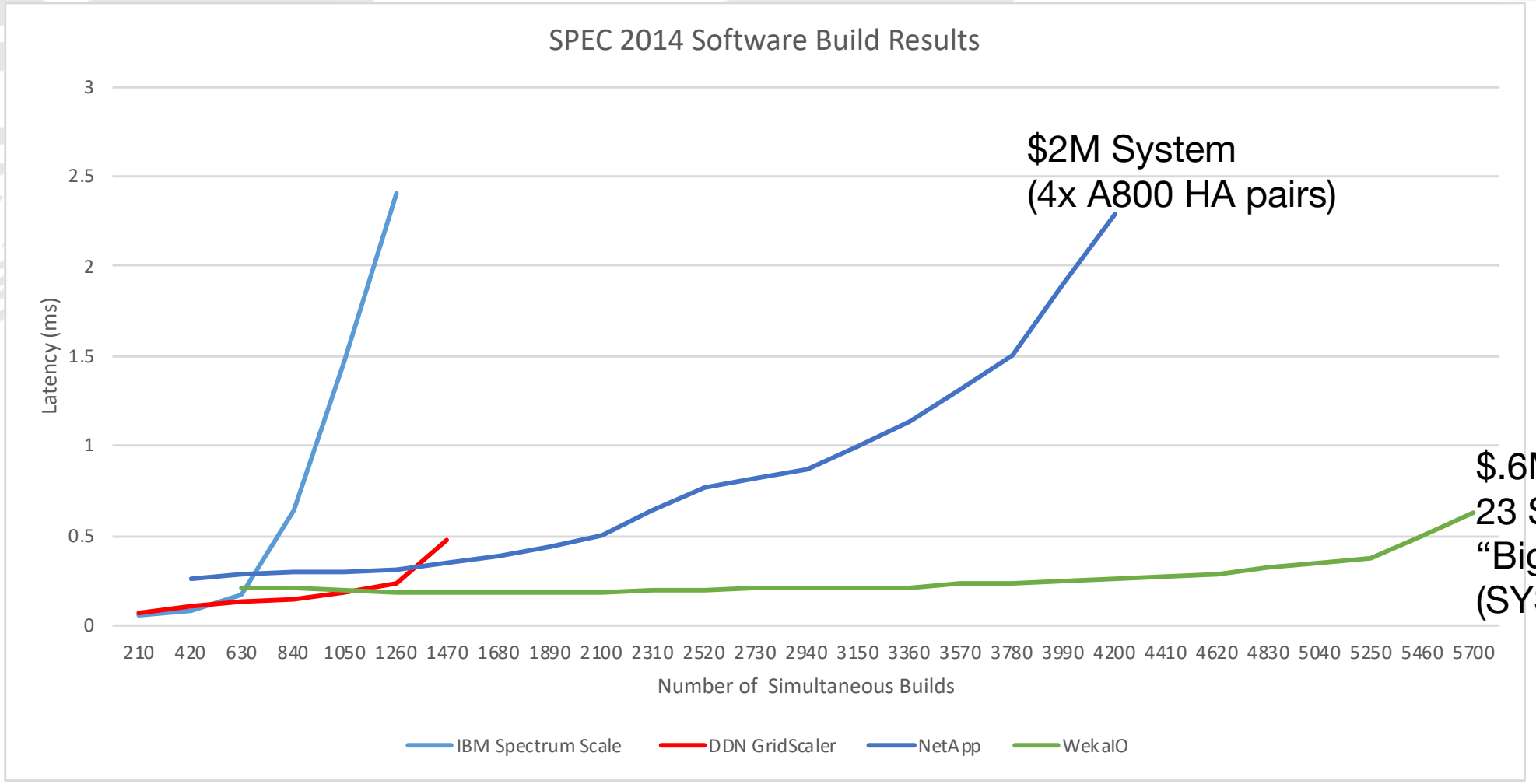
https://www.netapp.com/us/media/nva-1121-design.pdf
https://www.weka.io/promo/hpe-ai-tech-white-paper-oct-2018/

Why Data Locality is Irrelevant

- Modern networks on 10-GbE are 30x faster than SSD for Reads, and 10x faster than SSD for writes
- With WekaIO's networking implementation, our shared storage is **faster than local storage**



SPEC 2014 Results For Engineering SW Builds



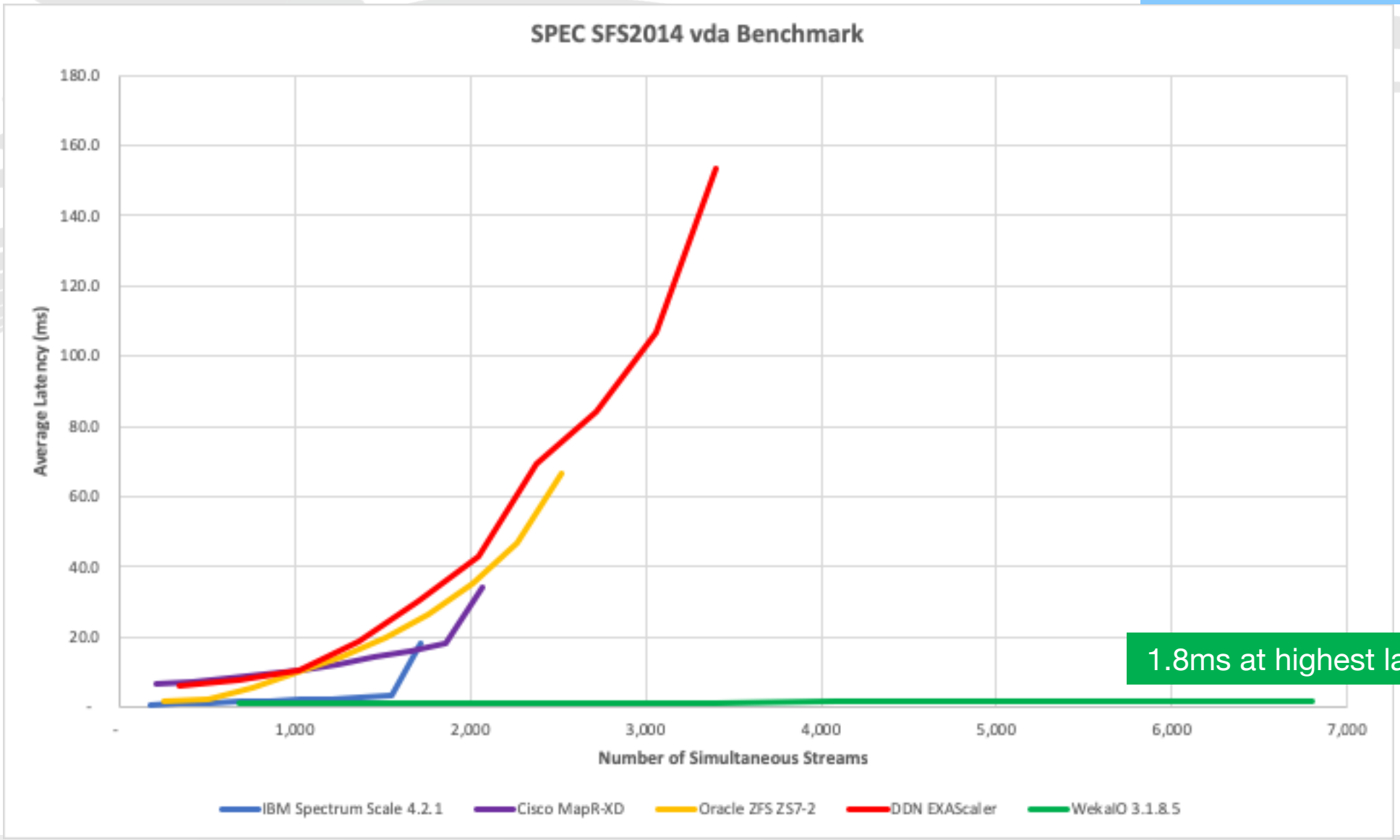
The Fastest, Most Cost-Effective Storage System

