SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

Tuning and Optimizing Ethernet-based NVMe over Fabric Transport Protocols

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

intel

Notices and Disclaimers

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 information go to www.intel.com/benchmarks.

Performance results are based on testing as of date disclosed in the system configuration and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at www.intel.com.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Optimization Notice: 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. Notice Revision #20110804.

Intel, the Intel logo, Intel Atom®, Xeon and Xeon logos, are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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

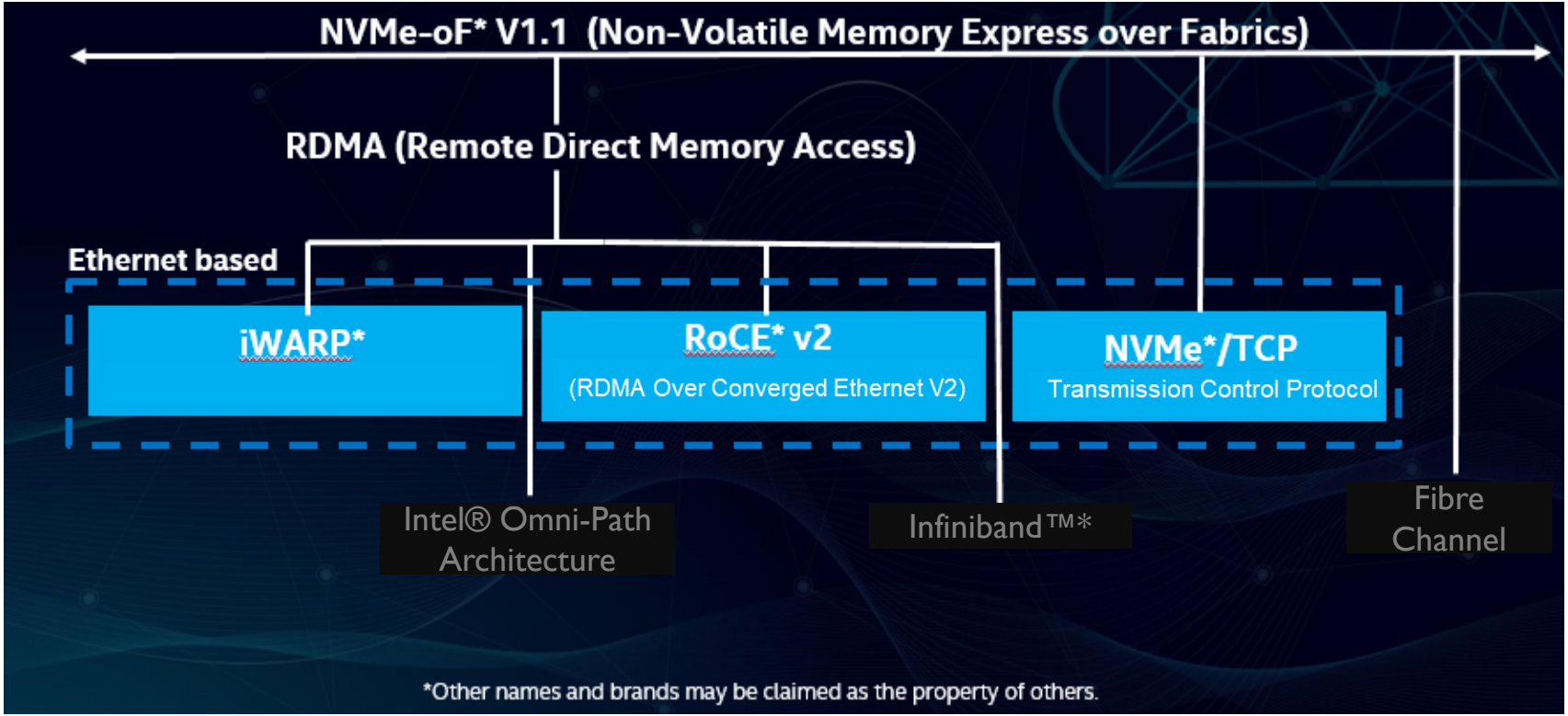
Agenda

- NVMe-oF Ethernet Transport Review
- NVMe* Host-Side CPU Efficiency Characterization
- NVMe-oF Target-Side CPU Efficiency Characterization
- Conclusion

