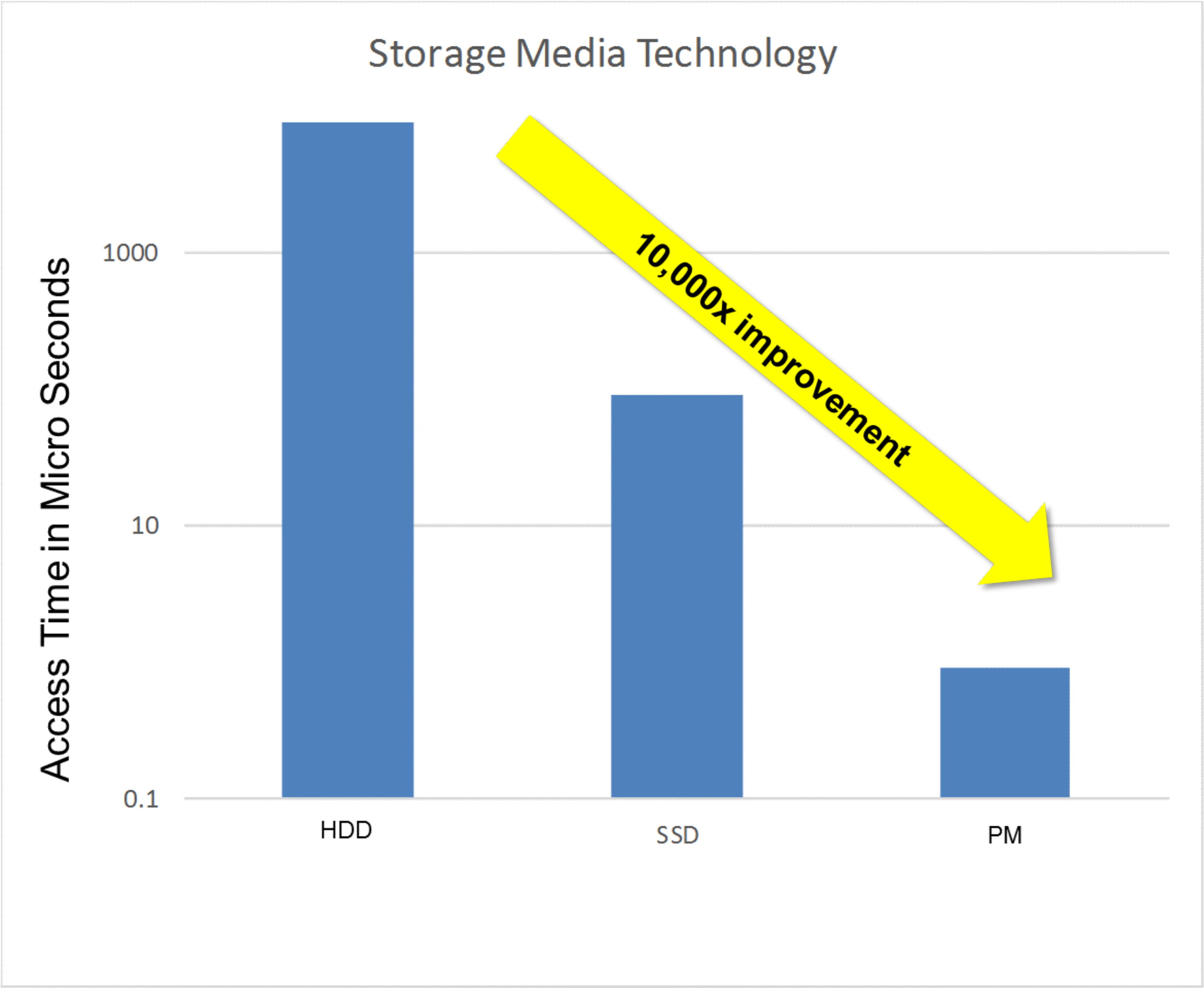
SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2016

