NVMe over Fabrics Architecture and HPC Applications

Idan Burstein, Storage @ Mellanox CTO

Mellanox TECHNOLOGIES
Connect. Accelerate. Outperform.™

Exponential Data Growth Everywhere

Indigo

Network Processors

BlueField

System on a Chip

Innova ConnectX

Adapters

Spectrum Quantum

Switches

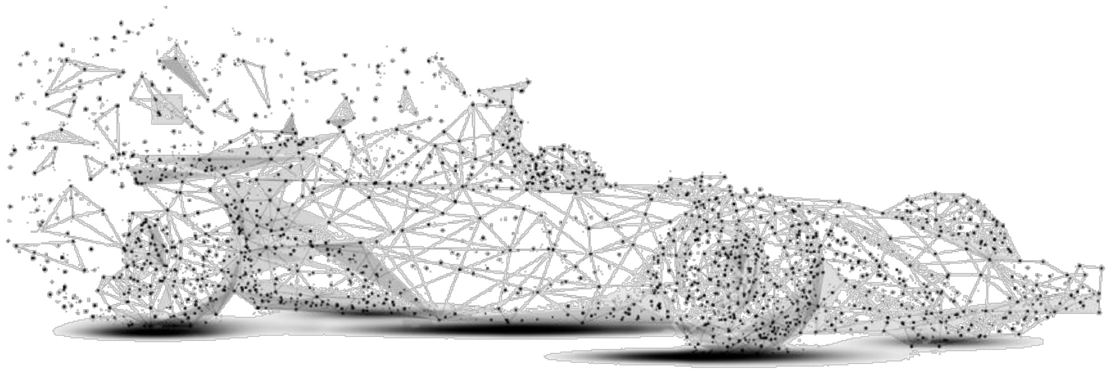
LinkX

Cables & Transceivers

Higher Data Speeds

Faster Data Processing

Better Data Security



HPC Storage Efficient Storage IO

- Total bandwidth available to an application from centralized storage solution is limiting applications
  - The higher the bandwidth, the faster a well-optimized application can read/write a large amount of data.
  - the IOPS (Input/Output Operations per Second) may become the limiting factor of performance
  - Amount of data for analytics is growing

- Improved I/O performance can help science applications in many ways:
  - Improved reliability through checkpoints
  - Accelerated application I/O performance
  - High performance staging area for large data analytics
  - Overlap application processing and data transition

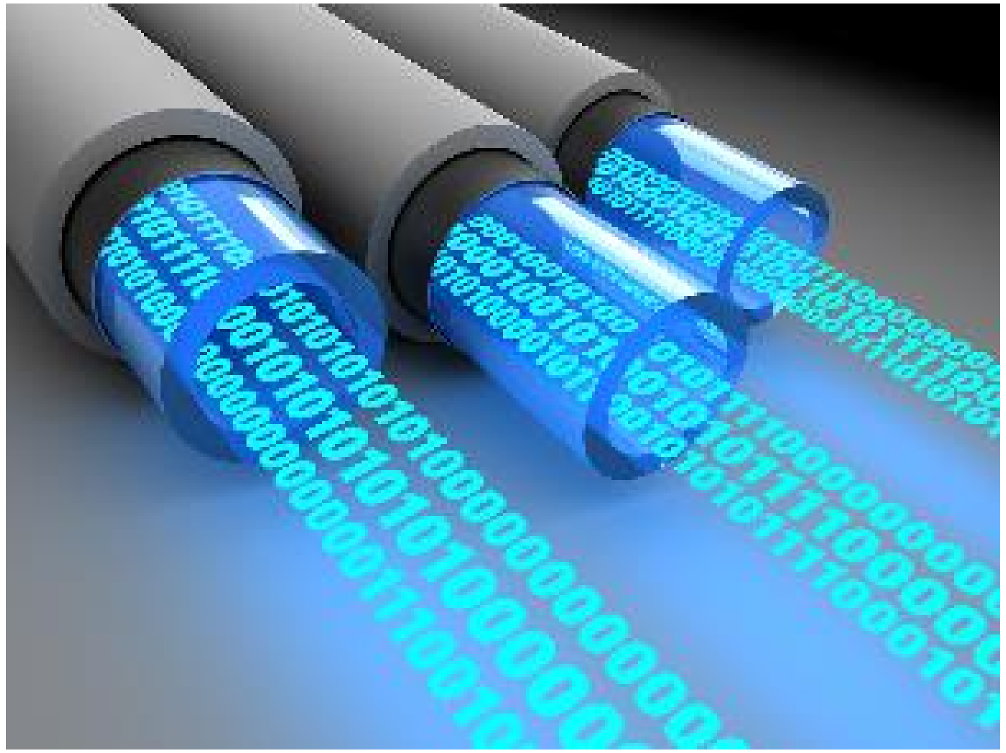
HPC Solution: Burst Buffer

Burst Buffer

- Why?
  - Absorb spikes in IO demand (i.e. checkpoints)
  - Drain/sink data from FS asynchronously (i.e. overlap)
  - On the fly analysis of data (i.e. plot the data on the fly)
  - Extend compute memory for data intensive analytics (i.e. artificial intelligence)

- What?
  - Higher performance storage tier from the parallel file system
  - More efficient access interface

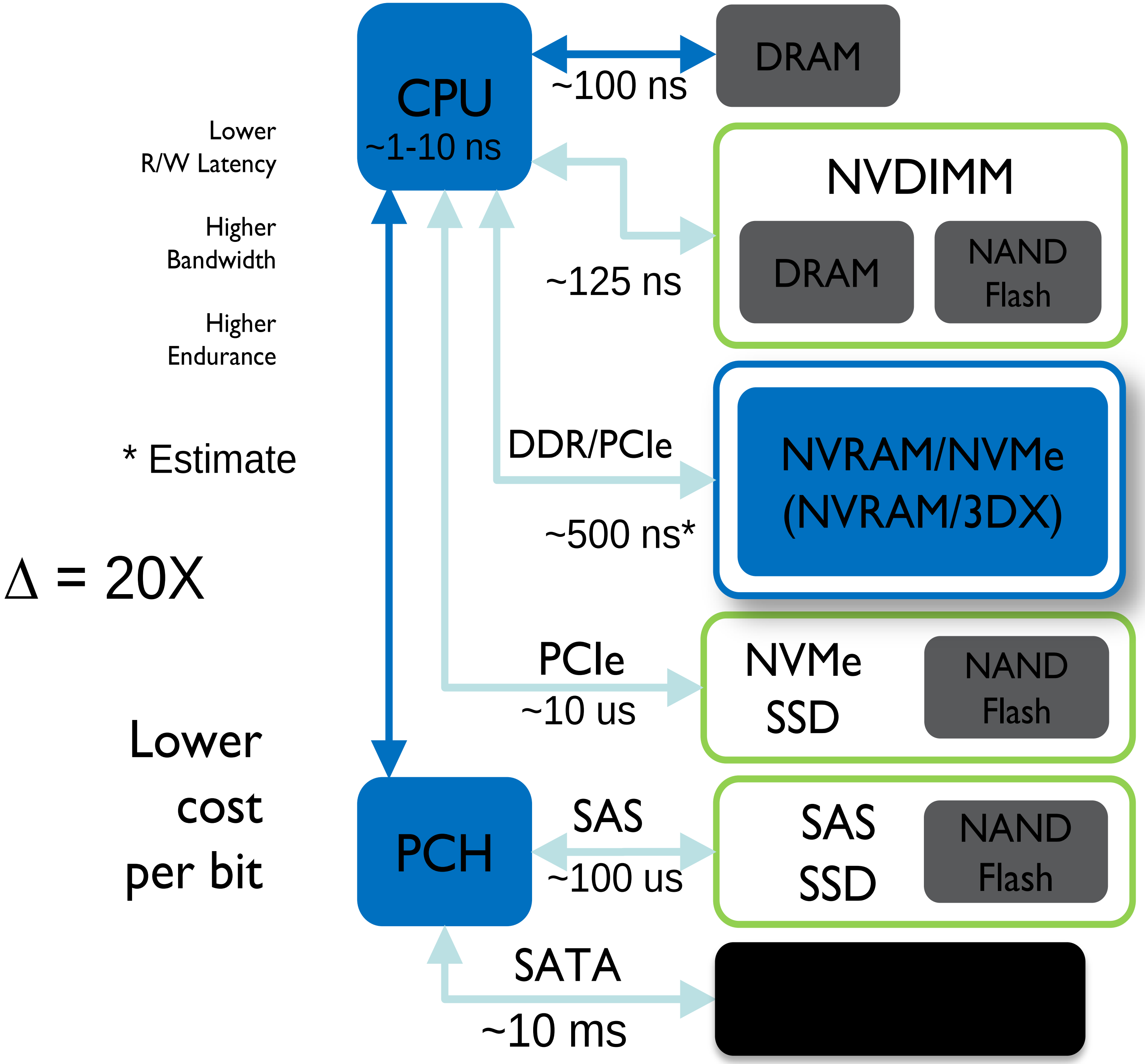
- How?
  - Overlapped async interface to the application
  - Storage could be
    - Locally attached in the nodes
    - Centralized within the compute cluster
    - High performance storage tier (highly tuned all flash array)