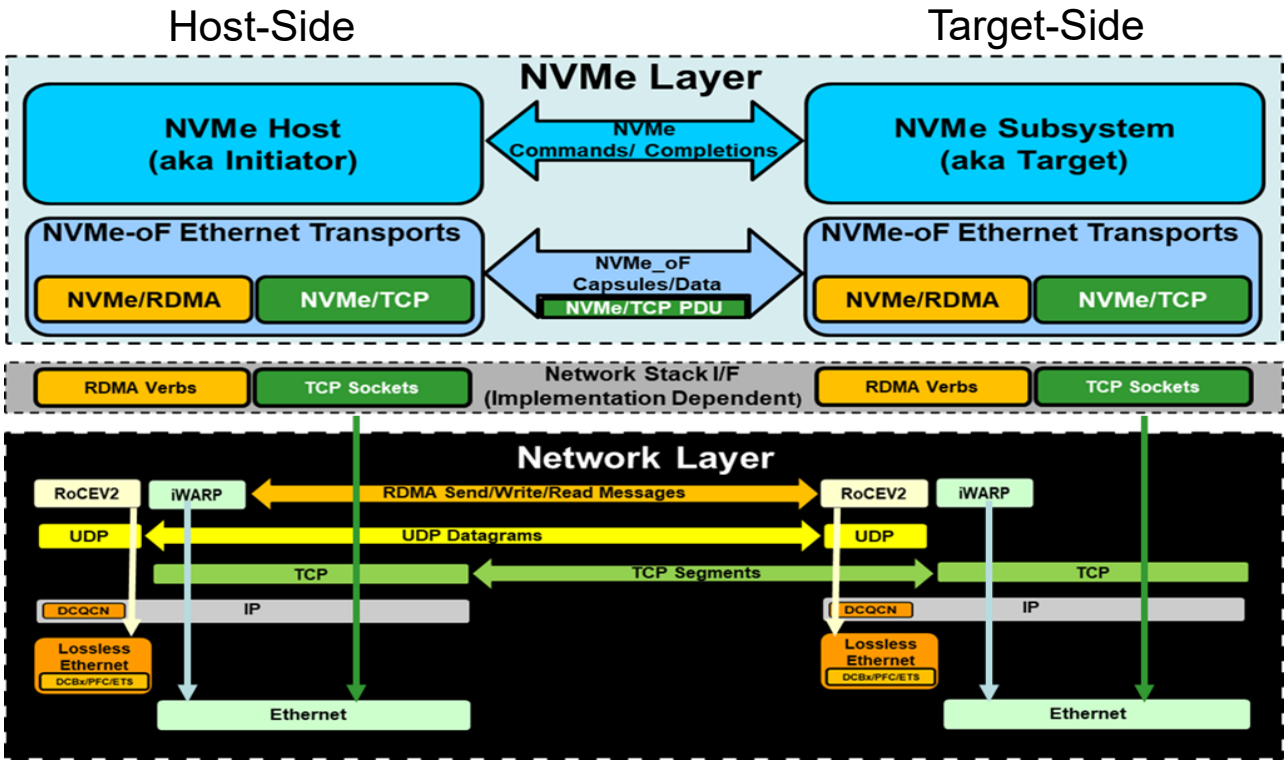
NVMe-oF Transports Review

- Full material in SDC ’19 presentation on
  Selecting an NVMe-oF™ Ethernet Transport RDMA or TCP?

NVMe-oF Ethernet Transports



NVMe-oF Ethernet Transports Layering

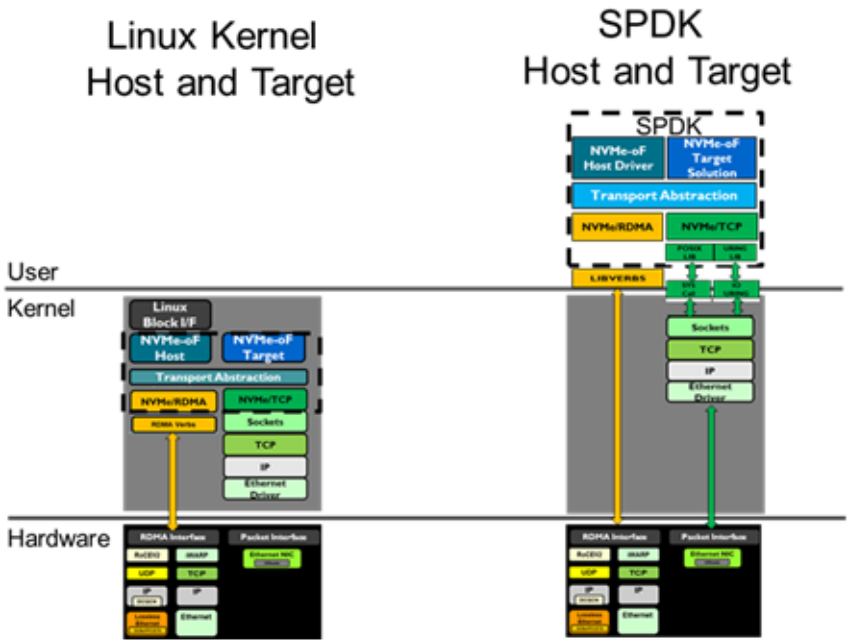


NVMe-oF Ethernet Transport Comparison

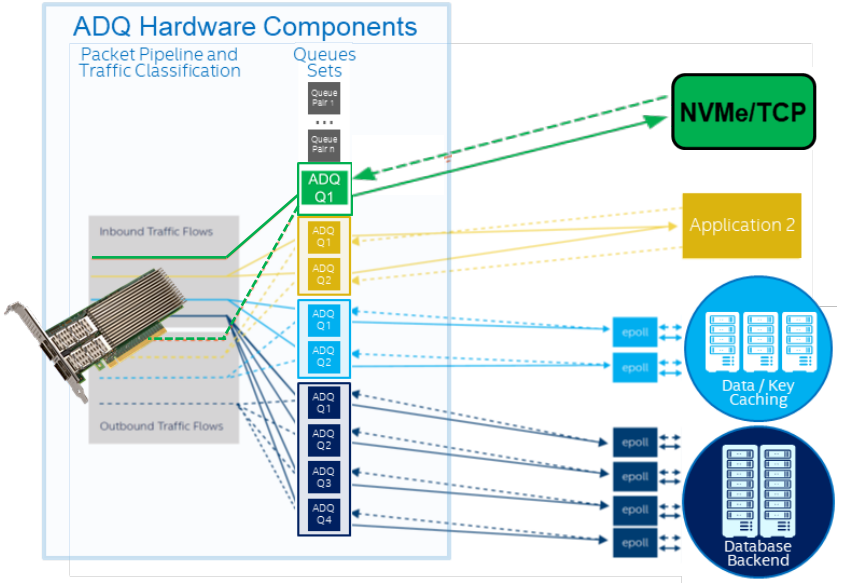
	NVMe/TCP	NVMe/RDMA iWARP	NVMe/RDMA RoCEV2	NVMe/TCP with ADQ*
Network Infrastructure	Standard NIC(s) + Standard Ethernet switches	RDMA Enabled NIC(s) + Standard Ethernet switches	RDMA Enabled NIC(s) + Lossless Ethernet switches	Standard NIC(s) + Standard Ethernet switches
Performance	Baseline IOPS and CPU efficiency, Highest tail latency	High IOPS and CPU efficiency, lowest tail latency	High IOPS and CPU efficiency, lowest tail latency	High IOPS and CPU efficiency, low tail latency
O/S Network Software	In-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

