Flash Memory Summit

Earle F. Philhower, III
Technical Marketing Engineer
Pavilion Data

Stephen Daniel
Distinguished Technologist
HPE

John Kim
Director, Storage Marketing
Mellanox

Molly Presley
Global Product Marketing
Qumulo

Flash Memory Summit

Go Parallel or Go Home
Parallel Storage Architecture for NVMeoF

Earle F. Philhower, III
earle.philhower@paviliondata.io

Flash Memory Summit

Living in a Parallel Dimension

- Compute has become massively parallel
  - How many cores does your phone have?
- Workloads have become massively parallel
  - Cloud-Scale distributed systems run the world
  - SQL Database -> NoSQL Cluster
  - Data Mining -> AI training
- Parallel applications have unique storage needs.

Parallel Application Storage Needs

- Multiple disparate high-perf storage units
  - 10,000 shard DBs instead of 1 large DBF
- Resiliency
  - Application or hardware needs to preserve data
- Reconfigurability
  - Cluster hardware will run many different workloads

Flash Memory Summit

Parallelizing Storage in the Server

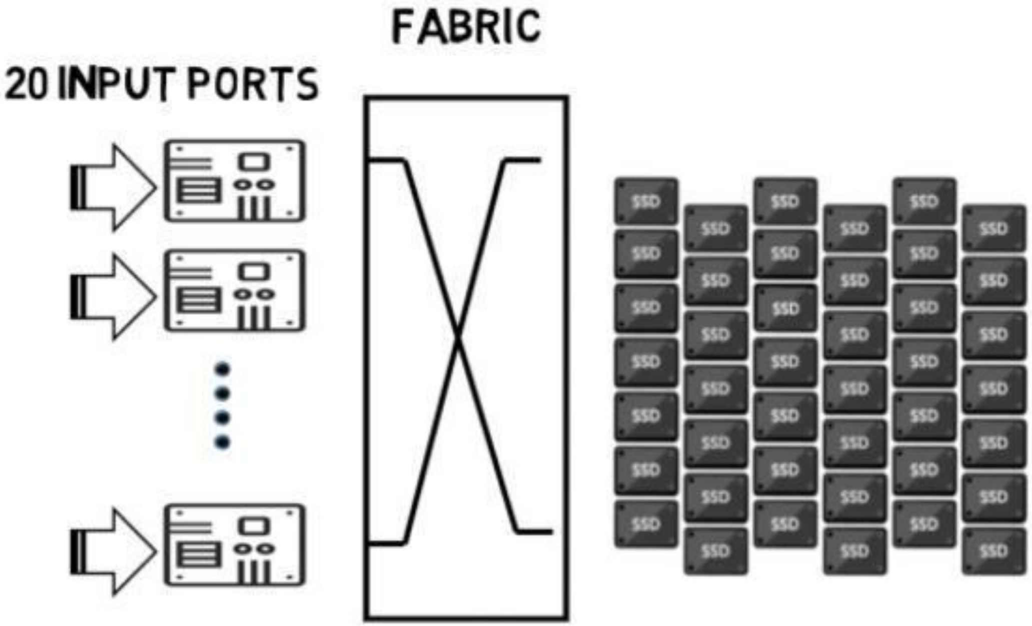
- NVMe drives in server, parallelism across servers
- Replicas for data durability and accessibility
  - 2X-3X the amount of flash , 2x-3x the cost
- Rebuilds over the network from peers
  - Saturates the network, reduces entire application performance
- Wasted storage dollars on overprovisioning
  - Lowest $/bit drives are often too large to be fully used