Storage Media/Controller Tiers

- NVDIMM / NVRAM
  - Byte addressable, low latency
  - RDMA Enables transparent access through the network
  - Low capacity

- NVMe
  - High performance
  - Asynchronous producer/consumer interface
  - Data is transferred by the device
  - Multi queue lockless interface



Lower R/W Latency

Higher Bandwidth

Higher Endurance

* Estimate

Δ = 20X

Lower cost per bit

CPU ~1-10 ns

~100 ns

DRAM

~125 ns

NVDIMM

DRAM

NAND Flash

DDR/PCIe

~500 ns*

NVRAM/NVMe (NVRAM/3DX)

PCIe ~10 us

NVMe SSD

NAND Flash

PCH

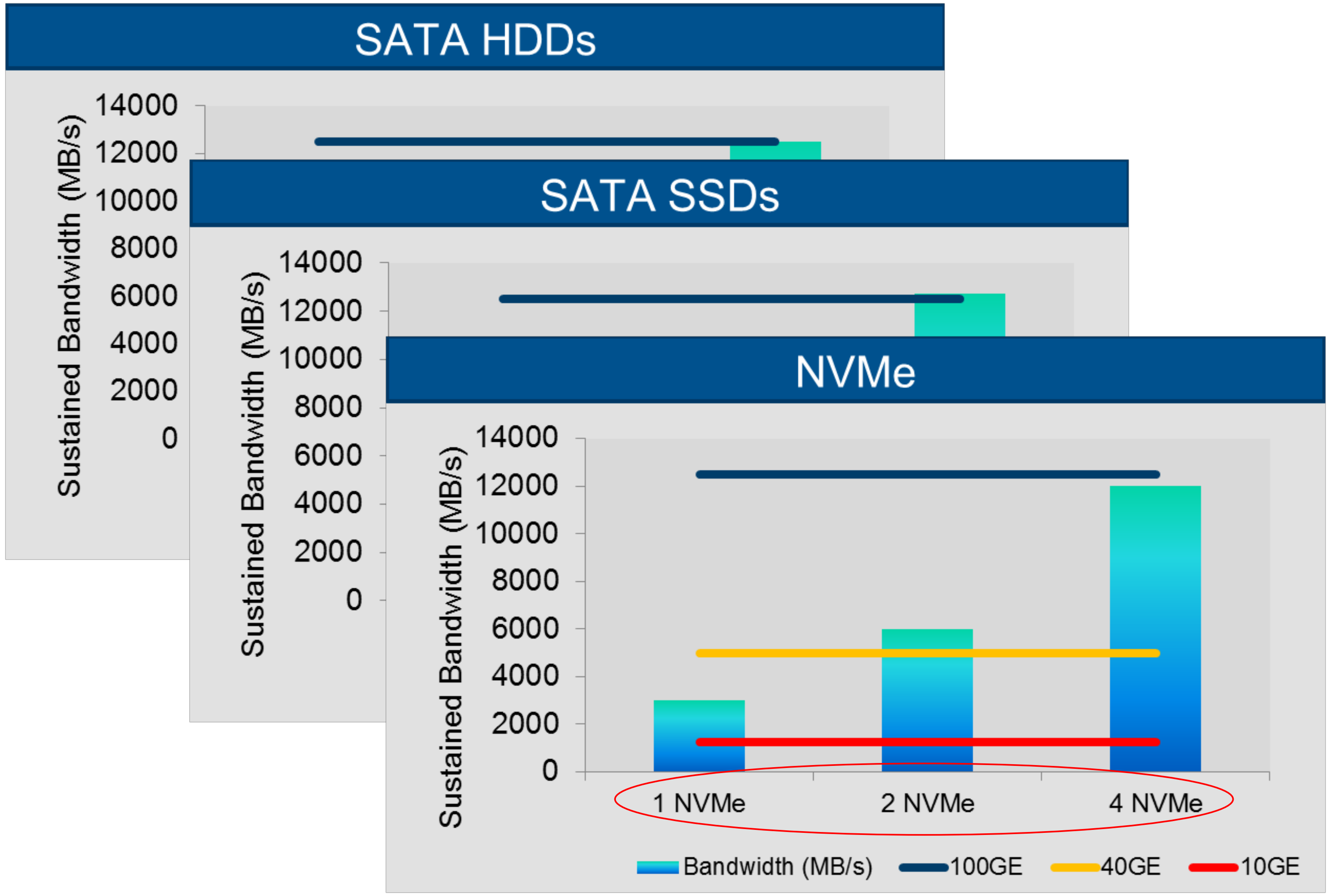
SAS ~100 us

SAS SSD

NAND Flash

SATA ~10 ms

Storage Performance Characteristics



SSDs move the Bottleneck from the Disk to the Network

nvm EXPRESS

NVMe OVER FABRICS POWERED BY RoCE

Faster Storage Needs Faster Networks

InfiniBand RDMA – Remote Direct Memory Access

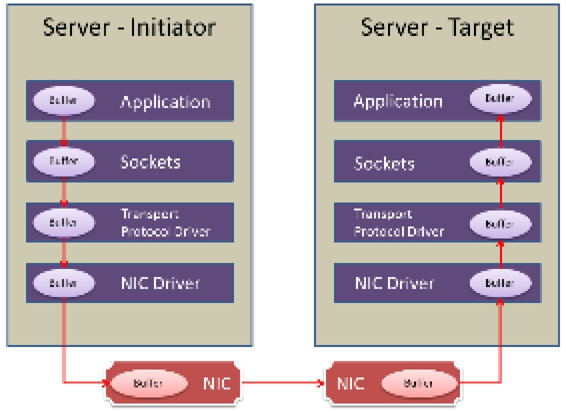
What is RDMA?

- Transport that enables
  - Direct Memory Access from Memory of one computer to another in hardware (READ/WRITE)
  - Transport offload for messaging (SEND/RCV)
- With interface that bypasses OS and TCP/IP stack, saves CPU cycles
- Results in Low Latency , high throughput and low CPU utilization

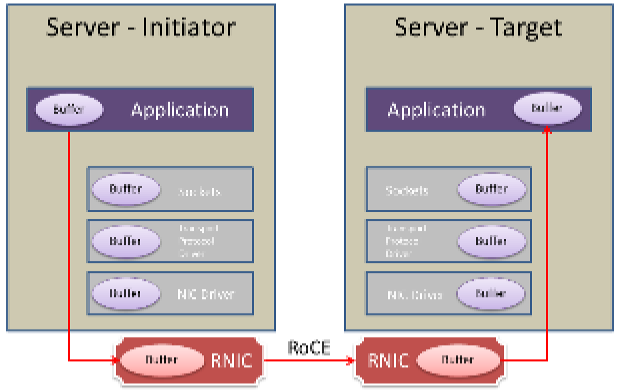
Why?