	NetApp 8 Nodes	NetApp 12 Nodes	WekaIO
Number of Builds	4,200	6,200	5,700
Latency (ORT)	0.78	0.83	0.26
$/Build	$604	$615	$105
Number of SSDs	192	288	138
Number of Clients	48	72	19
Builds/Client	88	86	300

- WekaIO latency is a third of the competition
  - Using 52% less SSDs
  - Using 74% less clients
  - Resulting in 249% more builds per client

SPEC SFS2014 vda Results

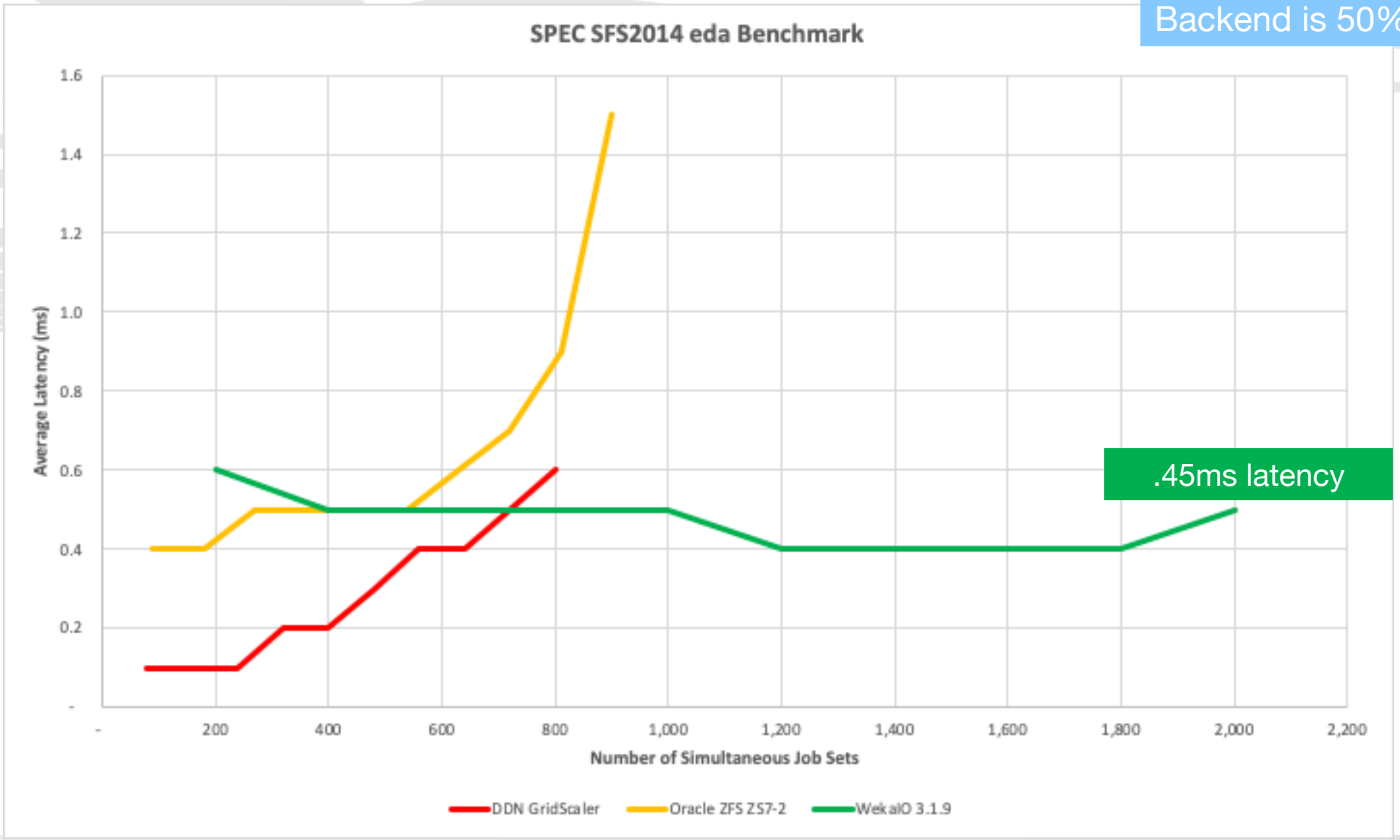
Test is 90% write intensive



1.8ms at highest latency

SPEC SFS2014 eda Results

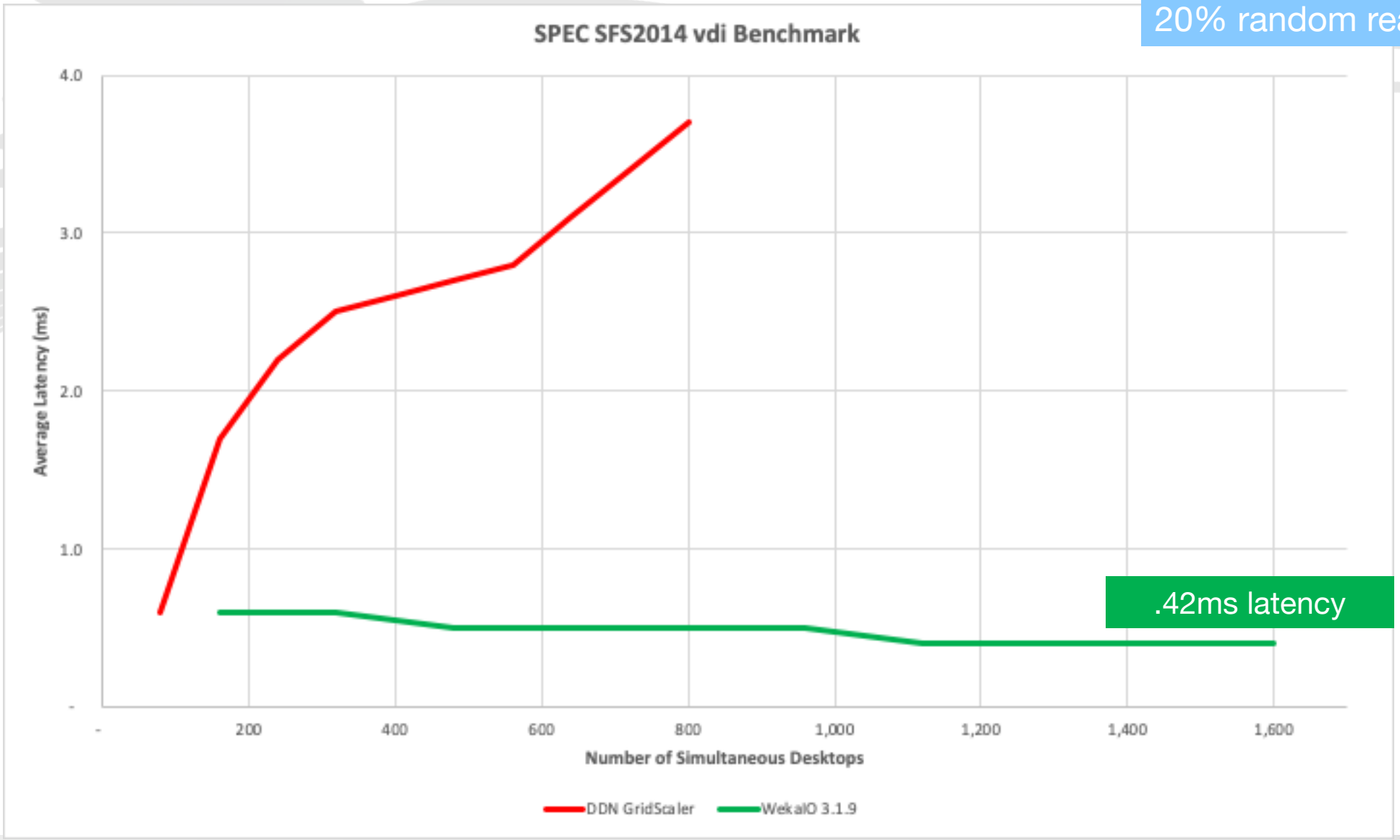
Frontend is 50% stats test
Backend is 50% R and W