Available NVMe-oF Open Source Stacks

- Linux Kernel
  - In-Kernel Host and Target
  - RDMA, TCP, TCP+ADQ
- SPDK *(SPDK)*
  - User-level Host and Target
  - RDMA, TCP, TCP+ADQ
- Mix/Match of Host and Target stacks interoperate using standard NVMe-oF protocol and transports



Application Device Queues (ADQ)



ADQ Basics

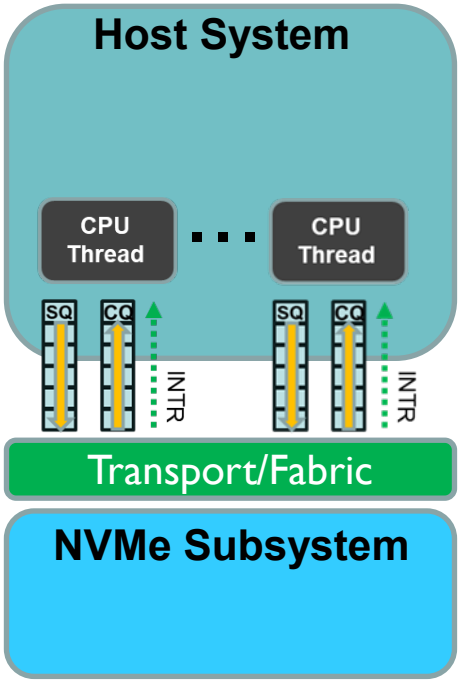
- Filters application traffic to a dedicated set of queues
- Application threads of execution are connected to specific queues within the ADQ queue set
- Bandwidth control of egress (Tx) network traffic per application

	Capability
Application	Align Application Threads and ADQ's
Kernel	Busy Polling Device Queues (e.g, epoll(), recv(), poll()) 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.

NVMe Host-Side CPU Efficiency

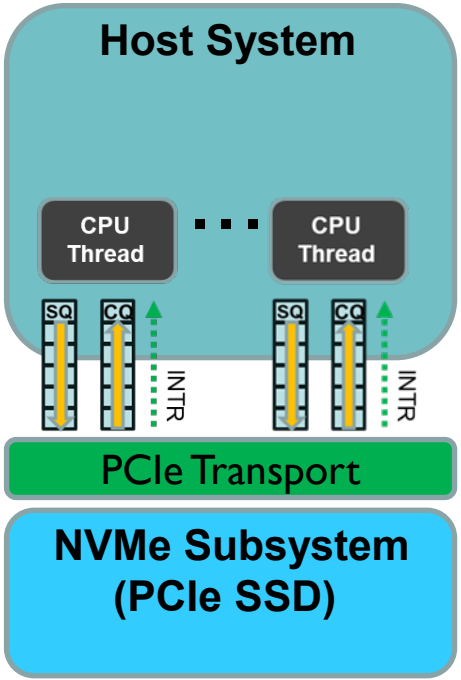
NVMe Host-Side CPU Efficiency



NVMe Host CPU Efficiency Contributors

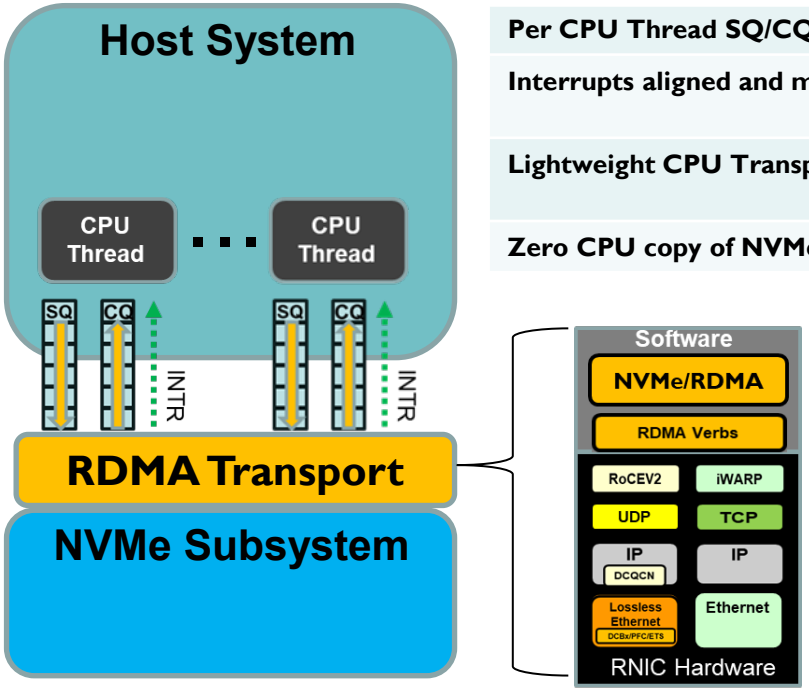
- Per CPU Thread Submission/Completion Queues
- Efficient completion model
  - Interrupt aligned between CQ and CPU Thread and interrupts moderated (Kernel Stack)
  - CPU polling of CQ (SPDK Stack)
- Lightweight Transport Stack
- Zero CPU copy of NVMe Data