- CPU% is valuable for application, expensive to spend on data transfer
- Real time applications require low predictable latency
- Scalability, congestion control and QoS is being done in hardware
- The move to SSD has made Latency a factor in storage

TCP/IP



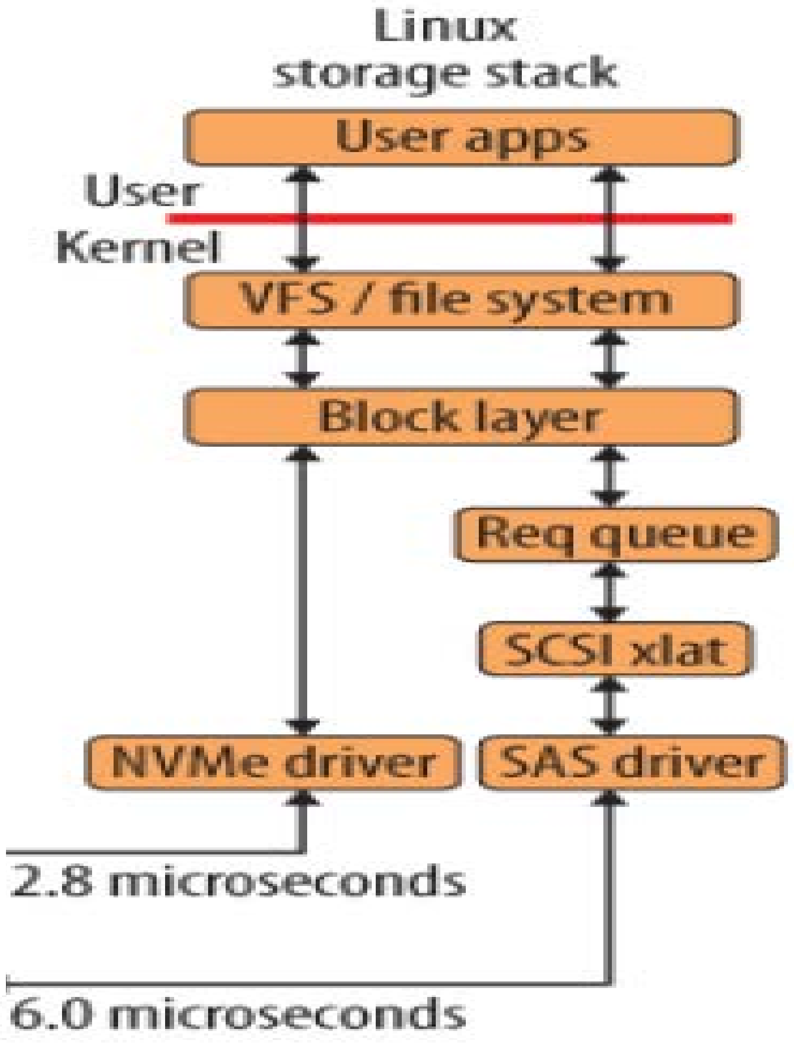
RDMA



Storage + RDMA = Awesome

NVMe Technology – Background

- Optimized for flash
  - Traditional SCSI designed for disk
  - NVMe bypasses unneeded layers
- NVMe outperforms SCSI stack
  - 2x-2.5x more bandwidth, 40-50% lower latency, Up to 3x more IOPS
- PCIe based Standardized API
  - Single optimized driver
  - No need for HBA
  - Interoperable with networking

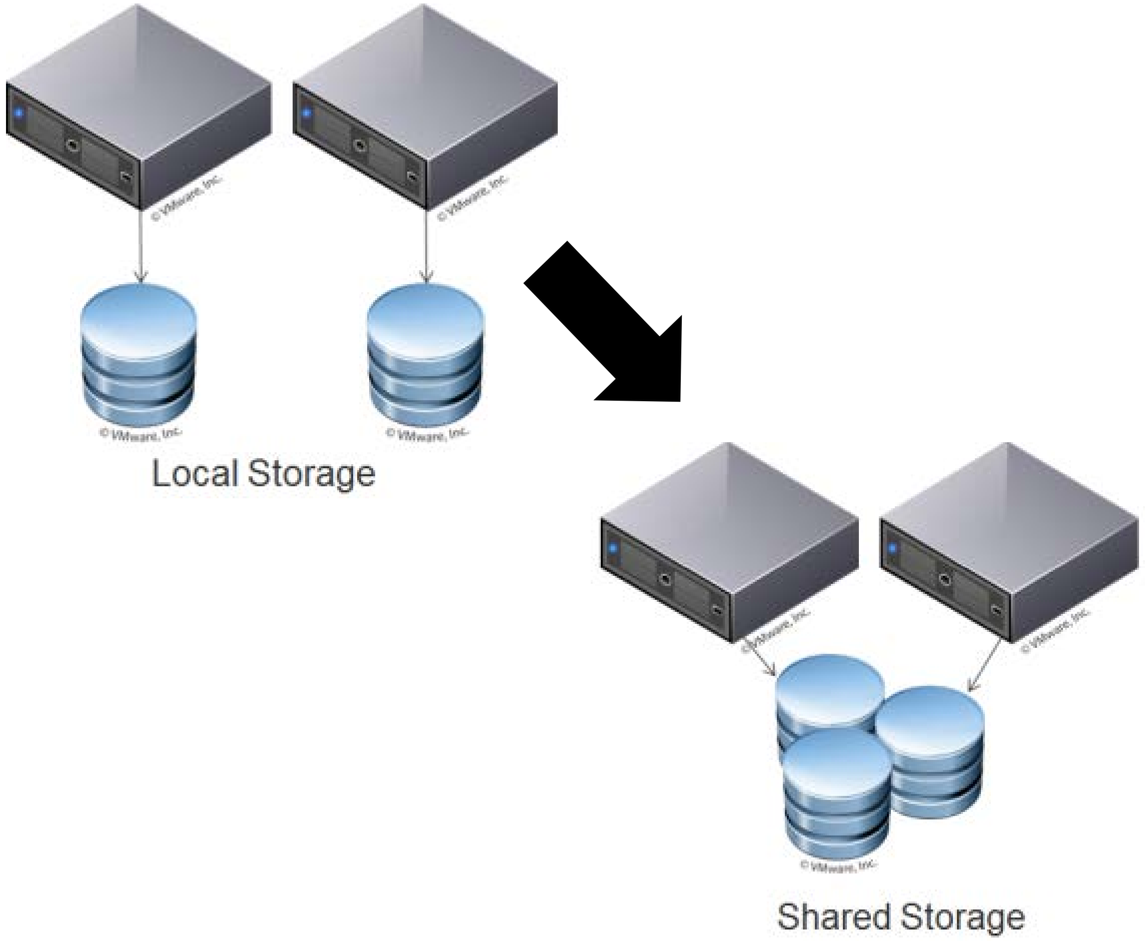


“NVMe Over Fabrics” Enables Storage Networking of NVMe

- Sharing NVMe based storage across multiple servers/CPUs
  - Better utilization: capacity, rack space, power
  - Scalability, management, fault isolation

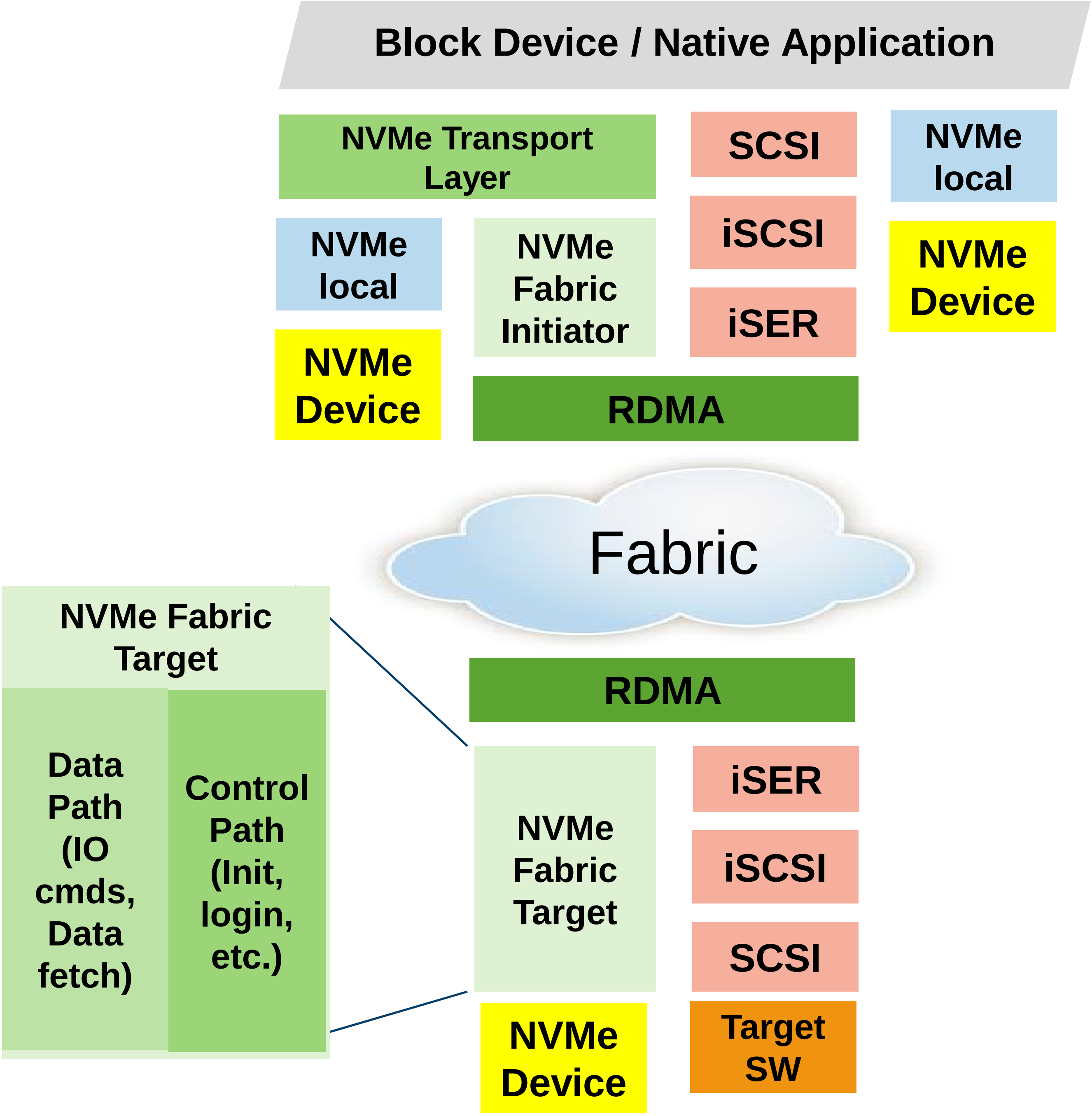
- NVMe over Fabrics industry standard developed
  - Version 1.0 completed in June 2016