.45ms latency

SPEC SFS2014 vdi Results

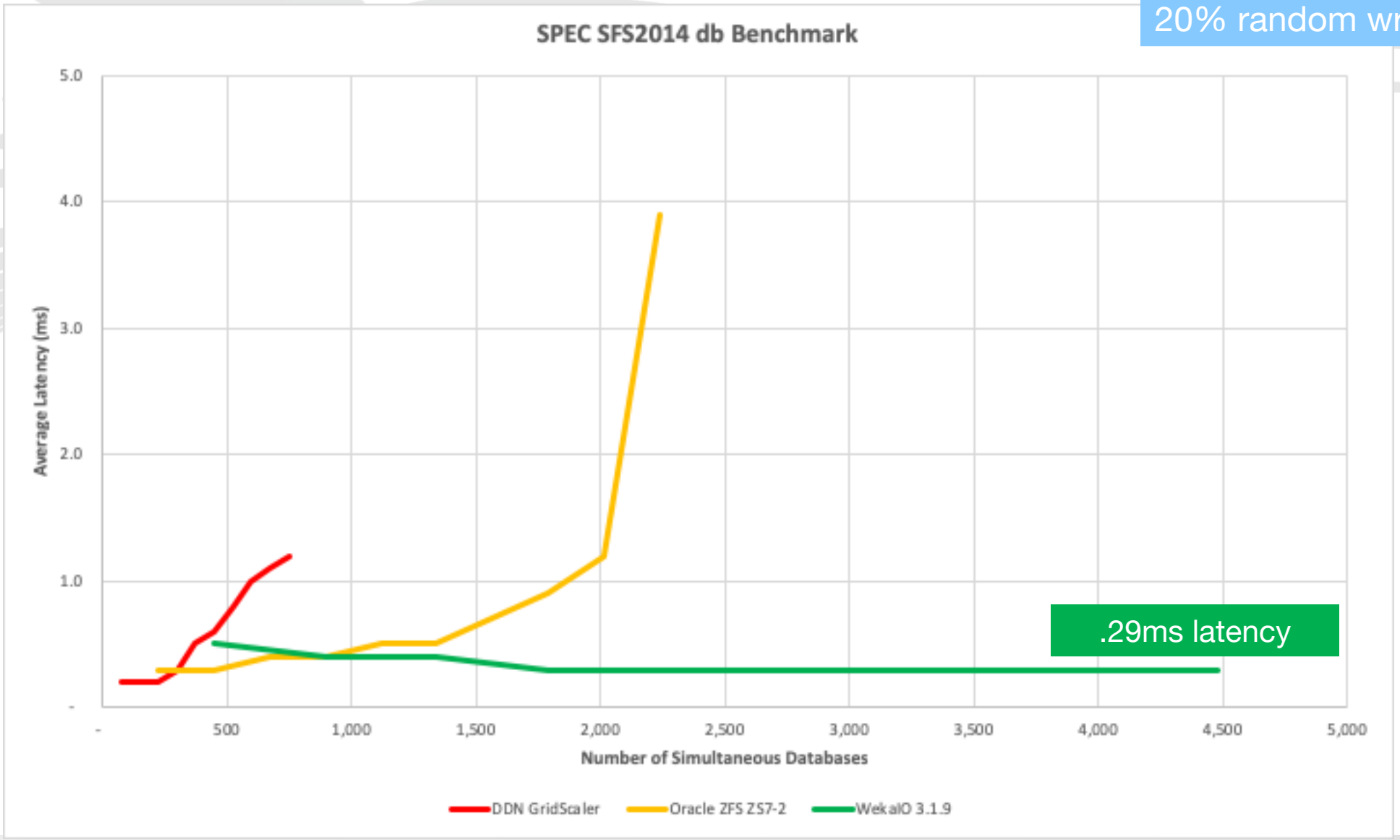
Test is 60% random write
20% random read



.42ms latency

SPEC SFS2014 db Results

Test is 80% random read
20% random write



Virtual Institute for I/O

https://www.vi4io.org/io500/list/19-01/10node

Recent Changes Sitemap

You are here: Virtual Institute for I/O » IO-500 » Lists » 2019-01 » **10 Node Challenge**

10 Node Challenge | Full List

10 Node Challenge

IO500

This is an unofficial intermediate list based on corrected calculations [1] for the IO-500 **10 Node Challenge ranked list**[2] containing the submissions from November 2018 (from SC 2018). The list shows all qualifying 10 node results.

#	information							io500		
	institution	system	storage vendor	filesystem type	client nodes	client total procs	data	score	bw	md
									GiB/s	kIOP/s
1	* WekaIO		WekaIO		10	700	zip	58.25	27.05	125.43
2	** Oak Ridge National Laboratory	Summit	IBM	Spectrum Scale	10	160	zip	44.30	9.84	199.48
3	DDN	Bancholab	DDN	Lustre	10	240	zip	31.50	6.33	156.69
4	IBM	Sonasad	IBM	Spectrum Scale	10	10	zip	24.24	4.57	128.61
5	KAUST	Shaheen II	Cray	DataWarp	10	80	zip	13.99	14.45	13.53
6	Google and DDN	Lustre on GCP	Google	Lustre	10	80	zip	12.82	4.30	38.23
7	Clemson University	ofsdev	Dell	BeeGFS	10	80	zip	10.17	2.32	44.67
8	Queen Mary; University Of	Apocrita	E8	GPFS	10	240	zip	9.65	4.32	21.55

