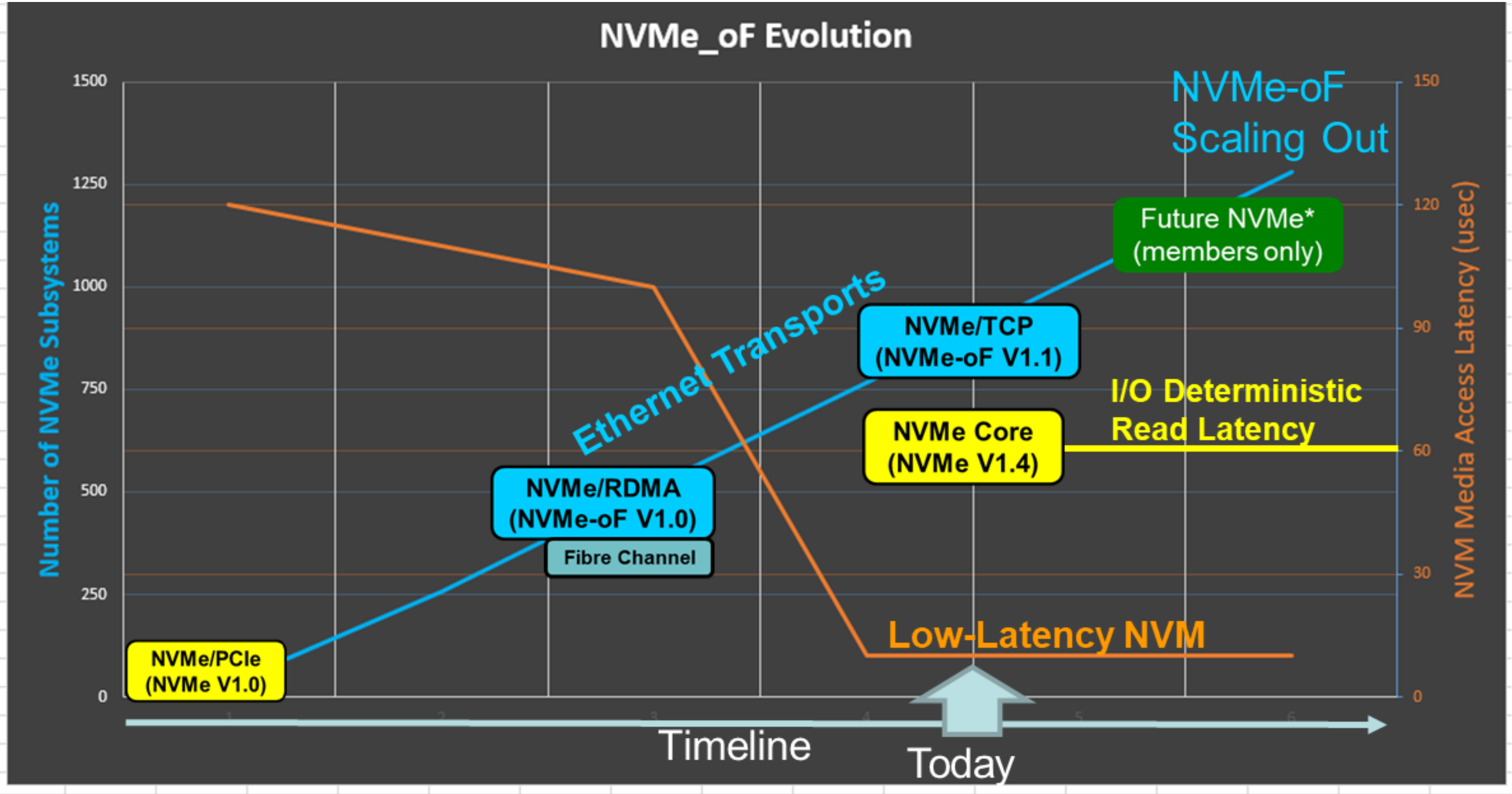
SDC 19 | September 23-26, 2019
Santa Clara, CA

Selecting an NVMe-oF™ Ethernet Transport RDMA or TCP?

Dave Minturn	Principal Engineer
Anil Vasudevan	Sr. Principal Engineer
Intel Corporation	