Flash Memory Summit

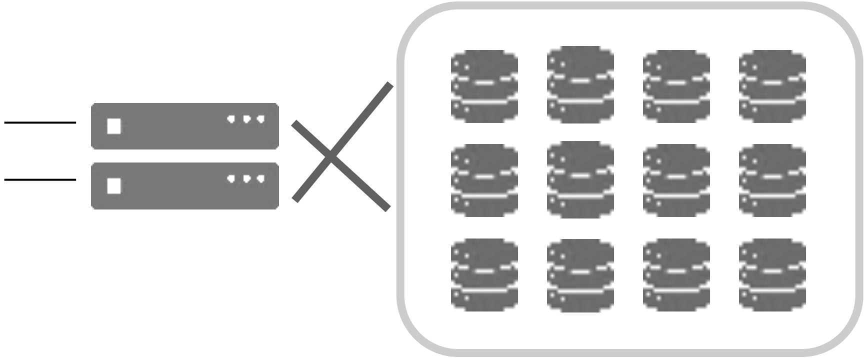
100G Changes the NVMe-oF Equation

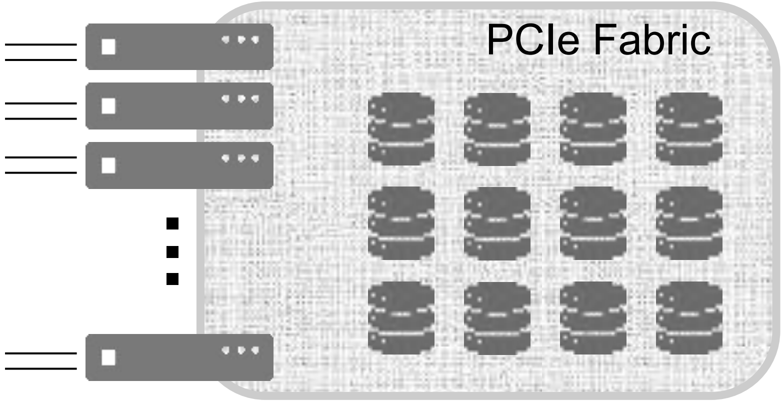
- RDMA + 100G Ethernet perfect for NVMe
  - uS-level additional IO latency
  - 10GB/s+ bandwidth per link
- Requires unique, parallel disaggregated storage architectures

Learning from Modern Networking



Parallel vs. Serial Storage Design





- Single/few head nodes
- Limited uplinks
- JBOF/JBOD + dual-path
- SAS/SATA-centric design

- 10s of line cards
- Multiple 100G per head
- PCIe fabric connectivity
- NVMe-centric design