Updated the week of Jan 21, 2019

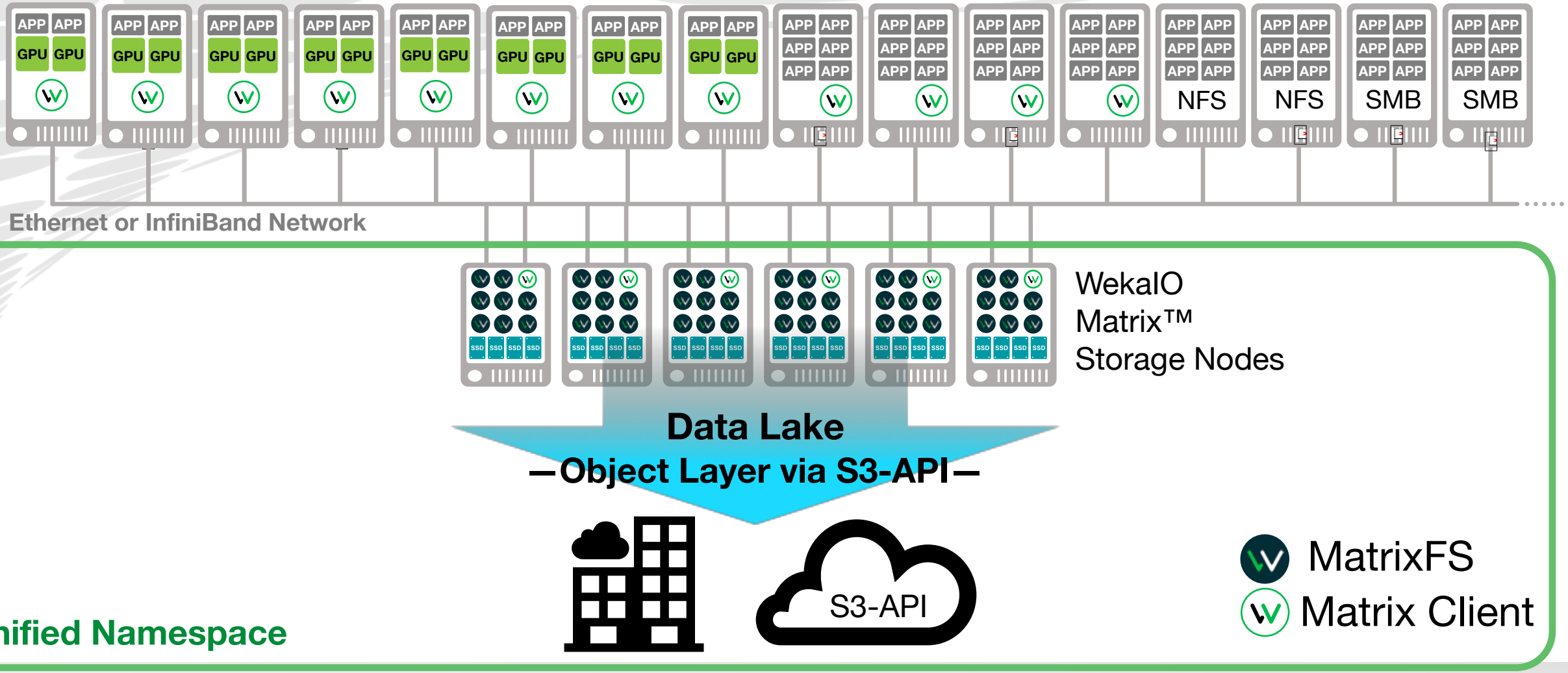
* This WekaIO cluster was about one half-rack of SuperMicro “Big Twin”

** The Oak Ridge “Summit” supercomputer system includes about 40 racks of IBM ESS

WEKA.IO

Data Path

Big Picture

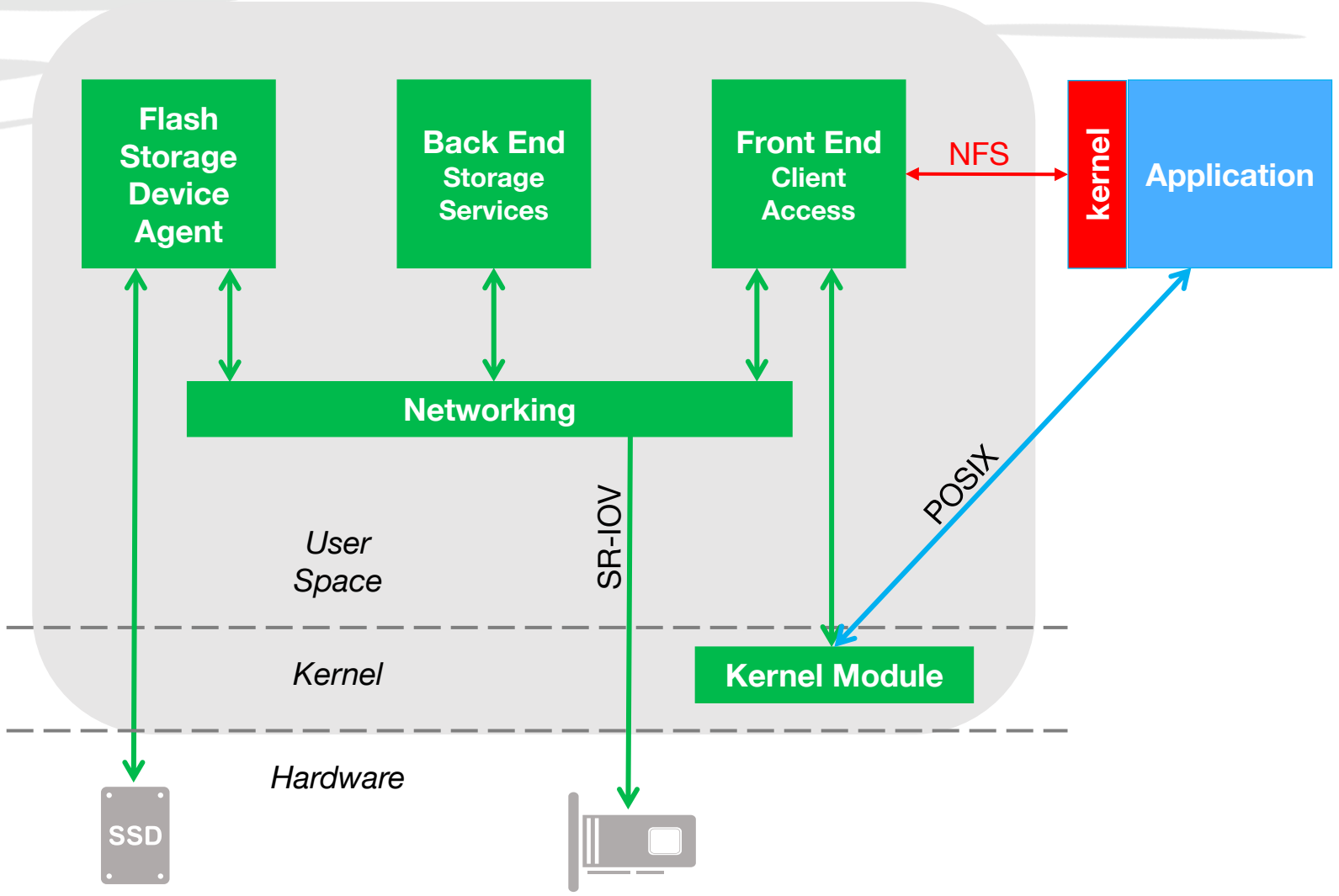


WekaIO Data Path

- Application IO (file operations)
  - Access WekaIO Client *as* Local FS
  - User-Space, Low-Latency
  - POSIX-complete, high-perf
  - Kernel Module for VFS integration