# NVMe over Fabrics - High Performance Flash moves to Ethernet

Rob Davis & Idan Burstein

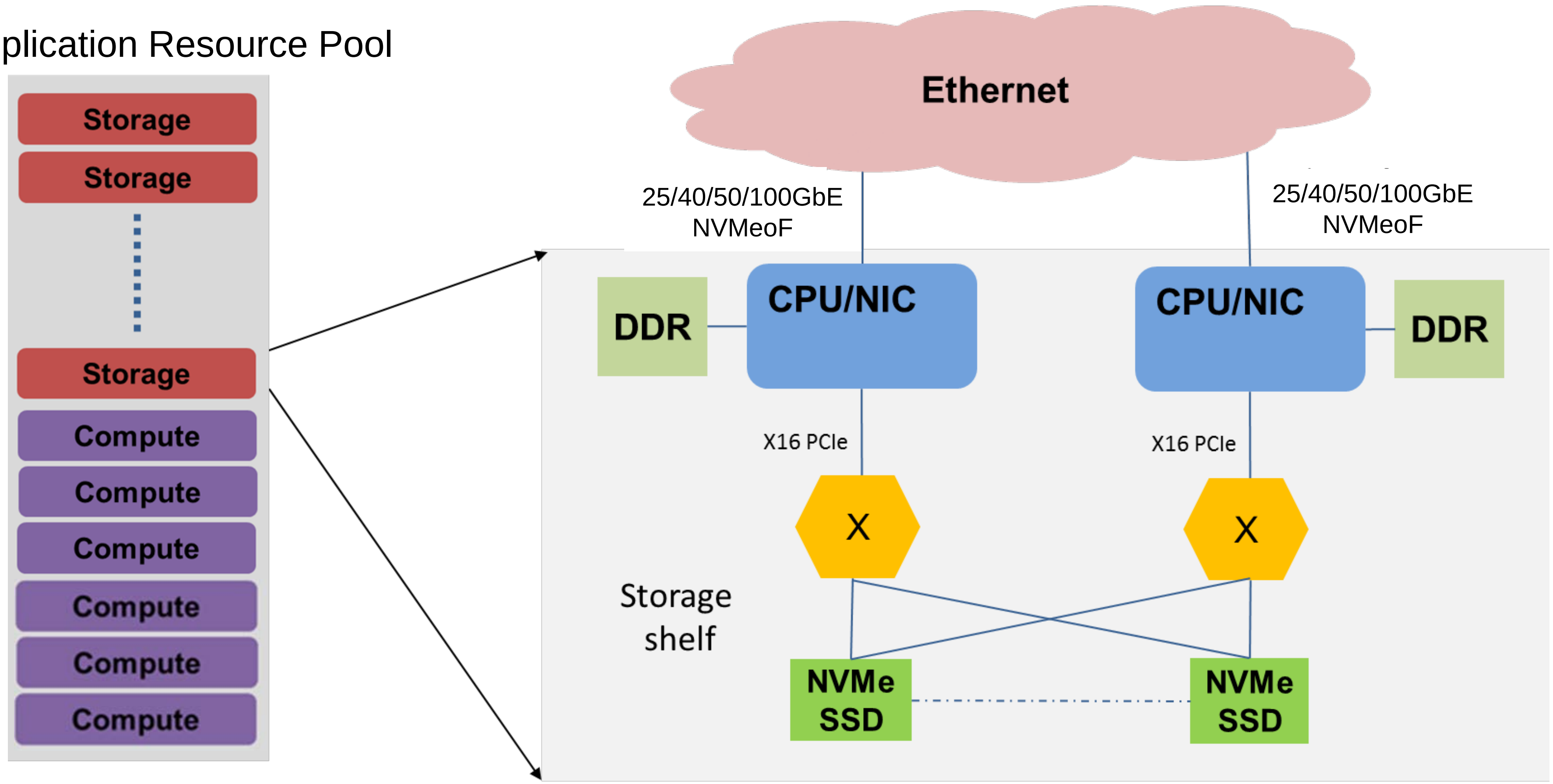
Mellanox Technologies

# Why NVMe and NVMe over Fabrics(NVMf)



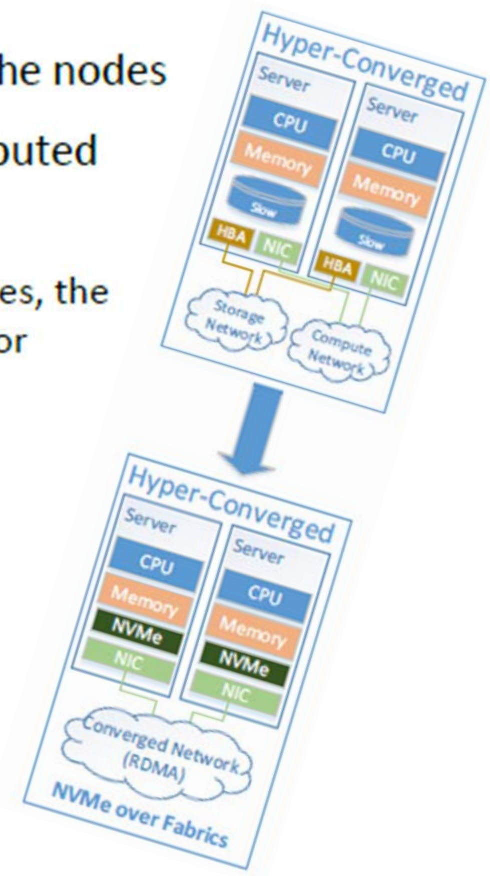
# Compute/Storage Disaggregation

Application Resource Pool

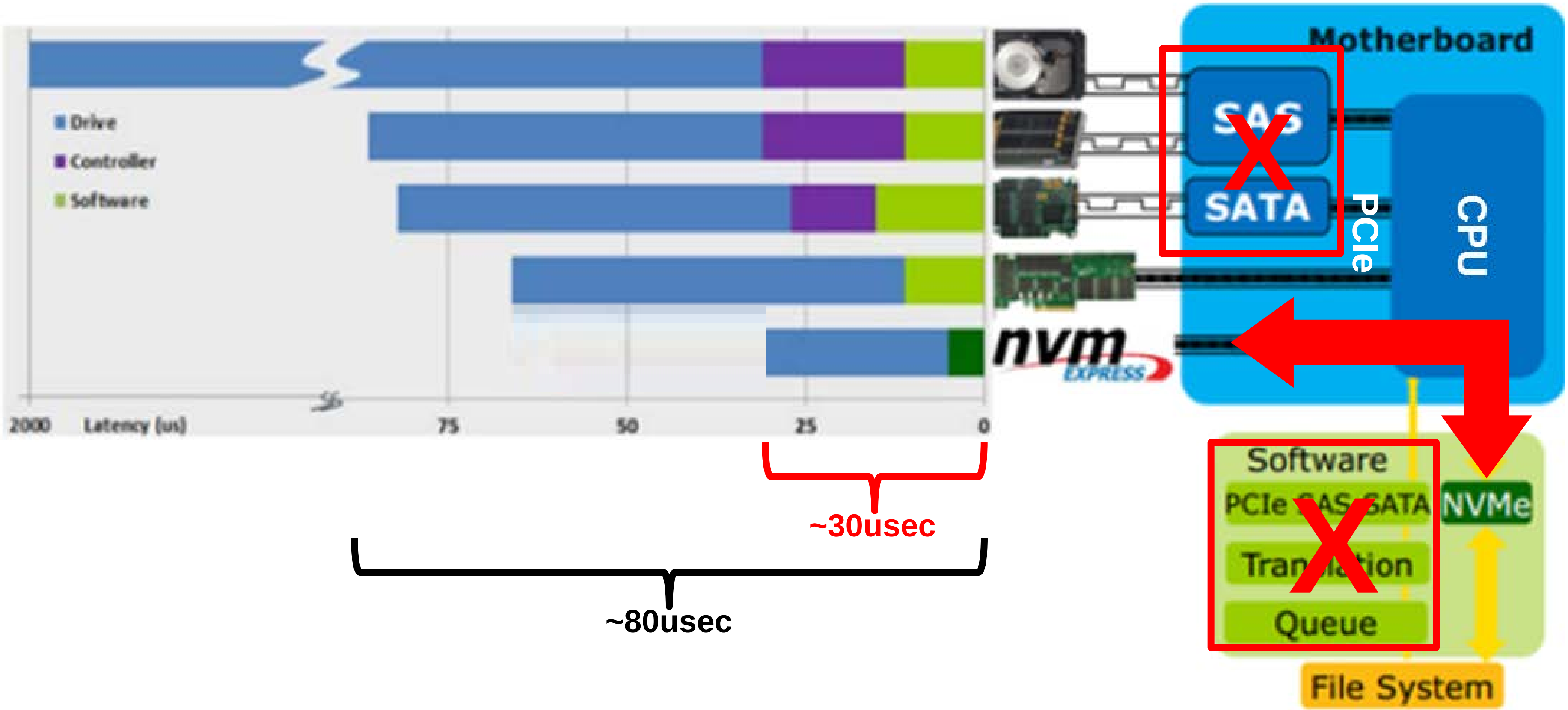


# Hyperconverged with NVMf

- Storage is distributed across the compute nodes and shared among the nodes
- Storage management and provisioning is software defined and distributed
- Benefits of NVMe over Fabrics
  - The most important: major reduction in CPU utilization while sharing devices, the compute nodes are not disrupted by storage → more compute resources for applications
  - Locally attached like performance
  - Scaling of RDMA network
  - Converged network
    - No protocol translation and no additional dedicated hardware

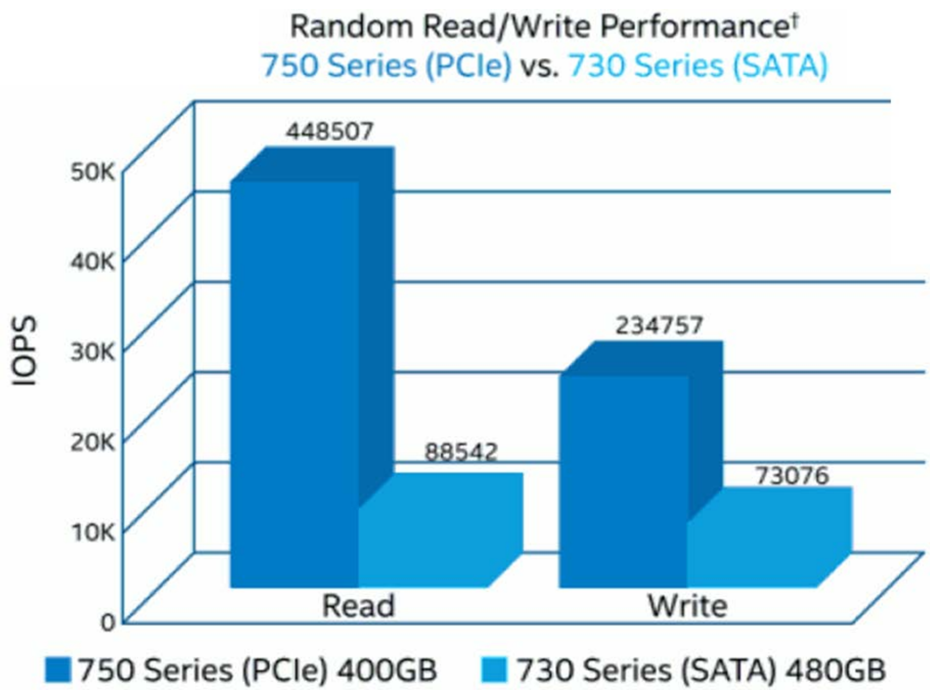


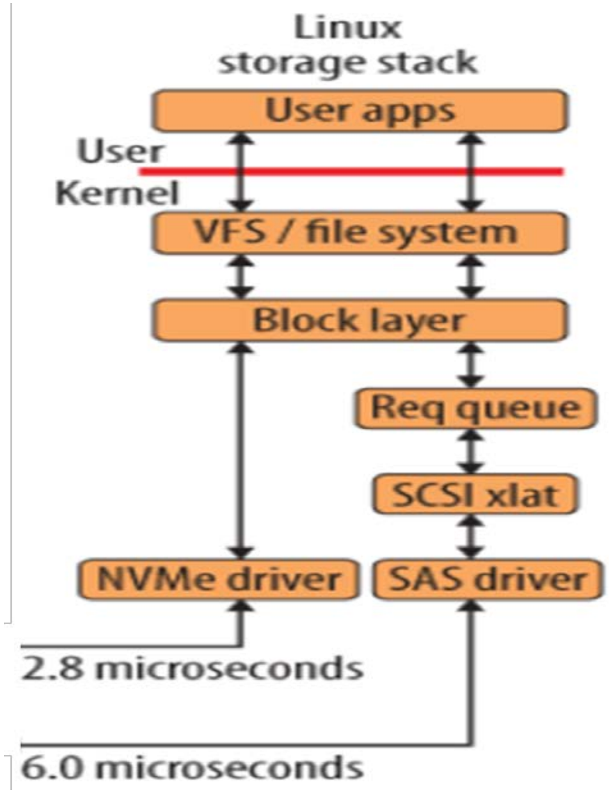
# What Makes NVMe Faster



# NVMe Performance

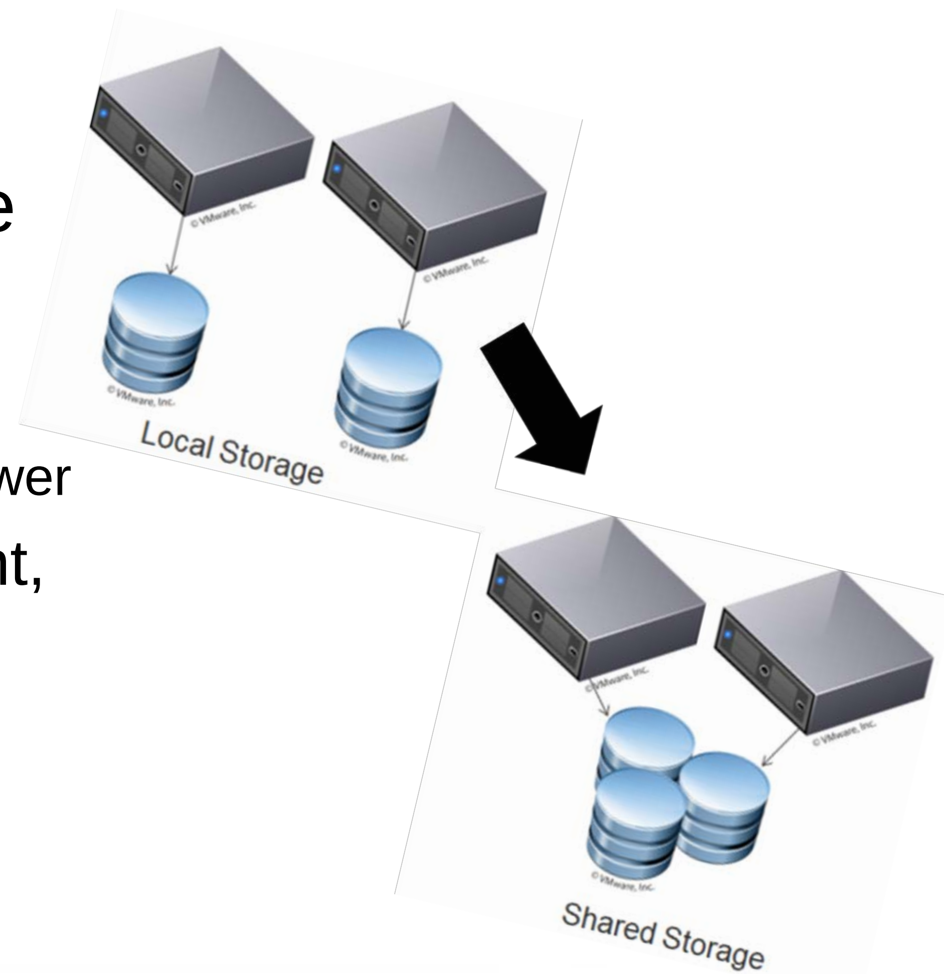
- 2x-4x more bandwidth, 50-60% lower latency, Up to 5x more IOPS