Flash Memory Summit

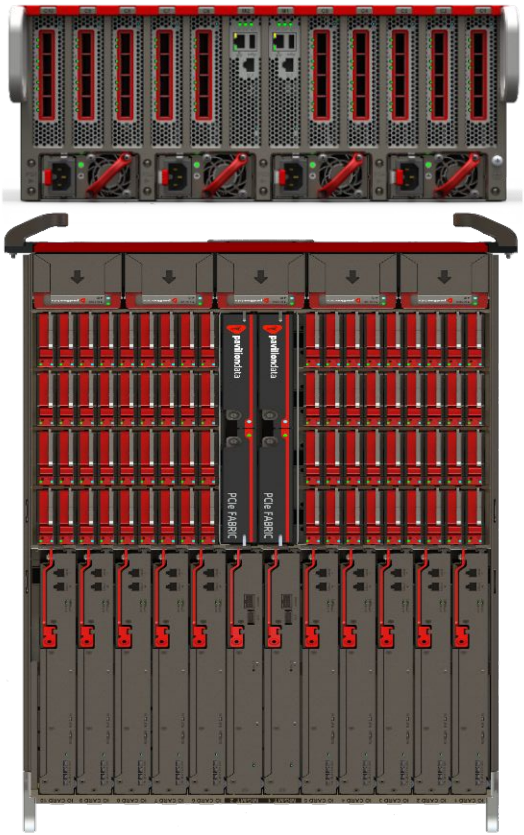
Parallelizing NVMe with NVMe-oF

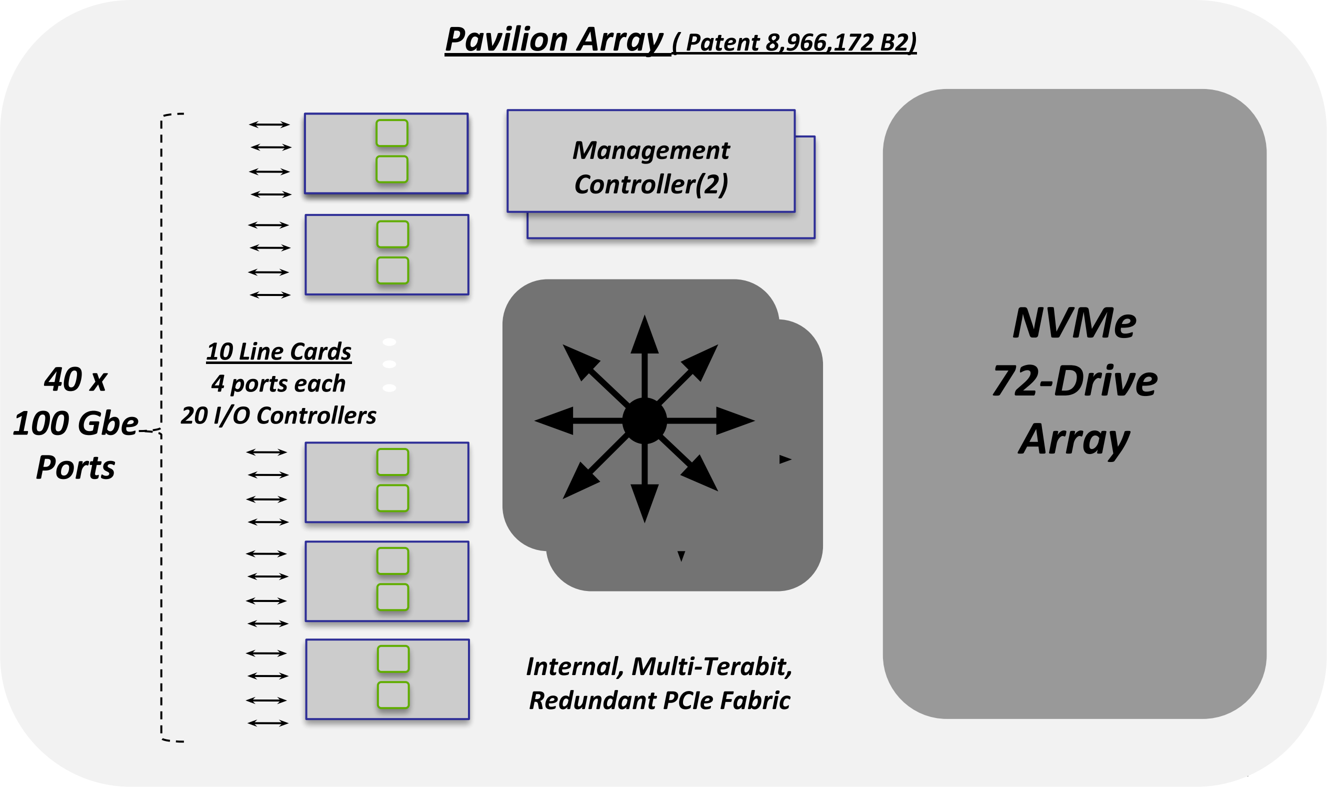
- Large numbers of SSDs == performance
- PCIe fabric helps avoid SPOF
  - Line card fails? Access SSD through another
- Spread perf. hot spots over many SSDs
  - Individual server can use BW/IOPS of many SSDs
- Standards-based, no custom drivers or SDS
  - Easier adoption, less management overhead

Parallel Disaggregated Side Benefits

- Disaggregation allows higher flash utilization
  - Install best $/bit or $/IOP flash, carve multiple LUNs
- RAID-5 or RAID-6 possible over larger stripes
  - Significant space savings vs. RAID-1 mirrors
- Allows single server SKU to perform disparate tasks
  - Simplifies operations, allows software to define the needed hardware via automation (K8s, VM, etc.)

Parallel NVMeoF Array Example





Drawing Parallel Conclusions

- NVMe deserves parallel storage architectures
- Key points for maximum NVMe-oF utility
  - Run many storage head nodes in parallel
  - Think fabric, not point-to-point
  - Multiple, high-speed network links to rack
  - CAPEX/OPEX of parallel disaggregated flash