- RDMA protocol is part of the standard
  - NVMe-oF version 1.0 includes a Transport binding specification for RDMA
  - InfiniBand or Ethernet(RoCE)



How “NVMe over Fabrics” works?

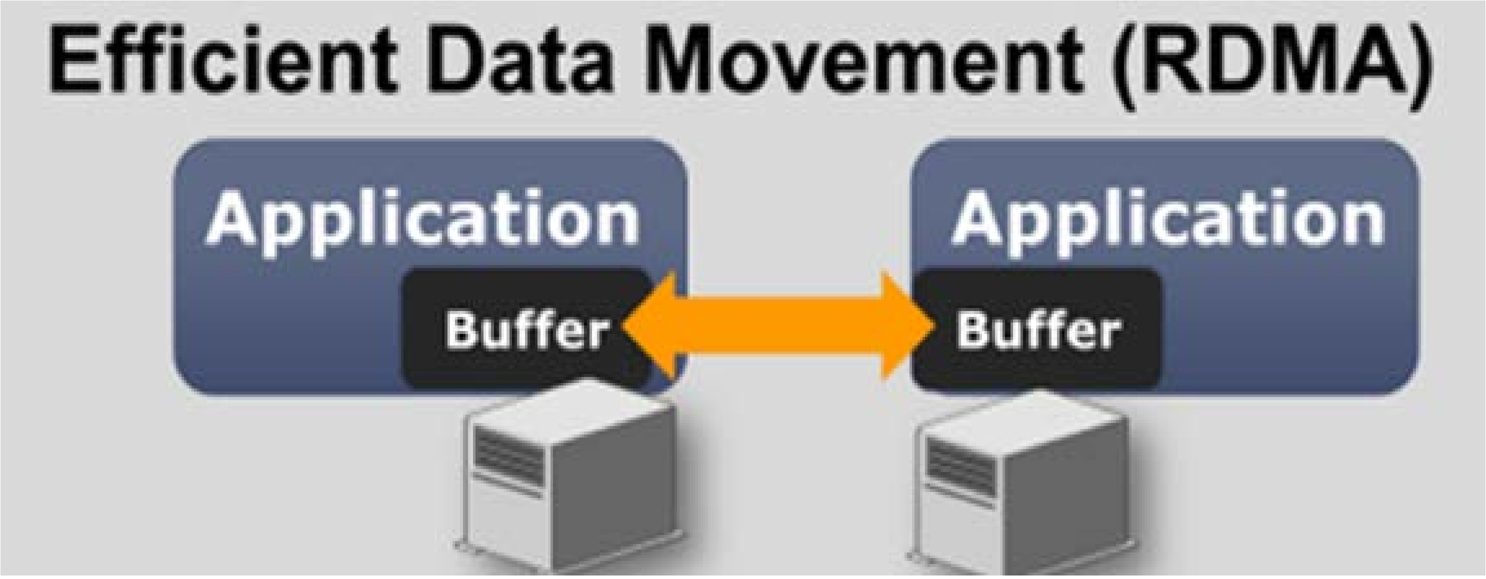
- The idea is to extend the efficiency of local NVMe interface over the fabric
  - NVMe commands and data structures are transferred end to end
  - RDMA one sided data transfer
  - Lockless multi-queue design
  - Ordering relaxation
- Mellanox ConnectX-5 will have target offload
  - Current ASICs already offload RoCE, IB and the Data Path moves
  - Initiator driver will be inbox with major OSes vendors after standard 1.0