OR

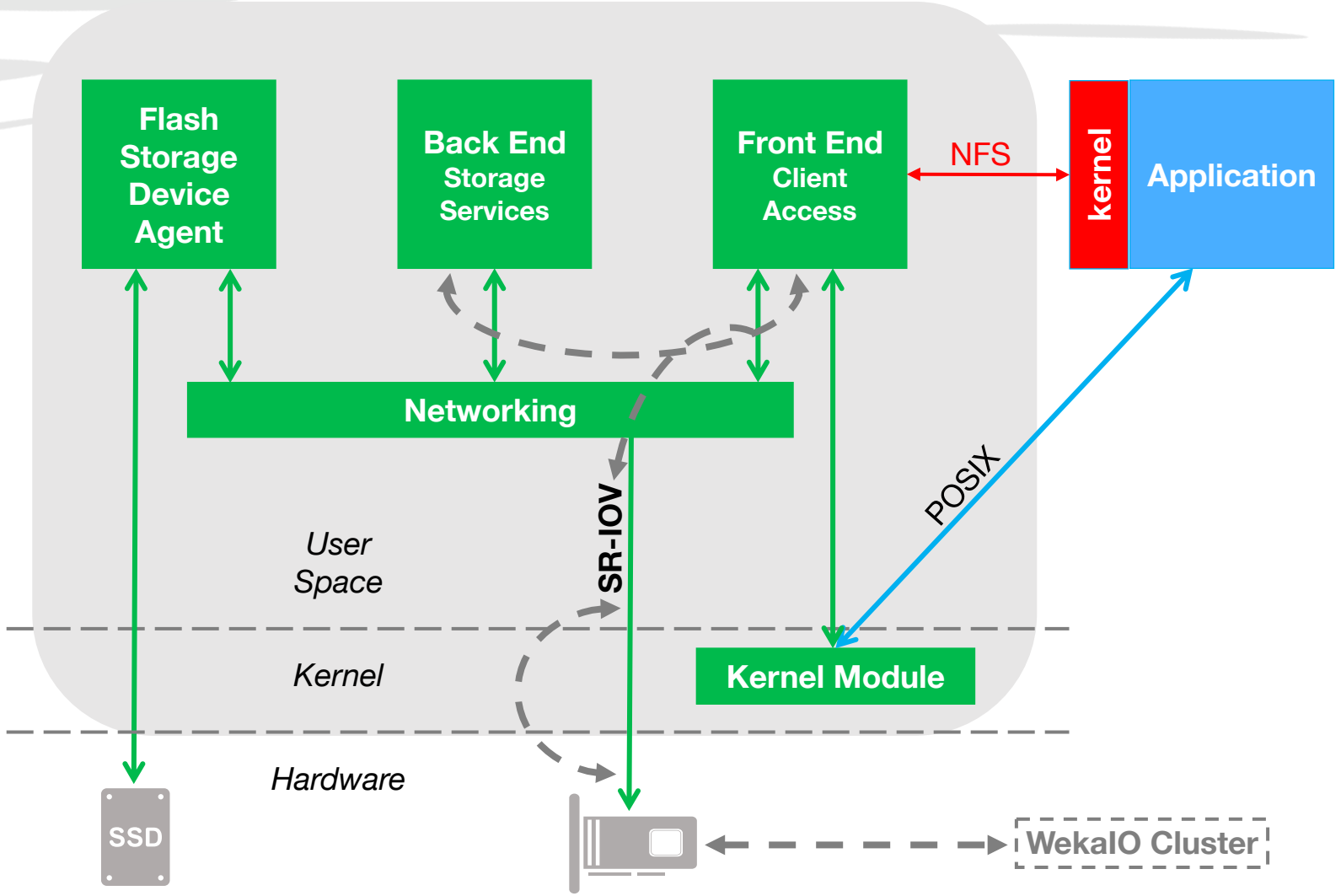
  - Client-side NFS
  - Bottlenecked by Kernel
  - Handled by WekaIO's Front End



WekaIO Data Path

- Application IO (file operations)
  - Access WekaIO Client *as* Local FS
  - User-Space, Low-Latency
  - POSIX-complete, high-perf
  - Kernel Module for VFS integration

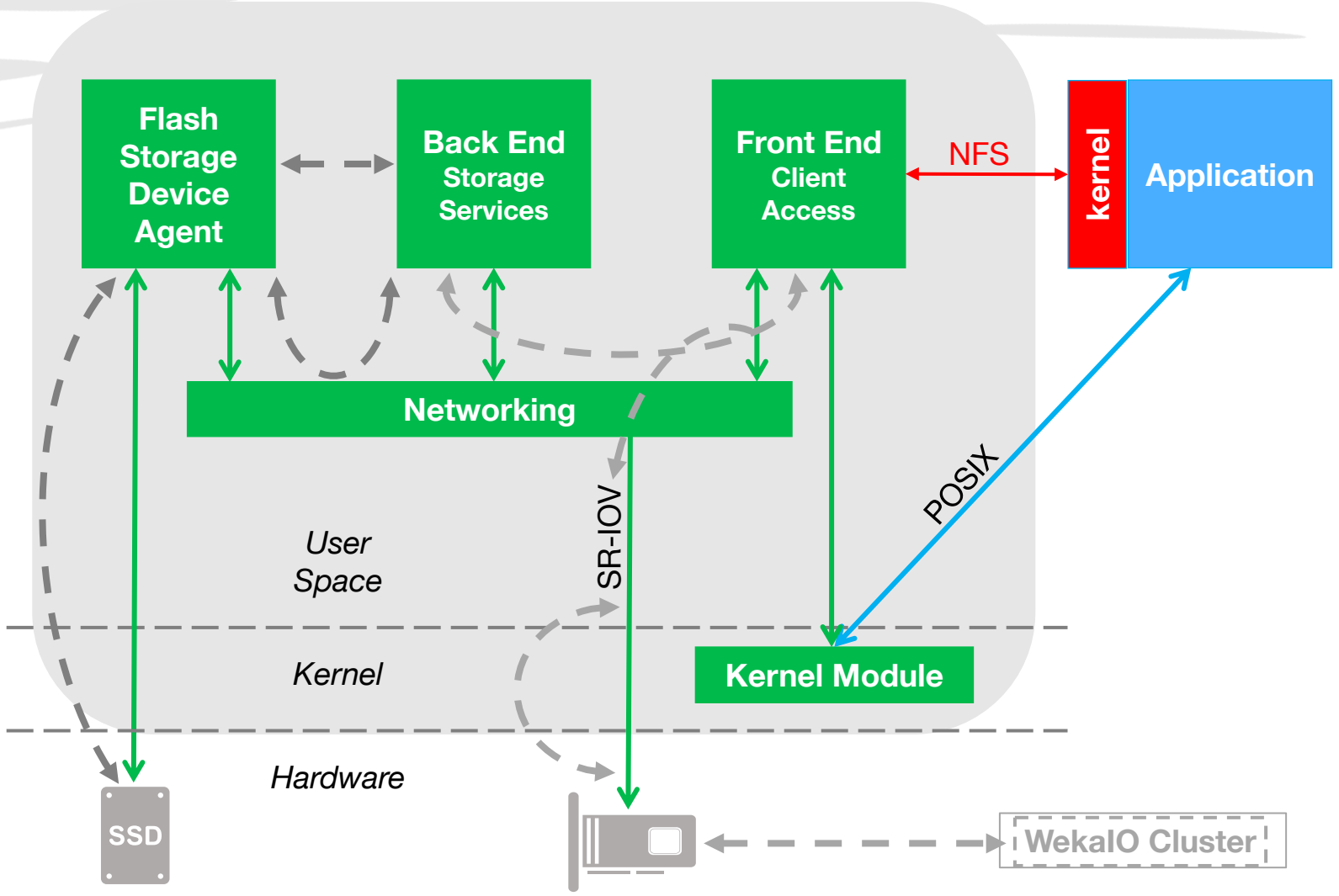
  OR
  - Client-side NFS
  - Bottlenecked by kernel
  - Handled by WekaIO's Front End
- WekaIO Front-Ends are Cluster-Aware
  - Incoming Read Requests optimized re Location & Loading Conditions
  - Incoming Writes can go anywhere
  - Metadata fully distributed
  - No redirects required
- SR-IOV optimizes Network access



