



Flash Memory Summit

NVMe-over-Fabrics: Enabling Next Generation Infrastructure

David L. Black, Ph.D.
Senior Distinguished Engineer
Dell EMC



About NVM Express

- NVM Express (NVMe™) is an open collection of standards and information to fully expose the benefits of non-volatile memory in all types of computing environments from mobile to data center.
- NVMe™ is designed from the ground up to deliver high bandwidth and low latency storage access for current and future NVM technologies.

NVM Express Base Specification

The register interface and command set for PCI Express attached storage with industry standard software available for numerous operating systems. NVMe™ is widely considered the defacto industry standard for PCIe SSDs.

NVM Express Management Interface (NVMe-MI™) Specification

The command set and architecture for out of band management of NVM Express storage (i.e., discovering, monitoring, and updating NVMe™ devices using a BMC).

NVM Express Over Fabrics (NVMe-oF™) Specification

The extension to NVM Express that enables tunneling the NVM Express command set over additional transports beyond PCIe. NVMe over Fabrics™ extends the benefits of efficient storage architecture at scale in the world's largest data centers by allowing the same protocol to extend over various networked interfaces.



What is NVM Express (NVMe)?

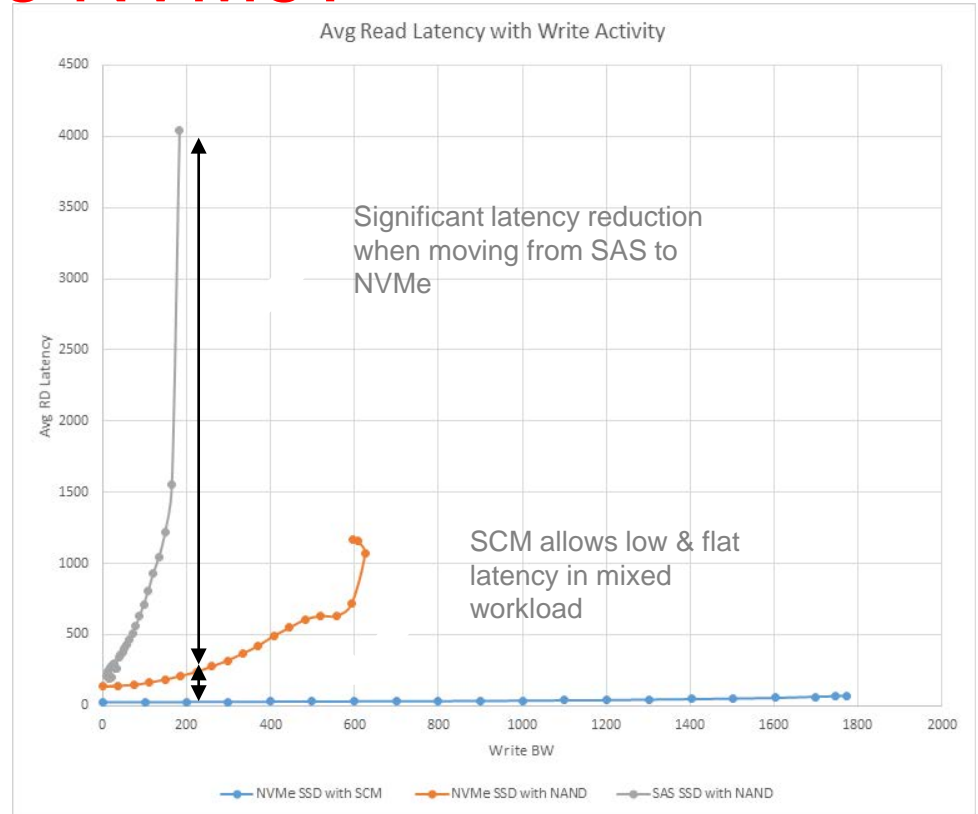
- Specification for SSD access via PCI Express (PCIe), initially flash media
 - Extended to Fabrics (e.g., InfiniBand, RDMA/Ethernet)
 - Designed to scale to any type of Non Volatile Memory, including Storage Class Memory
- Design target: high parallelism and low latency SSD access
 - Does not rely on SCSI (SAS/FC) or ATA (SATA) interfaces: New host drivers & I/O stacks
 - Common interface for Enterprise & Client drives/systems: Reuse & leverage investments
- New modern command set
 - 64-byte commands (vs. typical 16 bytes for SCSI)
 - Administrative vs. I/O command separation (control path vs. data path)
 - Small set of commands: Small fast host and storage implementations
- NVMe SSD support in all major operating systems






Why Move to NVMe?

- Parallelism fits multi core CPUs
 - Also reduces host CPU load
- Exploits low latency media
 - 3D XPoint, Z NAND, etc.
- Remove extra cost
 - SAS controller + expander
- Storage system benefits:
 - Lower average latency
 - Lower tail latency (consistency)
 - Higher bandwidth
- NVMe-over-Fabrics motivation: Extend these benefits end-to-end
 - Remove protocol translation overheads





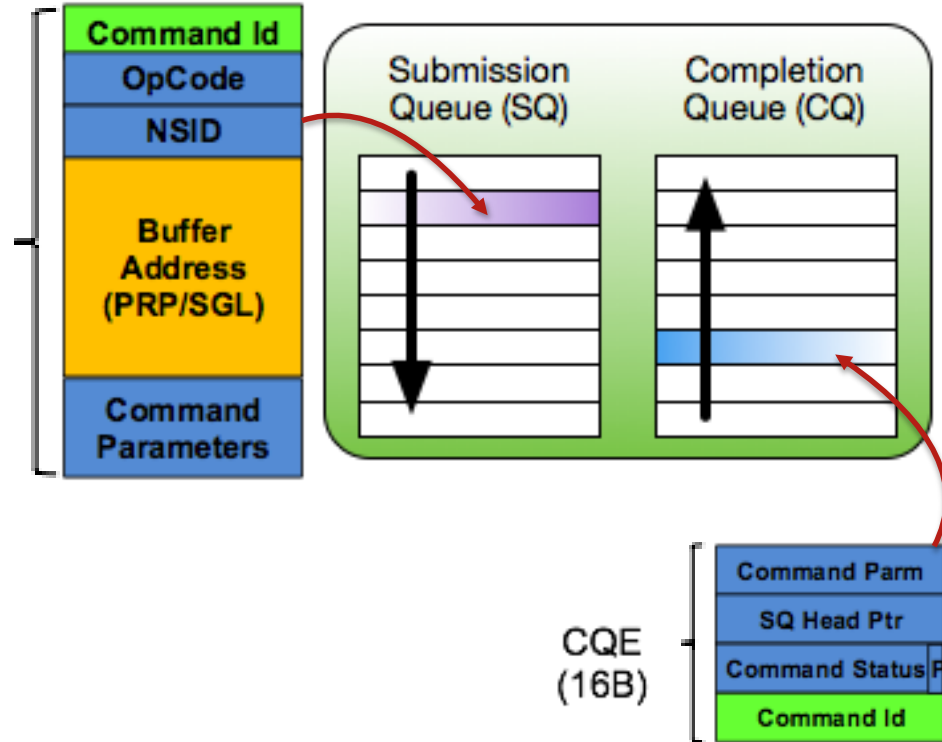
Talk Outline

- NVMe Background 
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports
 - RDMA-based, Fibre Channel, TCP
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers
- I/O Path Performance



NVMe Commands and Completions

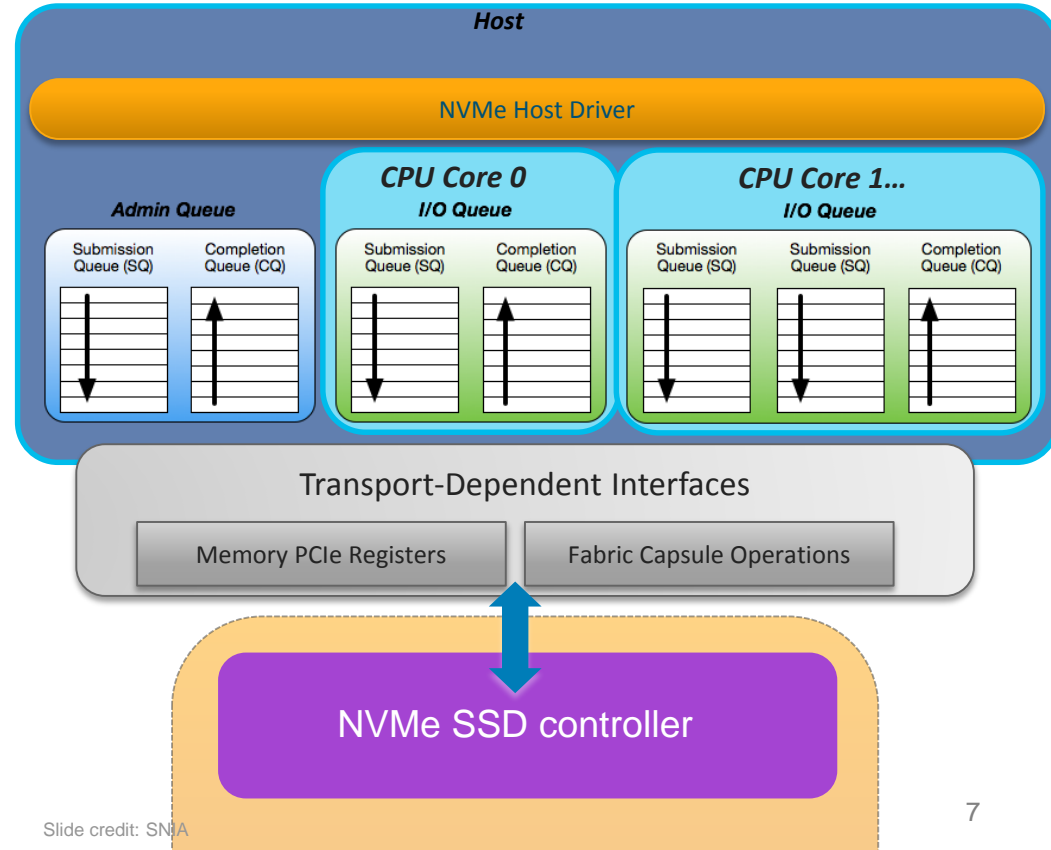
- Submission Queue (SQ):
 - Host to SSD Commands
 - 64B Submission Queue Entry (SQE)
 - Admin and IO Commands **SQE (64B)**
 - Separate Admin vs. IO queues
 - Enables IO path optimization
- Completion Queue (CQ)
 - SSD to Host Completions
 - 16B Completion Queue Entry (CQE)
 - Out of order completion
 - Completion interface optimized for success (common case)





NVMe Multi-Queue Interface

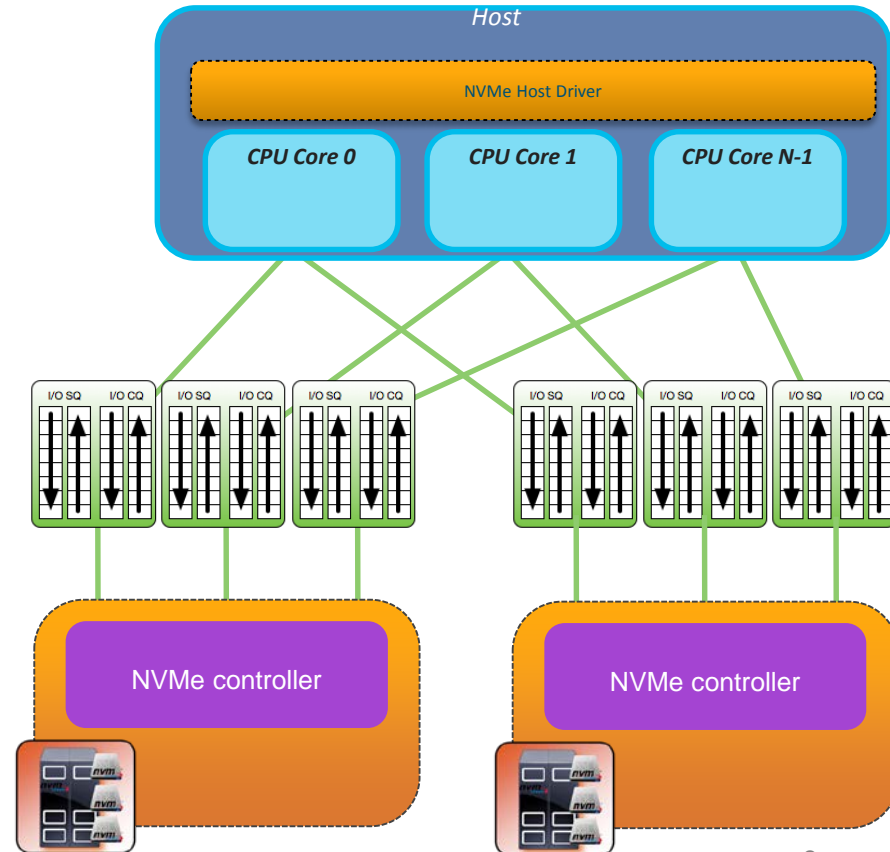
- Memory-based deep queues
 - Up to 64k queues, 64k commands/queue
 - Typical product limits are smaller
- I/O Submission and Completion Queue Pairs can be aligned to Host CPU Cores
 - Independent per-queue operations
 - No inter-CPU locks on command Submission or Completion
 - Per Completion Queue Interrupts enables source core interrupt steering