THANKS

Using Storage Class Memory to Accelerate All Flash Storage: Lessons Learned

Stephen Daniel

Distinguished Technologist, HPE

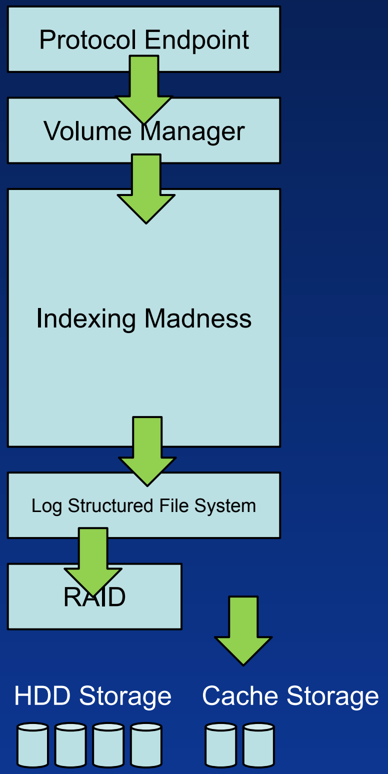
Flash Memory Summit

Project Goals

- Convert a NAND flash All-Flash Array to a hybrid array using Intel Optane as a read cache
- Drive read hit response time close to 100 µSec
- Maximize code reuse from HPE Nimble's SSD/HDD cache code
- Maximize scarce resources by using deduplication and compression in the Optane cache

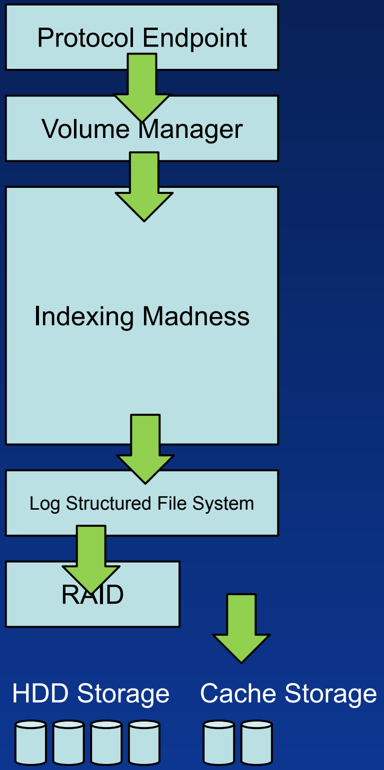
Flash Memory Summit

HPE Nimble Hybrid Array – Read Path



- Indexing:
  - Probe memory cache
  - Probe SSD cache
  - Read from HDD
- Data is deduplicated and compressed in all layers

HPE Nimble Hybrid Array – Read Path



Preliminary measurements:

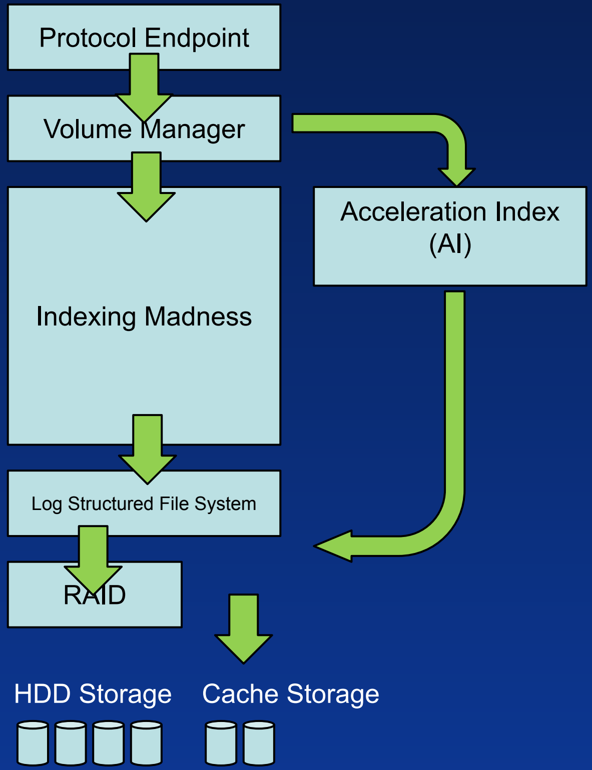
Component	µSec
Host Stack	25
Protocol, HBA, VM	50
Indexing Madness	60
LFS + async I/O stack	35
SSD	150
Total	320

Solution:

Get a fast SSD

Add another index!