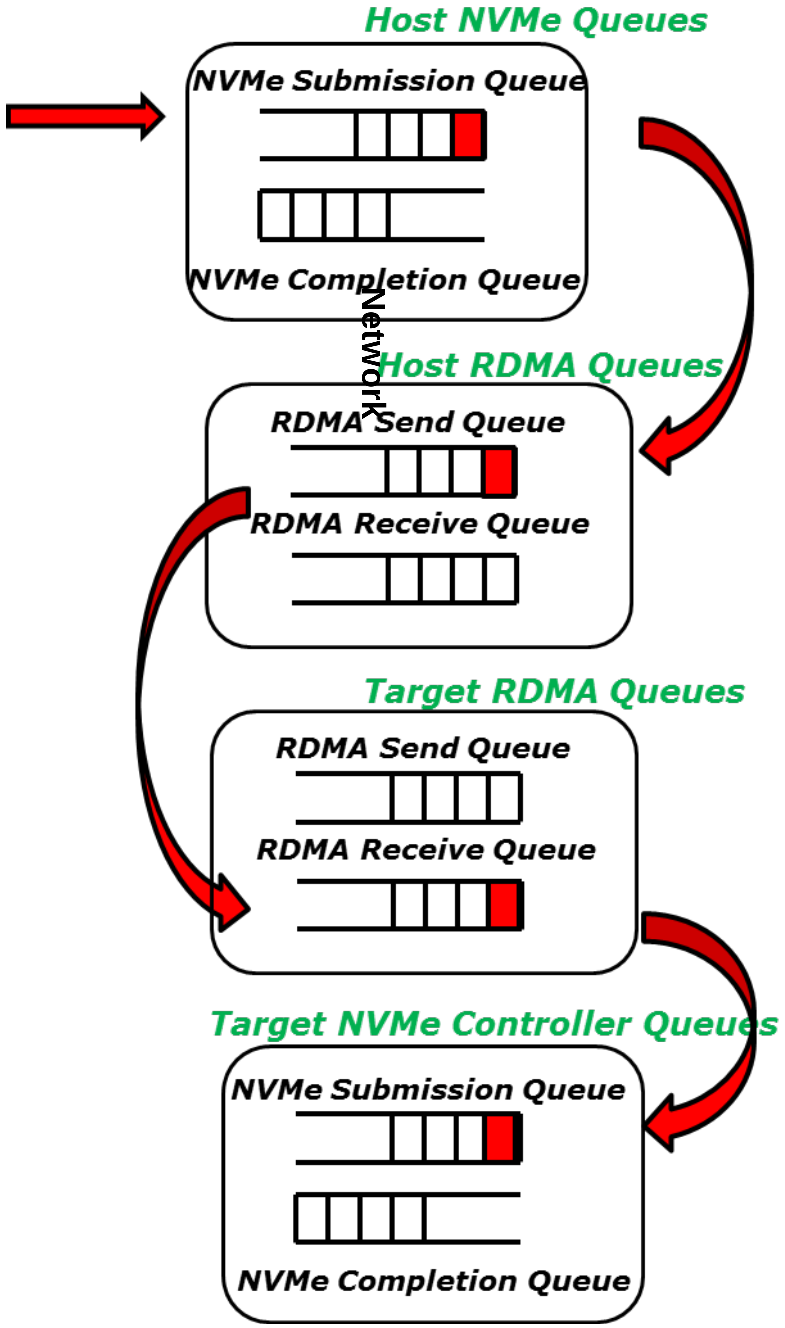


NVMe over Fabrics Protocol Highlights

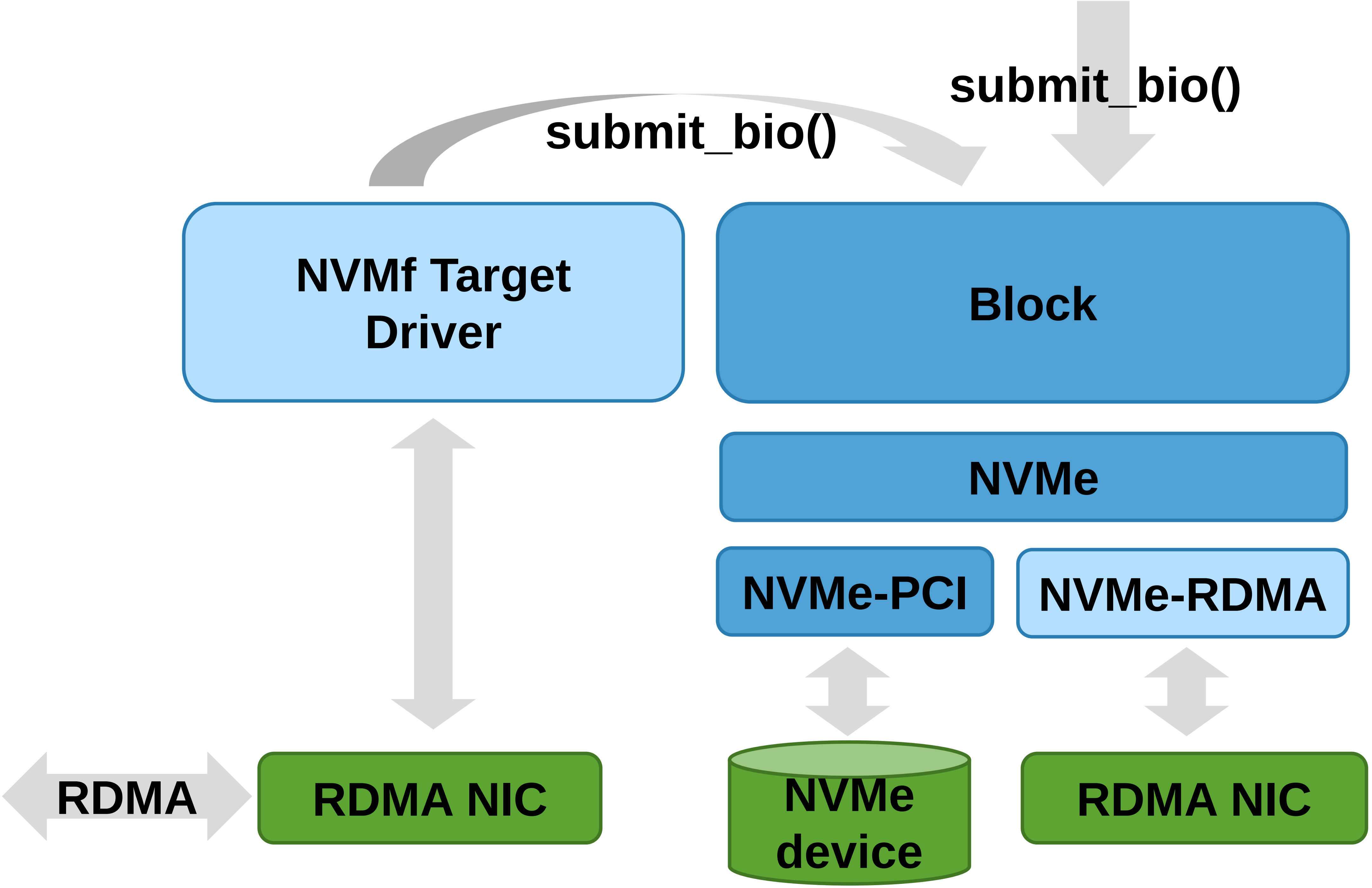
- NVMe queues are mapped to RDMA queues
  - Extend the parallelism of Multi-Queue NVMe
  - Lockless design
  - Direct access from user space?
    - Implementation dependent

- NVMe command are encapsulated into SEND messages
- Data transfer is fully offloaded using RDMA READ/WRITE