Queues Scale With Controllers

- Each Host/Controller pair may have its own set of NVMe queues
 - Controllers and queues operate independently
- NVMe Controllers may be local PCIe or remote Fabric
 - Common NVMe Commands and Queuing Model





Talk Outline

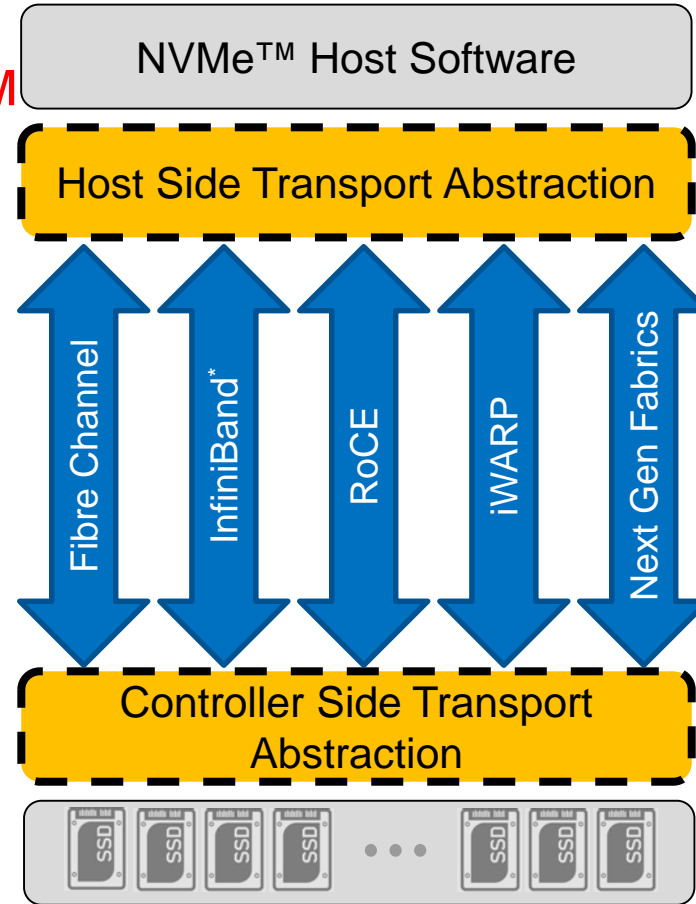
- NVMe Background
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports
 - RDMA-based, Fibre Channel, TCP
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers
- I/O Path Performance





NVMe over Fabrics™

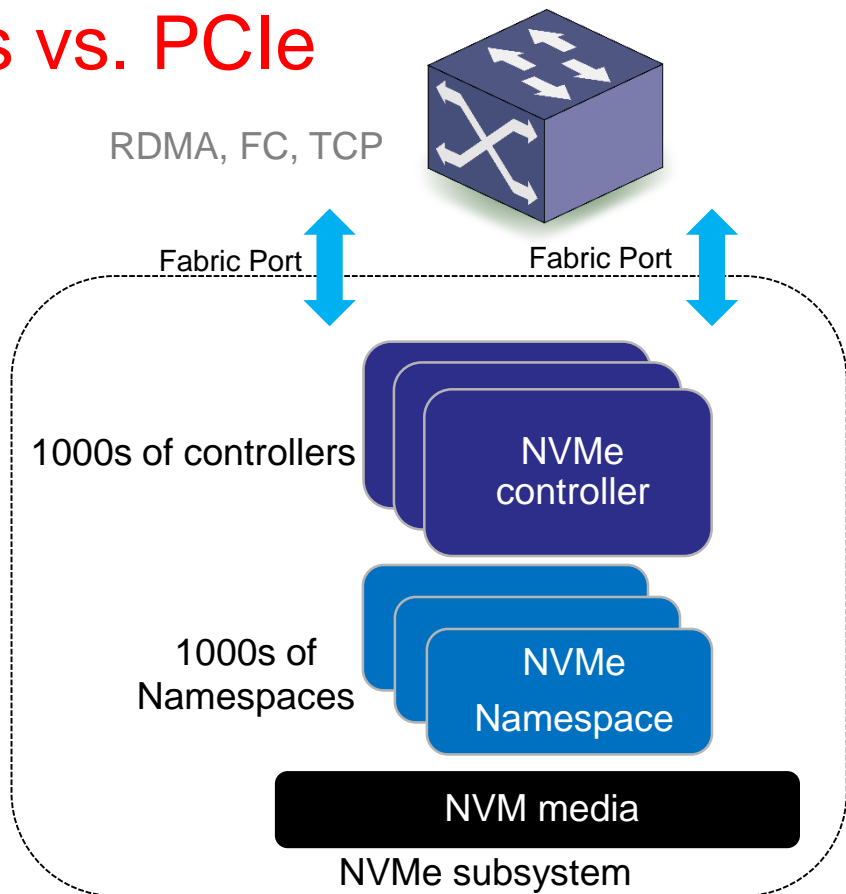
- Use NVMe™ end-to-end to get the simplicity, efficiency and low latency benefits
- NVMe over Fabrics™ is a thin encapsulation of the base NVMe™ protocol across a fabric
 - No translation to another protocol (i.e. SCSI)
- NVMe-oF™ Fabrics v1.0 includes RDMA-based transports and Fibre Channel
 - RDMA-based, e.g. InfiniBand™, RoCE, iWARP





NVMe over Fabrics vs. PCIe

- Extend NVMe subsystem scale & reach
- All I/O commands are the same
 - Some differences in Admin commands
- Architecture independent of specific fabric
- Primary changes from PCIe:
 - Discovery controller replaces PCIe enumeration
 - Queues: Fabric messages replace shared memory
 - Commands & completions use messages (capsules), instead of shared memory
 - Fabric-specific data transfer mechanisms
- NVMe over Fabrics performance goal:
 - ~10us or less added latency vs. local PCIe






NVMe-oF Ecosystem

- Context: High-availability high-reliability storage systems
 - Customer Requirements: No single point of failure, just don't lose the data
- NVMe-oF functionality maturing, some more work needed, e.g.:
 - Asymmetric Multipathing (ANA): New functionality, needs “soak time” experience
 - Cross-subsystem multipathing (Dispersed Namespaces): Work in progress
 - Use cases: Online storage migration, active-active storage replication
 - Persistent Reservations (for HA server clusters): Not yet battle-tested
 - Similar to SCSI, with some extensions and improvements, every corner case matters!
 - In-band authentication (for non-embedded fabrics): Work in progress
- Host Operating System Support: Primarily Linux
 - Additional OS/Hypervisor support – work-in-progress



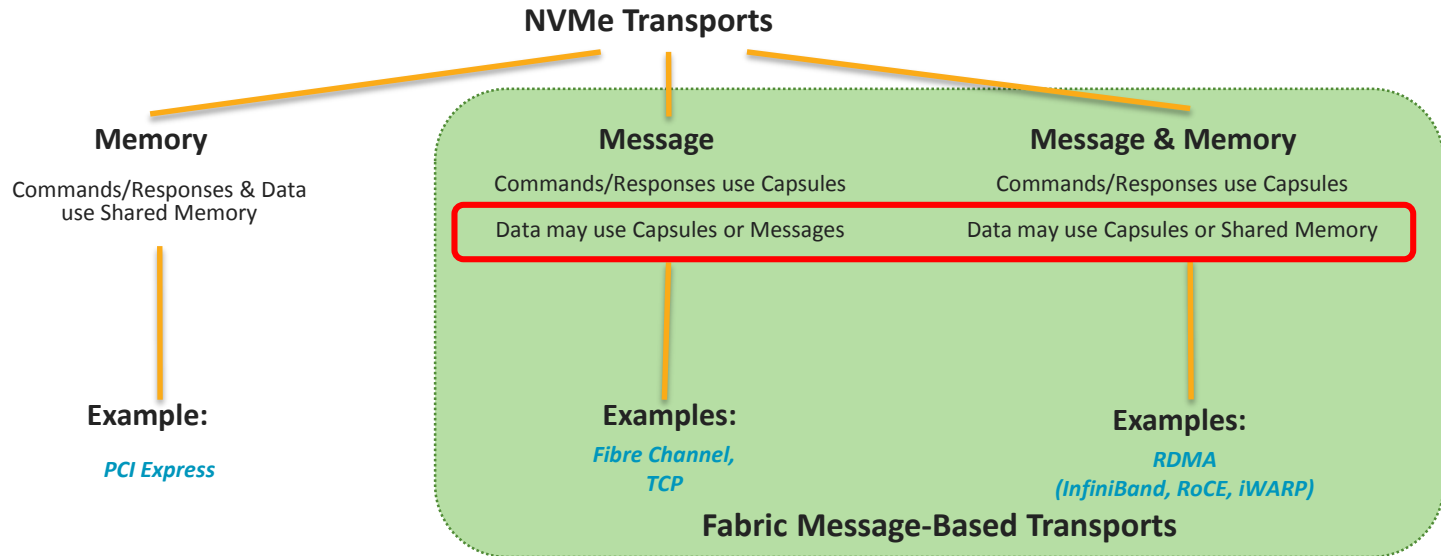
Talk Outline

- NVMe Background
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports 
 - RDMA-based, Fibre Channel, TCP
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers
- I/O Path Performance



NVMe Transport Models

- NVMe is a Memory-Mapped, PCIe Model
- Fabrics is message-based, shared memory is optional



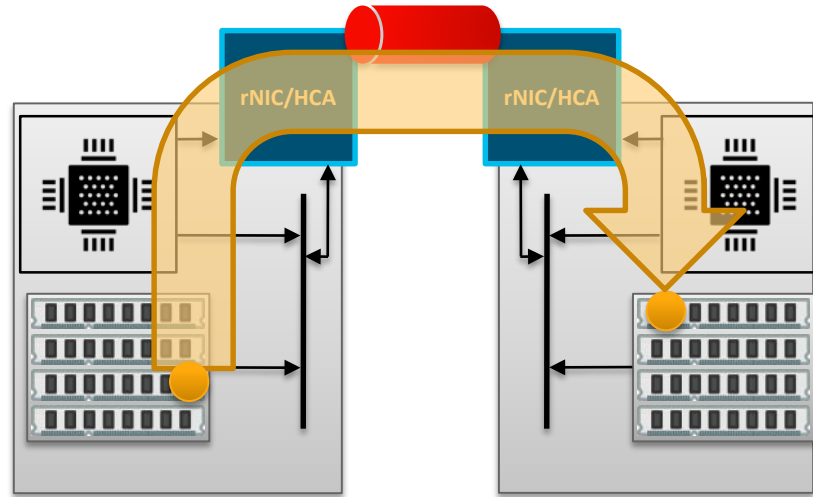
Capsule = Encapsulated NVMe Command/Completion within a transport Message
Data = Transport data exchange mechanism (if any)

Slide credit: NVM Express and SNIA



What is Remote Direct Memory Access (RDMA)?

- CPU-offload and bypass technology:
 - Low-latency messaging
 - Direct memory-to-memory data transfers
- Hardware handles the details
 - RDMA NICs (RNICs) or HCA adapters
 - Reliable source-destination connections
- Common application API (RDMA Verbs)
 - Queue Pairs (QPs)
 - QP = Send Queue (SQ) + Receive Queue (RQ)
 - Completion Queues (CQs)
 - CQ can be associated with multiple SQs or RQs





RDMA: Non-Ethernet Fabrics

- InfiniBand: Native RDMA functionality
 - Embedded environments, e.g., inside storage systems
 - High Performance Computing (HPC)
 - High Frequency Trading (HFT)
 - Full NVMe over Fabrics support
- Omni-Path: HPC-focused
 - NVMe over Fabrics works, support not complete in spec
 - Support is relatively easy to complete
- Speeds up to 100 Gb/sec supported by both fabrics



RDMA: Ethernet Fabrics

- RDMA Protocols for Ethernet (speeds up to 100 Gb/sec)
 - RDMA usage can be shared with other uses of Ethernet
 - Ecosystem maturing, configuration and management still need attention
- RoCE: InfiniBand RDMA protocol on Ethernet
 - Industry Focus: RoCEv2, based on UDP/IP, hence routable
 - Ignore RoCEv1, IP not used, not routable, not interesting ...
 - Requires “lossless” DCB Ethernet (DCB = Data Center Bridging)
 - Enhanced Transmission Selection, Priority Flow Control and DCBX config protocol
 - Not enabled by default: Careful NIC and network switch configuration required
- iWARP: TCP/IP-based RDMA, hence routable
 - TCP provides excellent robustness to “stupid network tricks”
 - Simpler network switch configuration than RoCE



RDMA and Ethernet

- RoCE and iWARP: Hardware implementations favored
 - Can be implemented in software ... but that was not the point ...
 - I/O hardware: NICs with RDMA = RNICs (RDMA enabled NICs)
- Ethernet hardware transition in progress
 - 10/40 Gb/sec Ethernet and prior: Most NICs are not RNICs.
 - 25/50/100 Gb/sec Ethernet: Most NICs are RNICs.
 - Increased RDMA usage expected with 25/50/100 Gb/sec Ethernet
- Some additional RDMA storage protocols:
 - SMB Direct (Windows file sharing)
 - iSER (iSCSI over RDMA),
 - SRP (SCSI RDMA Protocol)



Talk Outline

- NVMe Background
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports
 - RDMA-based, Fibre Channel, TCP ←
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers
- I/O Path Performance