HPE Nimble Hybrid Array – Read Path



Preliminary measurements:

- Indexing consumed 60 μSec of CPU time on a cache hit

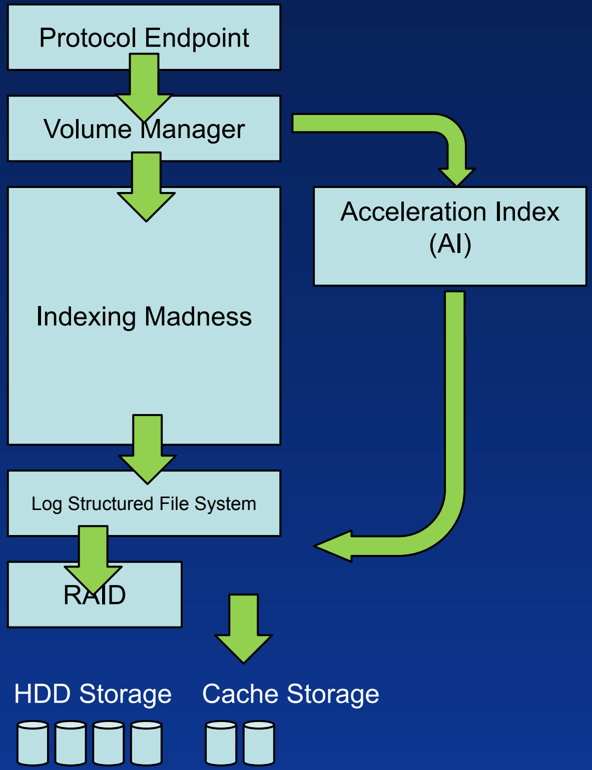
Solution:

- Add Acceleration Index (AI)
- AI is a hash table that maps

  `(vol_id, offset) ⇒ LFS location`

- Goal: Cut at least 50 μSec from CPU path length

HPE Nimble Hybrid Array – Read Path



Estimates at project start:

Component	Baseline µSec	Accelerated µSec
Host Stack	25	25
Protocol, HBA, VM	50	50
Indexing Madness	60	10
LFS + async I/O stack	35	35
SSD	150	15
Total	320	135

Should get us close to goal …

Maintaining the Optane Cache

Data in Optane is managed by existing hybrid cache code

- Using the existing hybrid caching code gives us a cache that is deduplicated and compressed
- Cache blocks are managed by the Cache Index (CI)
- Blocks are inserted on random write or random read that misses cache
- Heat maps and LFS garbage collection evict old data

Flash Memory Summit

Maintaining the Acceleration Index

On AI miss during read:

- Follow normal read path
- If the data is found in the cache, populate the AI
- If the data is not in the cache, read from flash SSD and populate the cache

Maintaining the Acceleration Index

Evicting AI entries:

- AI is a 4-way set-associative cache. An insert may evict the oldest entry in the set
- Eviction from cache (CI) by overwrite triggers AI invalidation
- Eviction by aging and garbage collection will create stale AI entries. These are found when referenced, and evicted

Managing Optane Bandwidth

Optane SSD throughput is media bandwidth limited

- It is possible to run out of Optane bandwidth
- When reading from Optane, if queuing delay would cause excessive latency, bypass Optane and read from flash

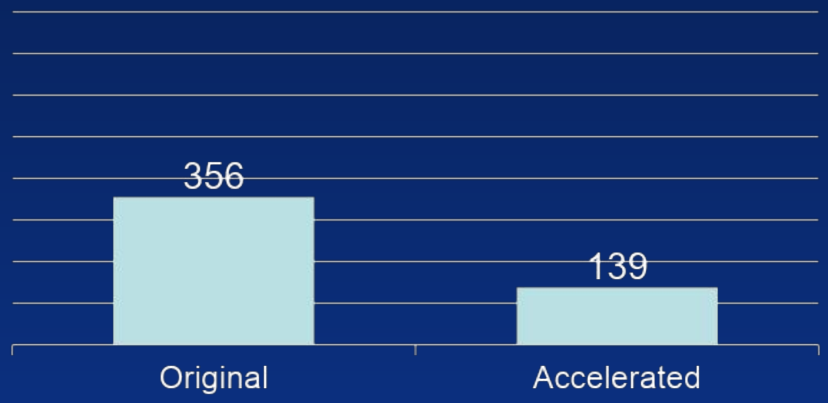
Flash Memory Summit

Other Notes

AI is a fixed sized hash table, initialized at boot

- AI is not persisted to stable storage
- The cache (and associated CI) persists across reboots, the AI does not
- AI entries not deduplicated. High reference-count blocks are cached in memory to prevent AI thrashing