NVMe Host-Side CPU Efficiency (PCIe Transport Example)

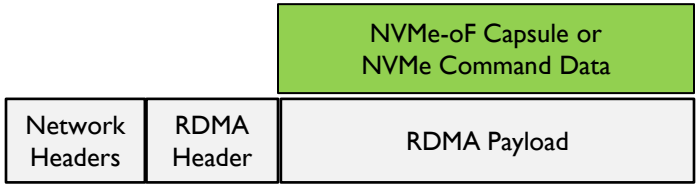


NVMe Efficiency Contributors	PCIe Transport
Per CPU Thread SQ/CQ	Yes, Up to the NVMe device limit
Interrupts aligned and moderated	Yes, MSI-X per NVMe CQ per CPU (Polling for SPDK)
Lightweight CPU Transport Stack	Yes, due to shared memory queues
Zero CPU copy of NVMe Data	Yes, due to shared memory buffers

NVMe Host-Side CPU Efficiency (RDMA Ethernet Transport)



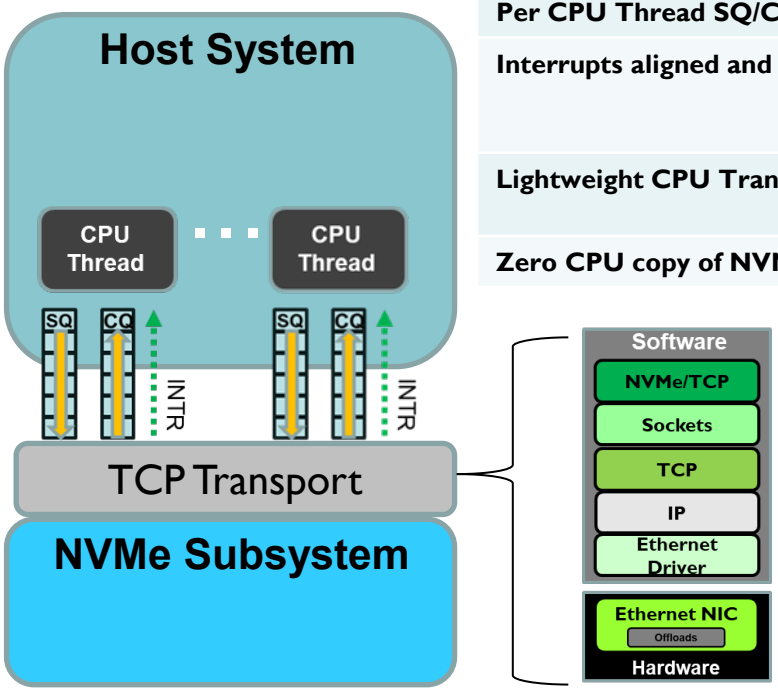
NVMe Efficiency Contributors	NVMe/RDMA Transport
Per CPU Thread SQ/CQ	Yes, NVMe CQ/SQ mapped to RDMA QPs
Interrupts aligned and moderated	Yes, MSI-X per RDMA_EVQ, Polling RDMA_CQ for SPDK
Lightweight CPU Transport Stack	Yes, due to H/W network transport and kernel bypass (for SPDK)
Zero CPU copy of NVMe Data	Yes, due to RDMA READ/WRITE

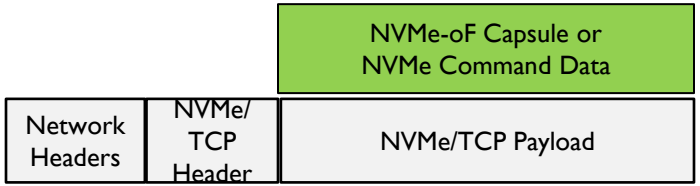


RDMA Header used by RNIC Hardware to place RDMA payload into host memory-resident NVMe CQs and NVMe command data buffers

NVMe Host-Side CPU Efficiency (TCP Ethernet Transport)

NVMe Efficiency Contributors	NVMe/TCP Transport
Per CPU Thread SQ/CQ	NVMe CQ/SQ mapped to TCP Connections
Interrupts aligned and moderated	Alignment based on Kernel and NIC H/W TCP Flow Steering Logic Socket-level polling (for SPDK)
Lightweight CPU Transport Stack	Heavier due to transport in software Utilizes NIC TCP H/W Offloads
Zero CPU copy of NVMe Data	Yes on transmit, No on reception