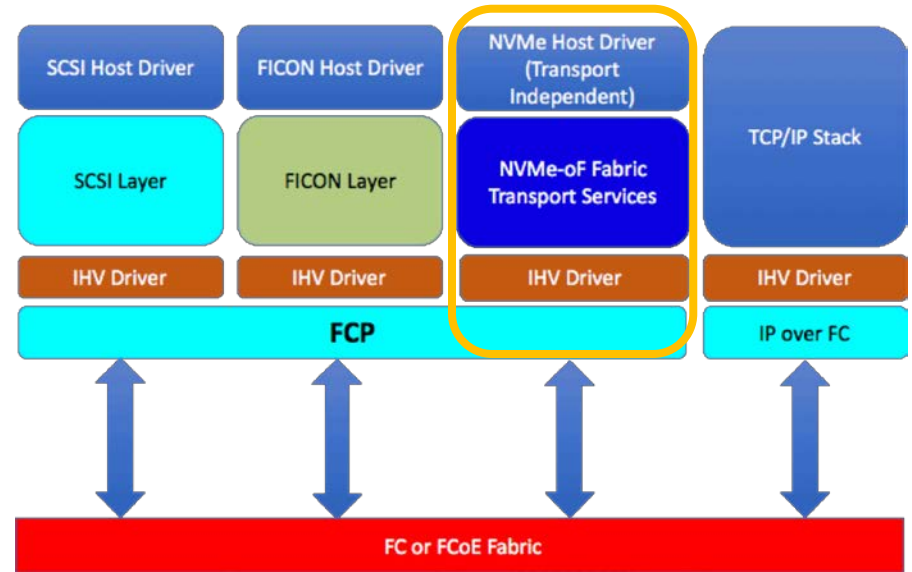
Fibre Channel Overview

- Widely deployed SAN (Storage Area Network) Fabric
 - Optimized for storage traffic: FCP (SCSI over FC protocol)
 - Also supports mainframe storage: FICON protocol
- Fabric-based Architecture: Fabric is a first-class entity
 - Set of switches and end node ports managed as a “Fabric”
 - End node ports log into the “Fabric” (e.g., not to a switch)
- Fabric-wide services (important)
 - Nameserver, including update functionality
 - Example: Notify servers about new storage system
 - Zoning: Fabric switch hardware blocks misdirected traffic.
 - Complements storage system access control (LUN masking/mapping)
- Dedicated hardware (HBAs, switches), optical cabling
 - Highly reliable, redundant SANs typically used for availability



NVMe/FC and FCP: Data Transfer

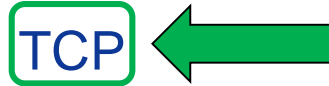
- FCP: Original SCSI over FC protocol
 - Implementations optimized FCP data transfer for SCSI
- Now used for all FC data transfer
 - SCSI, NVMe over Fabrics, and FICON
- FCP data transfer: Back-to-back DMA
 - Source: DMA gather
 - Destination: DMA scatter
 - No scatter-gather list “on the wire”
- NVMe over FC uses FCP for data
 - FCP data optimizations “just work”
 - Optimized HBAs in particular
 - Enables NVMe/FC to coexist with SCSI, and even FICON





Talk Outline

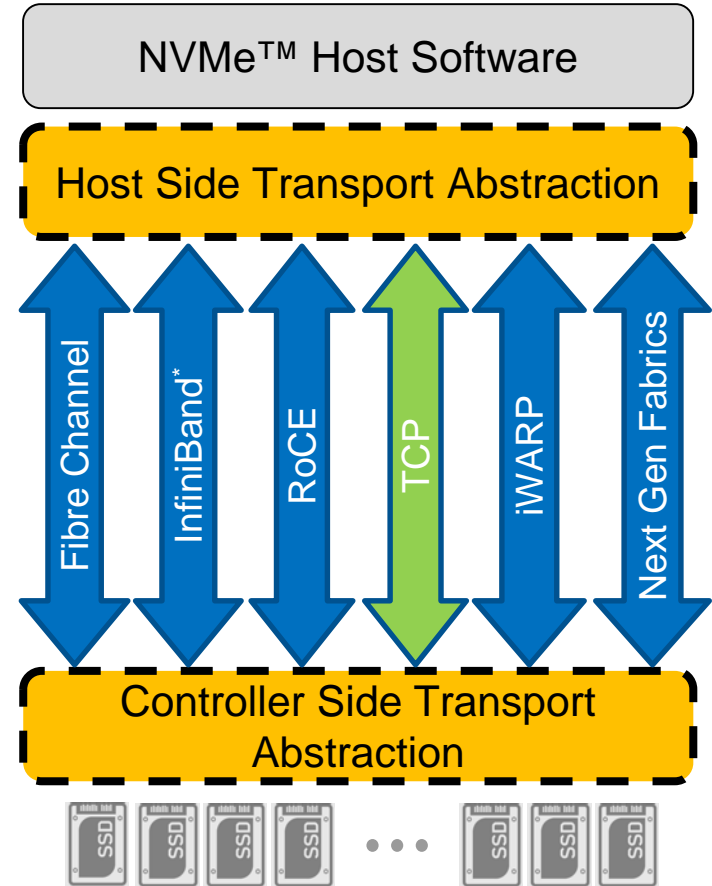
- NVMe Background
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports
 - RDMA-based, Fibre Channel, TCP
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers
- I/O Path Performance





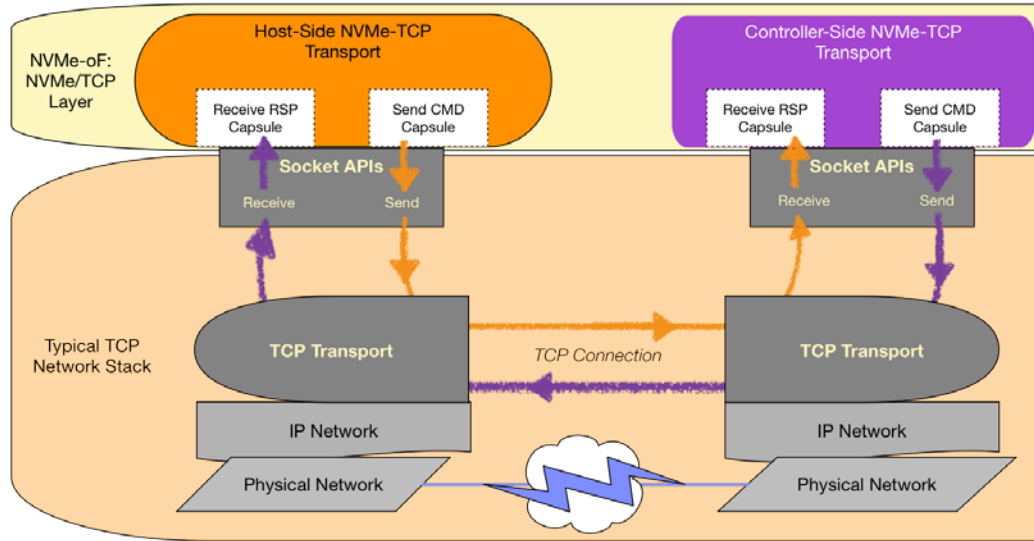
NVMe/TCP

- NVMe™ block storage protocol over standard TCP/IP transport
- No changes required to network infrastructure
 - Benefits from NIC optimizations for TCP
- Independently scale storage & compute to maximize resource utilization and optimize for specific workload requirements
- Maintains NVMe™ model: subsystems, controllers, namespaces, Admin & I/O Queues





NVMe/TCP in a Nutshell

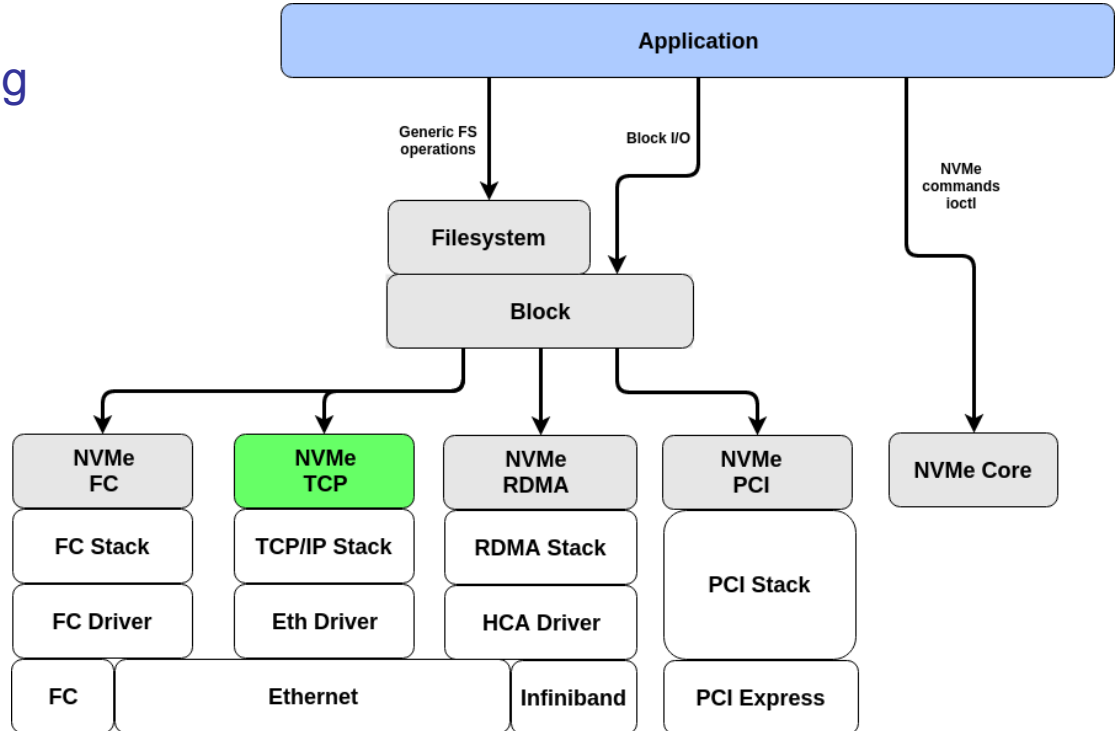


- NVMe-oF Commands sent over standard TCP/IP sockets
- Each NVMe queue pair mapped to a TCP connection
- TCP provides a reliable transport layer for NVMe queues
- Designed for software implementation on existing hardware



NVMe/TCP Standardization

- NVMe Technical Working Group adding TCP to NVMe Fabrics spec alongside RDMA
 - NVMe over FC: Separate T11 standard
- NVMe over TCP ratification expected by end of this year (2018)





TCP for NVMe – Surely you jest??

Not at all, this is completely serious ...

Absolute latency higher than RDMA?

Head-of-line blocking could increase latency?

Delayed ACKs could increase latency?

Incast could be an issue?

Lack of hardware acceleration?

Only matters if the application is extremely latency-sensitive

Protocol breaks up large transfers

ACKs pace packet transmission, result is TCP “self-clocking.”

Can be mitigated by network switch functionality

Not an issue for initial NVMe/TCP use-cases, may change in longer term.



Fabrics for NVMe-oF

- InfiniBand: Primarily an embedded fabric, also HPC & HFT
- Omni-Path: HPC-focused niche fabric
- RoCE and iWARP: RDMA over Ethernet
 - Effective implementations: hardware-based (RDMA NICs)
 - RoCE vs. iWARP: Stay tuned
 - But ignore RoCEv1 (not routable, superseded by RoCEv2)
- Fibre Channel: Mature SAN technology
 - NVMe is a peer to SCSI: Can use both on same hardware
- TCP: TCP/IP over Ethernet
 - New protocol, initial target: software implementations



Talk Outline

- NVMe Background
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports
 - RDMA-based, Fibre Channel, TCP
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers
- I/O Path Performance

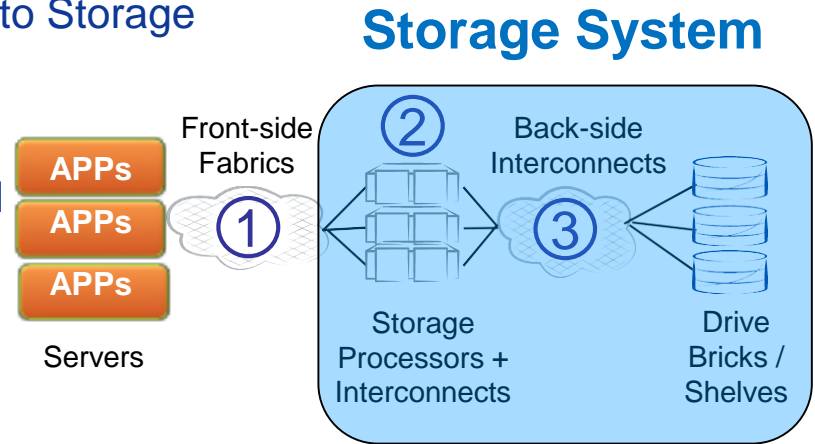




Storage System Architecture (Simplified)

Includes Converged Infrastructure (CI): Server and Storage in same rack(s)

- External Connectivity
 1. Front-side Fabric connects Servers to Storage
 - Examples: FC, iSCSI
- Internal Connectivity
 2. Storage Processors inter-connected
 - Examples: InfiniBand, Ethernet
- Drive Connectivity
 3. Storage Processors connected to Drive Bricks or Shelves
 - Examples: SAS, NVMe/PCIe

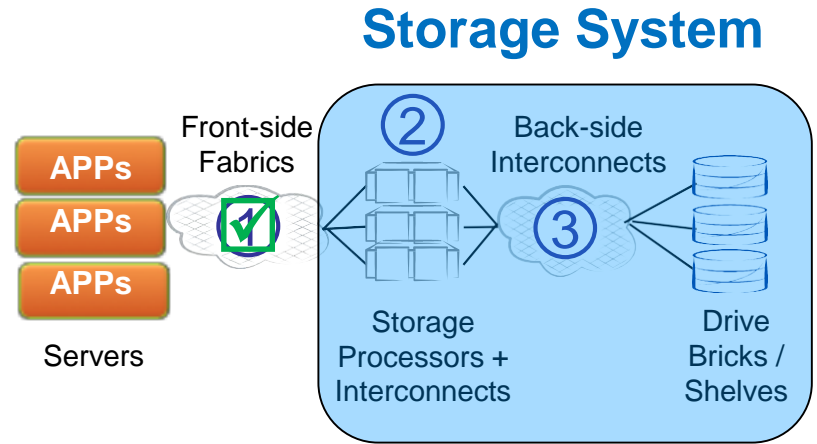




Storage System Architecture (Simplified)

1. Front-side Fabric connects Servers to Storage

- Examples: FC, iSCSI
- NVMe-oF: Likely
- Use case: SAN functionality
- Likely Fabrics for NVMe-oF
 - Ethernet (RoCEv2, iWARP, TCP)
 - Fibre Channel





Storage System Architecture (Simplified)

2. Storage Processors inter-connected

- Examples: InfiniBand, Ethernet

- NVMe-oF: Unlikely ?