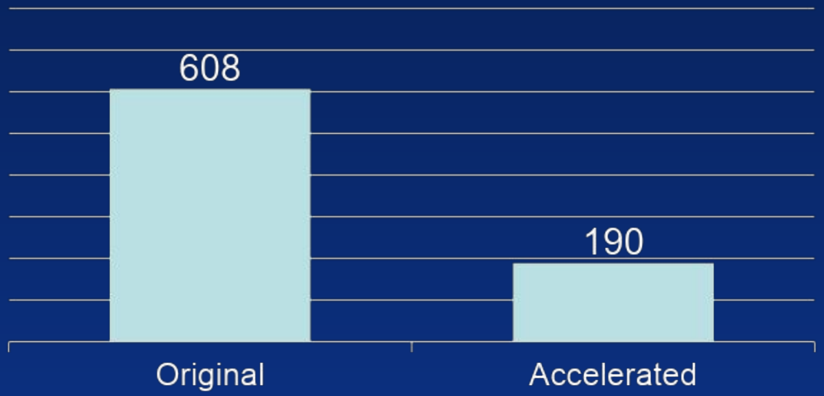
Preliminary Measurements



Measurements with a single read thread on an array capable of about 500,000 IOPS

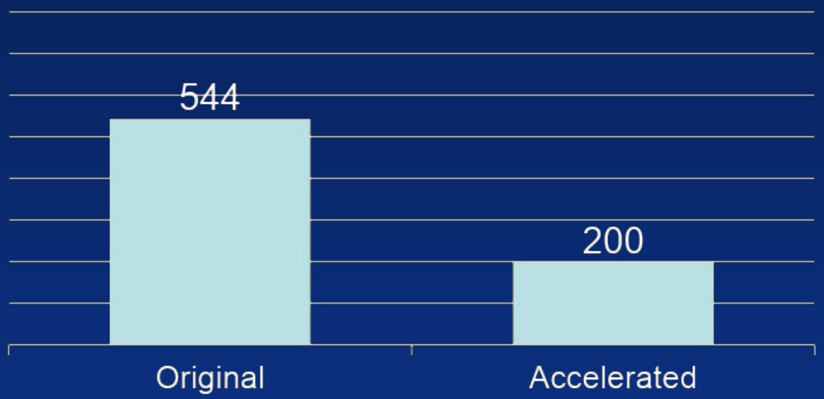
Flash Memory Summit

Preliminary Measurements



Measurements at 150,000 IOPS on an array capable of about 500,000 IOPS

Preliminary Measurements



Measurements at 150,000 IOPS on an array capable of about 500,000 IOPS

Q & A

Stephen Daniel

Stephen.Daniel@HPE.com

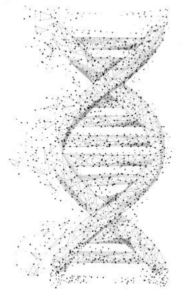
Dramatically Increasing File System Performance with Flash

ENST-102A-1 on Tuesday August 6

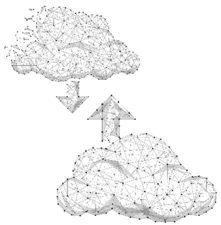
John F. Kim, Mellanox Technologies

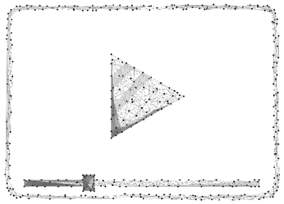
Flash Memory Summit

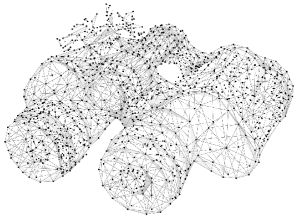
Why Accelerate File Systems?