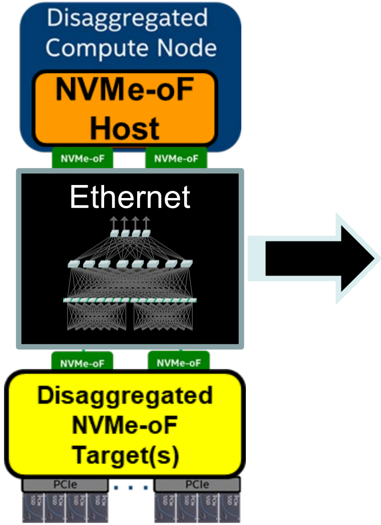
intel

NVMe-oF Ethernet Evolution

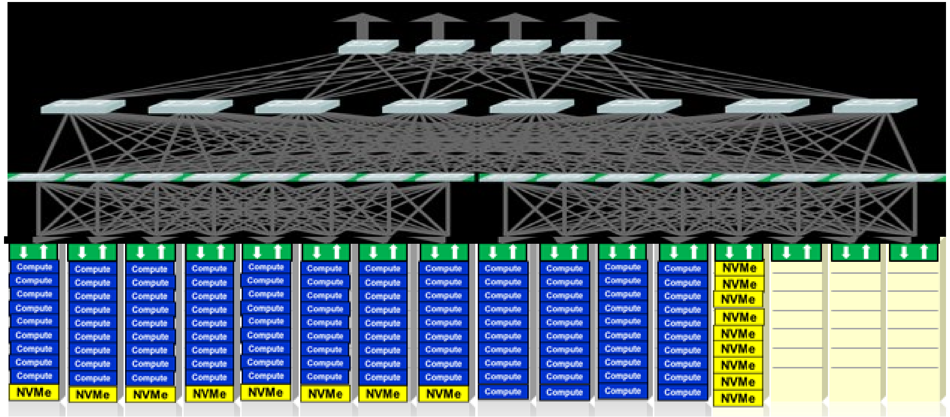


Scaling-out NVMe-oF Storage on Ethernet

NVMe-oF Disaggregation Model



Disaggregated NVMe-oF Deployment Models



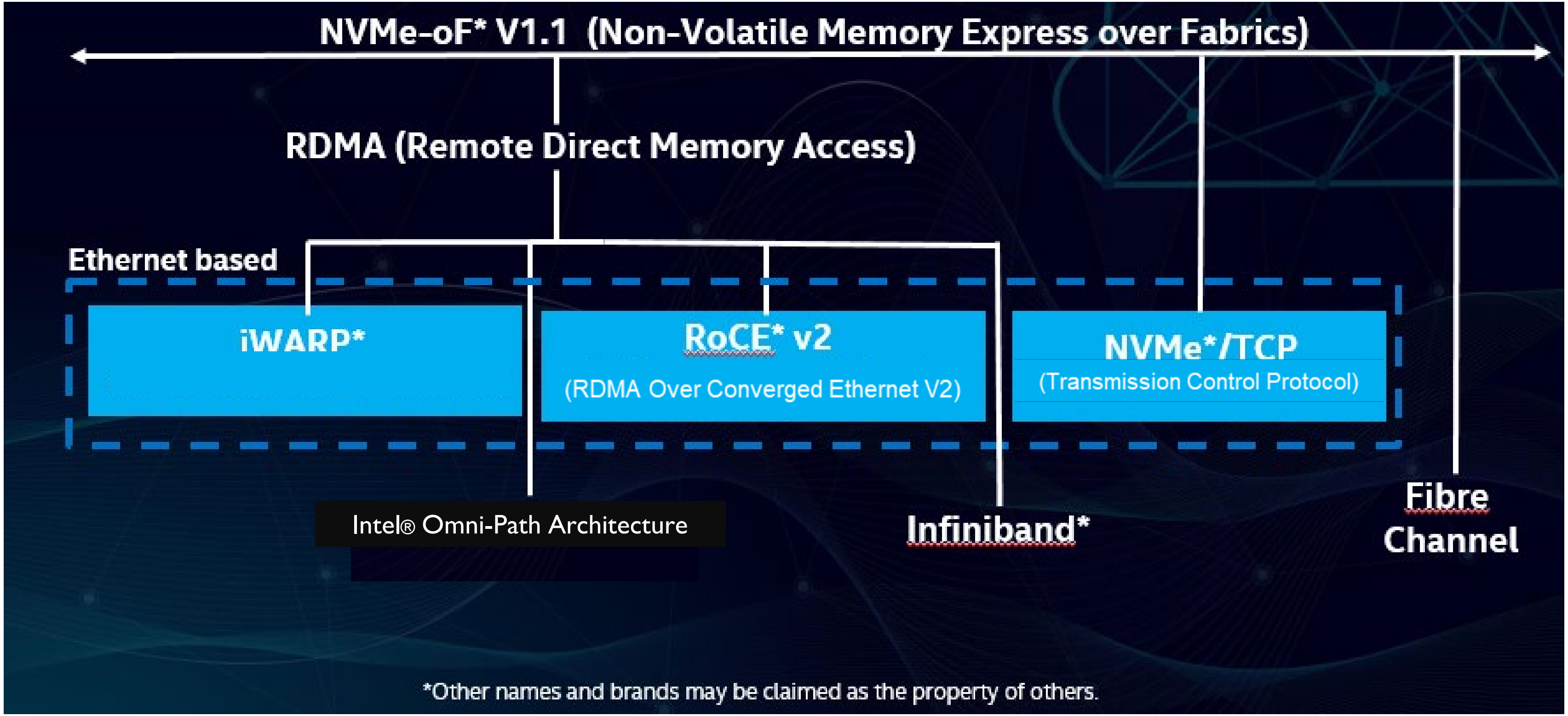
Per-Rack NVMe-oF Storage

- Ephemeral (JBOFs, Targets, ..)
- Back-end to storage nodes

NVMe-oF Storage in remote racks

- Durable (Scale-out)

NVMe-oF Ethernet Transports



NVMe-oF* V1.1 (Non-Volatile Memory Express over Fabrics)

RDMA (Remote Direct Memory Access)