- Use case: Internal Functionality
- More complex than NVMe commands

- Fabrics likely to be used

- But not with NVMe-oF as primary protocol
- Storage arrays have rather different internal protocols
- Need more functionality than NVMe commands.
- Cross-array storage replication is similar.

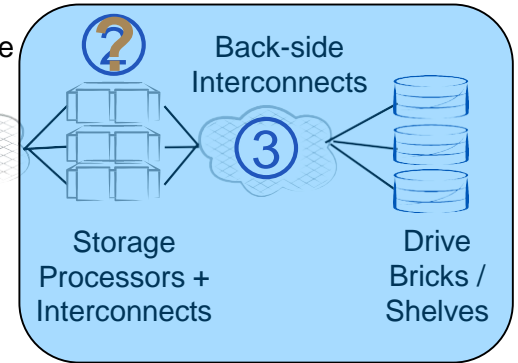


Servers

Front-side Fabrics



Storage System





Storage Processors and NVMe-oF

- Storage system: Cluster of storage processors
 - Peer-to-peer interactions, sophisticated coordination protocols
- NVMe commands: Originally designed for SSDs
 - Master-slave structure (these are commands, not requests)
 - Host tells SSD what to do, SSD does it.
- Master-slave structure: Poor fit to peer-to-peer cluster
 - Storage processor interaction requires more than drive commands
 - Already the case for SCSI commands
- Fabrics used among storage processors, but ...
 - NVMe-oF not sufficient for storage processor interaction
- Storage replication is similar
 - Sophisticated peer-to-peer functionality, not just remote writes




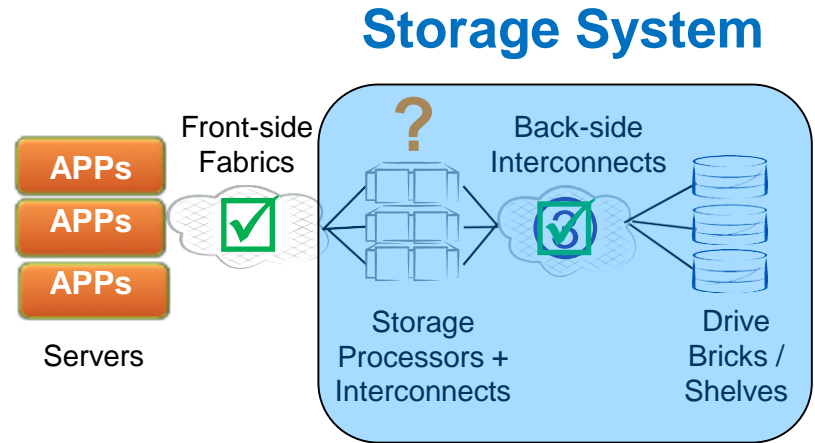


Storage System Architecture (Simplified)

3. Storage Processors connected to Drive Bricks or Shelves

- Examples: SAS, NVMe/PCIe

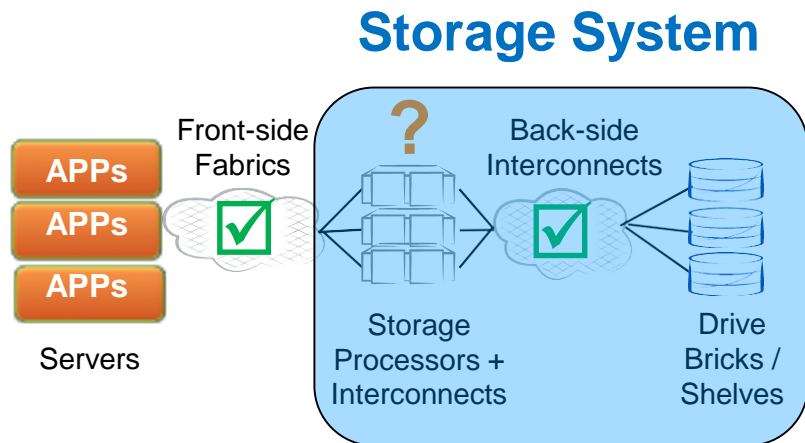
- NVMe-oF: Likely 
 - Use case: Fabric access to drives
 - SAS usually limited to 2-way connectivity
 - NVMe-oF provides connectivity to more than 2 storage processors
- Likely Fabrics for NVMe-oF
 - InfiniBand
 - Ethernet (RoCEv2, iWARP, TCP)



Storage System Architecture (Simplified)

Includes Converged Infrastructure (CI): Server and Storage in same rack(s)

1. Front-side Fabric connects Servers to Storage
 - NVMe-oF: Likely
 - Likely Fabrics: FC, Ethernet
2. Storage Processors inter-connected (and storage replication)
 - NVMe-oF: Unlikely
3. Storage Processors connected to Drive Bricks or Shelves
 - NVMe-oF: Likely
 - Likely Fabrics: InfiniBand, Ethernet





Talk Outline

- NVMe Background
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports
 - RDMA-based, Fibre Channel, TCP
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers ←
- I/O Path Performance

Rack and Modular Servers: Infrastructure Scaling



- Many servers in same enclosure or rack
- Storage: Drives in servers + SDS software
 - SDS = Software Defined Storage
 - Examples: VSAN, Ceph, Storage Spaces, ...
- Scale-out via multi-server building blocks
 - Uniform hardware configurations
 - Common IT procedures
- But ... workloads aren't uniform





Rack and Modular Servers: Infrastructure Scaling (2)

Includes Hyper-Converged Infrastructure (HCI): Servers with Storage Drives

- But ... workloads aren't uniform
 - Hardware compute-to-storage ratio hard to change
 - Especially after installation
- Now what?
- Storage Fabrics to the Rescue!
 - More compute than storage:
 - Add fabric-attached drive bricks (JBOFs)
 - Assign drives to any server
 - More storage than compute:
 - Add fabric storage targets
 - Provide SDS storage to external servers

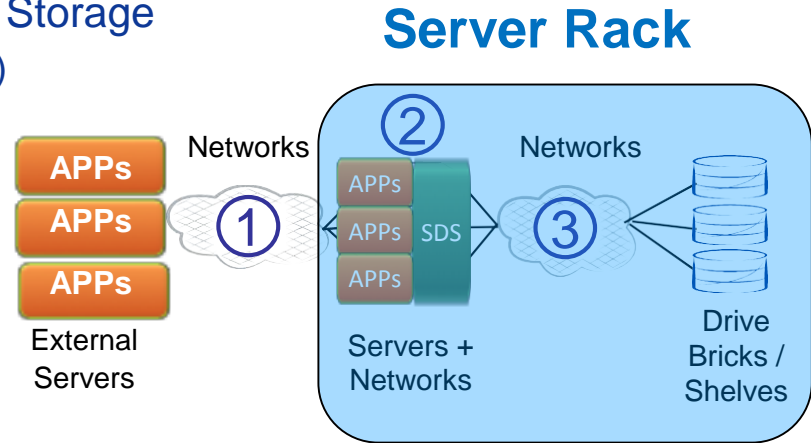




Server Infrastructure (Simplified)

Includes Hyper-Converged Infrastructure (HCI): Servers with Storage Drives

- External Connectivity
 1. External Servers connected to SDS Storage
 - Example: iSCSI, FCoE (via Ethernet)
- Internal Connectivity
 2. Servers and SDS components inter-connected
 - Usually Ethernet
- Drive Connectivity
 3. Servers connected to External Drive Bricks or Shelves
 - Example: SAS, shared drives not typical

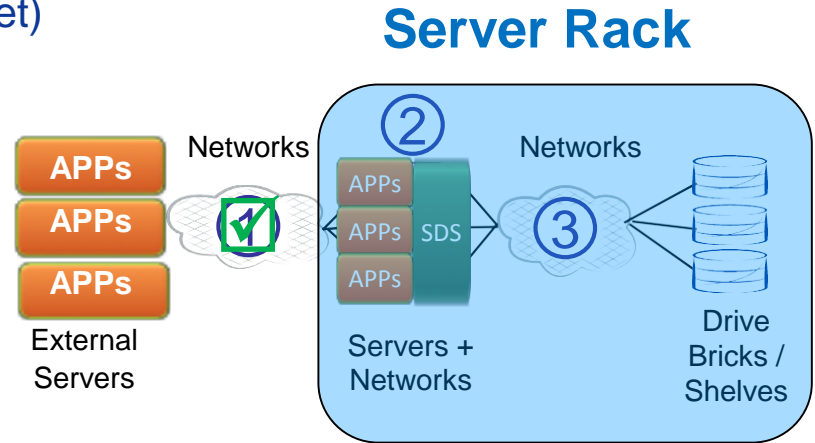




Server Infrastructure (Simplified)

1. Networks connect External Servers to SDS Storage

- Examples: iSCSI, FCoE (via Ethernet)
- NVMe-oF: Likely
- Use case: SAN functionality
- Likely Fabrics for NVMe
- Ethernet (RoCEv2, iWARP, TCP)





Server Infrastructure (Simplified)

2. Servers and SDS components inter-connected

- Usually Ethernet

- NVMe-oF: Unlikely ?

- Use case: Internal Functionality
- SDS: More complex than NVMe cmds.

- Networks likely to be used

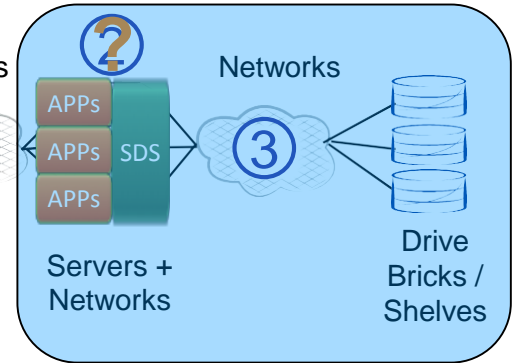
- But not with NVMe-oF protocol
- SDS systems have rather different internal protocols

- SDS storage replication is similar

- Need more functionality than NVMe commands.



Networks



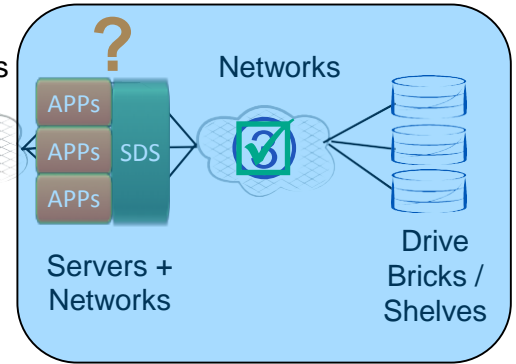
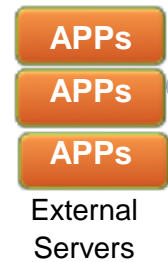


Server Infrastructure (Simplified)

3. Servers connected to External Drive Bricks or Shelves

- Example: SAS, shared drives are unusual

- NVMe-oF: Likely
 - Use case: Fabric access to drives
 - SAS usually limited to 2-way connectivity
 - NVMe-oF can connect to more servers
- Likely Fabrics
 - Ethernet (RoCEv2, iWARP, TCP)
 - InfiniBand





Server Infrastructure (Simplified)

Includes Hyper-Converged Infrastructure (HCI): Servers with Storage Drives

1. External Servers connected to SDS Storage

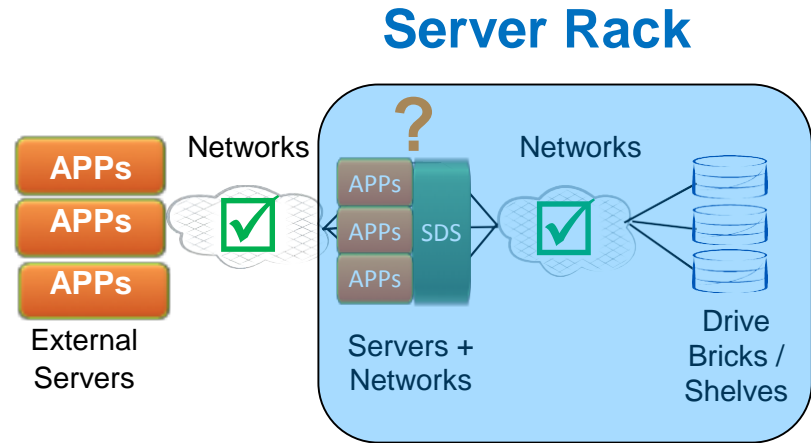
- NVMe-oF: Likely
- Likely Fabrics: Ethernet

2. Servers and SDS components inter-connected

- NVMe-oF: Unlikely

3. Servers connected to External Drive Bricks or Shelves

- NVMe-oF: Likely
- Likely Fabrics: Ethernet, InfiniBand



Does this look familiar 😊 ??