- Required for many demanding workloads
  - HPC, AI, ML
  - Technical computing
  - High-res video editing or special effects
- Increasingly want/need to use flash





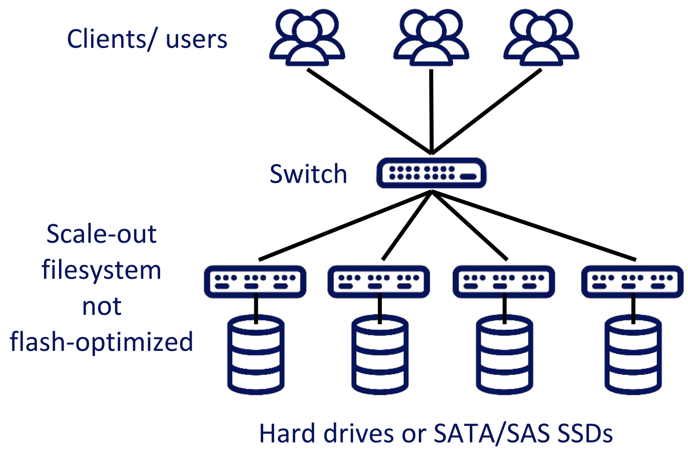


Challenges of with FS

- Many new flash arrays are block-only
- File systems performance overhead
- File storage not optimized for flash
  - File system designed for hard drives
  - NFS over RDMA – limited support
  - SMB Direct in Windows only

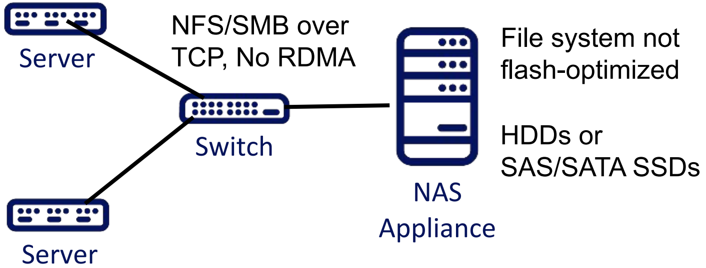
Flash Memory Summit

File Storage, not Flash-Optimized

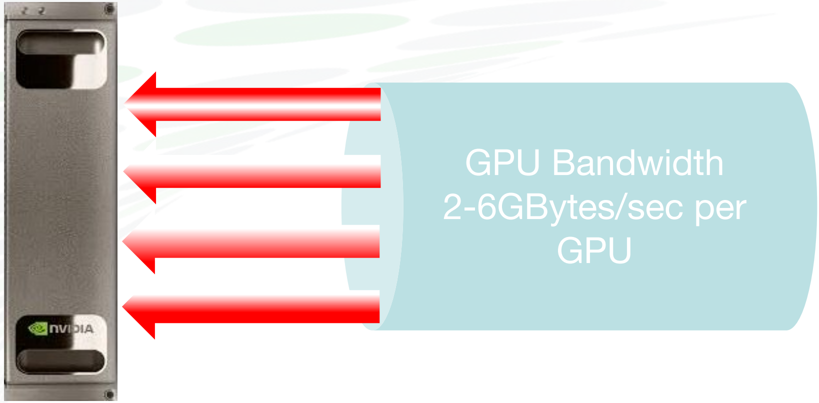


FCP
PCIe
SAS
SRP
iSER
iSCSI
NVMe-oF

High-performance NVMe-oF array:
Block access only



WekaIO: NFS = Not For Speed



NFS Bandwidth
1 - 1.5GBytes/sec

- A protocol invented in 1984 trying to solve a 2019 problem
- pNFS tried to fix NFS but failed when metadata workloads exploded
- Legacy parallel file systems like Lustre and GPFS cannot handle billions of small files

Flash for Files: Four Solutions

- Flash in NAS or in scale-out FS nodes
- NVMe arrays behind scale-out FS
- Updated file system
- Alternatives to a file system
  - Object storage
  - Hyperconverged infrastructure
  - Cloud

Flash in the NAS

- Put SSDs inside NAS appliances
- Examples:
  - NetApp AFF, Dell EMC Isilon, Oracle ZFS
- Question: Is it fully optimized?
  - Filesystem might not be optimized
  - Storage protocols lack RDMA

NVMe Arrays Behind Scale-out FS

- Put NVMe/NVMe-oF arrays behind FS
  - Examples: Lustre, IBM Spectrum Scale
  - High performance demonstrations with SPEC SFS2014
- Examples
  - DDN + Lustre/IBM SS; E8 + IBM Spectrum Scale
  - IBM FlashSystem + Spectrum Scale; Excelero+Lustre
- Question: Is it fully optimized?