Ethernet based

iWARP*

RoCE* v2

(RDMA Over Converged Ethernet V2)

NVMe*/TCP

(Transmission Control Protocol)

Intel® Omni-Path Architecture

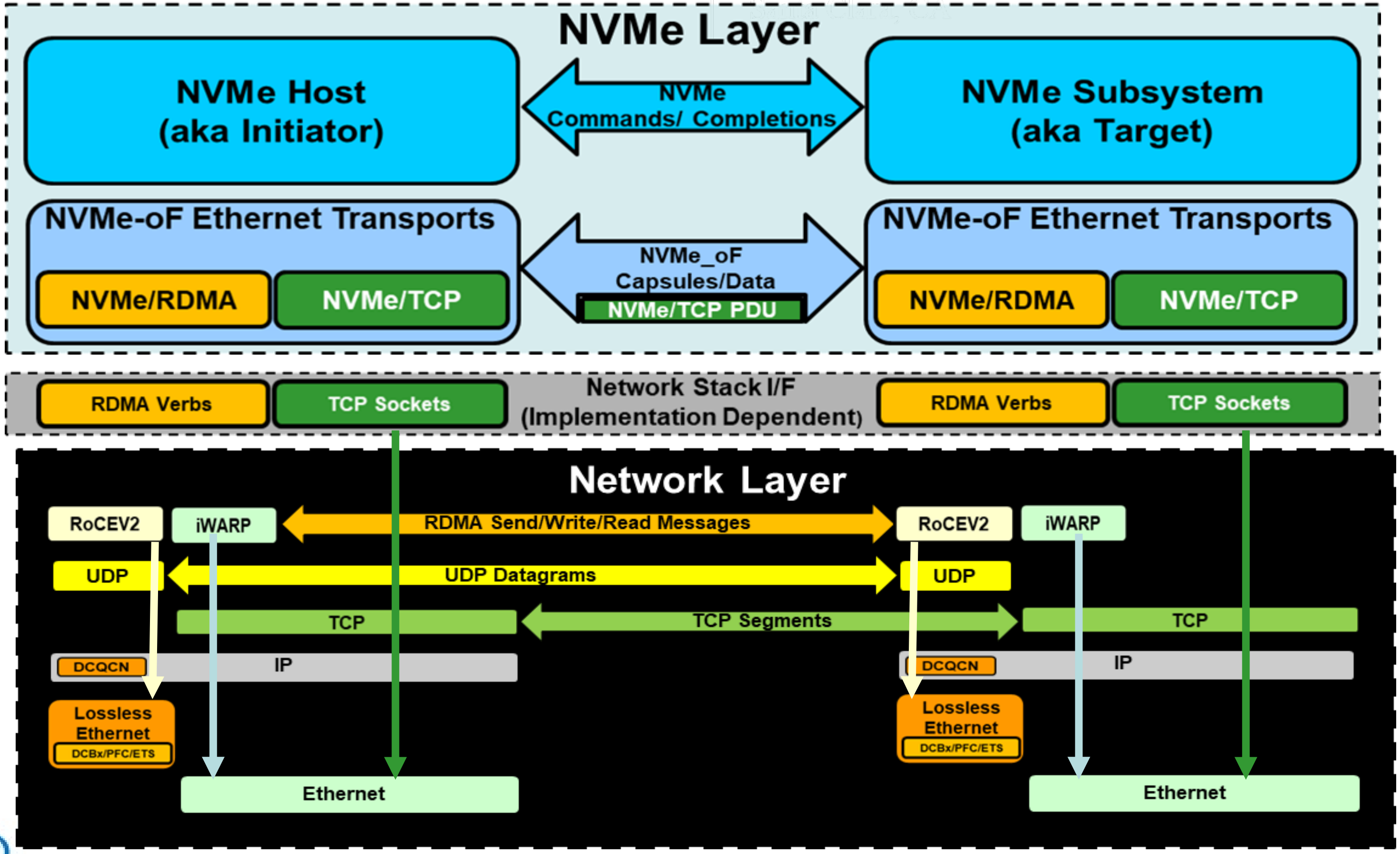
Infiniband*

Fibre Channel

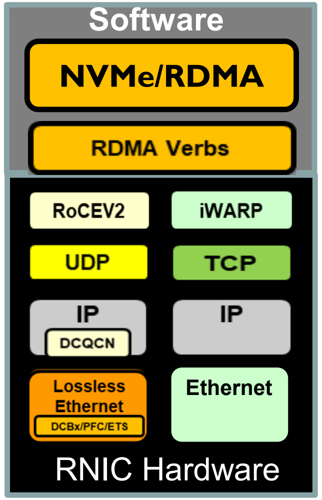
*Other names and brands may be claimed as the property of others.