Storage System Architecture (Simplified)

Includes Converged Infrastructure (CI): Server and Storage in same rack(s)

1. Front-side Fabric connects Servers to Storage

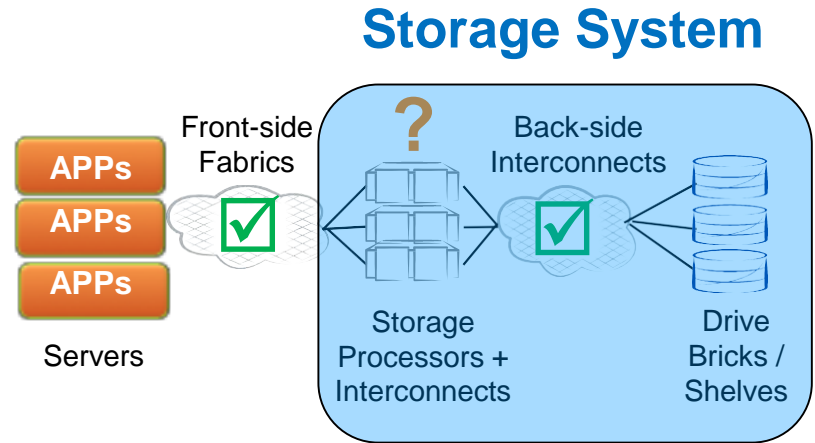
- NVMe-oF: Likely
- Likely Fabrics: FC, Ethernet

2. Storage Processors inter-connected (and storage replication)

- NVMe-oF: Unlikely

3. Storage Processors connected to Drive Bricks or Shelves

- NVMe-oF: Likely
- Likely Fabrics: InfiniBand, Ethernet

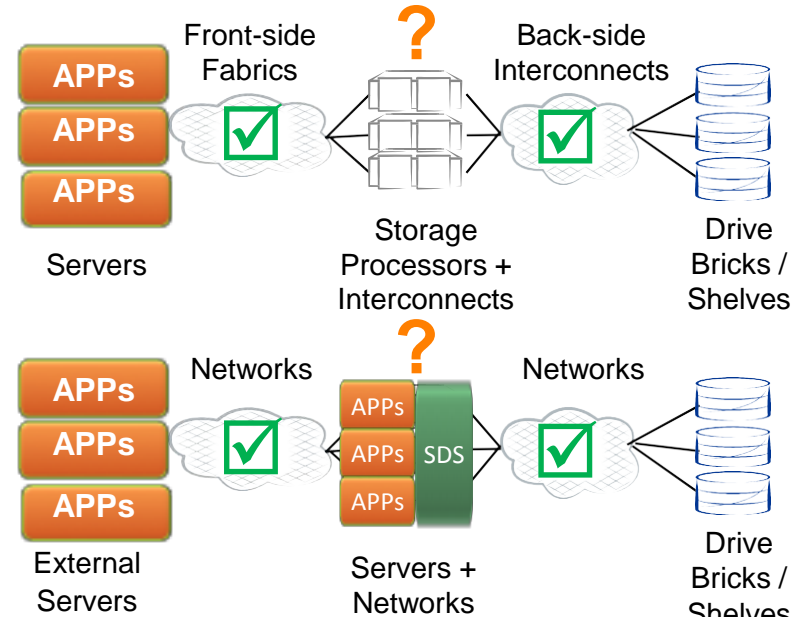


Aha, this looks familiar 😊 !!



NVMe-oF: Fabric Perspectives

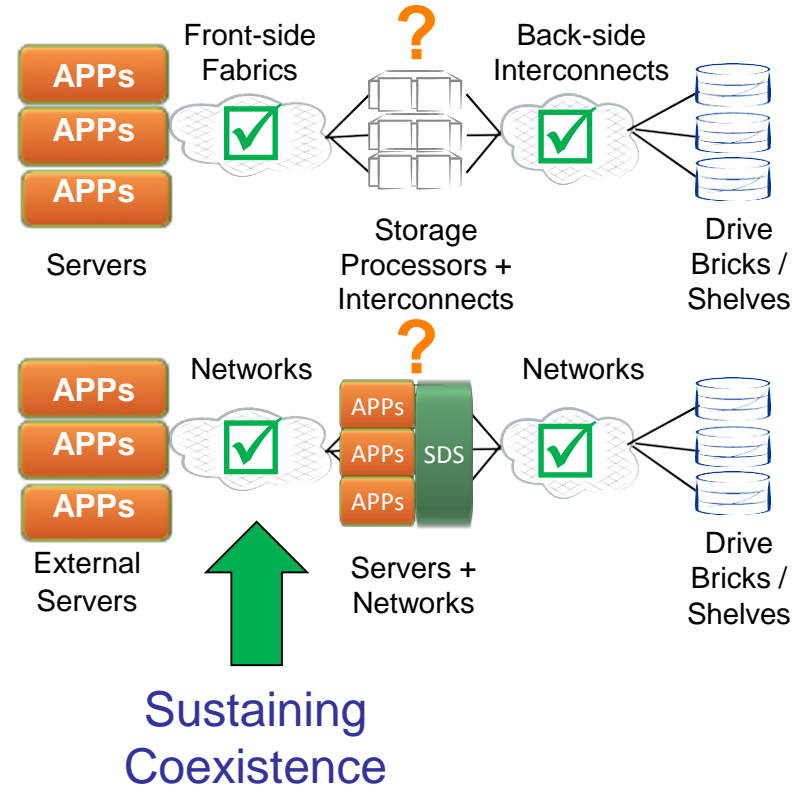
- Ethernet applies everywhere
 - RDMA: Favors hardware RNICs
 - TCP: Favors software, NICs are fine
- Fibre Channel
 - Servers to external storage only
- InfiniBand
 - Embedded usage in storage systems
- Storage System or SDS internals: NVMe-oF insufficient
 - Constrained Master-slave architecture
 - Same is true of SCSI, e.g., iSCSI
 - Also applies to Storage Replication





NVMe-oF: Front-side Impacts

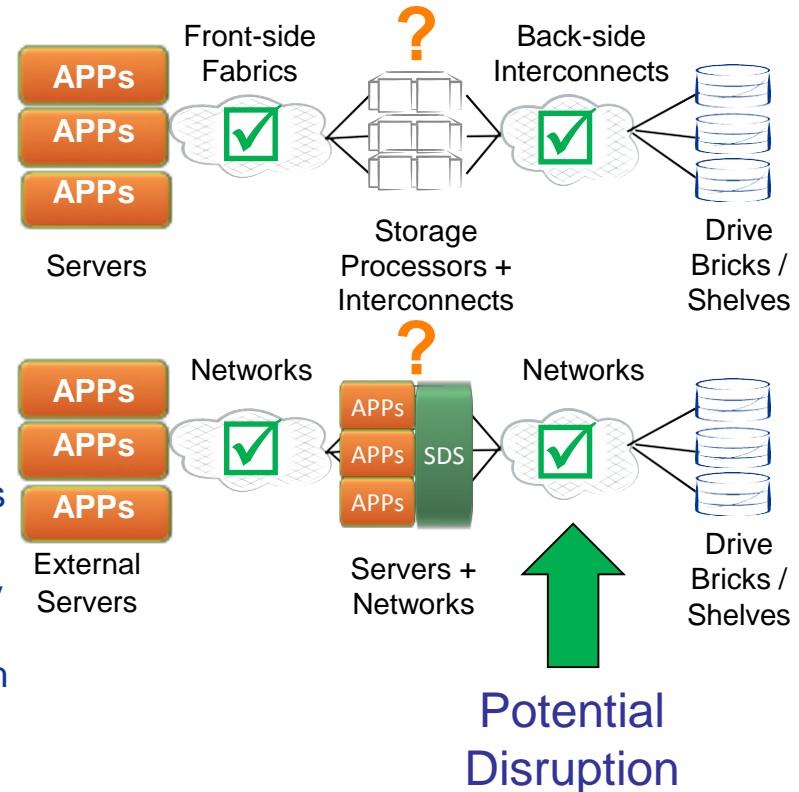
- Fibre Channel SANs
 - NVMe and SCSI coexist
 - Along with mainframe FICON
- TCP/IP networks
 - iSCSI and NVMe/TCP coexist
 - Along with many other protocols
- RDMA: Challenger
 - Limited current storage use
 - Coexists with iSCSI, RDMA-based SCSI protocols
 - RNICs strongly preferred
 - Deployment will increase with 25/50/100 Gb/sec Ethernet





NVMe-oF: Back side impacts

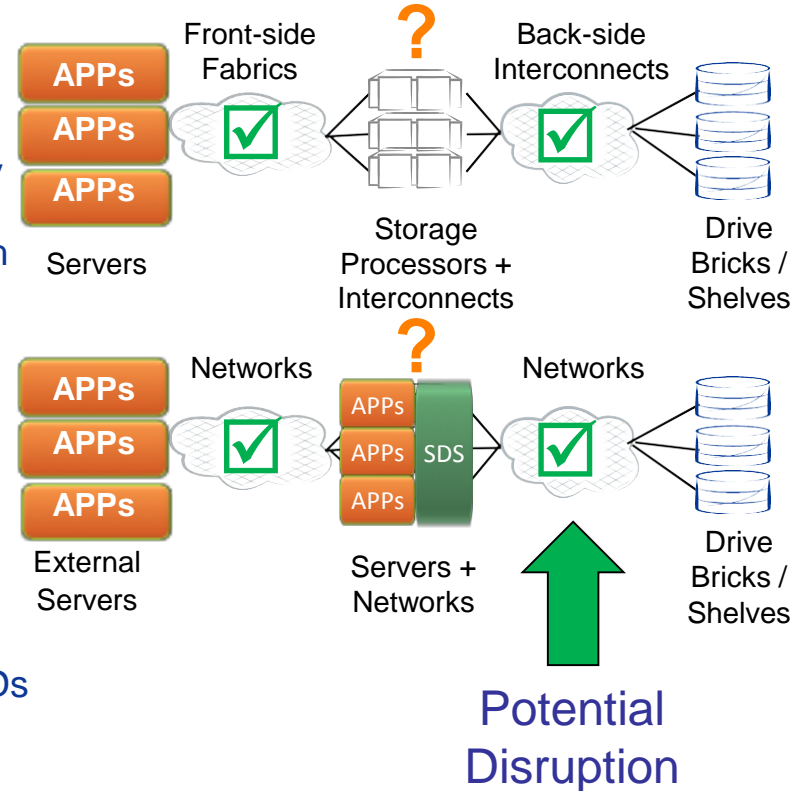
- Servers or Storage Processors connected to drive bricks or shelves
 - Current technology: SAS, limited to 2-way connectivity
- Servers: single connectivity common
 - Server failure can cause lengthy SDS rebuild
- Storage systems: 2-way connectivity common (to 2 storage processors)
 - No direct access from other storage processors
- NVMe-oF: N-way connectivity
 - Any server or storage processor can talk to any drive brick or shelf
 - Potentially disruptive when 2-way is not enough
 - Functional change to drive usage with servers






NVMe-oF: Back side impacts (cont'd)

- NVMe-oF: N-way connectivity
 - Any server or storage processor can talk to any drive brick or shelf
 - Potentially disruptive when 2-way is not enough
 - Functional change to drive usage with servers
- Preferred fabric: TBD
 - InfiniBand
 - Ethernet: RoCEv2, iWARP, TCP
- Server requirement: Management software
 - Easy to use and automate drive bricks/shelves
 - Especially for HCI (rack-based) servers
- Protocol chips becoming available
 - NVMe-oF interface to multiple NVMe/PCIe SSDs





Talk Outline

- NVMe Background
- NVMe over Fabrics (NVMe-oF) Background
- NVMe-oF Transports
 - RDMA-based, Fibre Channel, TCP
- NVMe-oF Transport Usage
 - Storage Systems
 - Servers
- I/O Path Performance 