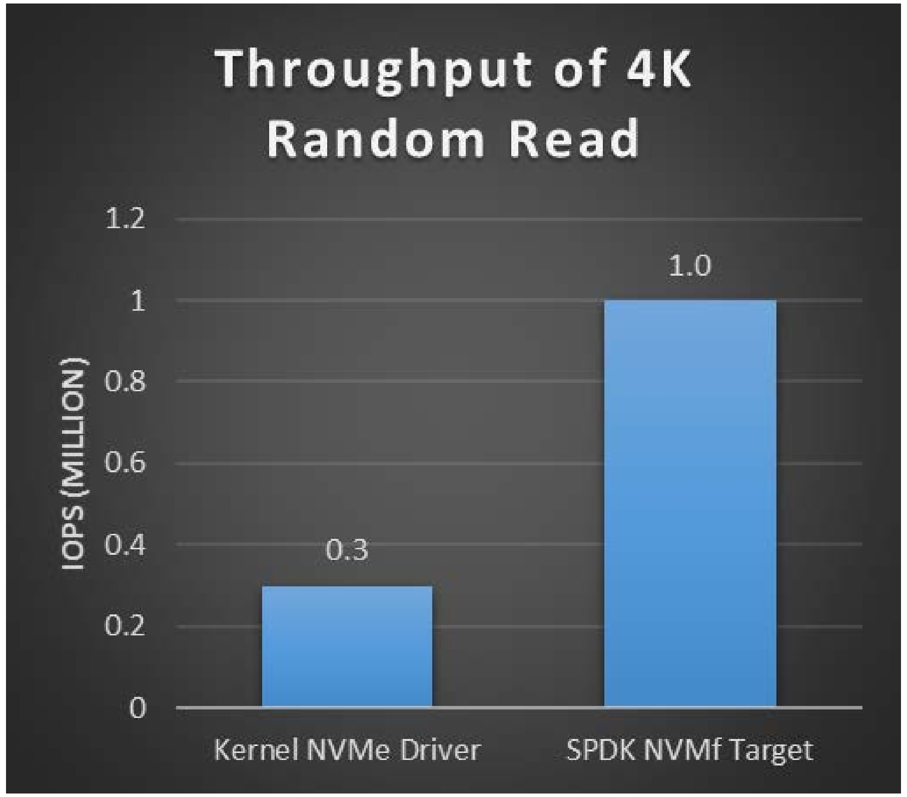


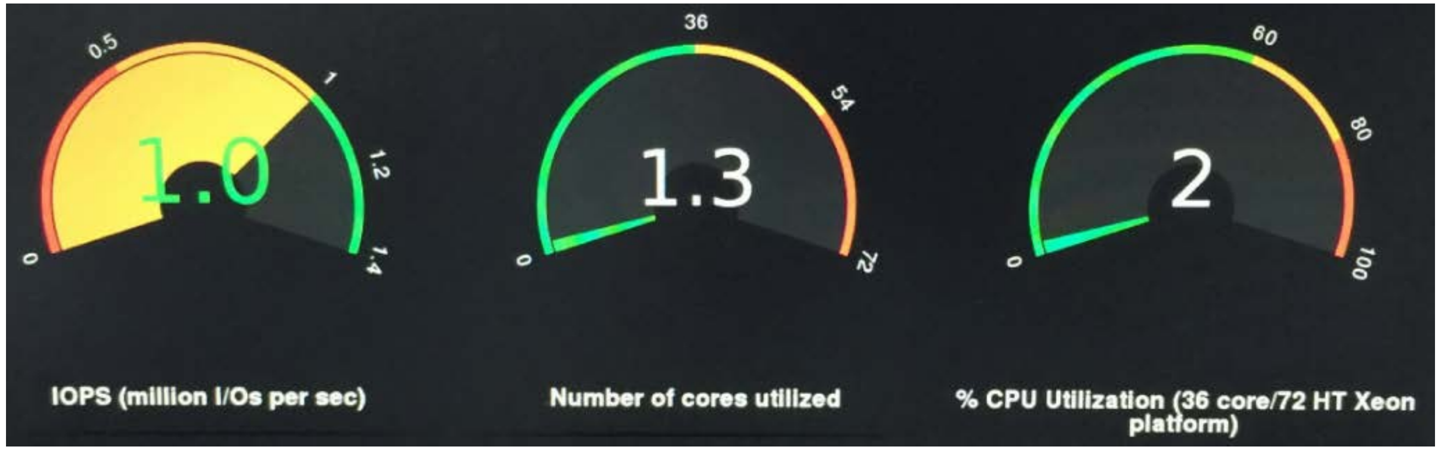


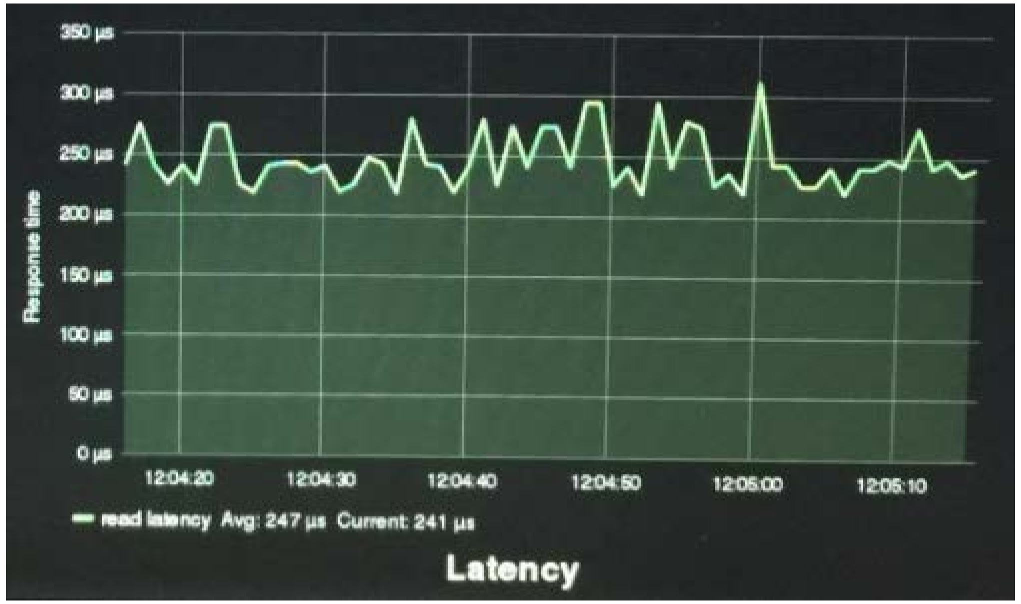
Linux 4.8 Software Architecture



SPDK NVMe over Fabrics demo performance




0
0.5
1
1.2
1.4
1.0
IOPS (million I/Os per sec)
0
36
54
72
1.3
Number of cores utilized
0
60
80
100
2
% CPU Utilization (36 core/72 HT Xeon platform)



Fabric added latency ~8usec

Mellanox RDMA fabric can greatly improve CPU efficiency and optimize application latency

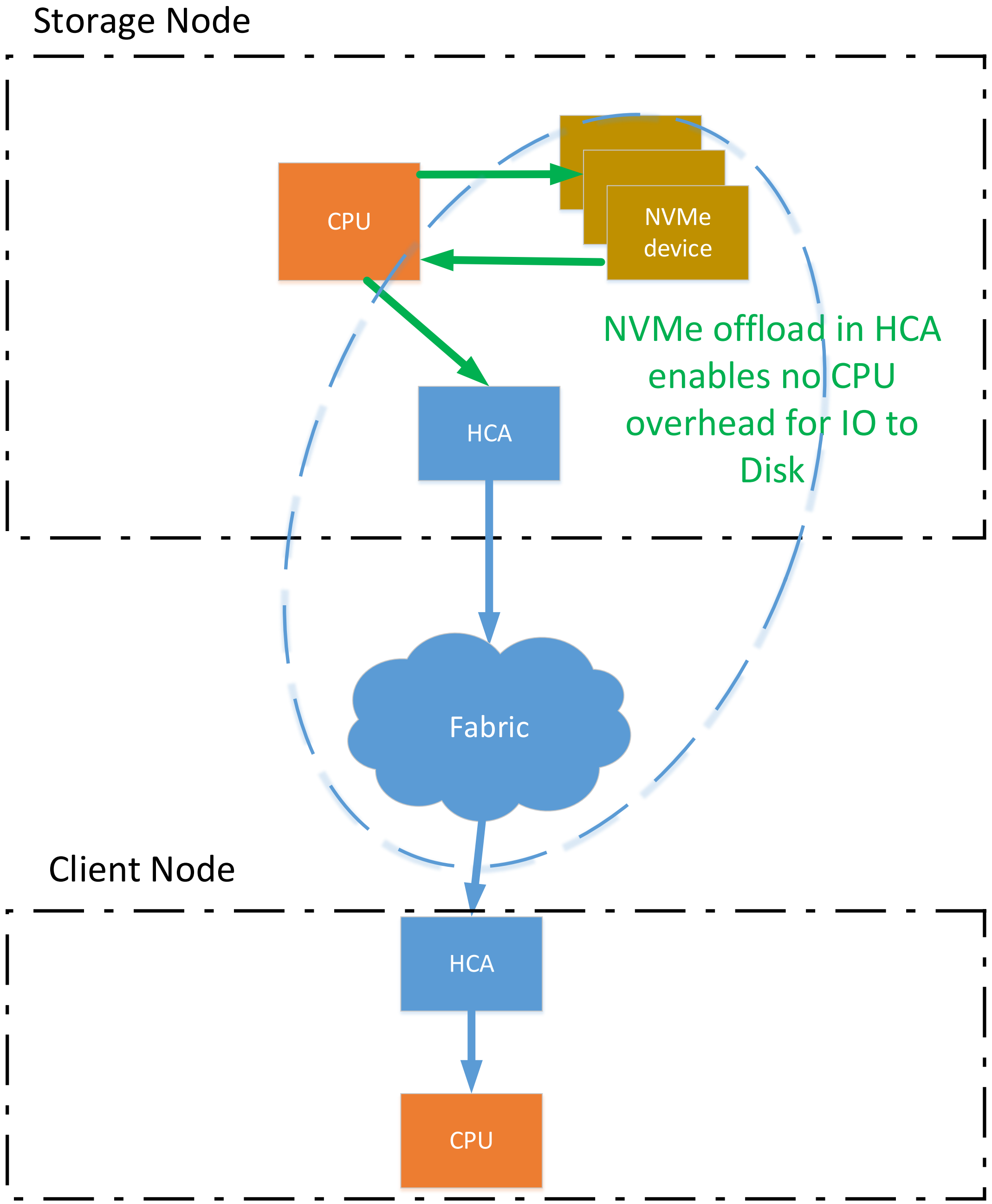
Mellanox ConnectX-5 NVMe over Fabrics Target Offload

- In NVMf, SSD access is exposed to the network

- With NVMf offload HCA can read/write/flush directly to the NVMe SSD **without CPU interrupts**
  - Reduction of latency
  - Reduction of CPU utilization
  - Reduction of cost