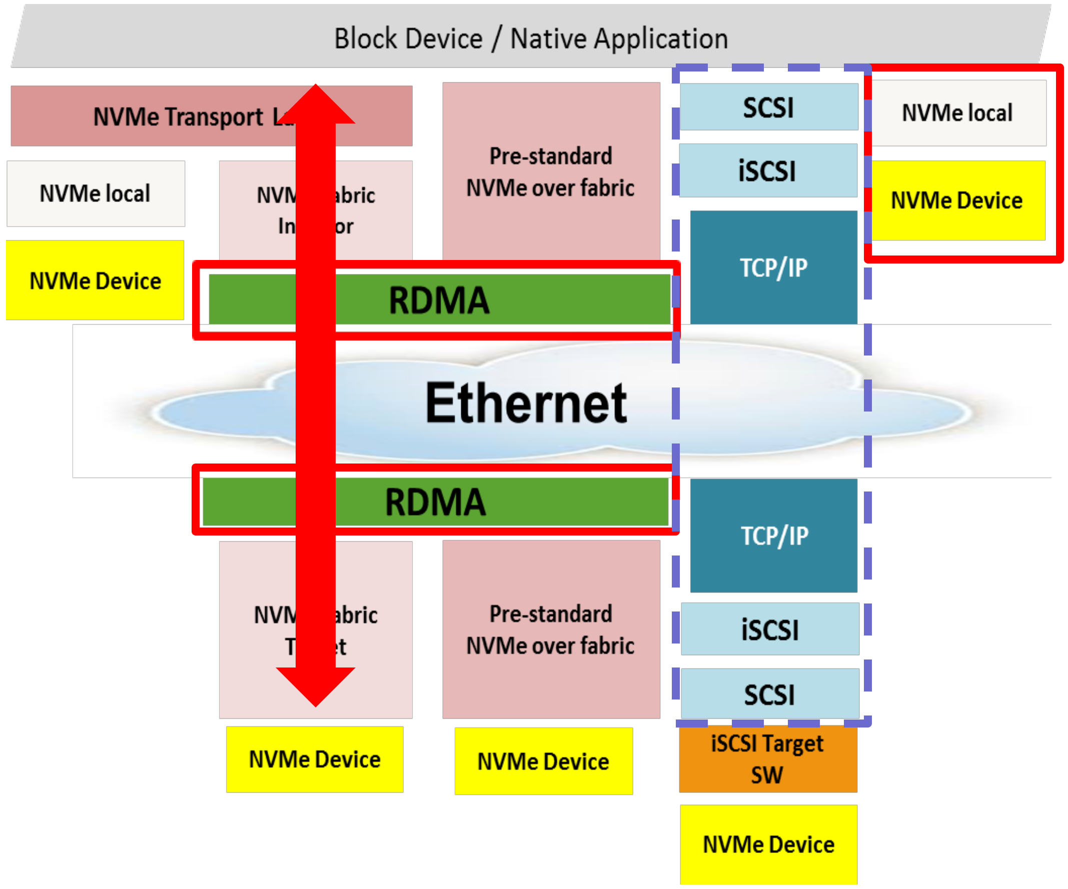
# NVMf is the Logical and Historical next step

- Sharing NVMe based storage across multiple servers/CPUs
  - Better utilization
    - capacity, rack space, power
  - Scalability, management, fault isolation
- NVMf Standard 1.0 was completed in early June 2016

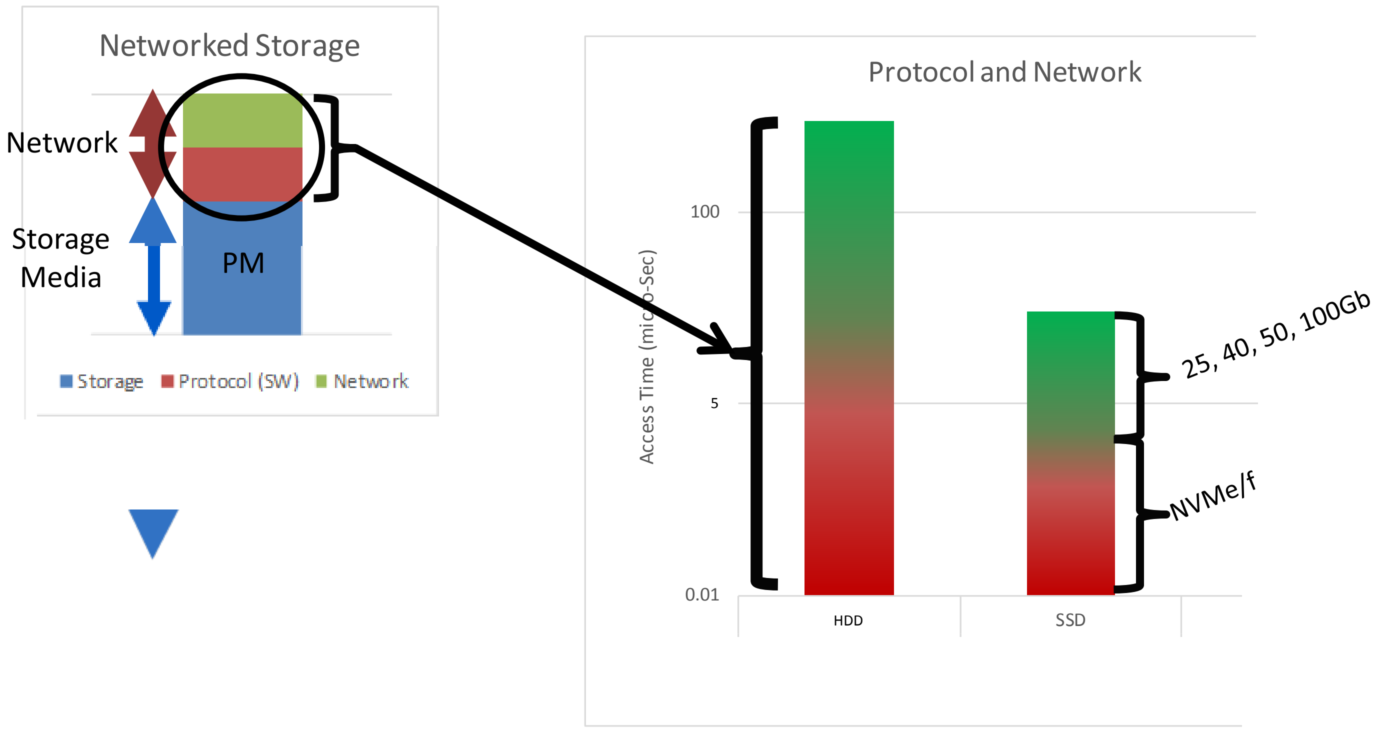


# How Does NVMf Maintain Performance

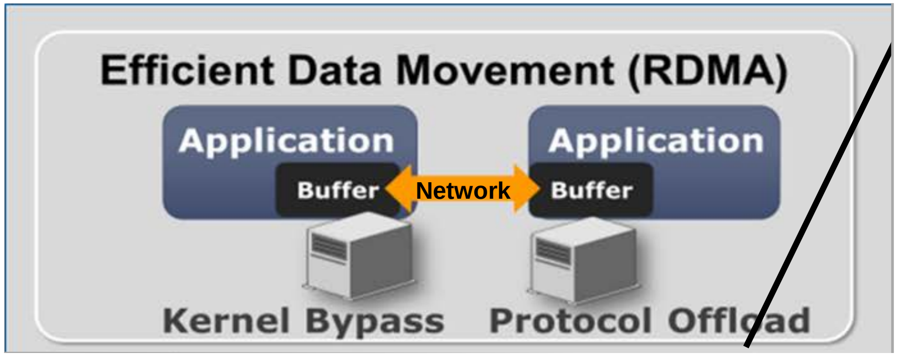
- The idea is to extend the efficiency of the local NVMe interface over a fabric
  - Ethernet or IB
  - NVMe commands and data structures are transferred end to end
- Relies on RDMA to match DAS performance
  - Bypassing TCP/IP



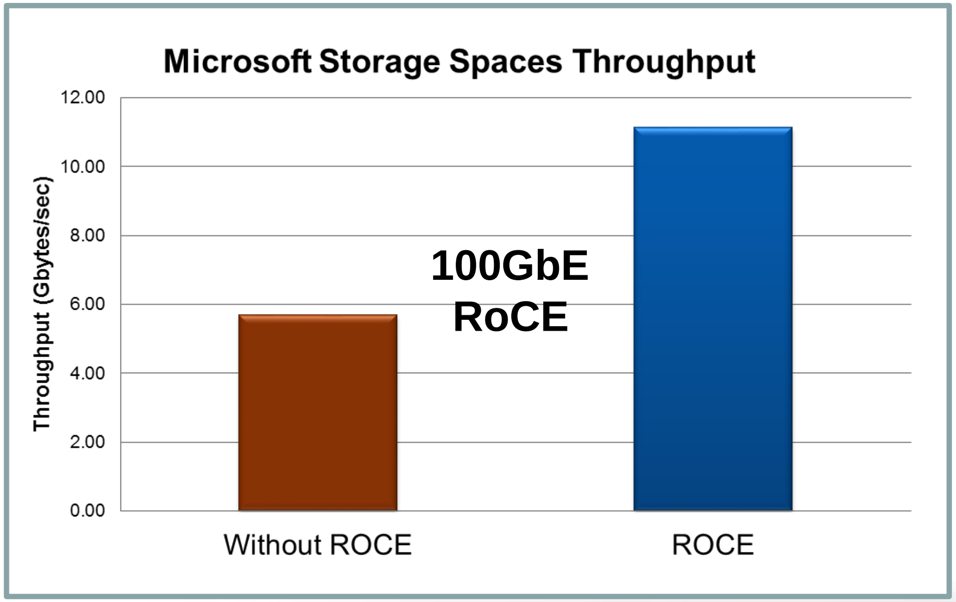
# Why not Traditional TCP/IP Network Stack



# What is RDMA



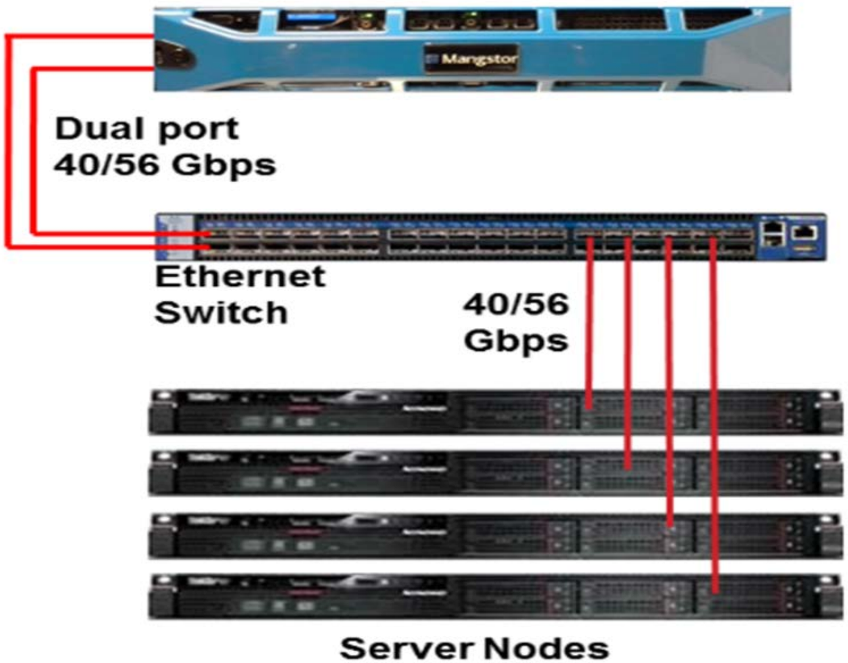
Efficient Data Movement (RDMA)
Application
Application
Buffer
Network
Buffer
Kernel Bypass
Protocol Offload