WekaIO Data Path

- Application IO (file operations)
  - Access WekaIO Client *as* Local FS
  - User-Space, Low-Latency
  - POSIX-complete, high-perf
  - Kernel Module for VFS integration

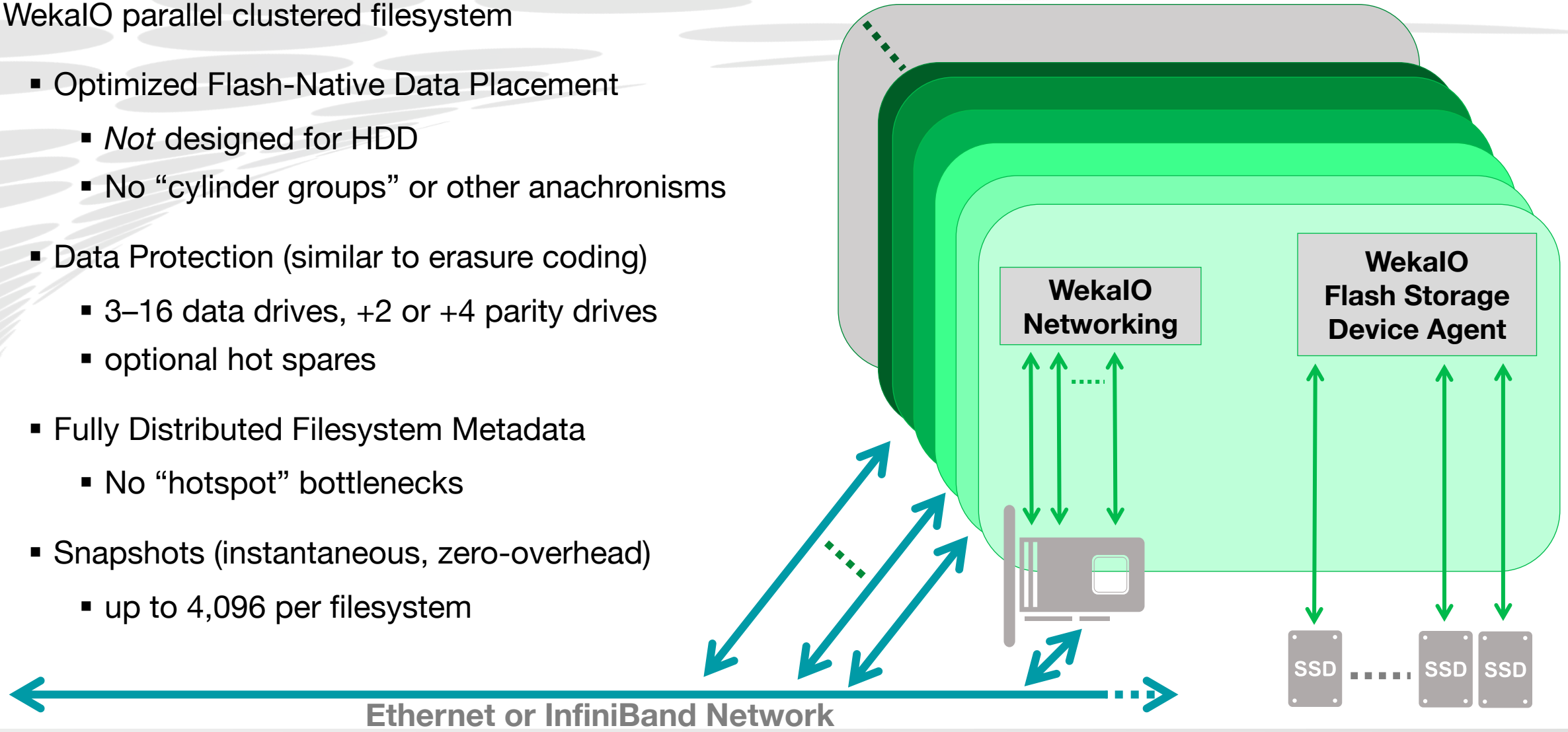
  OR
  - Client-side NFS
  - Bottlenecked by kernel
  - Handled by WekaIO's Front End
- WekaIO Front-Ends are Cluster-Aware
  - Incoming Read Requests optimized re Location & Loading Conditions
  - Incoming Writes can go anywhere
  - Metadata fully distributed
  - No redirects required
- SR-IOV optimizes Network access
- WekaIO directly accesses NVMe flash
  - Bypassing kernel, better perf



WekaIO Back End Storage Services (1 of 2)