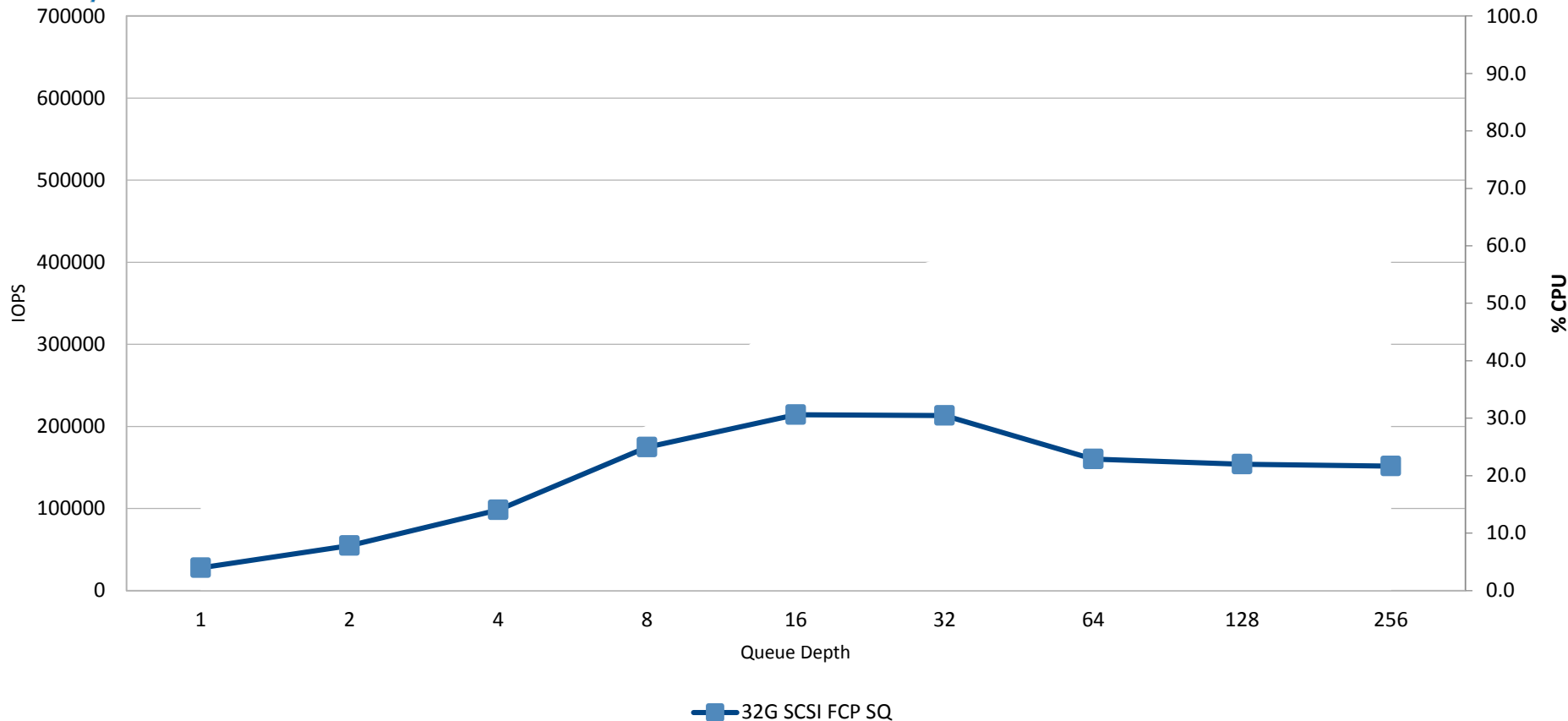


I/O Path: Performance Perspective

- SCSI over FC (FCP) & NVMe over FC (FC-NVMe)
- Single server Linux, 32GFC Fibre Channel
 - 32GFC: 28.05 GHz line frequency, 25.6 Gbit/sec bandwidth
- Workload: I/O Generator: small random I/Os
 - 4KB I/Os, 70:30 R:W mix to RAM disk target
 - Workload exercises I/O path only
- Plot: Sustained throughput (IOPS) vs. Queue Depth

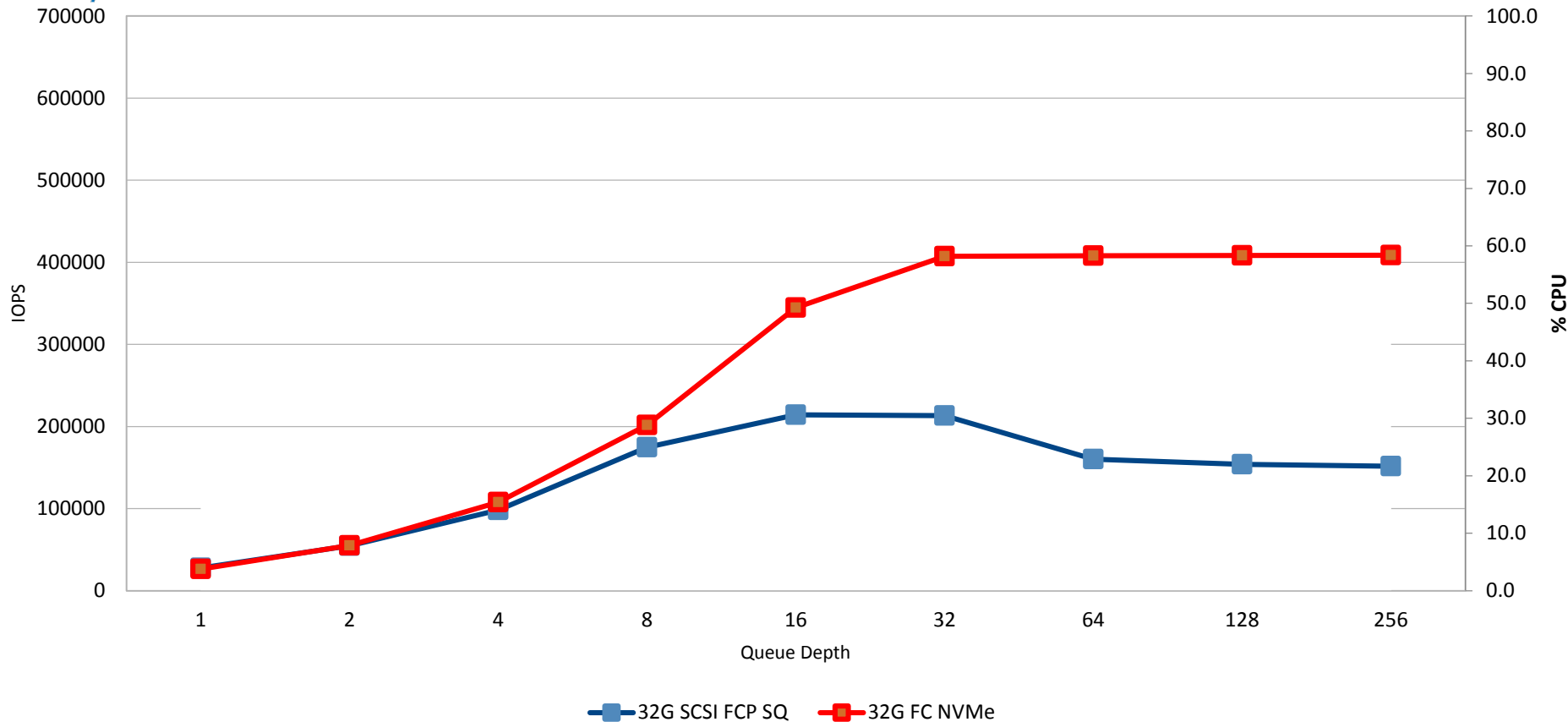


Sustained Random 4K Mixed (R70:W30)



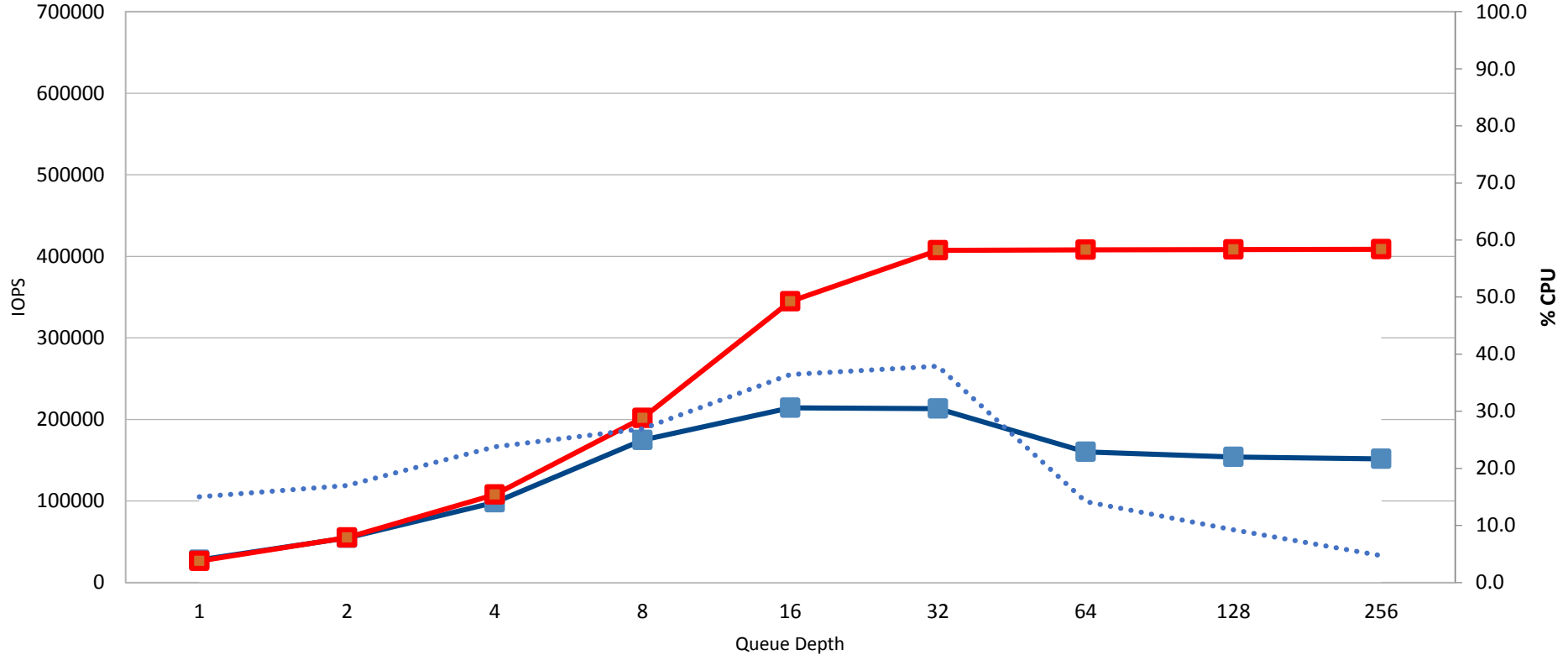


Sustained Random 4K Mixed (R70:W30)





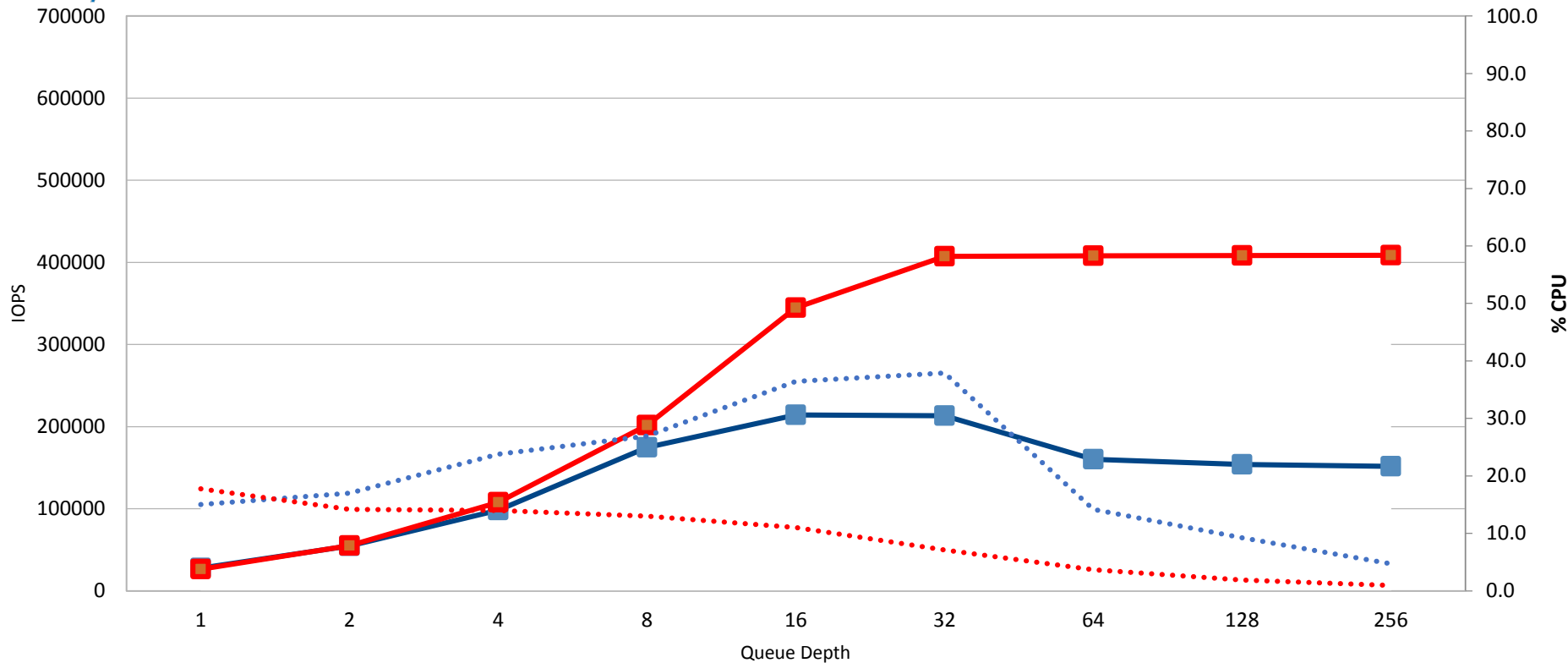
Sustained Random 4K Mixed (R70:W30)