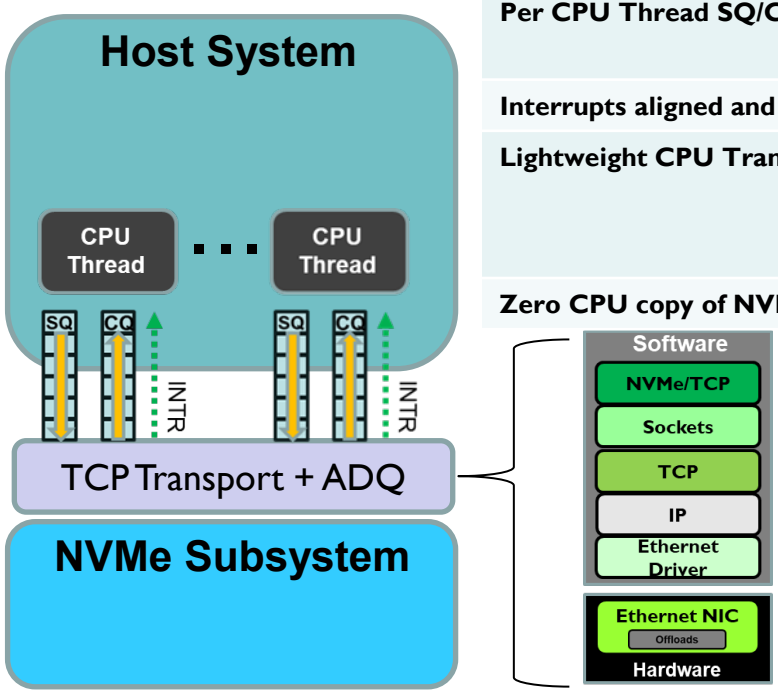


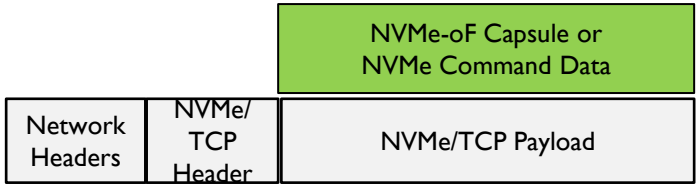


Network and NVMe/TCP headers processed by CPU software transports

NVMe Host-Side CPU Efficiency (TCP Ethernet Transport with ADQ)

NVMe Efficiency Contributors	NVMe/TCP Transport + ADQ
Per CPU Thread SQ/CQ	NVMe CQ/SQ mapped to TCP Connections
Interrupts aligned and moderated	Enhanced interrupt moderation
Lightweight CPU Transport Stack	Busy polling leverages block polling interface In context processing of requests and responses Efficient load distribution among dedicated NIC HW queues
Zero CPU copy of NVMe Data	Yes, on transmit, no on reception





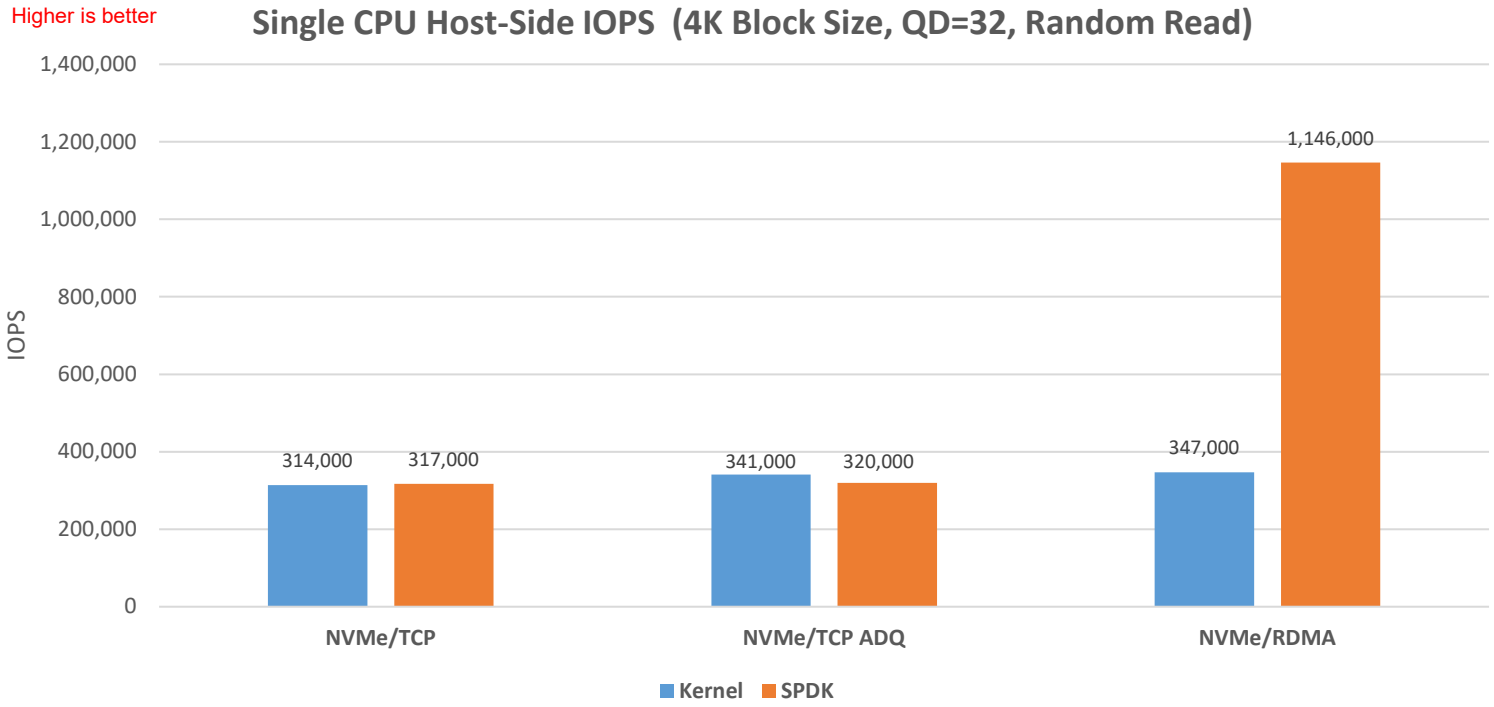
Network and NVMe/TCP headers processed by CPU software transports