NVMe-oF Ethernet Layering

September 23-26, 2019
Santa Clara, CA



NVMe-oF RDMA Ethernet Transports



NVMe/RDMA was standardized and implemented (Linux) to be RDMA provider type agnostic

- Implemented over common RDMA Verbs (Linux)
- Two common provider types are RoCEV2 and iWARP

NVMe-oF Capsules exchanged with RDMA_SEND and NVMe Data exchanged with RDMA_READ/RDMA_WRITE

- Full hardware offload of RDMA operations and underlying network protocol stack layers to reduce latency and CPU utilization
- Direct data placement of ingress NVMe command data

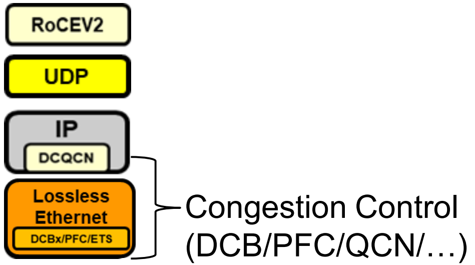
NVMe Queue Pairs (SQ/CQ) are mapped 1x1 with RDMA QPs

- Enables separation of NVMe QP flows to avoid HOL blocking

NVMe-oF RDMA deployments must use H/W RDMA enabled host and target endpoints of the same RDMA provider type

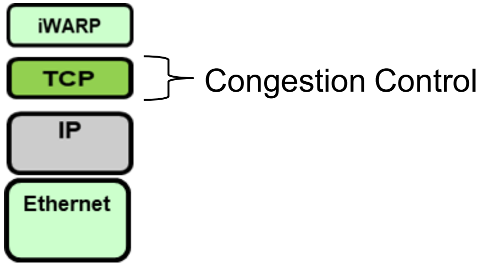
RDMA Ethernet Provider Types

RoCEV2: **R**DMA **o**ver **C**onverged **E**thernet



- Requires use of a Lossless Ethernet enabled network for efficient NVMe-oF
  - DCB enabled Ethernet switches for Lossless Ethernet
  - Literature describing deploying in large scale networks; (DCQCN, HPCC, ..)

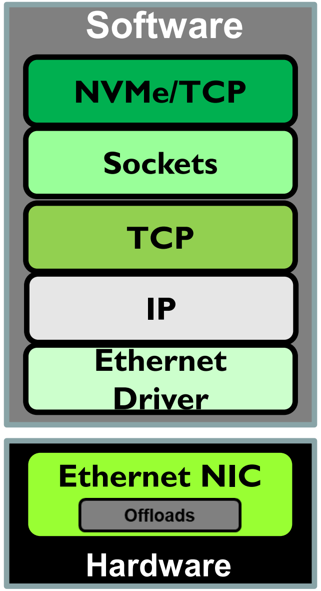
iWARP:



- Use of TCP enables use on any Ethernet network
  - Can benefit from Lossless Ethernet but does not require it
  - Not as widely deployed as RoCE based solutions

Choice of RDMA provider type mostly based on network infrastructure dependencies
RNIC product implementations may offer one or both RDMA Provider Types

NVMe-oF NVMe/TCP Ethernet Transport



Motivation for NVMe/TCP standardization and implementation (Linux)

- Enabls the use of NVMe-oF on existing datacenter networks
- Provide a more efficient alternative to iSCSI for NVMe SSD configured targets
- Facilitates both software (shown) and hardware based implementations

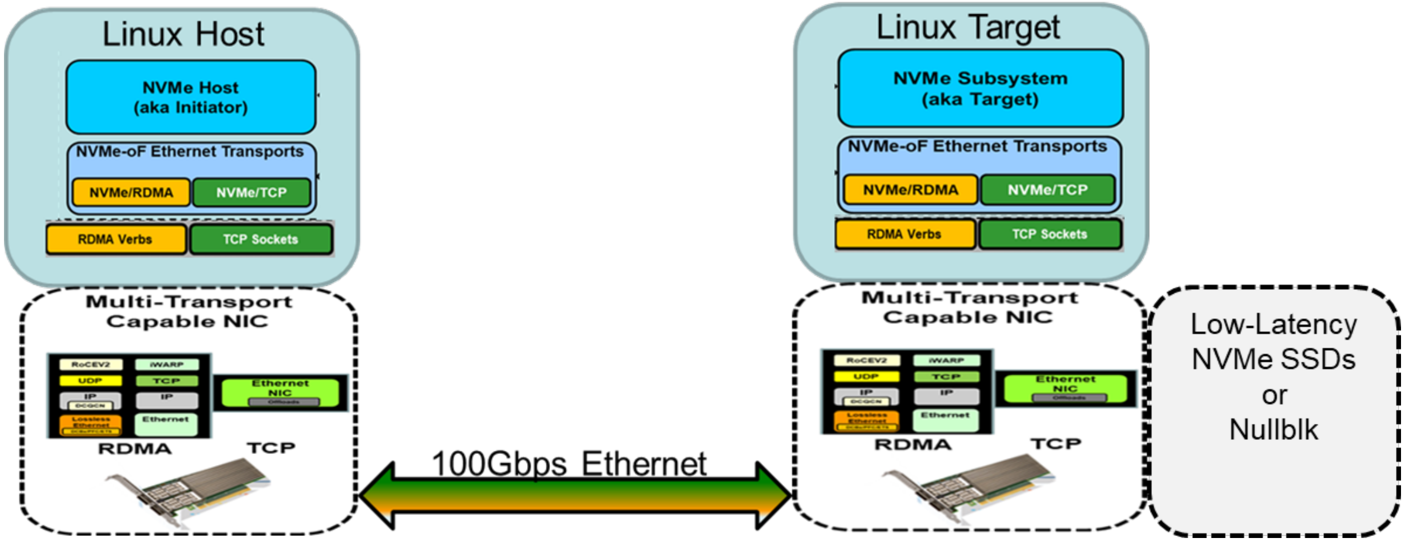
NVMe-oF Capsules exchanged using NVMe/TCP PDUs
NVMe Data exchanged either in-capsule or using NVMe/TCP R2T