New Scale-out File System

- File system optimized for flash
  - Billions of files, scalable metadata
  - Optimized networking, faster connections, RDMA?
- Examples
  - Qumulo, Weka-io, Pure FlashBlade
  - Updates to Lustre, IBM SS, NetApp, Isilon?

Flash Memory Summit

Qumulo Scale-out File System

Modern Scale-out File Systems

Built for **petabyte scale**

Handles billions of **files** both **large & small**

Never lose data
Full stop. No excuses

Delivers **highest performance** across many dimensions

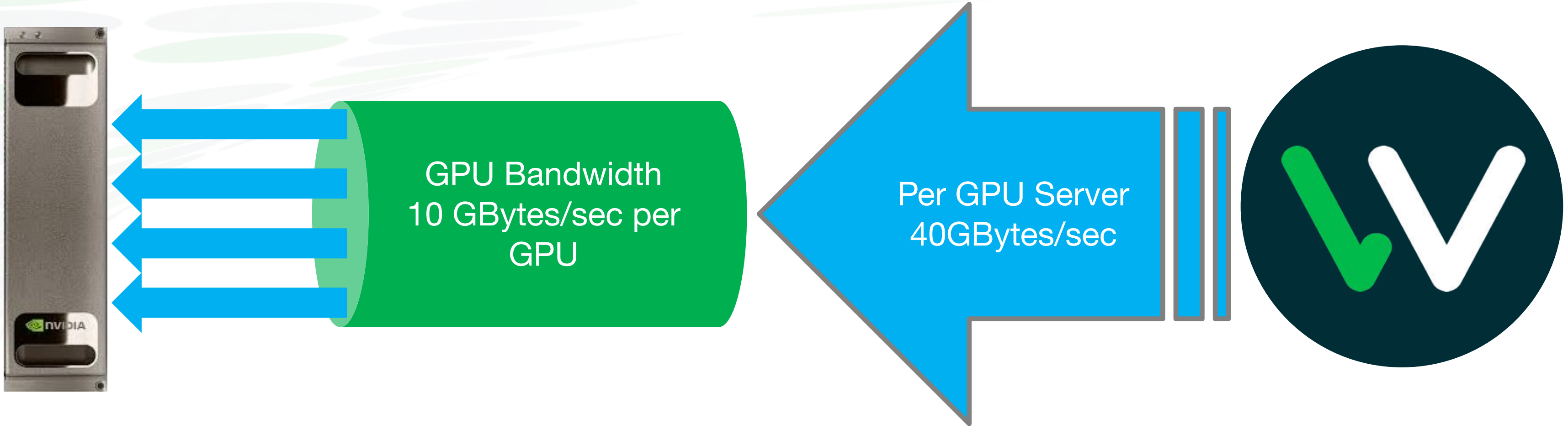
Hardware freedom
On-premise, cloud, platform of choice

Works with your existing applications & is **completely programmable**

Costs less & more efficient than legacy storage appliances

Qumulo

WekaIO Parallel File System



- Parallel file system written for NVMe, and modern networks
- Faster than local NVMe drives
- A file system as scalable as object storage

Or… No more File System

- High-speed object store; Key-value SSDs
- Hyperconverged; Computational storage
- Examples
  - Min.io (object)
  - Key-value SSDs
  - Nutanix, Microsoft S2D, Cohesity, Pivot3
  - Eideticom, Pliops, NGD, ScaleFlux, Samsung, etc.

Summary—Flash-Optimized Files

- Faster flash in the NAS
- NVMe/NVMe-oF behind the SOFS
- New/better file systems
- Alternatives to file storage
- Faster network connections

Flash Memory Summit

Thank You

Flash Memory Summit

Earle F. Philhower, III
Technical Marketing Engineer
Pavilion Data

Stephen Daniel
Distinguished Technologist
HPE

John Kim
Director, Storage Marketing
Mellanox