TCP Transport with ADQ Efficiency Optimizations

NVMe/TCP/ADQ Optimizations	What is it?	Benefit
Dedicated and isolated queue set with enhanced steering	NIC HW queues for NVMe/TCP traffic Queue identification visibility to application Optimized queue selection within queue set	Reduces jitter Enables targeted application specific queuing optimizations
NIC driver optimizations	Signaling and steering enhancements for NIC queues	Reduces jitter and latency, improves throughput
Egress rate shaping	HW accelerated shaping per queue set	Divides bandwidth among multiple applications
Socket priority	Socket priority mapped to queue set	Reduces jitter
In context Request/Response	NVMe-oF requests and responses without switching contexts	Reduces context switches and jitter
Busy Polling	Polls queues in application context	Reduces jitter, context switches, latency, improves throughput
Grouping	Groups multiple NVMe queues into a NIC HW queue	Reduces CPU utilization and latency

Host-Side Efficiency Measurements

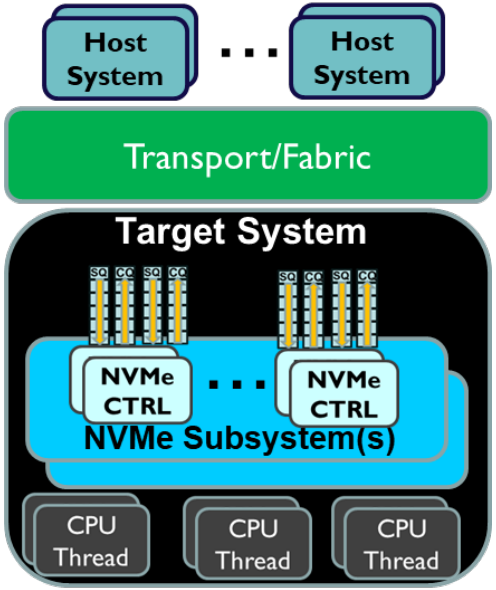
Higher is better



For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. See configuration disclosure for details. For more information regarding performance and optimization choices in Intel software products., please visit https://software.intel.com/en-us/articles/optimization-notice.

NVMe Target-Side CPU Efficiency

NVMe Target-Side CPU Efficiency



NVMe_oF Target CPU Efficiency Contributors

- CPU Thread submission/completion queue mappings
- Interrupt and/or polling scheduling techniques on transports
- Lightweight transport with queue separation
- Efficient distribution of multi-host NVMe-oF Queue processing onto target-side CPU Threads

NVMe Target-Side CPU Efficiency (RDMA Ethernet Transport)

NVMe Efficiency Contributors	NVMe/RDMA Transport
CPU Thread to SQ/CQ Mappings	NVMe CQ/SQ mapped to RDMA QP Distribution of RDMA CQ vectors across target CPU cores (Kernel)
Interrupt and polling techniques	RDMA CQ Polling Conservative interrupt moderation (Kernel) RDMA CQ Polling (for SPDK Target)
Lightweight CPU Transport Stack	H/W offload of NVMe_oF Capsule and Data transfers Kernel-bypass used (for SPDK target)

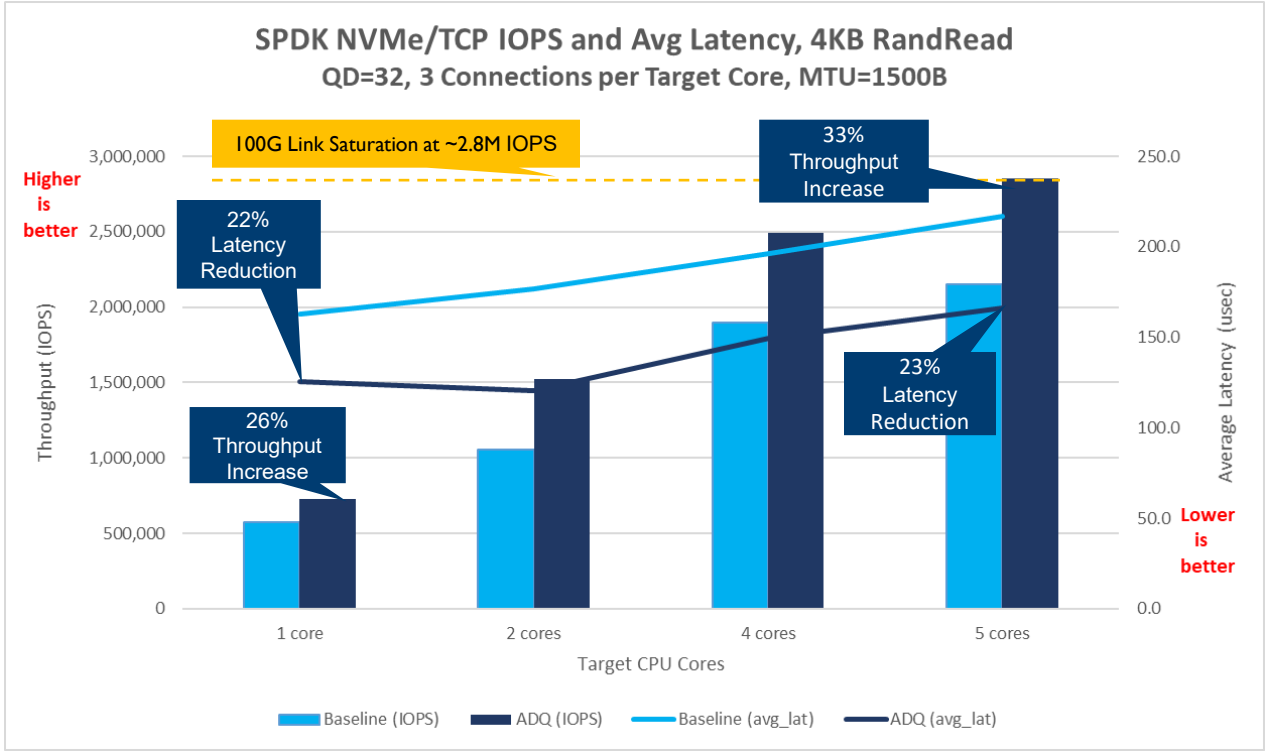
NVMe Target-Side CPU Efficiency (TCP Ethernet Transport)