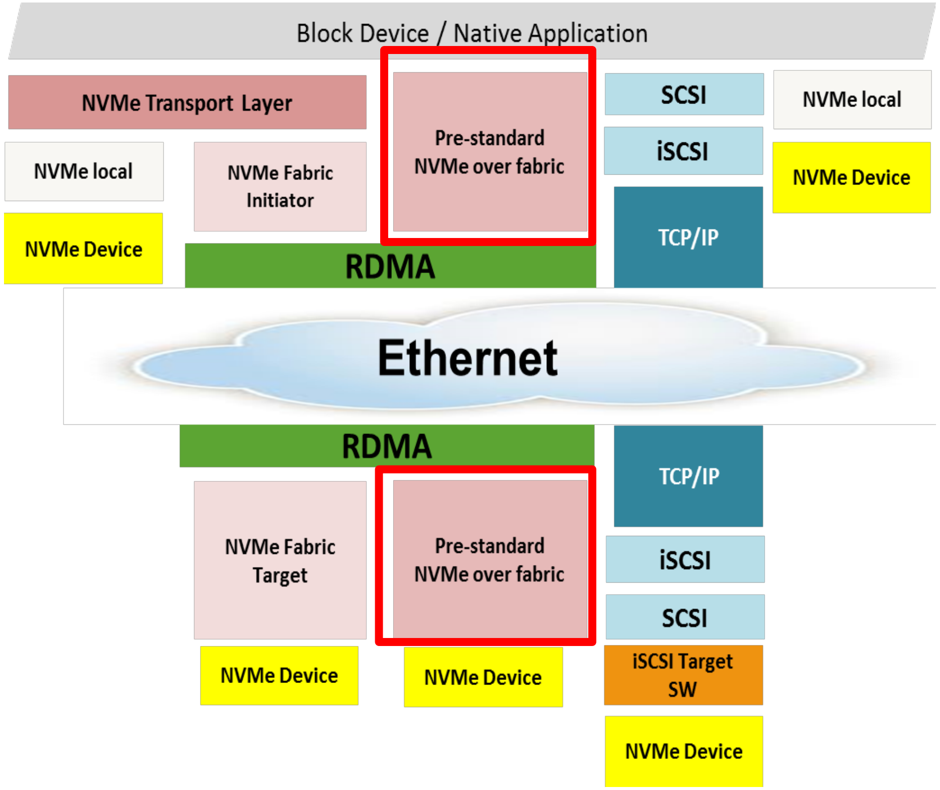



Microsoft Storage Spaces Throughput
Throughput (Gbytes/sec)
12.00
10.00
8.00
6.00
4.00
2.00
0.00
100GbE
RoCE
Without ROCE
ROCE

# Early Pre-standard Demonstrations

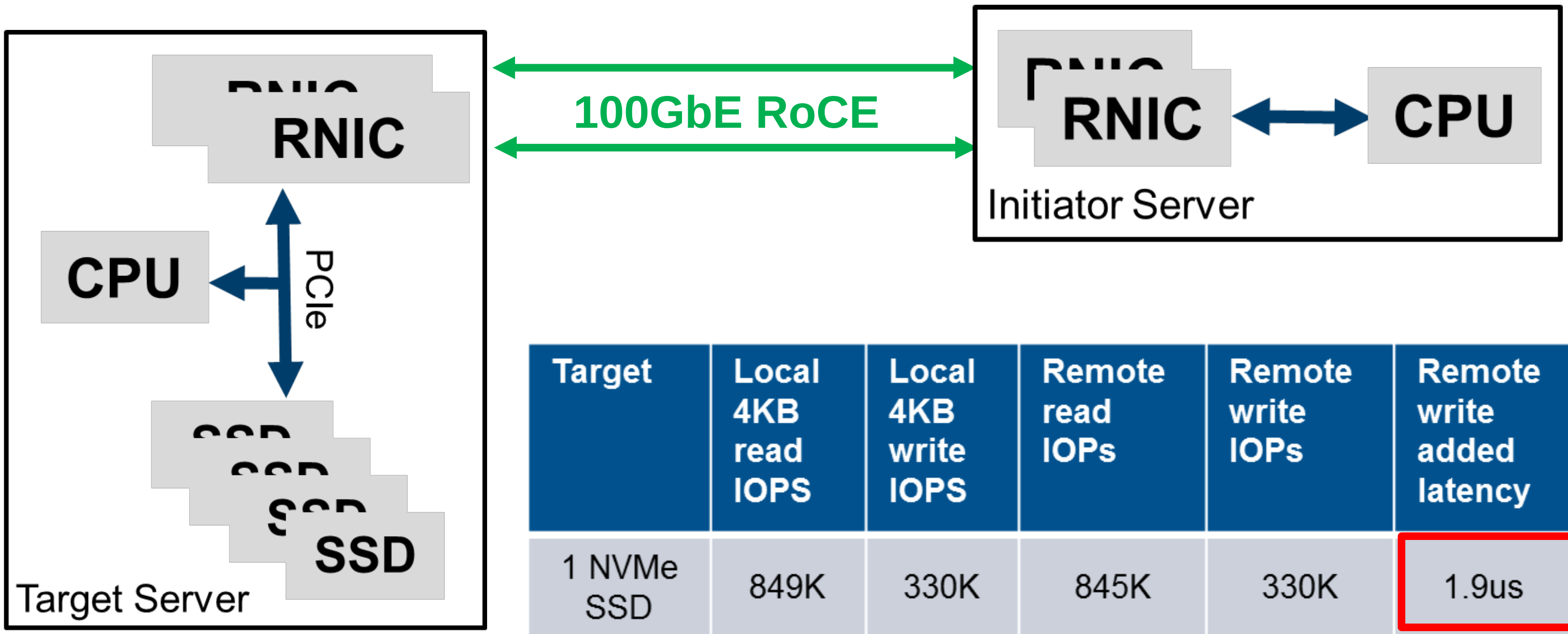
- April 2015
  - NAB Las Vegas




- 10Gb/s Reads, 8Gb/s Writes
- 2.5M Random Read 4 KB IOPs
- Latency ~8usec over local

# Micron FMS 2015 Demonstration




| Target | Local 4KB read IOPS | Local 4KB write IOPS | Remote read IOPs | Remote write IOPs | Remote write added latency | Remote read added latency |
|---|---|---|---|---|---|---|
| 1 NVMe SSD | 849K | 330K | 845K | 330K | 1.9us | 4.76us |
| 4 NVMe SSDs | 3406K | 1333K | 3388K | 1332K | N/A | N/A |

# Pre-standard Drivers Converge to V1.0

| Demo | NVMe Hardware | Software / Drivers | Network |
|---|---|---|---|
| Mangstor | Mangstor | Mangstor NVMeoF | RoCE or IB |
| PMC Sierra | PMC | PMC NVMeoF | 40Gb RoCE |
| HGST | HGST | HGST NVMeoF | 56Gb InfiniBand |
| Micron | Micron | Mellanox NBDx | 100Gb RoCE |
| Memblaze | Memblaze | Mellanox NBDx | 40Gb RoCE |
| Samsung at FMS15 | Samsung | iSER / Ceph / SMB Direct | 40Gb RoCE |
| Intel at IDF14 | Intel | Intel/Chelsio NVMeoF | 40Gb iWARP |
| Stealth startups | Any / Intel NVMe | Startup's NVMeoF | 40Gb RoCE |

# NVMf Standard 1.0 Community Open Source Driver Development

Groups ▾ ProjectView ▾

Workspace » All Groups » My Groups » Working Group - Fabrics Linux Driver

## Working Group - Fabrics Linux Driver

**Group Info**

Group Chair: Bob Beauchamp, EMC

**Group Email Addresses**

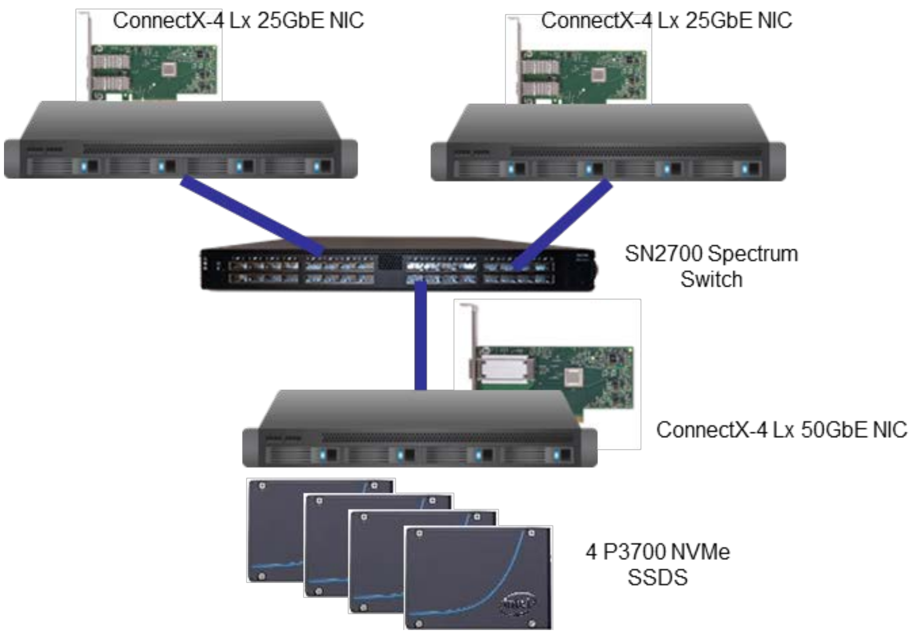
Post message: fabrics_linux_driver@nvmexpress.org

Contact chair: fabrics_linux_driver-chair@nvmexpress.org

- Mellanox
- Intel
- HGST
- EMC
- Apeiron Data Systems
- Broadcom Corporation
- Chelsio Communications, Inc
- Excelero
- Hewlett Packard Enterprise
- Kazan Networks
- Kenneth Okin Consulting
- Mangstor
- NetApp
- Oracle America Inc.
- PMC
- Qlogic Corporation
- Samsung
- SK hynix Inc.