NVMe Queue Pairs (SQ/CQ) are mapped 1x1 with TCP Connections

- Enables TCP connection based NVMe queue to CPU core association

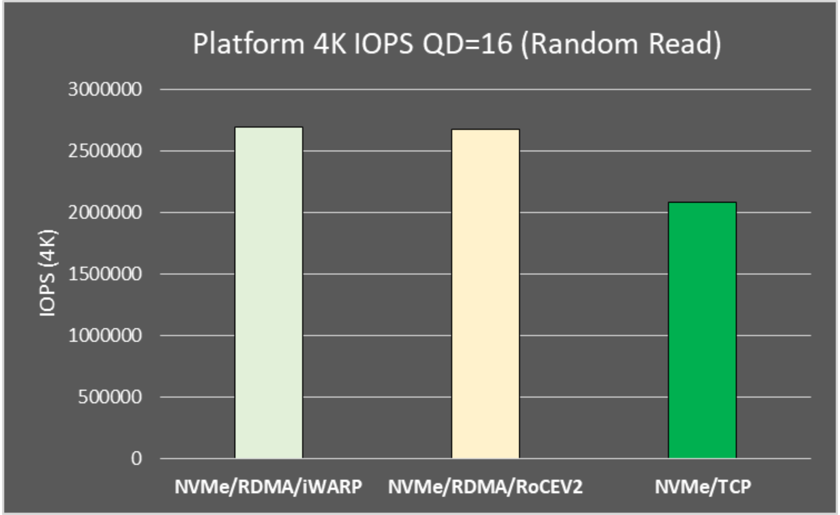
NVMe Ethernet Transport Performance

(Linux Host and Target Configuration)

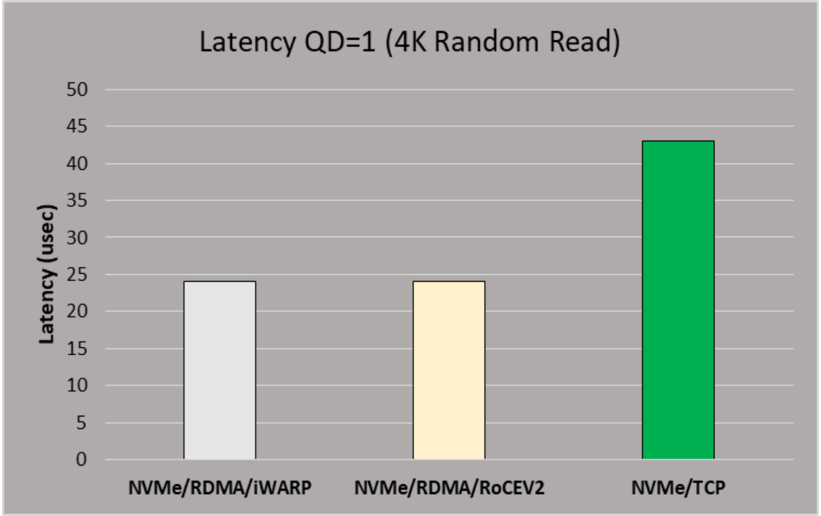


NVMe-oF Ethernet Transport Performance

IOPS
(Higher is Better)



Latency
(Lower is Better)



Performance results are based on testing as of September 2019 and may not reflect all publicly available security updates. See configuration disclosure on slide 21 for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

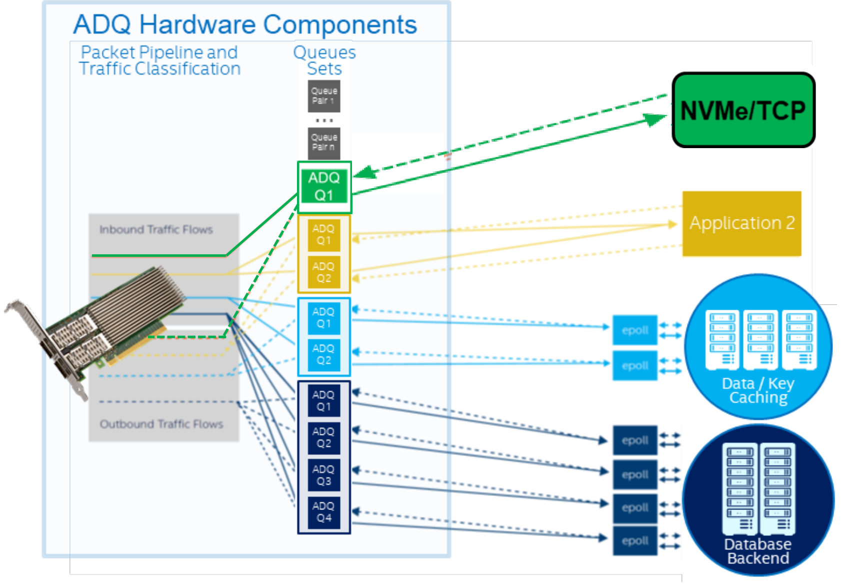
Improving Linux NVMe/TCP Performance

Identifying the NVMe/TCP Inefficiencies

System attributes influenced by the network stack and NIC

System attribute	Problem(s) to influence/solve
Interrupts	How to prevent an interrupt from firing when the NVMe stack is doing useful work?
Context Switches	How to minimize context switches?
Synchronization	How to minimize/eliminate synchronization operations e.g. locks, multi-thread sharing?
Application & TCP Protocol Processing	How can the application, NVMe stack, and TCP processing operate in the same context?
Data Movement	How to improve working set locality?