■ 32G SCSI FCP SQ ■ 32G FC NVMe 32G SCSI FCP SQ CPU



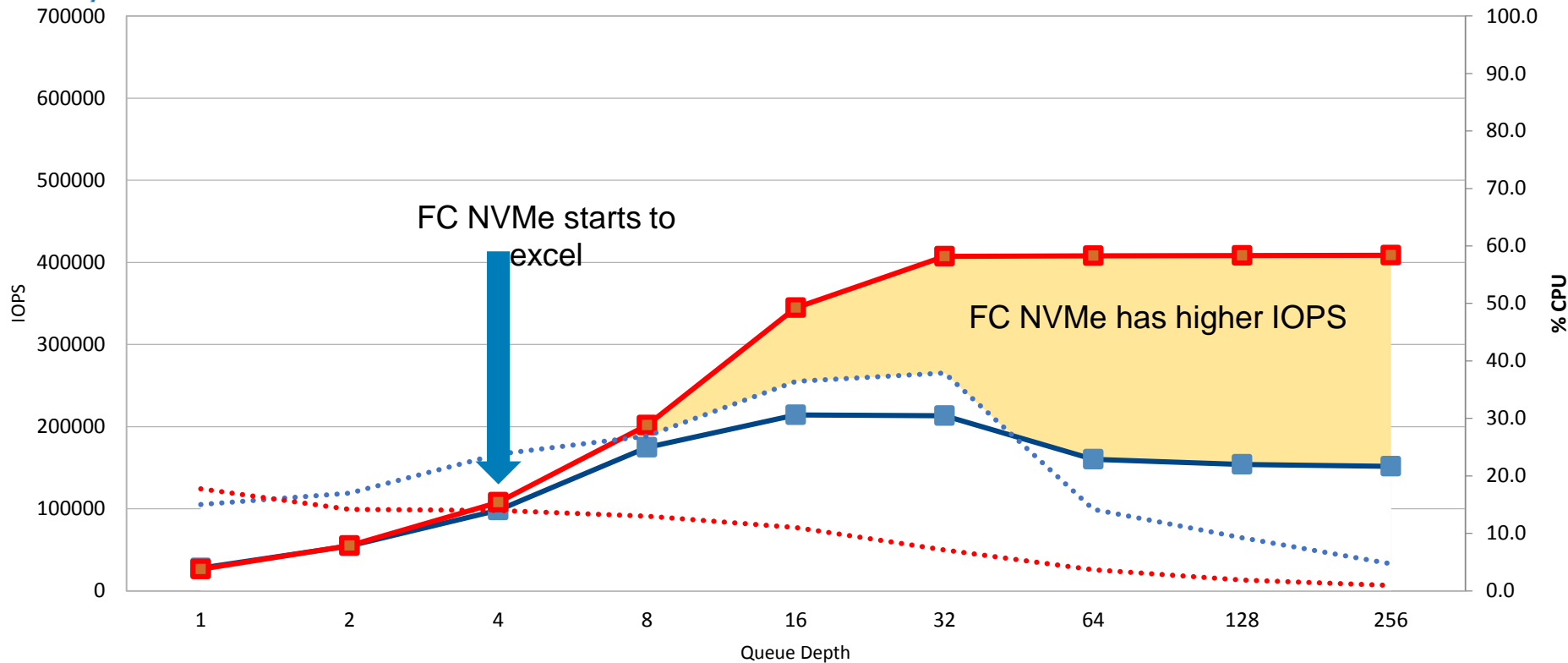
Sustained Random 4K Mixed (R70:W30)



—■— 32G SCSI FCP SQ —■— 32G FC NVMe 32G SCSI FCP SQ CPU 32G FC NVMe CPU



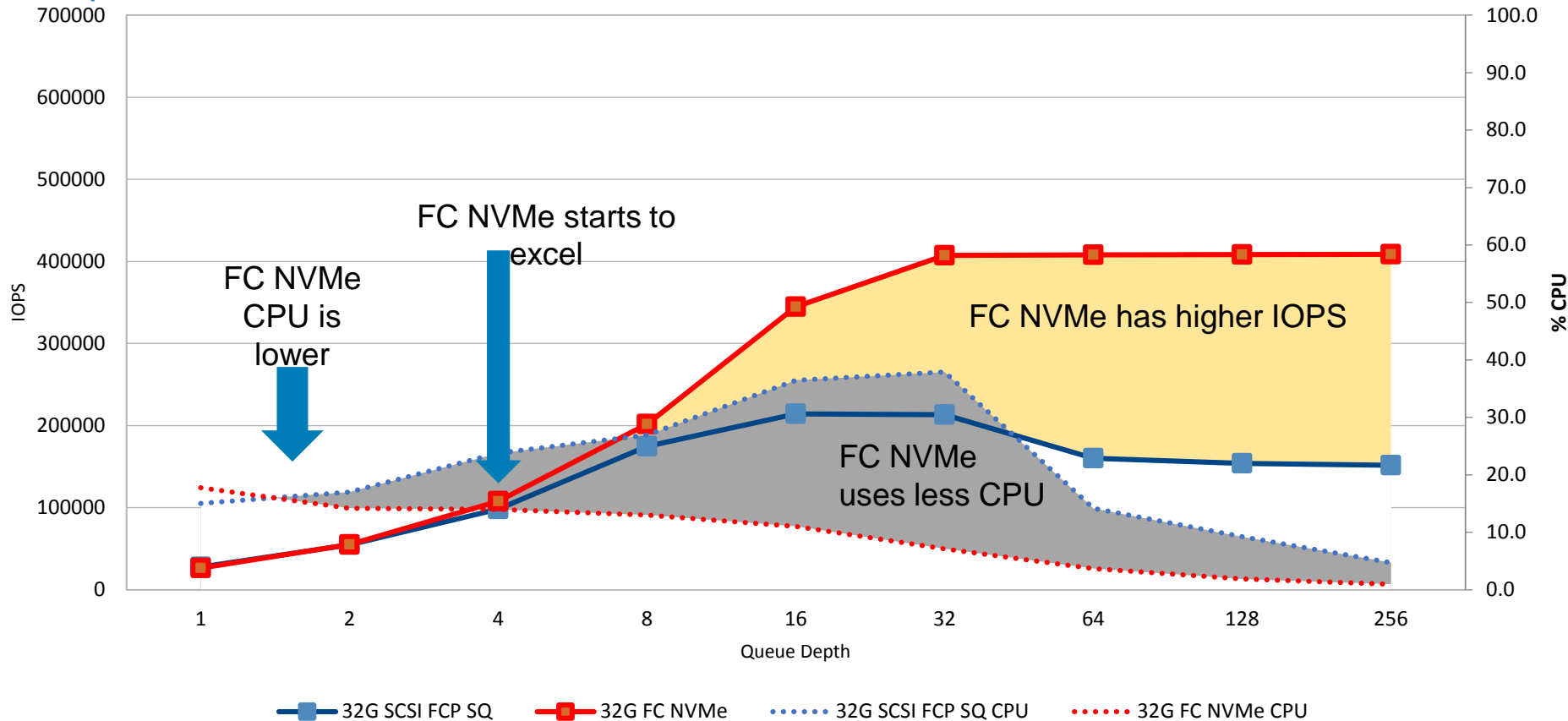
Sustained Random 4K Mixed (R70:W30)



—■— 32G SCSI FCP SQ —■— 32G FC NVMe 32G SCSI FCP SQ CPU 32G FC NVMe CPU

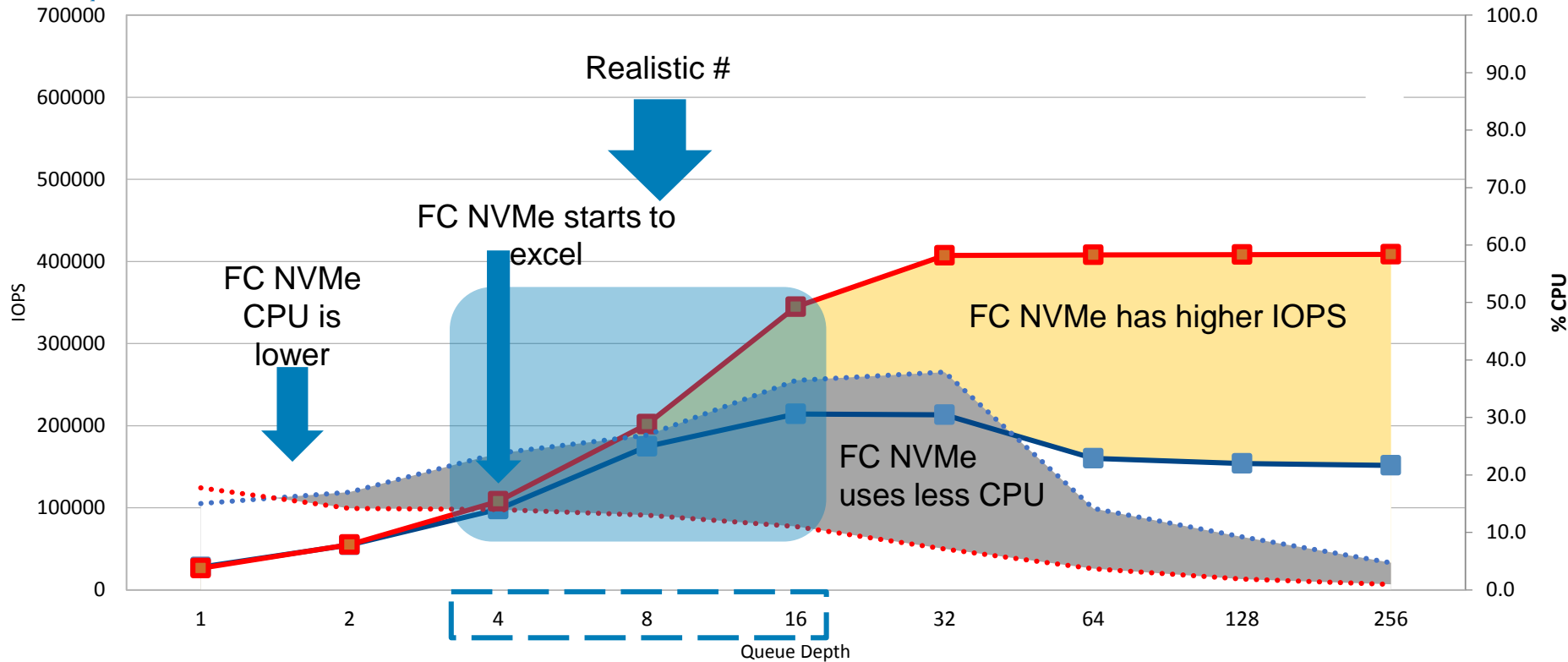


Sustained Random 4K Mixed (R70:W30)





Sustained Random 4K Mixed (R70:W30)



—■— 32G SCSI FCP SQ —■— 32G FC NVMe 32G SCSI FCP SQ CPU 32G FC NVMe CPU



I/O Path Performance: Initial Observations

- NVMe IOPS improves over SCSI at reasonable I/O queue depths
- NVMe uses less CPU (shorter I/O path)
- Reminder: I/O-only workload
 - I/O stack gets more efficient with larger queue depths
 - Easier for host driver and adapter to avoid each other
 - Batching of multiple I/Os in I/O stack



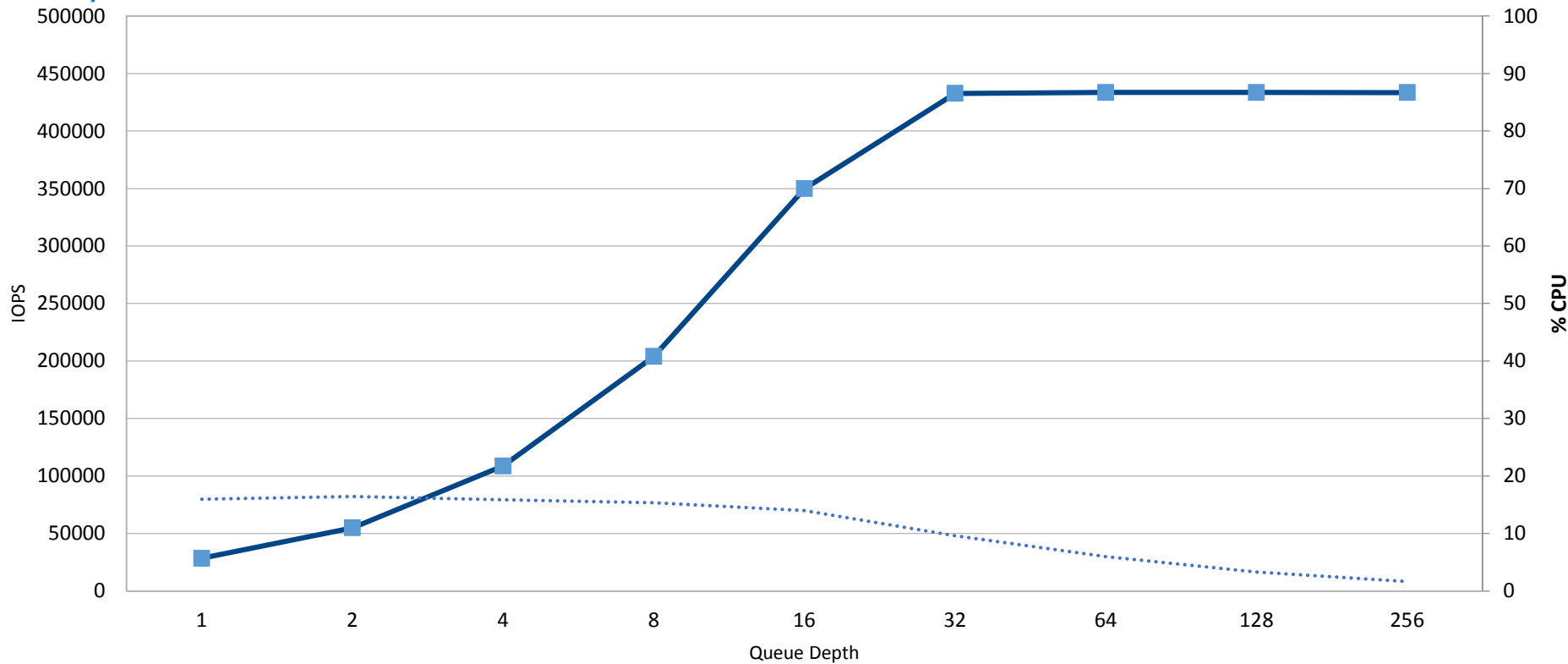
Separating Performance Factors

- Linux SCSI FCP stack: Single Queue for all I/Os
- Linux NVMe FC stack: Multiple I/O Queues
- Can we factor protocol vs. queue structure?