WekaIO parallel clustered filesystem

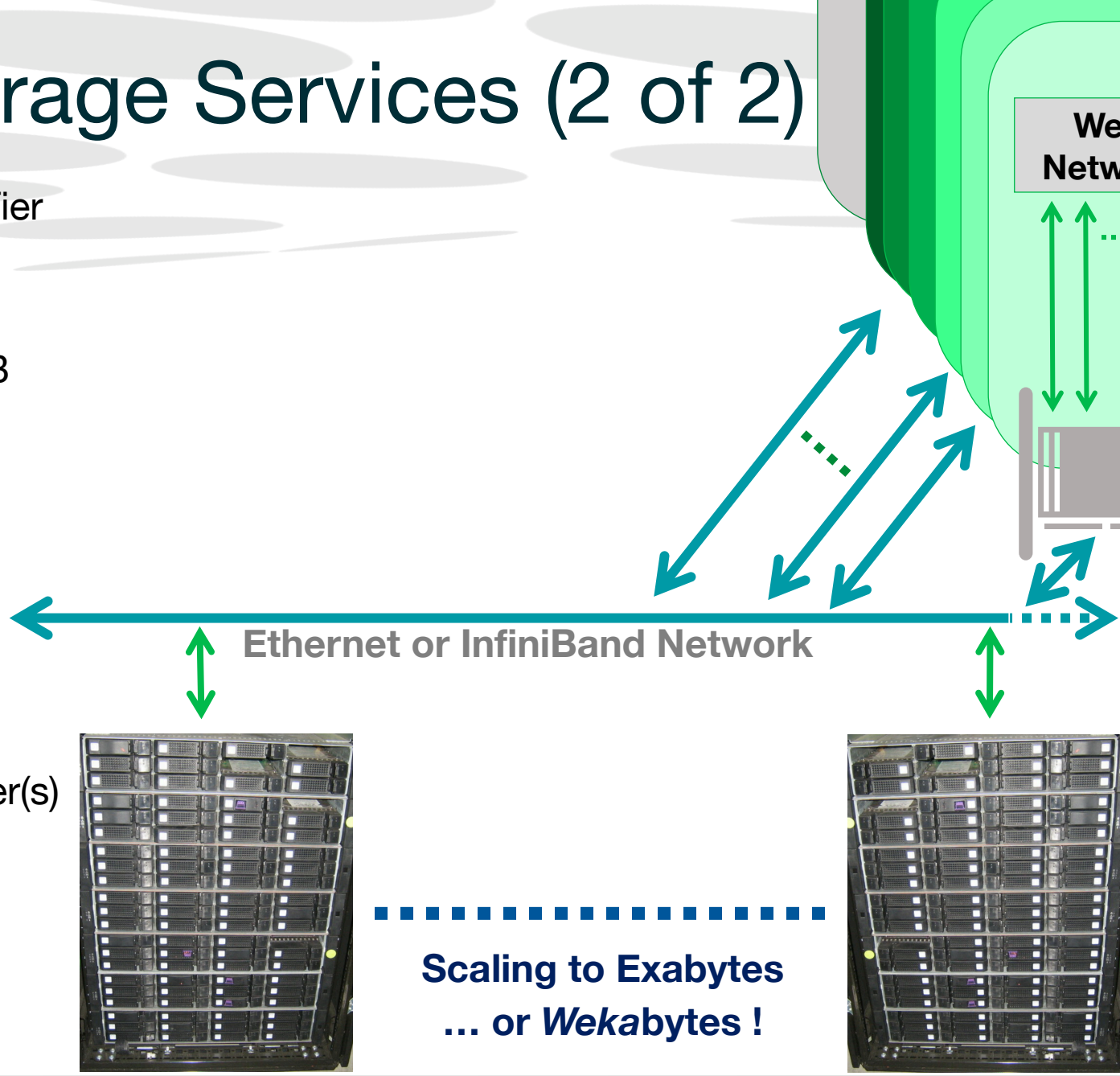
- Optimized Flash-Native Data Placement
  - *Not* designed for HDD
  - No “cylinder groups” or other anachronisms
- Data Protection (similar to erasure coding)
  - 3–16 data drives, +2 or +4 parity drives
  - optional hot spares
- Fully Distributed Filesystem Metadata
  - No “hotspot” bottlenecks
- Snapshots (instantaneous, zero-overhead)
  - up to 4,096 per filesystem



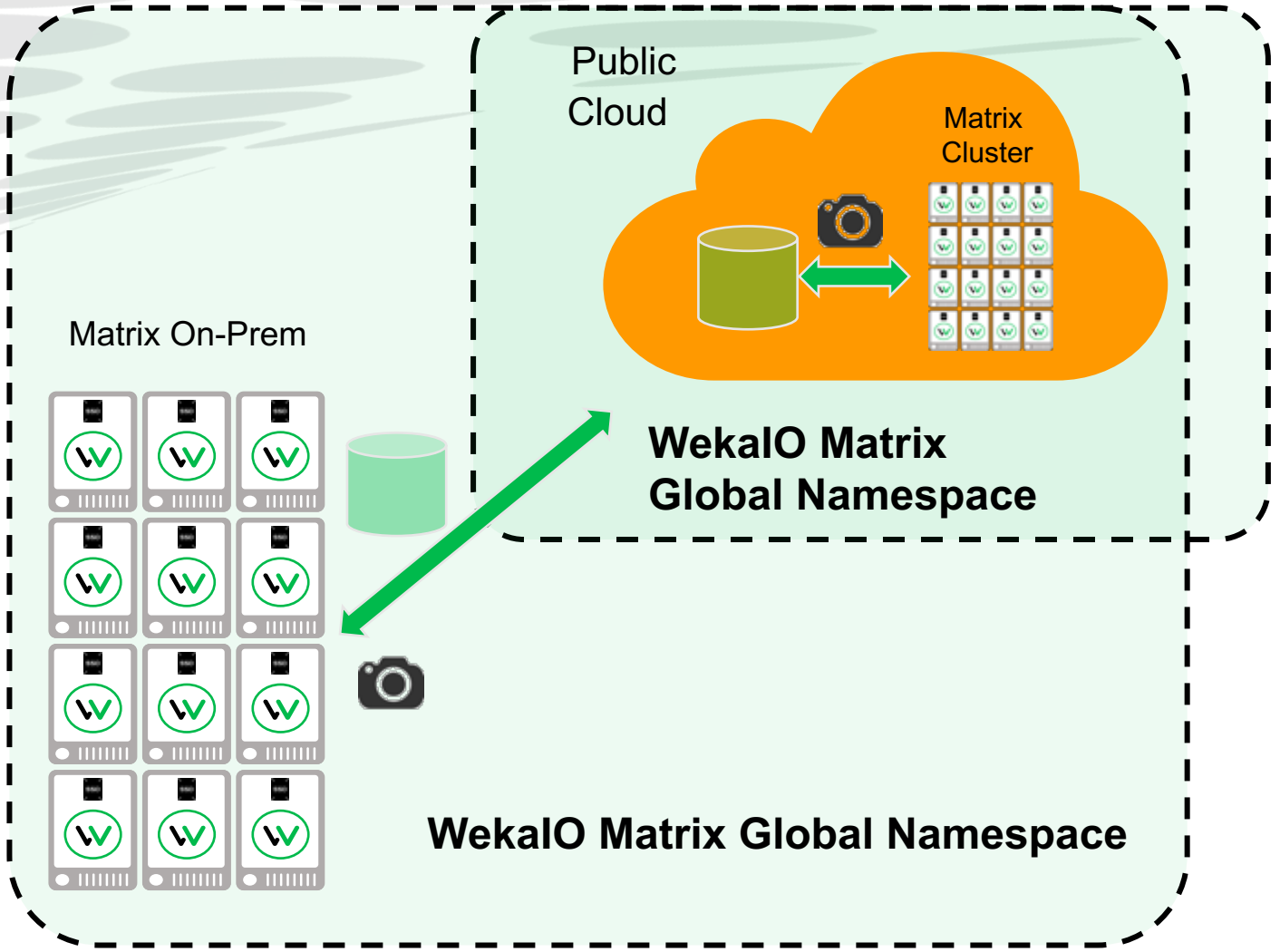
WekaIO Back End Storage Services (2 of 2)

Global Namespace = Hot Tier + Object Storage Tier

- Tiering to S3-API Object Storage
  - Additional capacity with lower cost per GB
  - Files sharded to object storage layer
    - Parallelized Access optimizes perf
    - Simplifies Partial or Offset Reads
- Snap-To-Object
  - Entire Filesystem (incl metadata) captured
  - Can be rehydrated by other WekaIO cluster(s)
  - Used for Backup, DR, Cloud-Bursting