Application Device Queues (ADQ)



ADQ Basics

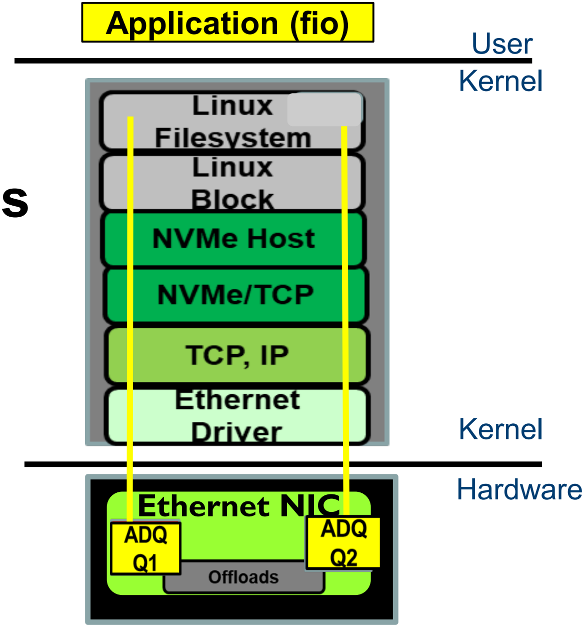
- Filters application traffic to a dedicated set of queues
- Application threads of execution are connected to specific queues within the ADQ queue set
- Bandwidth control of application egress (Tx) network traffic

	Capability
Application	Align Application Threads and ADQ's
Kernel	Event polling (Epoll) enabled with Busy Polling Device Queues Symmetric Queuing for receive and transmit Queue identification for Applications HW accelerated Application receive traffic steering configuration HW accelerated Application transmit traffic shaping configuration
Driver	Steering and signaling optimizations
NIC HW	Application specific traffic steering and queuing Application transmit traffic shaping

[1]Features & schedule are subject to change. All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.
Intel and the Intel Logo are trademarks of Intel Corporation in the U.S. and other countries. *Other names and brands may be claimed as the property of others.

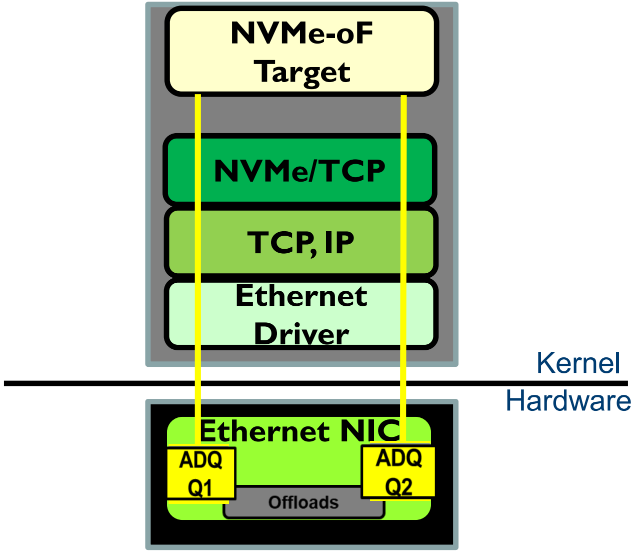
Modifications to NVMe/TCP (Initiator)

- **In context request submission**
  - Leverages ADQ's, 1 queue per core and optimizes application thread <->I/O submission
- **Leveraging polling enhancements**
- **Set socket priority**



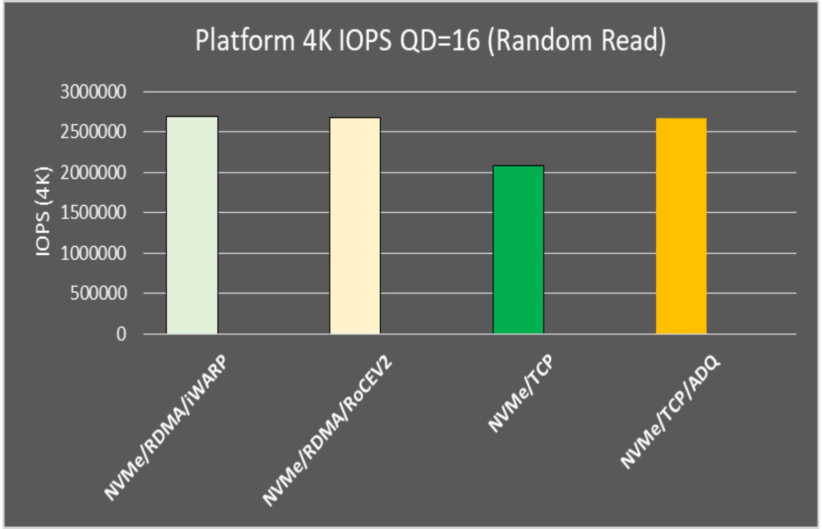
Modifications to NVMe/TCP (Target)