# Performance with Community Driver

- Two compute nodes
  - ConnectX4-LX 25Gbps port
- One storage node
  - ConnectX4-LX 50Gbps port
  - 4 X Intel NVMe devices (P3700/750 series)
- Nodes connected through switch

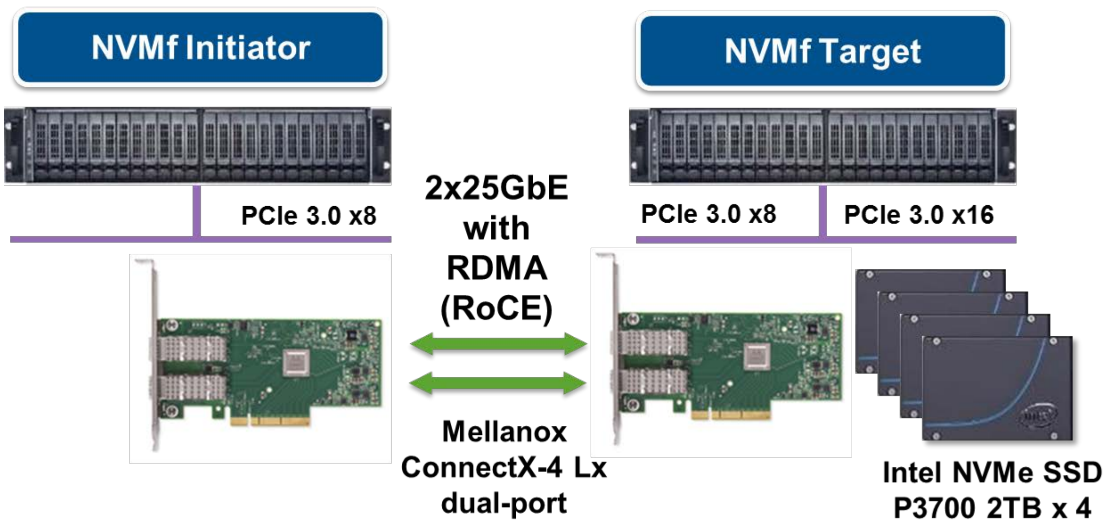


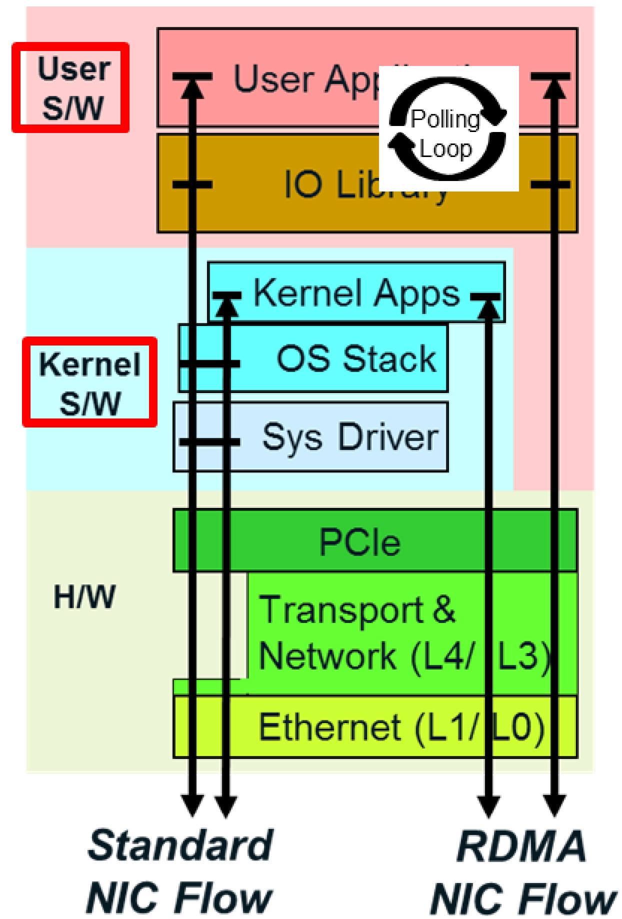

| Added fabric latency |
|---|
| ~12us |

| | Bandwidth (target) | IOPS (target) | # online cores | Each core utilization |
|---|---|---|---|---|
| BS = 4KB, 16 jobs, IO depth = 64 | 5.2GB/sec | 1.3M | 4 | 50% |

# Kernel & User Based NVMeoF

SPDK






SDC 16

# Some NVMeoF Demos at FMS & IDF 2016

**Flash Memory Summit**

- E8 Storage
- Mangstor
  - With initiators from VMs on VMware ESXi
- Micron
  - Windows & Linux initiators to Linux target
- Newisis (Sanmina)
- Pavilion Data
  - in Seagate booth

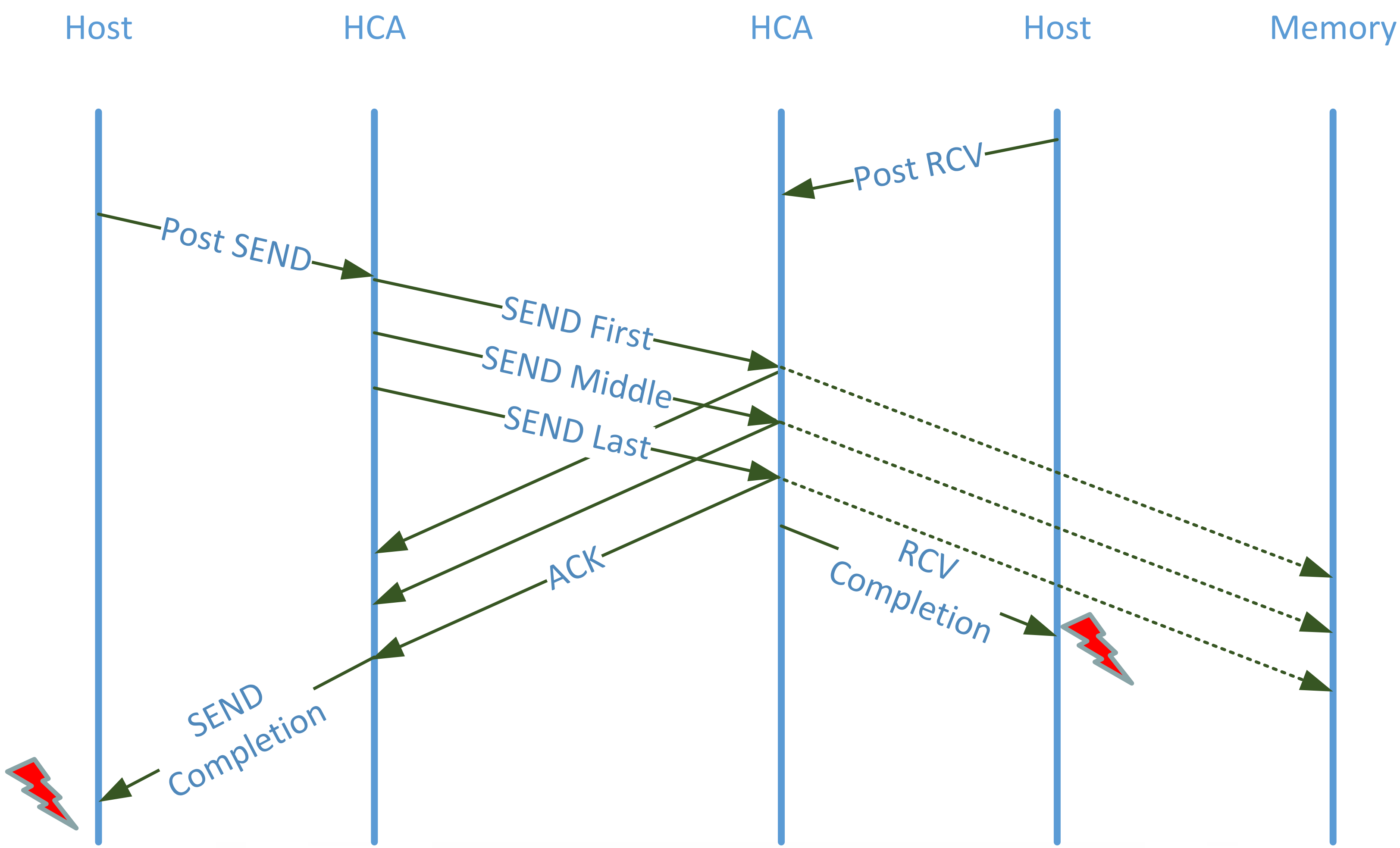
**Intel Developer Forum**

- E8 Storage
- HGST (WD)
  - NVMeoF on InfiniBand
- Intel: NVMe over Fabrics with SPDK
  - Mellanox 100GbE NICs
- Newisis (Sanmina)
- Samsung
- Seagate

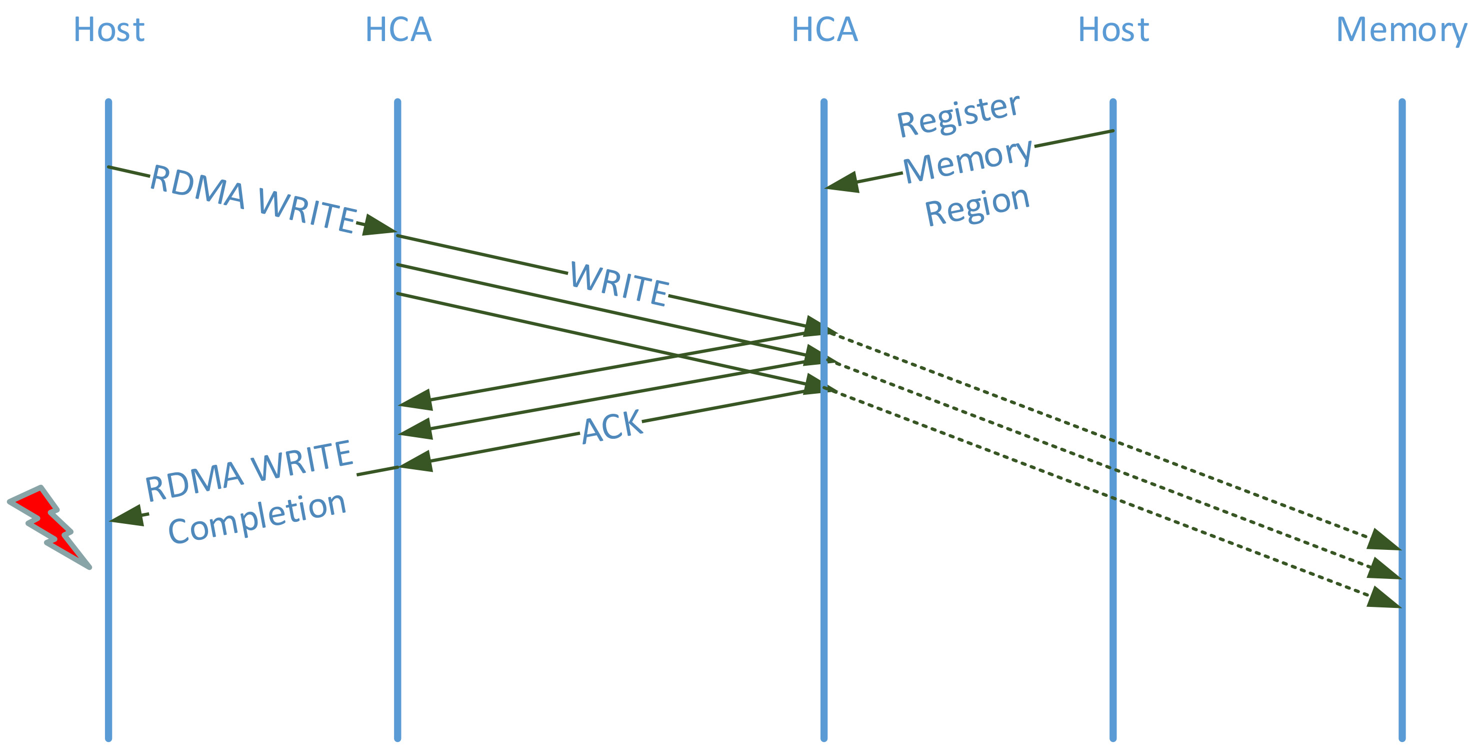
# RDMA

- Transport built on simple primitives deployed for 15 years in the industry
  - **Queue Pair (QP)** – RDMA communication end point
  - **Connect** for establishing connection mutually
  - RDMA **Registration** of memory region (REG_MR) for enabling virtual network access to memory
  - **SEND** and **RCV** for reliable two-sided messaging
  - RDMA **READ** and RDMA **WRITE** for reliable one-sided memory to memory transmission