Snapshot-to-S3 for Infrastructure Elasticity & DR

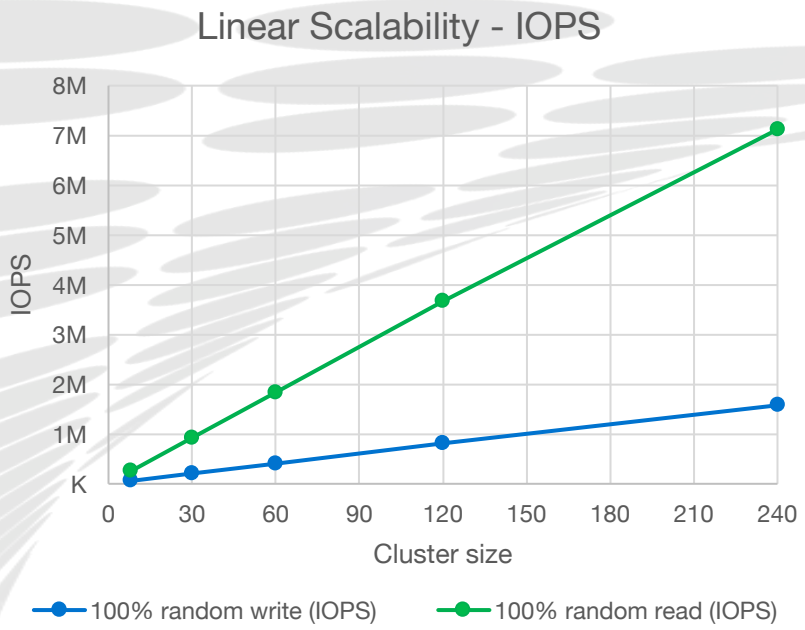


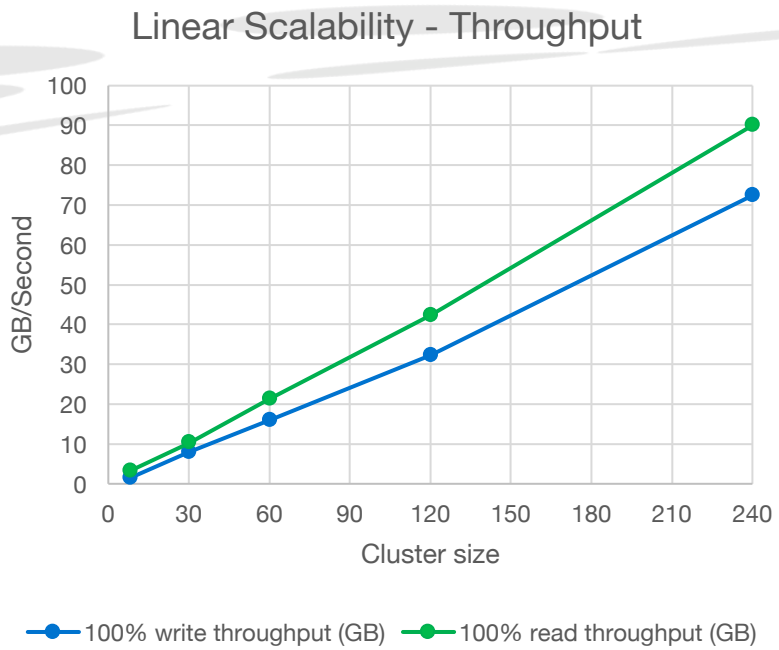
Thank You

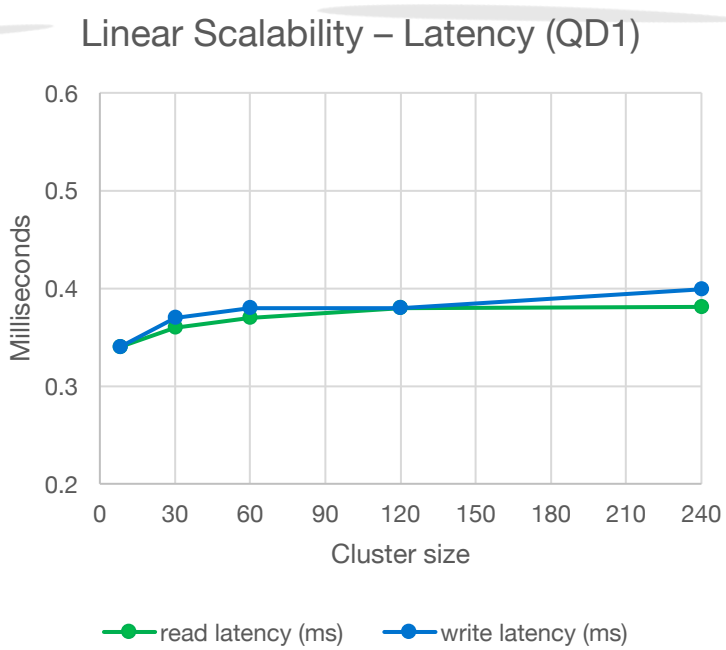
WEKA.IO

Backup Slide(s)

File System Scales **Linearly** with Cluster Size







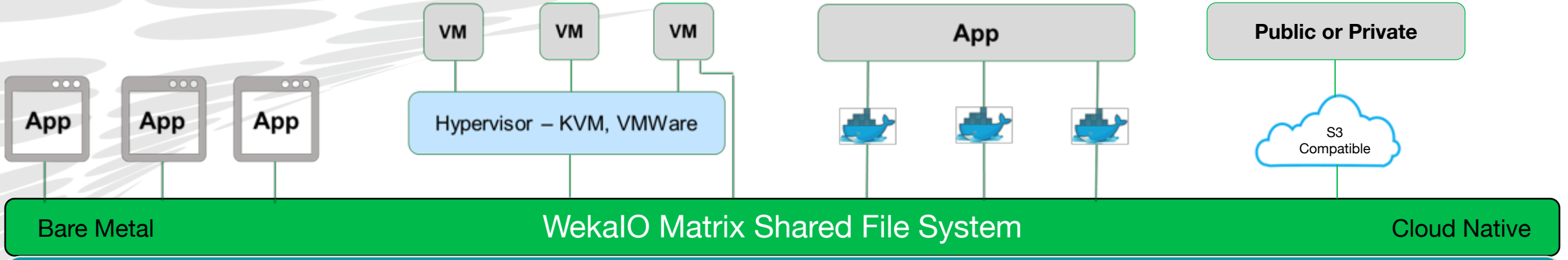
Near-Linear IOPS Scaling for Reads
Write IOPS Scaling Good but could be better (and has improved since these tests were run)

Super-Linear Overall *Throughput* Scaling

~Flat Latency Scaling for Reads
Very Good Scaling for Writes (remember to allow for AWS network fluctuations)

Test Environment – 30-240 R3.8xlarge cluster, 1 AZ, utilizing 2 cores, 2 local SSD drives & 10GB of RAM on each instance. About 5% of CPU/RAM.

WekaIO Matrix: Full-featured and Flexible



Fully Coherent POSIX File System That is Faster than Local File System

Distributed Coding, More Resilient at Scale, Fast Rebuilds, End-to-End DP

Instantaneous Snapshots, Clones, Tiering to S3, Partial File Rehydration

InfiniBand or Ethernet, Converged or Dedicated Storage Server