- Busy Polling enhancements
  - Add busy polling to the main target loop that has a non blocking recv() and send() call
- Set socket priority

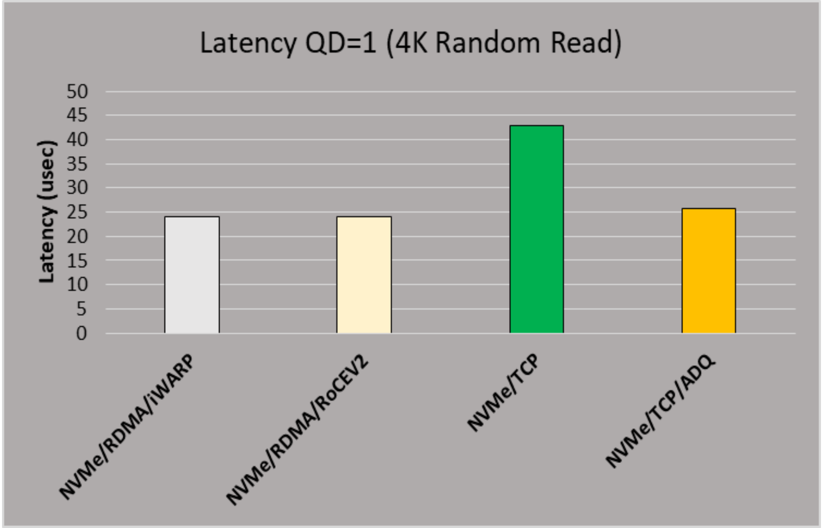


NVMe/TCP with ADQ Performance

IOPS (Higher is Better)



Latency (Lower is Better)



Performance results are based on testing as of September 2019 and may not reflect all publicly available security updates. See configuration disclosure on slide 21 for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Addressing Inefficiencies

System attribute	ADQ improvements
Interrupts	Optimizations assisted by event based busy polling
Context Switches	Busy polling keeps Application context alive during application related communications
Synchronization	Single producer consumer model creates a unique pipe between an application thread and a device queue
Application & Protocol Processing	Protocol processing triggered from the Application
Data Movement	Single producer consumer data flows and event based busy polling keep working set locality

NVMe-oF on Ethernet Summary

	NVMe/TCP	NVMe/RDMA iWARP	NVMe/RDMA RoCEV2	NVMe/TCP with ADQ
Network Infrastructure	Standard NIC(s) + standard Ethernet switches	RDMA Enabled NIC(s) + standard Ethernet switches	RDMA Enabled NIC(s) + lossless Ethernet switches	Standard NIC(s) + standard Ethernet switches
Performance	Baseline IOPS and efficiency, High tail latency	High IOPS and efficiency, lowest tail latency	High IOPS and efficiency, lowest tail latency	High IOPS and efficiency, low tail latency
O/S Network Software	Out of Box Linux host/target	RDMA Enabled	RDMA Enabled	ADQ Enabled
Ease of Use	Standard network Configuration	Standard network configuration	Requires additional network configuration	Standard network Configuration
NVMe-oF Usage Model	Data-Center wide	Rack-level, Data-Center wide	Rack-level, within lossless Ethernet domain	Data-Center wide

Please take a moment
to rate this session.

Your feedback matters to us.

Notices & Disclaimers

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

Performance results are based on testing as of September 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.   For more complete information visit http://www.intel.com/benchmarks .