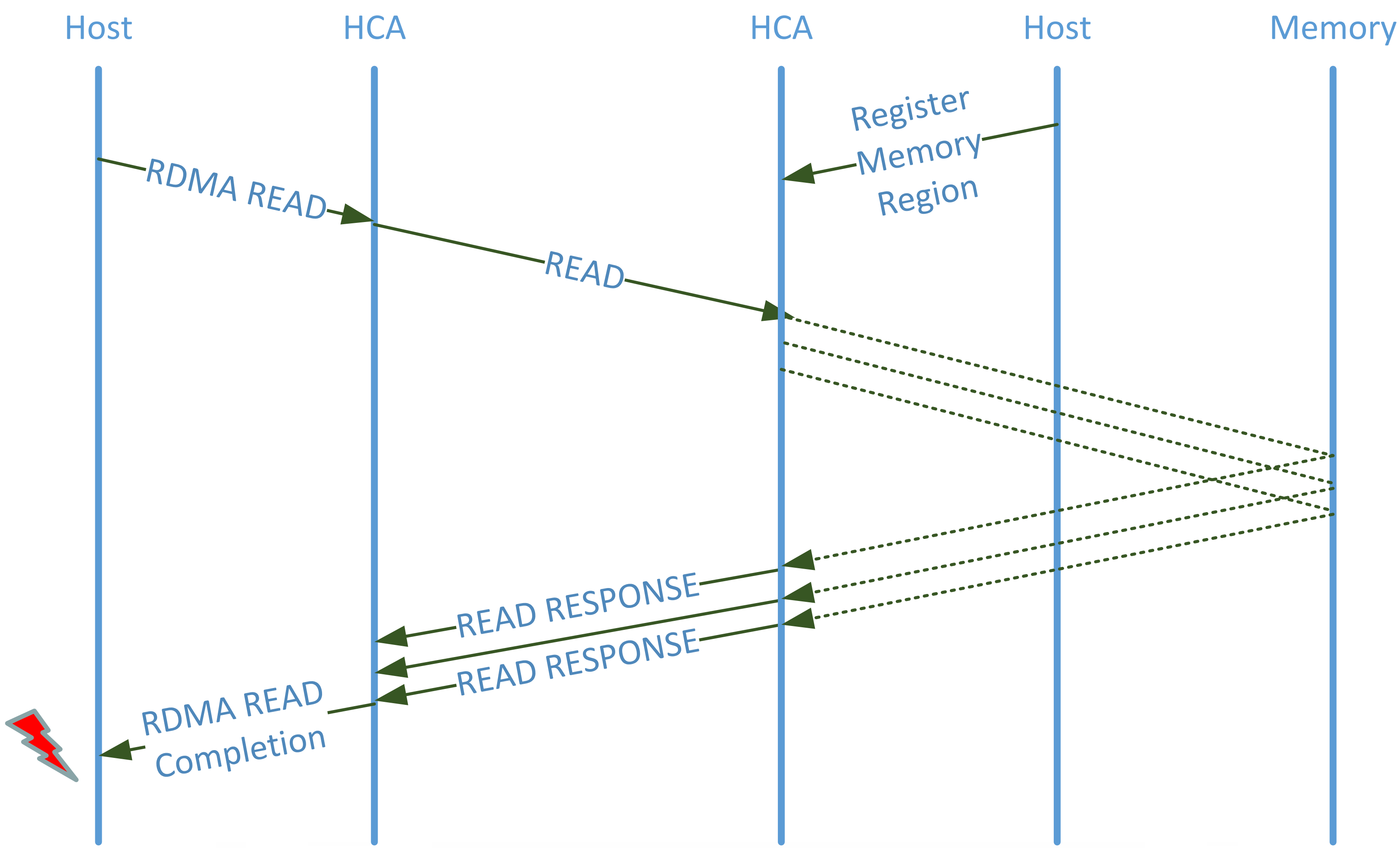
# SEND/RCV


Host
HCA
HCA
Host
Memory
Post RCV
Post SEND
SEND First
SEND Middle
SEND Last
ACK
RCV
Completion
SEND
Completion

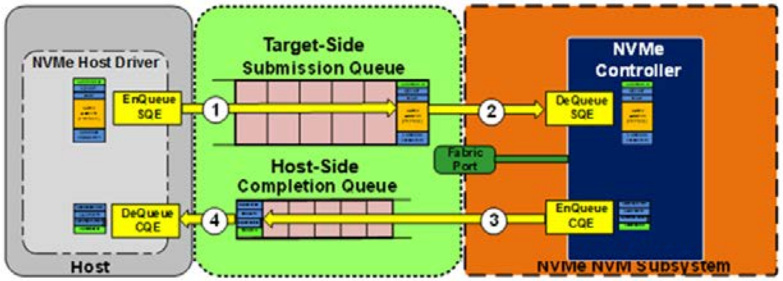
# RDMA WRITE

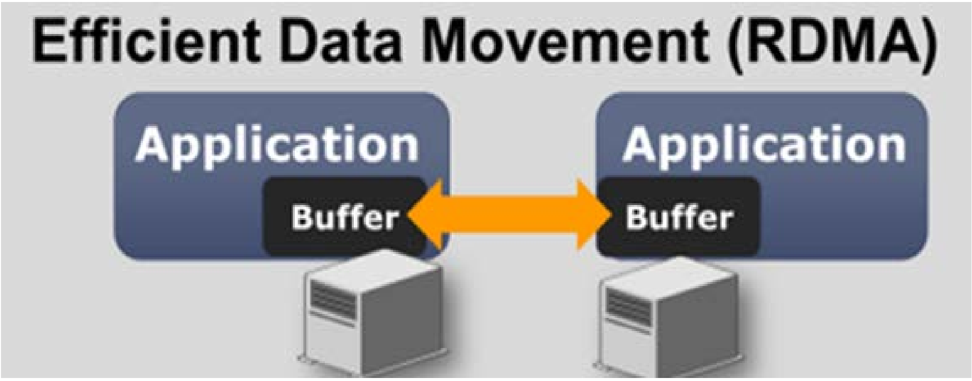

Host
HCA
HCA
Host
Memory
Register Memory Region
RDMA WRITE
WRITE
ACK
RDMA WRITE Completion

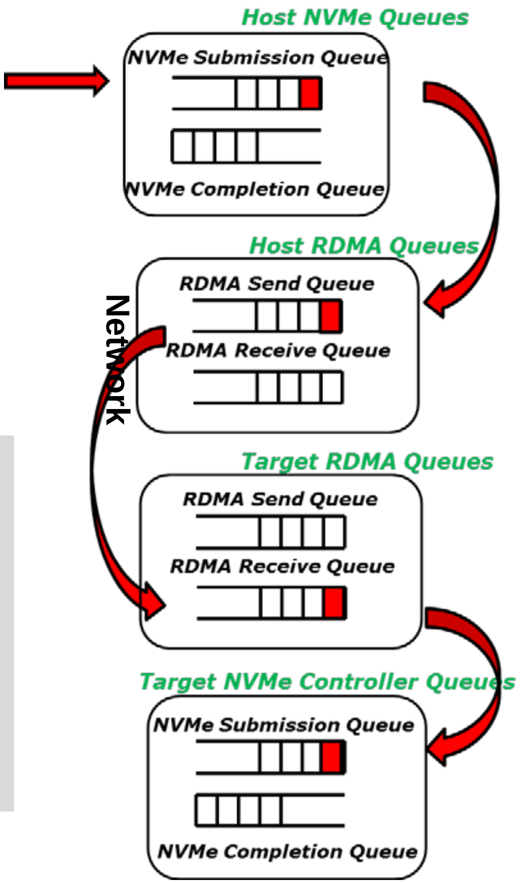
# RDMA READ


Host
HCA
HCA
Host
Memory
Register Memory Region
RDMA READ
READ
READ RESPONSE
READ RESPONSE
RDMA READ Completion

