NVMe over Fabrics - High Performance
Flash moves to Ethernet

Rob Davis & Idan Burstein

Mellanox Technologies
Why NVMe and NVMe over Fabrics (NVMf)

![Bar chart showing access time in microseconds for HDD, SSD, and PM with a 10,000x improvement for SSD compared to HDD and PM.](image-url)
Compute/Storage Disaggregation

Application Resource Pool

- Storage
- Compute
- Storage
- Compute
- Storage
- Compute
- Storage
- Compute

Ethernet

25/40/50/100GbE NVMeoF

DDR ➔ CPU/NIC ➔ DDR

25/40/50/100GbE NVMeoF

X16 PCIe

Storage shelf

X

NVMe SSD ➔ CPU/NIC ➔ NVMe SSD

SDC 2016

2016 Storage Developer Conference. © Insert Your Company Name. All Rights Reserved.
Hyperconverged with NVMe

- 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

PCIe

SATA

~30 usec

~80 usec
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

![Diagram showing Networked Storage and Protocol and Network performance comparison between HDD and SSD.](image)

- **Access Time (ms/µ-sec):**
  - HDD: 5, 7
  - SSD: 25, 40, 50, 100Gb

- **Protocol and Network Performance:**
  - **PM** indicates a specific protocol or network component.

- **Media:**
  - HDD
  - SSD
  - NVM

- **Access Time Metrics:**
  - Microseconds to milliseconds
What is RDMA

Efficient Data Movement (RDMA)

Application
Buffer → Network → Buffer

Kernel Bypass
Protocol Offload

Network adapter based transport
100GbE
RoCE

Microsoft Storage Spaces Throughput

Throughput (Gigabytes/sec)

Without ROCE

ROCE

100GbE 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

100GbE RoCE

Target Server

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

<table>
<thead>
<tr>
<th>Demo</th>
<th>NVMe Hardware</th>
<th>Software / Drivers</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mangstor</td>
<td>Mangstor</td>
<td>Mangstor NVMeoF</td>
<td>RoCE or IB</td>
</tr>
<tr>
<td>PMC Sierra</td>
<td>PMC</td>
<td>PMC NVMeoF</td>
<td>40Gb RoCE</td>
</tr>
<tr>
<td>HGST</td>
<td>HGST</td>
<td>HGST NVMeoF</td>
<td>56Gb InfiniBand</td>
</tr>
<tr>
<td>Micron</td>
<td>Micron</td>
<td>Mellanox NBDx</td>
<td>100Gb RoCE</td>
</tr>
<tr>
<td>Memblaze</td>
<td>Memblaze</td>
<td>Mellanox NBDx</td>
<td>40Gb RoCE</td>
</tr>
<tr>
<td>Samsung at FMS15</td>
<td>Samsung</td>
<td>iSER / Ceph / SMB Direct</td>
<td>40Gb RoCE</td>
</tr>
<tr>
<td>Intel at IDF14</td>
<td>Intel</td>
<td>Intel/Chelsio NVMeoF</td>
<td>40Gb iWARP</td>
</tr>
<tr>
<td>Stealth startups</td>
<td>Any / Intel NVMe</td>
<td>Startup’s NVMeoF</td>
<td>40Gb RoCE</td>
</tr>
</tbody>
</table>

**Proprietary Pre-standards Drivers**

**NVMeoF Standard V1.0**

**Community Initiators**

**Community Targets**

**Proprietary Targets**
NVMf Standard 1.0 Community Open Source Driver Development

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.

Working Group - Fabrics Linux Driver

Group Info
Group Chair: Bob Beauchamp, EMC

Group Email Addresses
Post message: fabrics_linux_driver@nvexpress.org
Contact chair: fabrics_linux_driver-chair@nvexpress.org
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

<table>
<thead>
<tr>
<th>Bandwidth (target)</th>
<th>IOPS (target)</th>
<th># online cores</th>
<th>Each core utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS = 4KB, 16 jobs, IO depth = 64</td>
<td>5.2GB/sec</td>
<td>1.3M</td>
<td>4</td>
</tr>
</tbody>
</table>
Kernel & User Based NVMeoF

SPDK

NVMf Initiator

NVMf Target

2x25GbE with RDMA (RoCE)

Mellanox ConnectX-4 Lx dual-port

PCle 3.0 x8

PCIe 3.0 x16

Intel NVMe SSD P3700 2TB x 4

User S/W

Kernel S/W

User App

IO Library

Kernel Apps

OS Stack

Sys Driver

H/W

PCle

Transport & Network (L4/L3)

Ethetnet (L1/L0)

Standard NIC Flow

RDMA NIC Flow
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 SEND
- SEND First
- SEND Middle
- SEND Last
- ACK
- Post RCV
- RCV Completion
- ACK

Host HCA Memory
RDMA WRITE
RDMA READ
NVMe and NVMeoF Fit Together Well

Efficient Data Movement (RDMA)
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

---

**Diagram:**
- NVMe Initiator
- RNIC
- RNIC
- NVMe Target
- Post Send (CC)
- Send – Command Capsule
- Ack
- Completion
- Post Send
- Write data
- Post Send (RC)
- Write first
- Write last
- Send – Response Capsule
- Ack
- Completion
- Free send buffer
- Post NVMe command
- Wait for completion
- Free receive buffer
- Free allocated buffer
- Free send buffer
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 its 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

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

New Capabilities

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

2003-2004
2.5Gb/s per lane
8/10 bit encoding

2007-2008
5Gb/s per lane
8/10 bit encoding

2011-2012
8Gb/s per lane
128/130 bit encoding

2015-2016
16Gb/s per lane
128/130 bit encoding
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