- NVMf target logic is terminated by the HCA

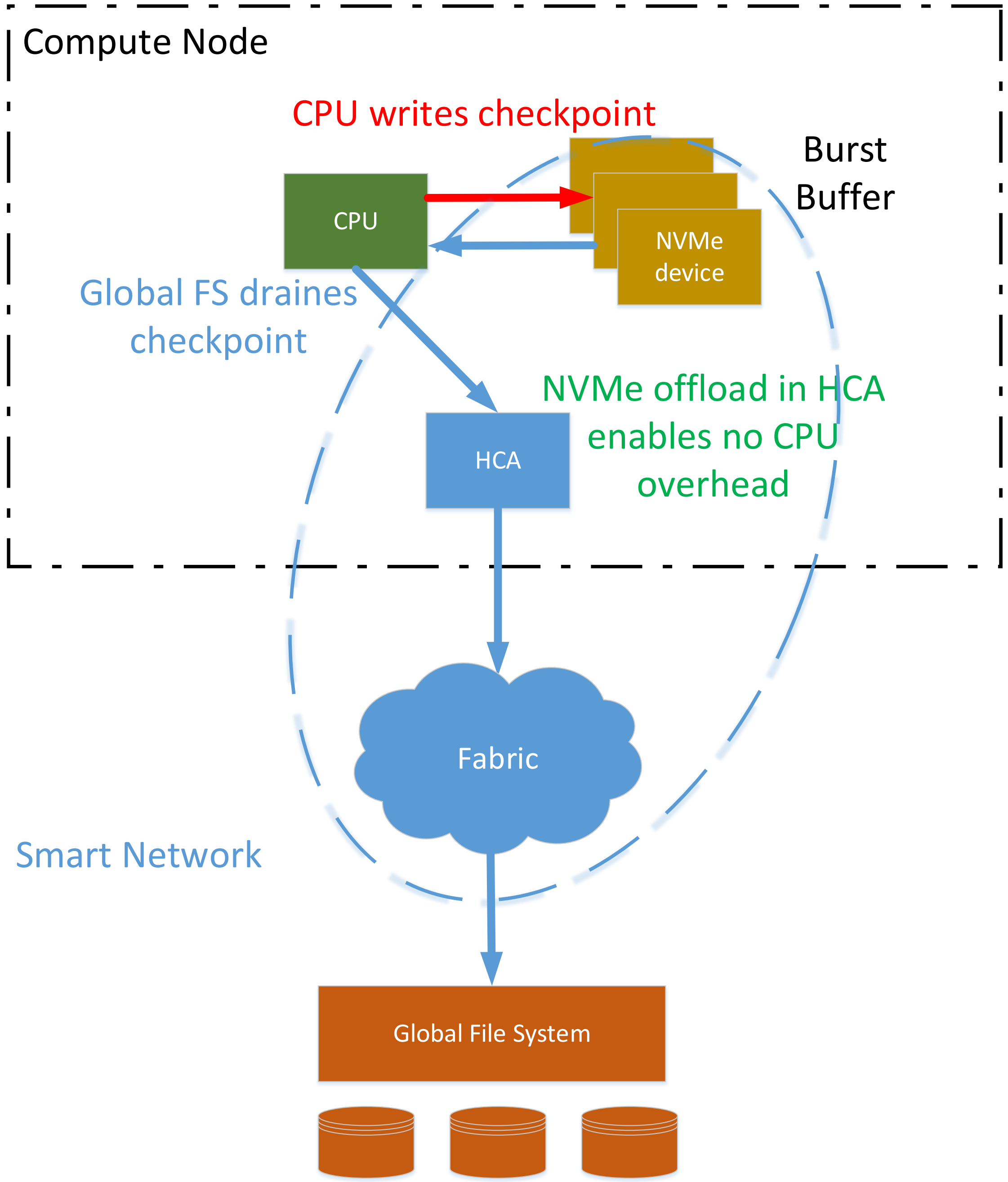
- Memory can be staged in system memory, HCA memory or SSD memory



Burst Buffer Use Case for Application Check Point

- Application checkpoints the data into the NVMe device
  - High BW PCIe connectivity

- Global file system sync with the check point using NVMe over Fabrics
  - Without CPU intervention on the compute node
  - Overlap of checkpoint and compute
  - Data rates could be provisioned for preserving the SLA of the compute networking needs



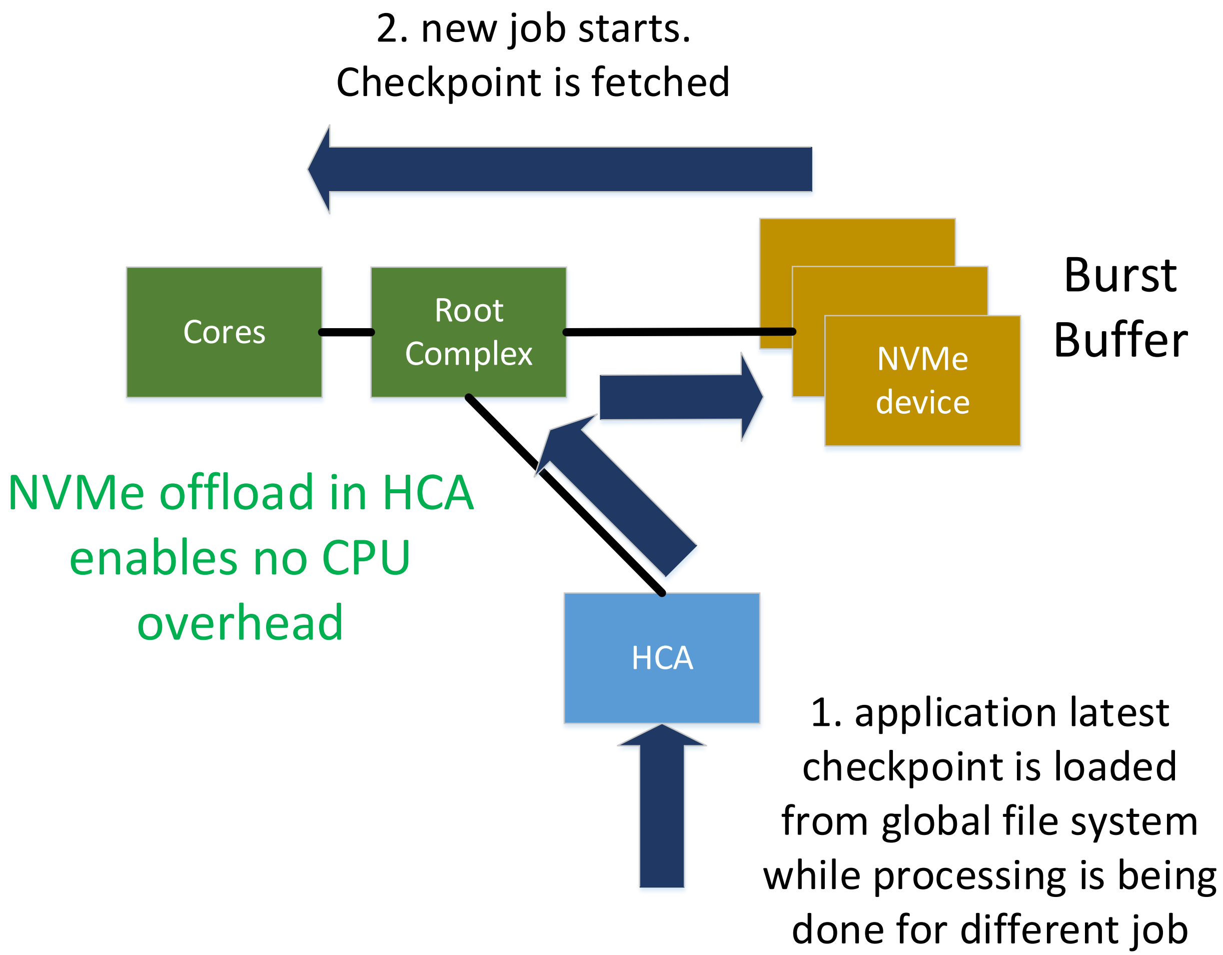
Burst Buffer Use Case for Asynchronous Data Fetch

- Global file system write the application data base into the burst buffer before run time
  - Overlapped with previous application run time
  - Data rates could be provisioned for preserving the SLA of the compute networking needs

- Data can be accessed locally by the application
  - Eliminating the need for IO on slow fabrics on run time