# NVMe and NVMeoF Fit Together Well

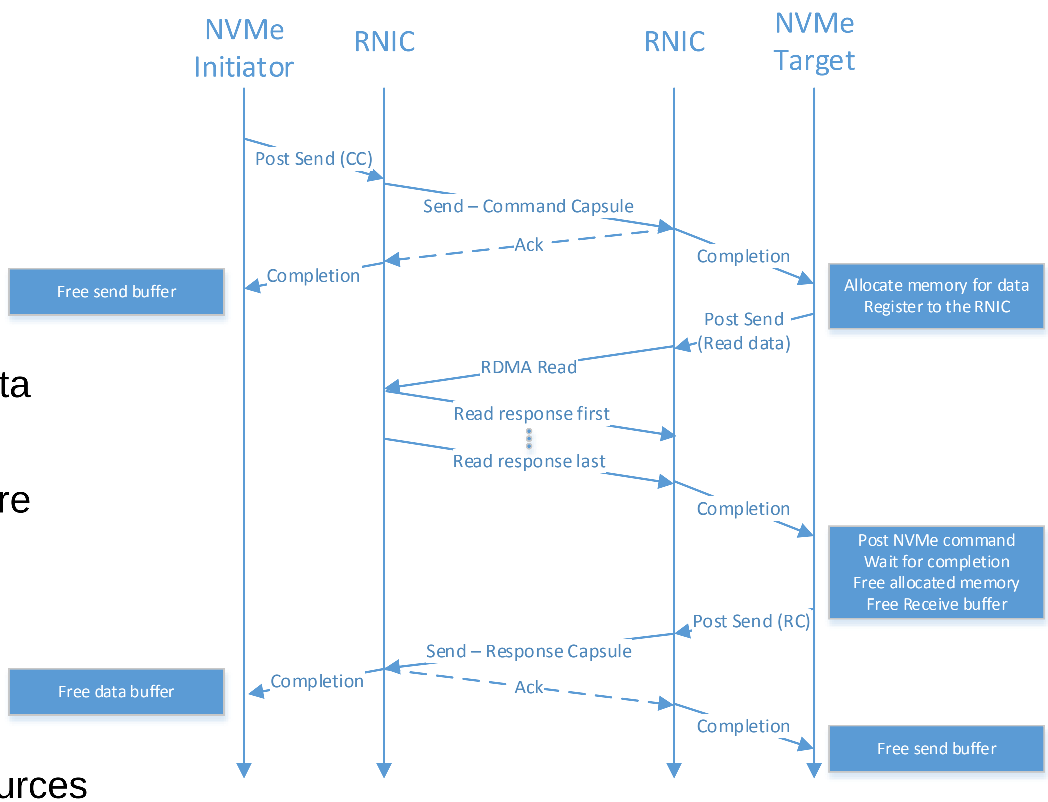


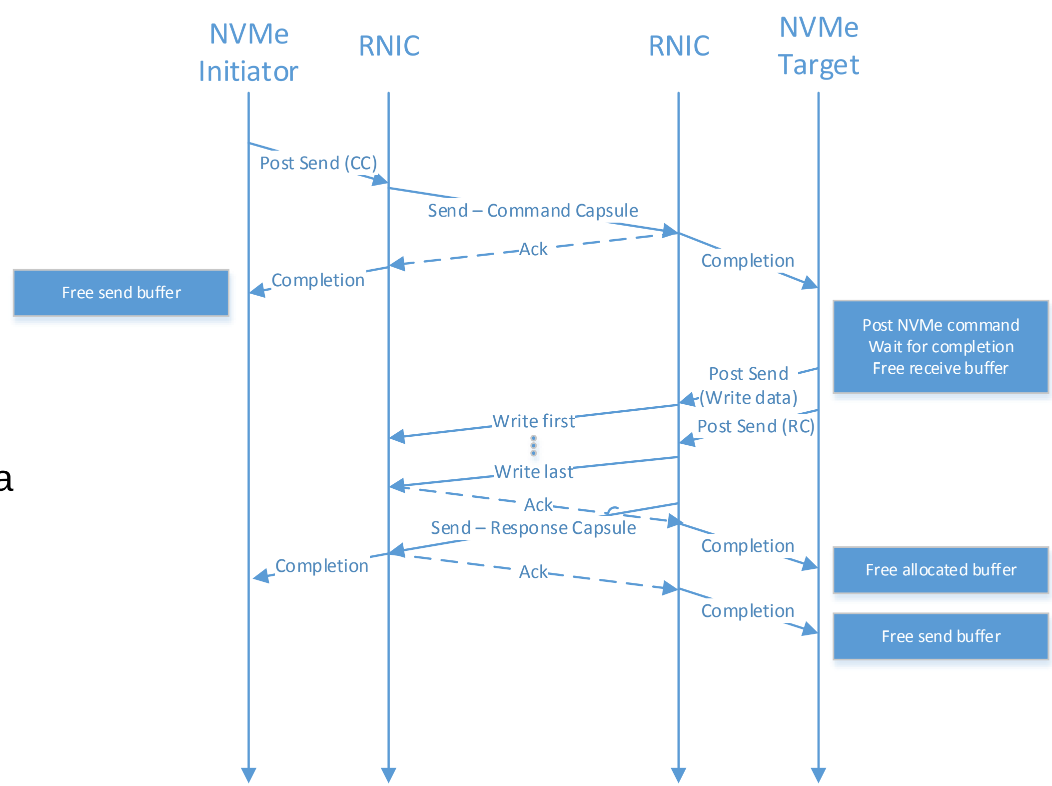
# NVMe-OF IO WRITE

- Host
  - Post SEND carrying Command Capsule (CC)
- Subsystem
  - Upon RCV Completion
    - Allocate Memory for Data
    - Post RDMA READ to fetch data
  - Upon READ Completion
    - Post command to backing store
  - Upon SSD completion
    - Send NVMe-OF RC
    - Free memory
  - Upon SEND Completion
    - Free CC and completion resources



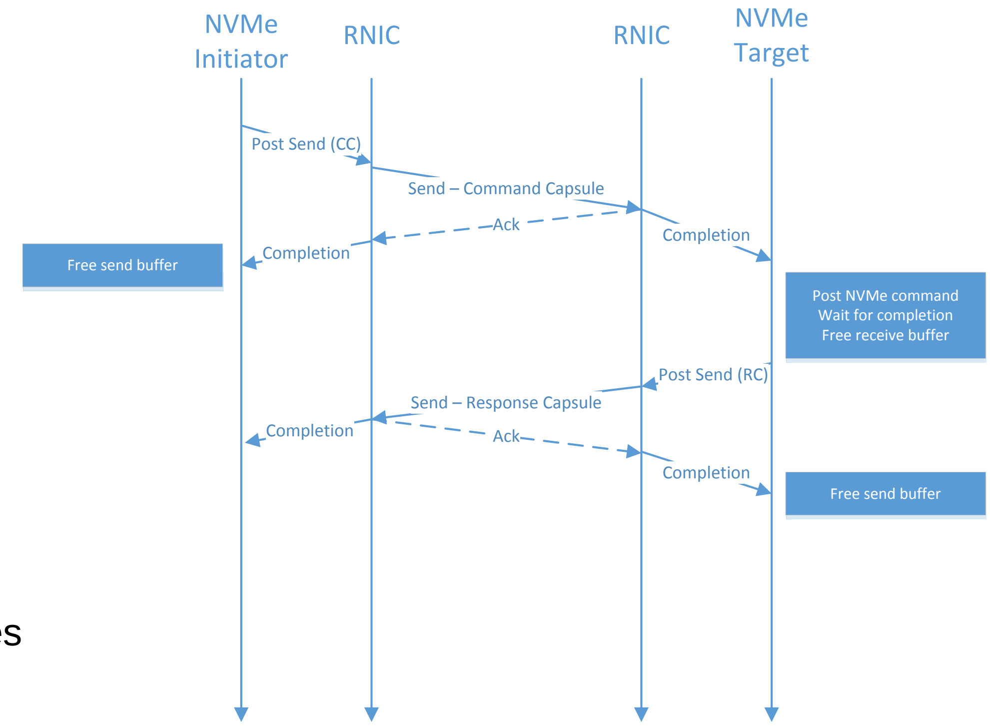
# NVMe-OF IO READ

- Host
  - Post SEND carrying Command Capsule (CC)
- Subsystem
  - Upon RCV Completion
    - Allocate Memory for Data
    - Post command to backing store
  - Upon SSD completion
    - Post RDMA Write to write data back to host
    - Send NVMe RC
  - Upon SEND Completion
    - Free memory
    - Free CC and completion resources



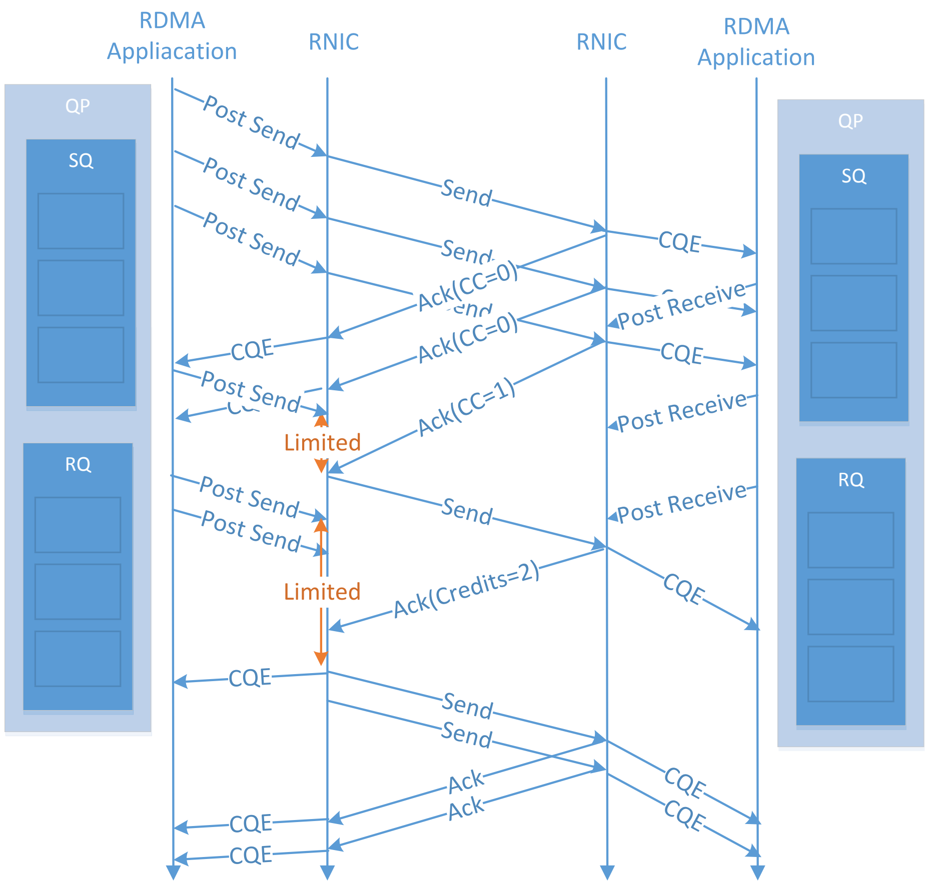
# NVMe-OF IO WRITE IN-Capsule

- Host
  - Post SEND carrying Command Capsule (CC)
- Subsystem
  - Upon RCV Completion
    - Allocate Memory for Data
  - Upon SSD completion
    - Send NVMe-OF RC
    - Free memory
  - Upon SEND Completion
    - Free CC and completion resources



# E2E Flow Control

- Receive resources credits are transported from RDMA responder to RDMA requester
  - Piggybacked in the acknowledgments
- Requester limit it's transmission to number of outstanding receive WQEs
- Benefits
  - Optimizes the amount of outstanding buffers assuming single connection, independently from the application flow control
  - Optimizes the use of the fabric