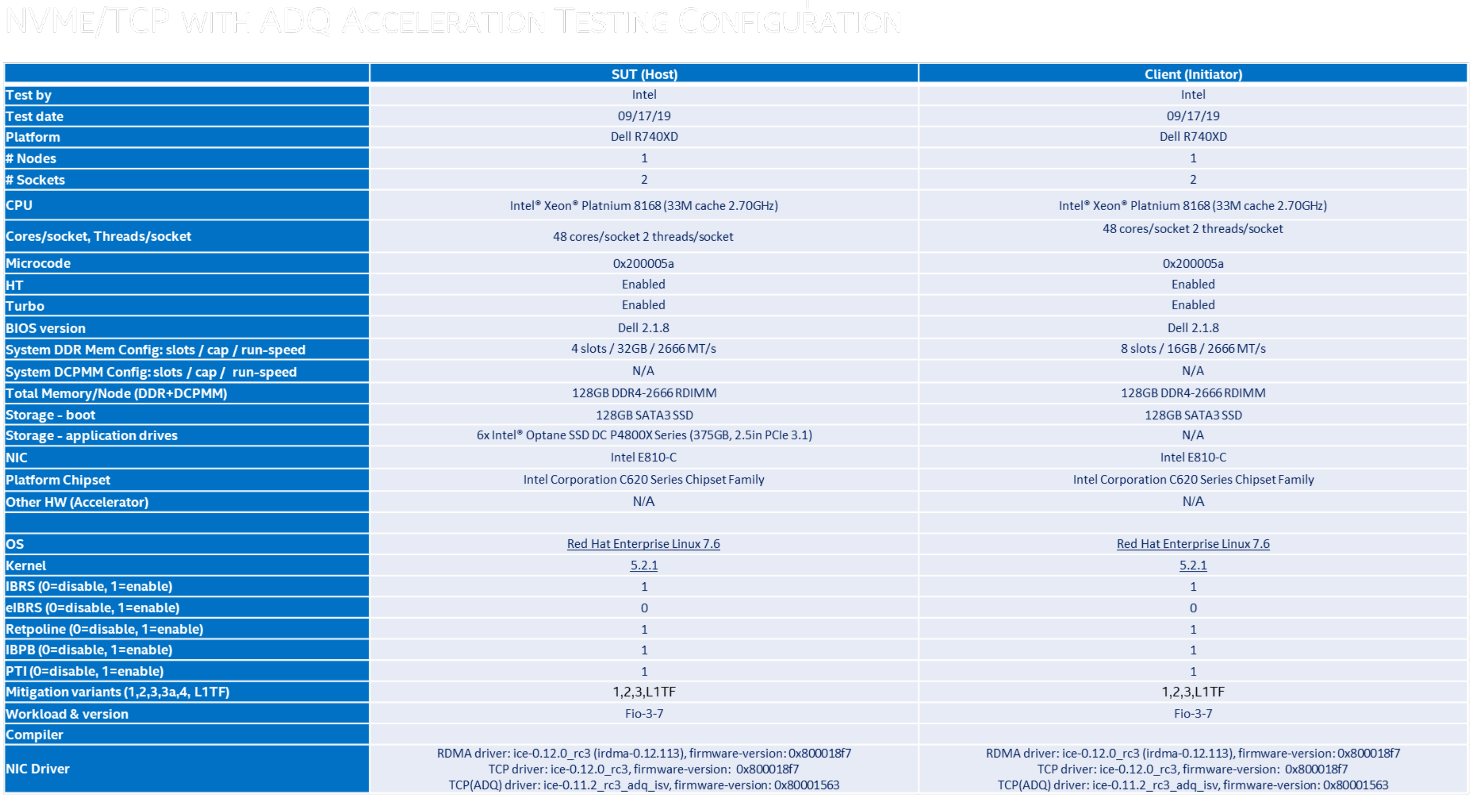
Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

NVMe-oF Test Configuration Info



NVMe/TCP with ADQ Acceleration Testing Configuration

	SUT (Host)	Client (Initiator)
Test by	Intel	Intel
Test date	09/17/19	09/17/19
Platform	Dell R740XD	Dell R740XD
# Nodes	1	1
# Sockets	2	2
CPU	Intel® Xeon® Platnium 8168 (33M cache 2.70GHz)	Intel® Xeon® Platnium 8168 (33M cache 2.70GHz)
Cores/socket, Threads/socket	48 cores/socket 2 threads/socket	48 cores/socket 2 threads/socket
Microcode	0x200005a	0x200005a
HT	Enabled	Enabled
Turbo	Enabled	Enabled
BIOS version	Dell 2.1.8	Dell 2.1.8
System DDR Mem Config: slots / cap / run-speed	4 slots / 32GB / 2666 MT/s	8 slots / 16GB / 2666 MT/s
System DCPMM Config: slots / cap / run-speed	N/A	N/A
Total Memory/Node (DDR+DCPMM)	128GB DDR4-2666 RDIMM	128GB DDR4-2666 RDIMM
Storage - boot	128GB SATA3 SSD	128GB SATA3 SSD
Storage - application drives	6x Intel® Optane SSD DC P4800X Series (375GB, 2.5in PCIe 3.1)	N/A
NIC	Intel E810-C	Intel E810-C
Platform Chipset	Intel Corporation C620 Series Chipset Family	Intel Corporation C620 Series Chipset Family
Other HW (Accelerator)	N/A	N/A
OS	Red Hat Enterprise Linux 7.6	Red Hat Enterprise Linux 7.6
Kernel	5.2.1	5.2.1
IBRS (0=disable, 1=enable)	1	1
eIBRS (0=disable, 1=enable)	0	0
Retpoline (0=disable, 1=enable)	1	1
IBPB (0=disable, 1=enable)	1	1
PTI (0=disable, 1=enable)	1	1
Mitigation variants (1,2,3,3a,4, L1TF)	1,2,3,L1TF	1,2,3,L1TF
Workload & version	Fio-3-7	Fio-3-7
Compiler		
NIC Driver	RDMA driver: ice-0.12.0_rc3 (irdma-0.12.113), firmware-version: 0x800018f7 TCP driver: ice-0.12.0_rc3, firmware-version: 0x800018f7 TCP(ADQ) driver: ice-0.11.2_rc3_adq_isv, firmware-version: 0x80001563	RDMA driver: ice-0.12.0_rc3 (irdma-0.12.113), firmware-version: 0x800018f7 TCP driver: ice-0.12.0_rc3, firmware-version: 0x800018f7 TCP(ADQ) driver: ice-0.11.2_rc3_adq_isv, firmware-version: 0x80001563