NVMe Efficiency Contributors	TCP Transport
CPU Thread to SQ/CQ Mappings	NVMe CQ/SQ mapped to TCP Connections Optimized I/O thread alignment
Interrupt and polling techniques	Based on NIC H/W capabilities Interrupt moderation
Lightweight CPU Transport Stack	Standard TCP socket select/service mechanism Packet-level transport S/W – H/W interface with stateless optimizations (TSO, GRO)

NVMe Target-Side CPU Efficiency (TCP Ethernet Transport with ADQ)

NVMe Efficiency Contributors	TCP Transport + ADQ
CPU Thread to SQ/CQ Mappings	NVMe CQ/SQ mapped to TCP Connections Optimized thread alignment NVMe TCP connections grouped on per CPU
Interrupt and polling techniques	Based on NIC H/W capabilities Enhanced interrupt moderation
Lightweight CPU Transport Stack	Efficient busy polling, per group Align connection group per NIC HW queue pair Efficient load distribution among NIC HW queues

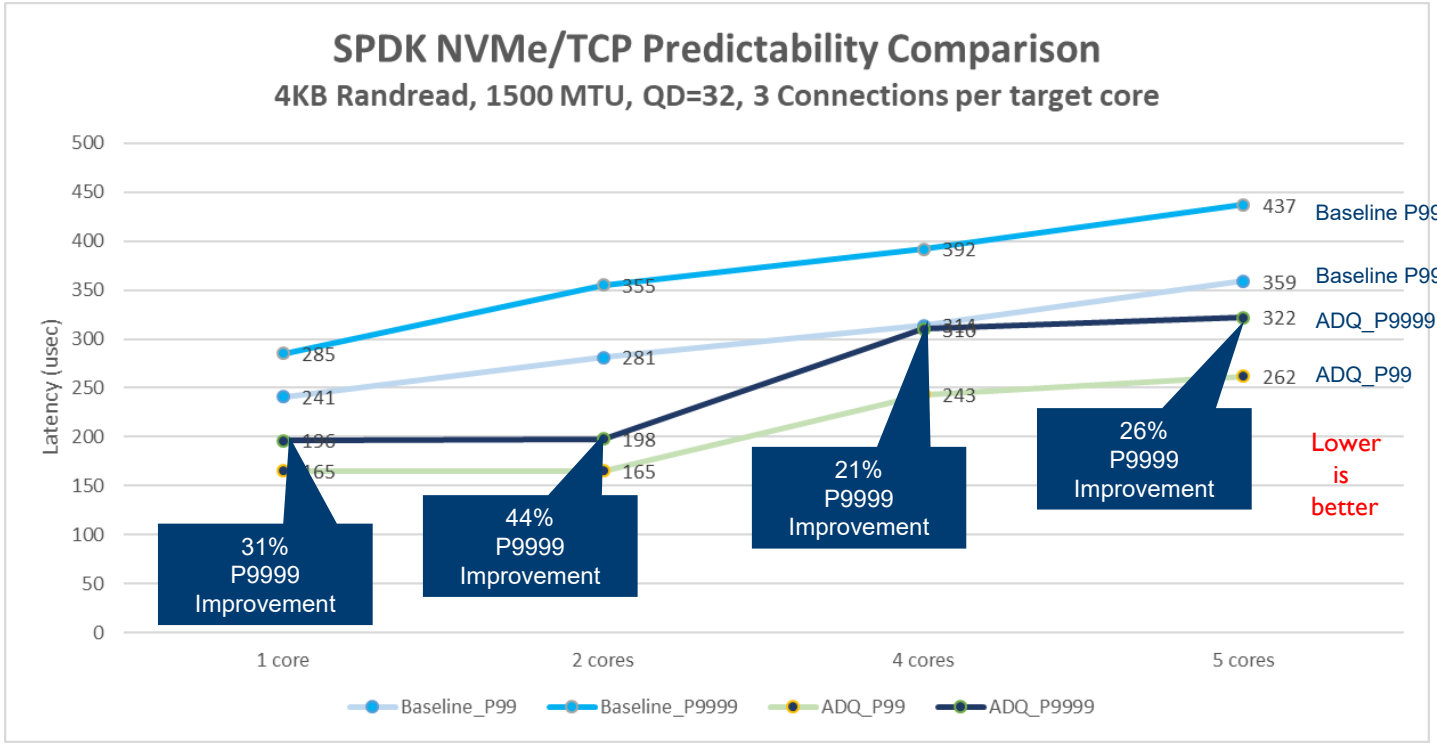
ADQ Improves IOPS with Reduced Latency - Read



- ADQ shows substantial performance improvement to SPDK NVMe/TCP implementation.
- With ADQ, 5 cores can saturate 100Gb link with 4KB IO size.