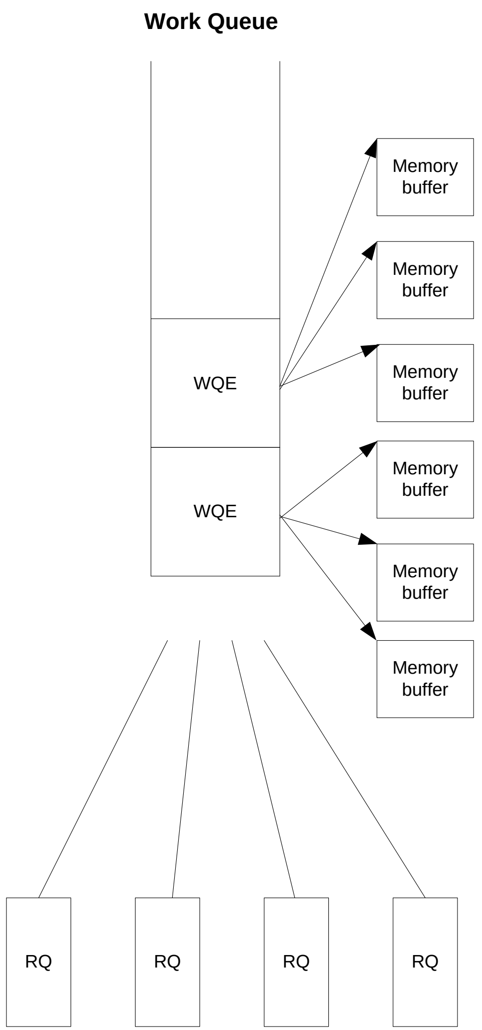


# Memory Consumption

- Staging Buffer
  - Intermediate buffer allocated for data fetch from Host / Subsystem
  - Allocated on demand, function of the parallelism of the backing store
- Receive Buffer
  - In RC QP, allocated according to the maximum parallelism of a single queue
  - Scales linearly with number of connections

# Shared Receive Queue

- Share receive buffering resources between QPs
  - According the the parallelism of the application
- E2E credits is being managed by RNR NACK with TO associated with the application latency
- Number of connections is the same as RC w/o SRQ



# Extended Reliable Connection Transport

- XRC SRQ
  - Application receive buffer
- XRC Initiator
  - Transport level reliability with a single XRC TGT
  - Capable of sending messages to the plurality of XRC SRQs in the target
- XRC Target
  - Transport level reliability with a single XRC INI
  - Spread the work across the XRC SRQ according to the packet

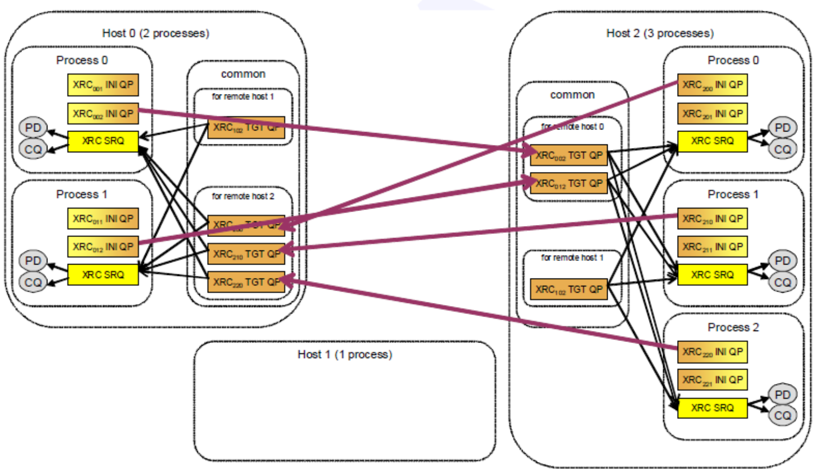



Figure 1 Extended Reliable Connected (XRC) Model

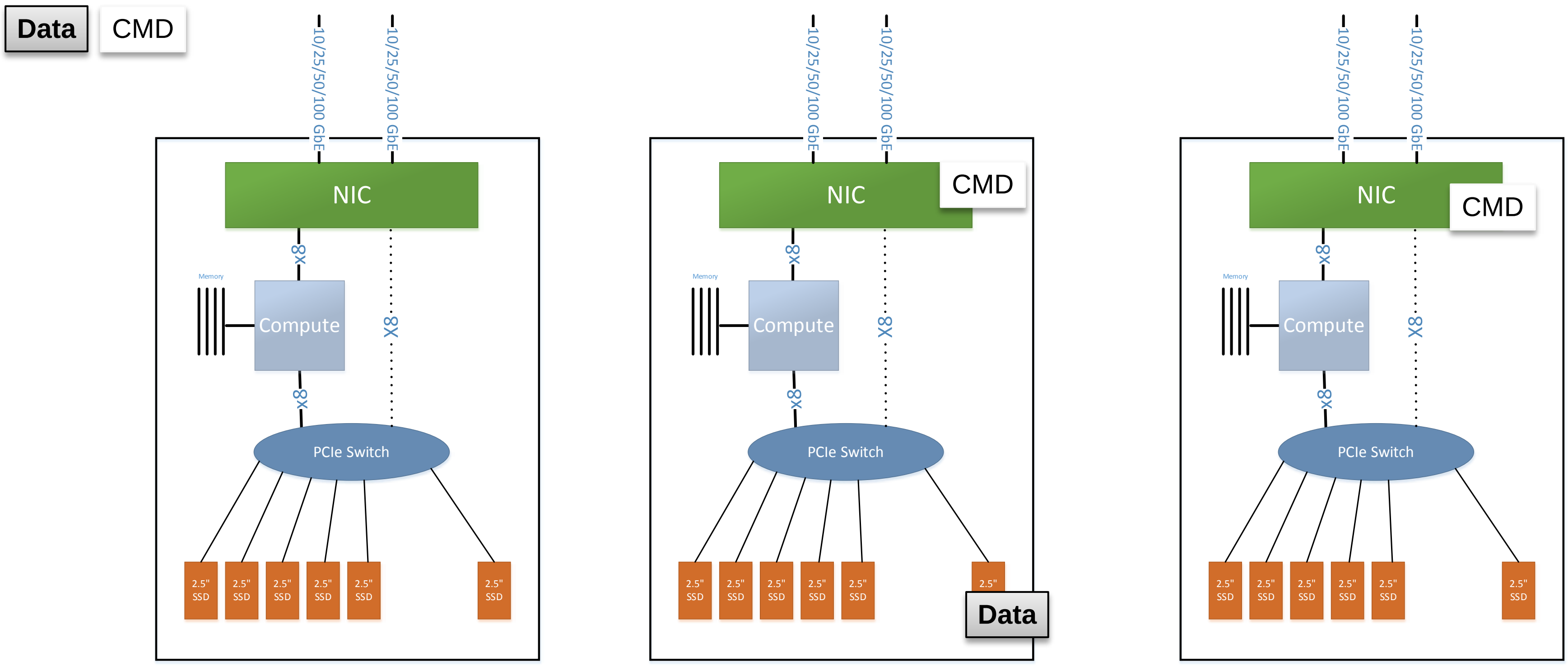
Scales down the amount of connection from per drive per core to per subsystem port times number of hosts

# CMB Introduction

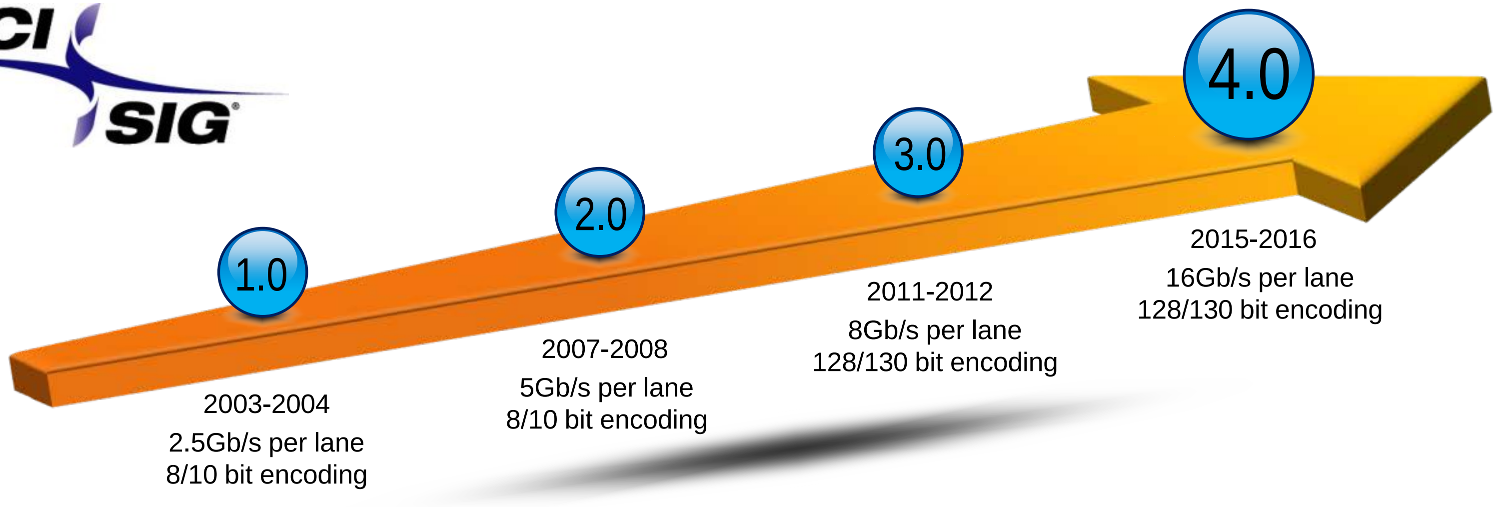
- Internal memory of the NVMe devices exposed over the PCIe
- Few MB are enough to buffer the PCIe bandwidth for the latency of the NVMe device
  - Latency ~ 100-200usec, Bandwidth ~ 25-50 GbE → Capacity ~ 2.5MB
- Enabler for peer to peer communication of data and commands between RDMA capable NIC and NVMe SSD

# Scalability of Memory Bandwidth Using Peer-Direct and CMB

IO Write with Peer-Direct, NVMf target offload and CMB



# PCIe 4.0 Accelerating CPU / Memory - Interconnect Performance




PCI EXPRESS 4.0

New Capabilities

- Higher Bandwidth (16 to 25Gb/s per lane)
- Cache Coherency
- Atomic Operations
- Advanced Power Management
- Memory Management…

# Conclusions

- Future Storage solutions will be able to deliver DAS storage performance over a network:
  - NVMe SSDs – new NVMe protocol eliminates HDD legacy bottlenecks
  - Fast network – "Faster storage needs faster networks!"
  - NVMf with RDMA – new NVMf protocol running over RDMA is within microseconds of DAS

# Thanks!

robd@mellanox.com
idanb@Mellanox.com