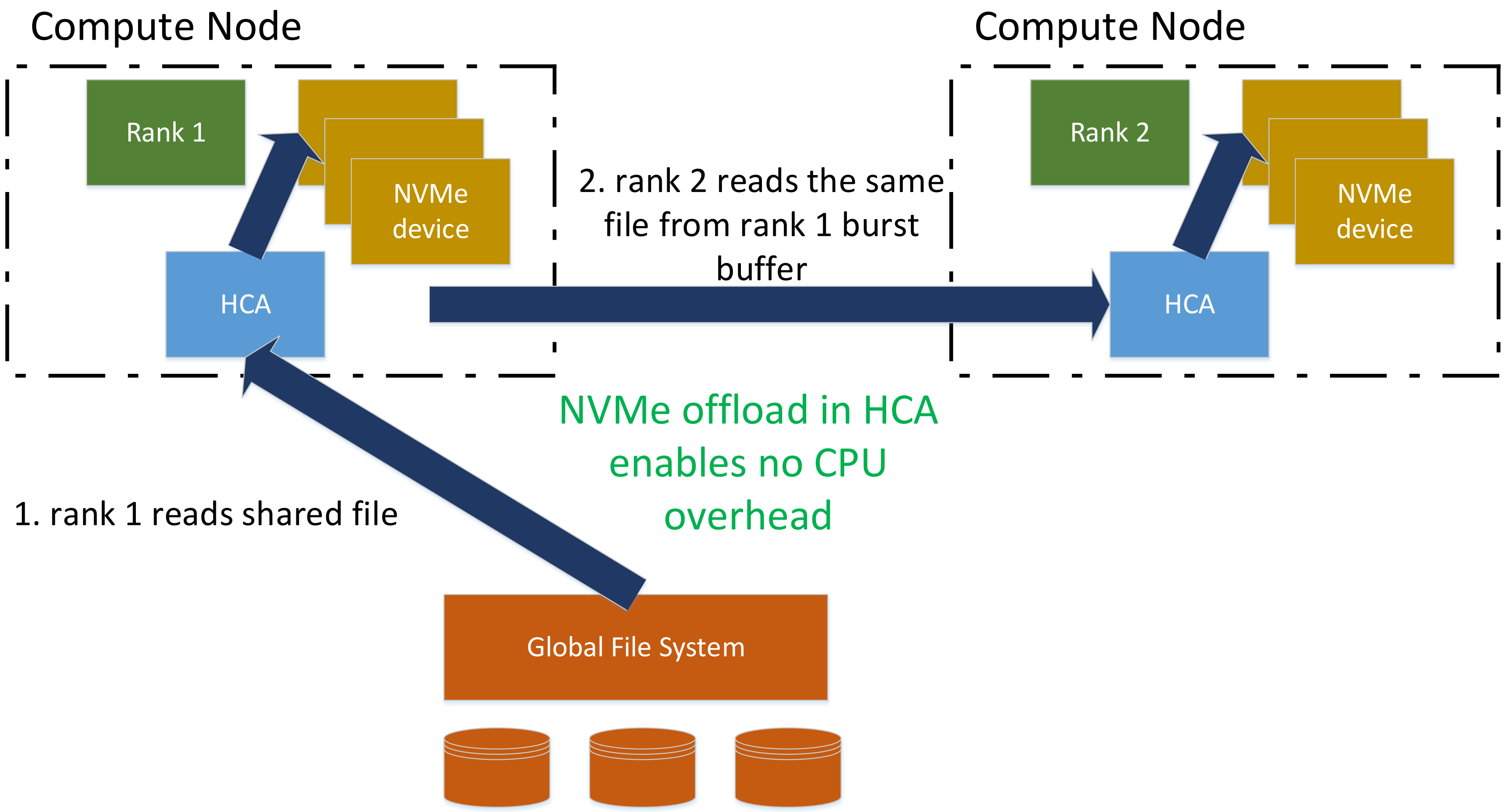
- Important for data intensive workloads
  - I.e. Machine Learning

No operating system noise on synchronizing checkpoint to local file system


2. new job starts.
Checkpoint is fetched
Cores
Root Complex
NVMe device
Burst Buffer
NVMe offload in HCA enables no CPU overhead
HCA
1. application latest checkpoint is loaded from global file system while processing is being done for different job

Sharing NVMe Burst Buffers

No operating system noise on sharing storage in between compute nodes for shared file systems



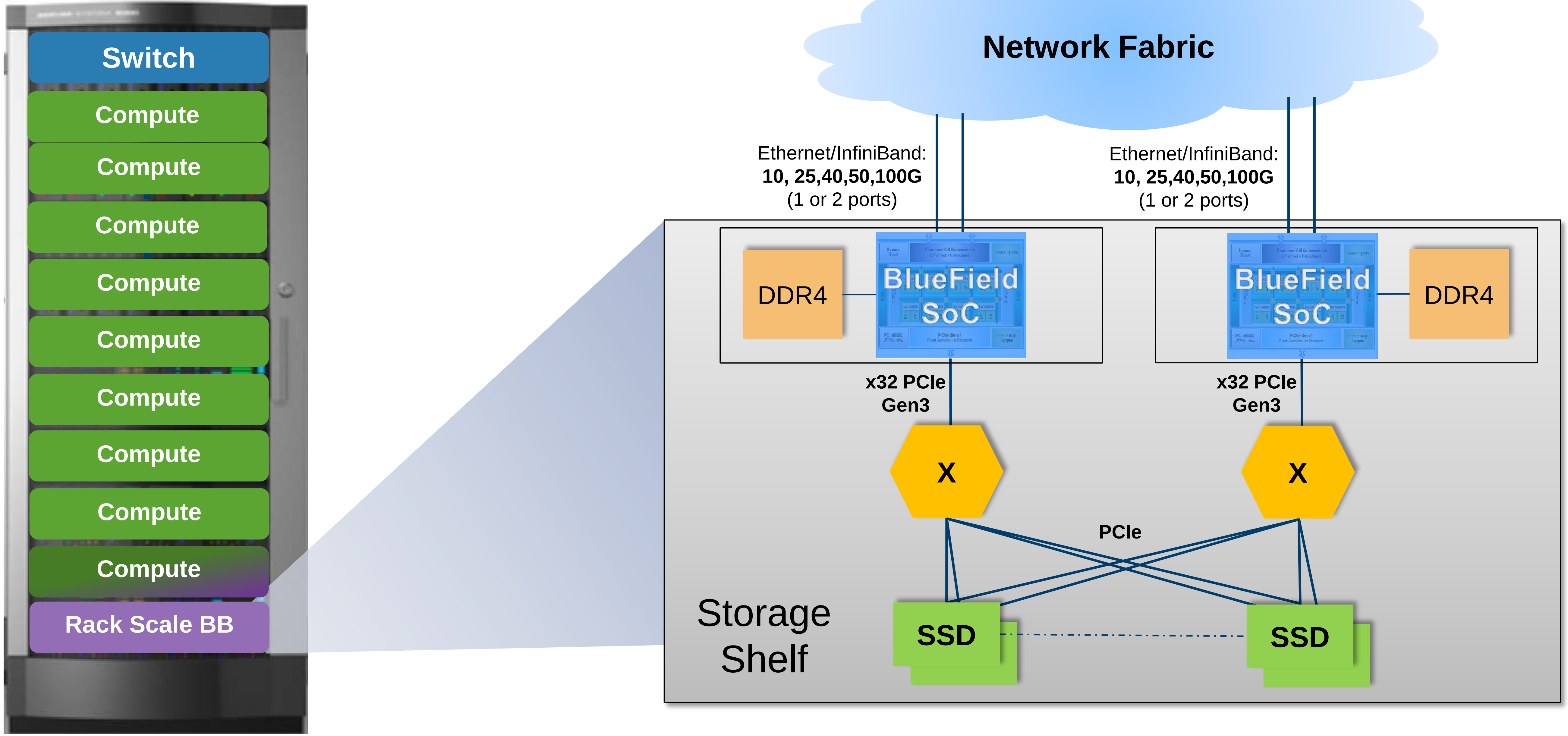
2. rank 2 reads the same file from rank 1 burst buffer

1. rank 1 reads shared file

NVMe offload in HCA enables no CPU overhead

Rack Scale Burst Buffer

Rack view



References

- NVMe over Fabrics Architecture
  - https://www.brighttalk.com/webcast/12367/181249
  - http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2015/20150811_FA11_Burstein.pdf
  - NVMe spec: http://nvmexpress.org/wp-content/uploads/NVM_Express_1_2_1_Gold_20160603.pdf
  - NVMe over Fabrics spec: http://www.nvmexpress.org/wp-content/uploads/NVMe_over_Fabrics_1_0_Gold_20160605-1.pdf
- NVMe_Linux
  - https://www.brighttalk.com/webcast/12367/202217?utm_campaign=communication_reminder_starting_now_registrants&utm_medium=email&utm_source=brighttalk-transact&utm_content=title
- Network direct access to NVMe
  - http://blog.pmcs.com/project-donard-peer-to-peer-communication-with-nvm-express-devices-part-1/
- “How to” guide
  - https://community.mellanox.com/docs/DOC-2504

Thank You

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.®