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. See configuration disclosure for details. For more information regarding performance and optimization choices in Intel software products., please visit https://software.intel.com/en-us/articles/optimization-notice.

ADQ Improves Predictability



- ADQ reduces tail of latency substantially across different configurations.

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. See configuration disclosure for details. For more information regarding performance and optimization choices in Intel software products., please visit https://software.intel.com/en-us/articles/optimization-notice.

Conclusions

- Multiple NVMe-oF transport options available depending on network storage requirements
- NVMe-oF host-side performance similar across kernel transports
  - SPDK user-level NVMe/RDMA offers highest performance for applicable systems
- NVMe/TCP transport with ADQ significantly improves performance
- For more information attend the following SDC20 sessions:
  - Optimizing user space NVMe-oF TCP transport solution with both software and hardware methodologies
  - Improving NVMe/TCP Performance by Enhancing Software and Hardware
  - Visit www.intel.com/adq for more information on ADQ

Please take a moment
to rate this session.

Your feedback matters to us.

intel

Linux Kernel NVMe/RDMA, NVMe/TCP with ADQ Testing Configuration

	SUT	Client
Test by	Intel	Intel
Test date	9/14/2020	9/14/2020
Platform	Dell R740XD	Dell R740XD
# Nodes	1	1
# Sockets	2	2
CPU	Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz	Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz
Cores/socket, Threads/socket	28 cores per socket, 56 threads per socket	28 cores per socket, 56 threads per socket
ucode	0x500001c	0x500001c
HT	Enabled*	Enabled*
Turbo	Enabled	Enabled
BIOS version	2.1.8	2.1.8
System DDR Mem Config: slots / cap / run-speed	8 slots / 16GB / 2666 MT/s + 4 slots / 8GB / 2666 MT/s	8 slots / 16GB / 2666 MT/s + 4 slots / 8GB / 2666 MT/s
System DCPMM Config: slots / cap / run-speed	N/A	N/A
Total Memory/Node (DDR+DCPMM)	160GB DDR4-2666 DIMM	160GB DDR4-2666 DIMM
Storage - boot	100GB SATA SSD	100GB SATA SSD
Storage - application drives	6x Intel® Optane ™ SSD DC P4800X, PCIe 3.0, x4	N/A
Network Adapter	Intel E810-C	Intel E810-C
PCH	Intel Corporation C620 Series Chipset Family	Intel Corporation C620 Series Chipset Family
Other HW (Accelerator)	N/A	N/A
OS	Red Hat Enterprise Linux 8.1 (Ootpa)	Red Hat Enterprise Linux 8.1 (Ootpa)
Kernel	5.8.0	5.8.0
Workload & version	Fio 3.21	Fio 3.21
Compiler	GCC 8.3.1 20191121 (Red Hat 8.3.1-5)	GCC 8.3.1 20191121 (Red Hat 8.3.1-5)
Network Adapter Driver	ice-1.1.3; irdma-1.1.21; FW: 2.10 0x8000433e 1.2789.0	ice-1.1.3; irdma-1.1.21; FW: 2.10 0x8000433e 1.2789.0

* HT enabled for RDMA. For kernel, turned off for benchmark purposes to not schedule FIO on threads of same physical core

SPDK NVMe/TCP with ADQ Testing Configuration

	SUT	Client
Test by	Intel	Intel
Test date	8/27/2020	8/27/2020
Platform	Dell R740XD	Dell R740XD
# Nodes	1	1
# Sockets	2	2
CPU	Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz	Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz
Cores/socket, Threads/socket	28 cores per socket, 56 threads per socket	28 cores per socket, 56 threads per socket
ucode	0x500001c	0x500001c
HT	Disabled *	Disabled
Turbo	Enabled	Enabled
BIOS version	2.1.8	2.1.8
System DDR Mem Config: slots / cap / run-speed	8 slots / 16GB / 2666 MT/s + 4 slots / 8GB / 2666 MT/s	8 slots / 16GB / 2666 MT/s + 4 slots / 8GB / 2666 MT/s
System DCPMM Config: slots / cap / run-speed	N/A	N/A
Total Memory/Node (DDR+DCPMM)	160GB DDR4-2666 DIMM	160GB DDR4-2666 DIMM
Storage - boot	100GB SATA SSD	100GB SATA SSD
Storage - application drives	6x Intel® Optane ™ SSD DC P4800X, PCIe 3.0, x4	N/A
Network Adapter	Intel E810-C	Intel E810-C
PCH	Intel Corporation C620 Series Chipset Family	Intel Corporation C620 Series Chipset Family
Other HW (Accelerator)	N/A	N/A
OS	Red Hat Enterprise Linux 8.1 (Ootpa)	Red Hat Enterprise Linux 8.1 (Ootpa)
Kernel	5.8.0	5.8.0
Workload & version	Fio 3.15	Fio 3.15
Compiler	GCC 8.3.1 20191121 (Red Hat 8.3.1-5)	GCC 8.3.1 20191121 (Red Hat 8.3.1-5)
Network Adapter Driver	Ice-1.2.0-rc4 FW ver: 0x8000433e	Ice-1.2.0-rc4 Fw ver: 0x8000433e

*HT was turned off for benchmark purposes to not schedule FIO on threads of same physical core