- Yes: Linux SCSI FCP Multi-Queue
 - Caveat: Test configuration only
 - Not enabled by default



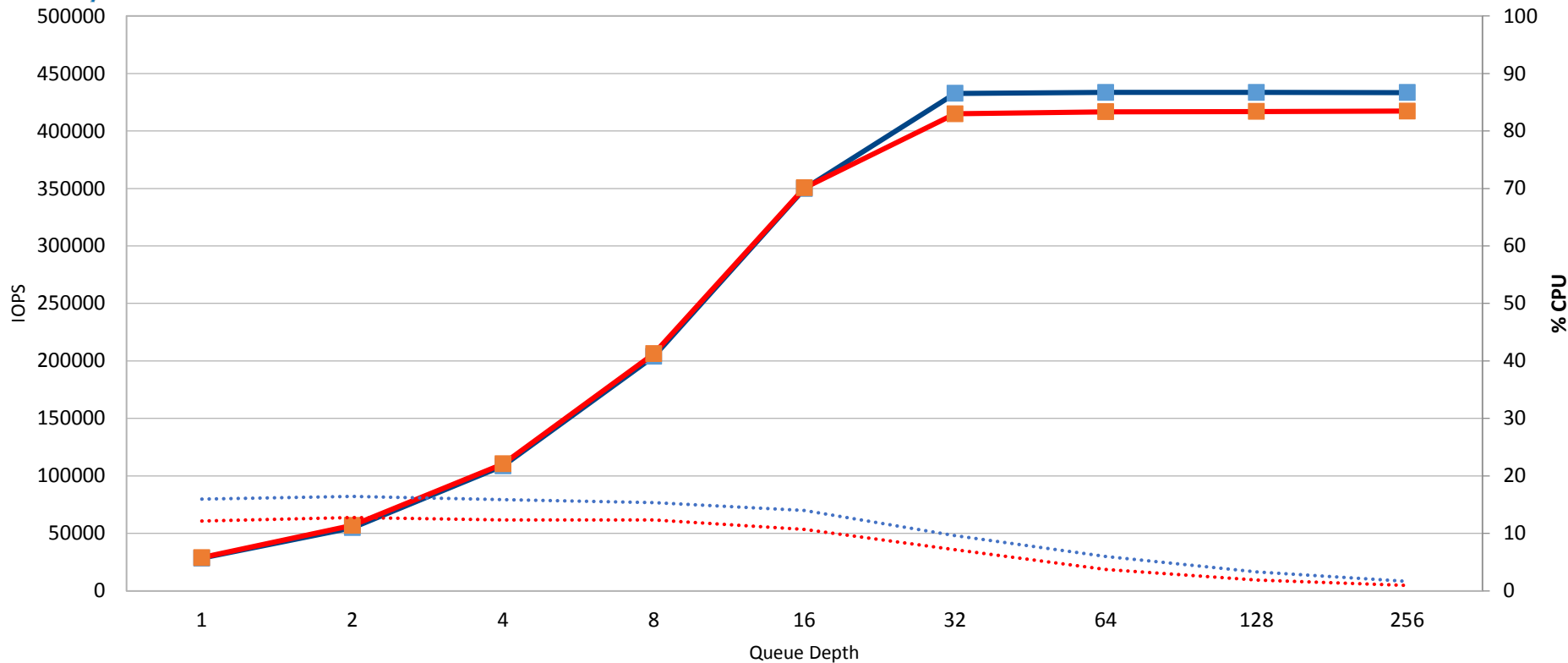
Sustained Random 4K Mixed (R70:W30)



■ 32G SCSI FCP MQ 32G SCSI FCP MQ CPU

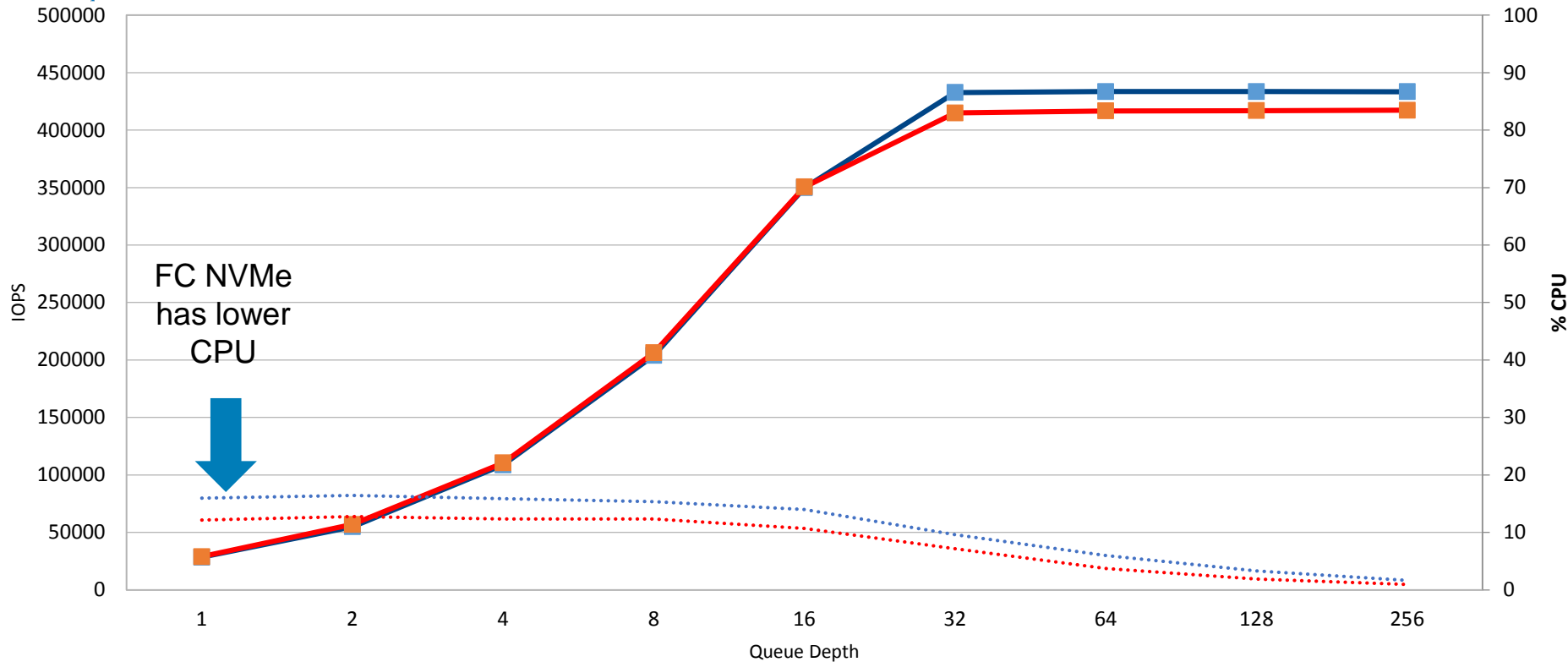


Sustained Random 4K Mixed (R70:W30)





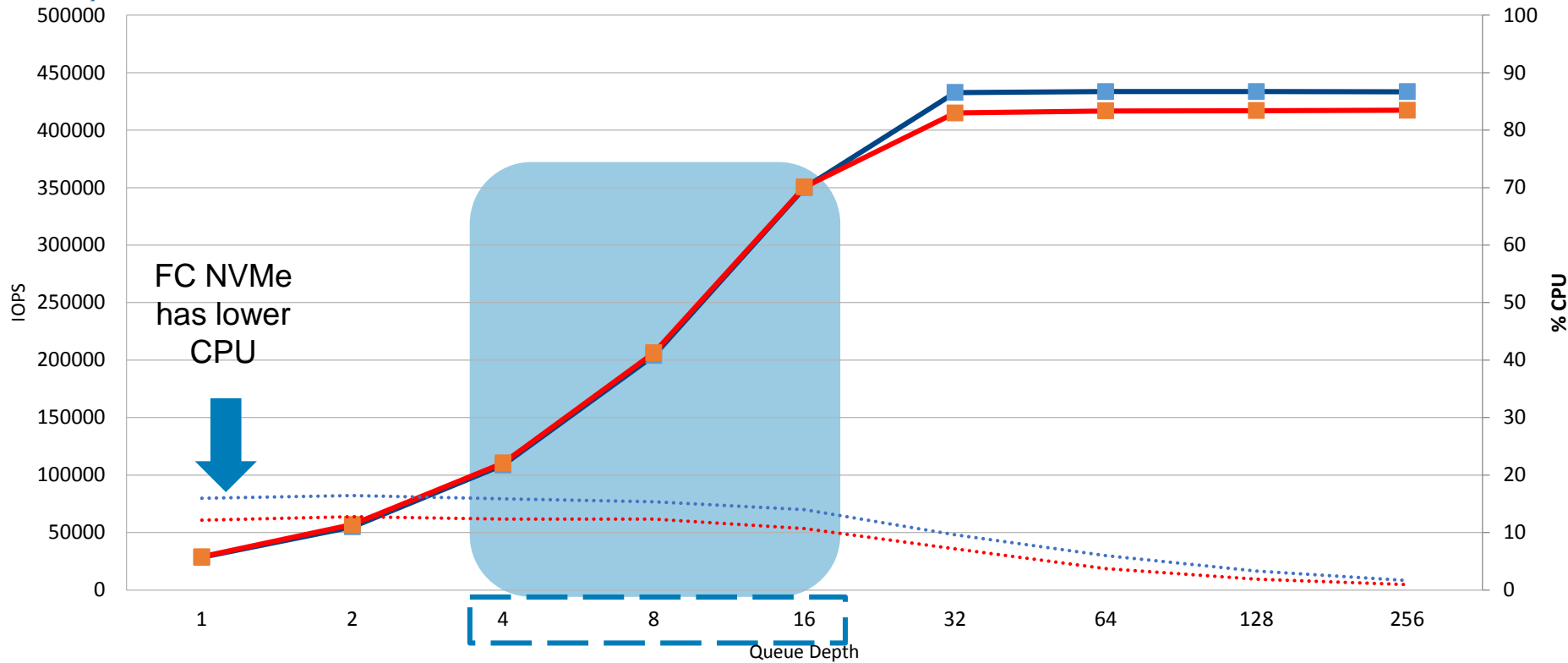
Sustained Random 4K Mixed (R70:W30)



■ 32G SCSI FCP MQ ■ 32G FC NVMe 32G SCSI FCP MQ CPU 32G FC NVMe CPU



Sustained Random 4K Mixed (R70:W30)





I/O Path Performance: More Observations

- NVMe and SCSI Multi-Queue:
 - Similar FC I/O path performance on I/O-bound workload
 - Single SCSI queue thrashed by this workload
- NVMe uses less CPU for I/O:
 - I/O-bound test workload
 - NVMe likely to perform better on actual application workloads
- Caveats
 - One I/O-bound test workload: 4K random I/O, 70:30 mix, RAM disk
 - Special test configuration: SCSI Multi-queue not enabled by default
 - IOPS-only throughput results, not latency
- So, what should we conclude? ...



Conclusion: That's Interesting ...

- Technology cross-pollination
- NVMe parallelism driving SCSI improvements
 - And vice-versa: FC-NVMe uses optimized SCSI (FCP) data path
- Better performance for both protocols
- We've seen this story before ...



Once Upon a Time in Networking ...

- Conventional Wisdom (1980s): TCP is throughput-limited
 - Faster network speeds require higher-performance protocol
 - Ethernet limits packet sizes to ~1.5k, too many CPU instructions per packet
- Conventional Wisdom: Proven wrong
 - Innovative TCP optimizations, Van Jacobson and many others
 - NIC Hardware offloads, especially segmentation and reassembly
- TCP continues to be a fine networking protocol
 - Hardware is now orders of magnitude faster
 - Coexists with other protocols that can out-run TCP in some environments
- Similar co-existence future for SCSI and NVMe
 - Cross-fertilization already in progress



Concluding Thoughts

- NVMe over Fabrics: Common framework
 - Different transports: RDMA, Fibre Channel, TCP
- NVMe-oF use cases
 - Front-side (server access to storage): 
 - Back-side (drive bricks/shelves): 
 - Middle (interconnect Storage Processors or SDS elements): 
- NVMe-oF transport applicability varies by use case
- NVMe and SCSI: Coexistence and cross-fertilization



Credits

- The NVM Express organization
- SNIA and J Metz (Cisco)
- Dell EMC colleagues
 - Amnon Izhar, Bill Lynn, Alan Rajapa, Tony Rodriguez, Erik Smith
- The entire NVMe Technical Working Group
 - For making all of this happen 😊