STORAGE DEVELOPER CONFERENCE
SDC 22
Fremont, CA
September 12-15, 2022
BY Developers FOR Developers

A SNIA Event

Protecting NVMe/TCP Data at 400 Gbps

With Intel® Data Streaming Accelerator

John Kariuki

Notices & Disclaimers

Results have been estimated or simulated.

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

All product plans and roadmaps are subject to change without notice.

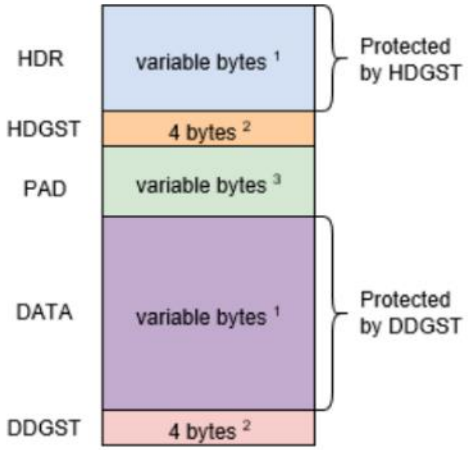
Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

Introduction and Test Configuration

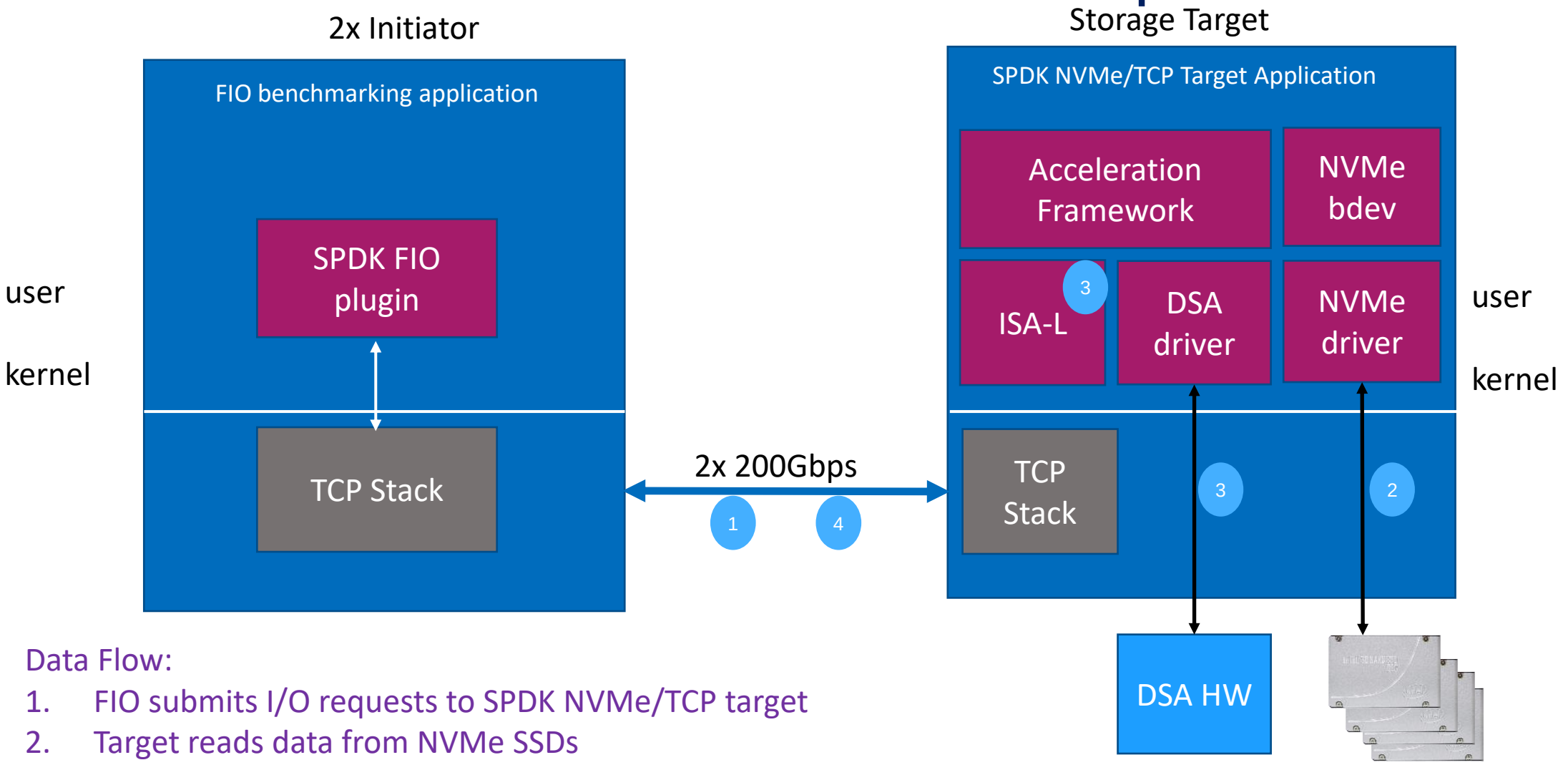
Introduction, test setup and configuration

NVMe/TCP Data Digest

- Host & Controller communicate over TCP by exchangin NVMe/TCP Protocol Data Unit (PDU).
- NVMe/TCP PDU has 5 parts.
- Data Digest (DDGST) field: Protects PDU data if enabled
- Data Digest calculated using CRC32C algorithm in RFC3385
- CRC32C algorithm provides strong error detection on communication networks



NVMe-oF/TCP Test Setup



Data Flow:

1. FIO submits I/O requests to SPDK NVMe/TCP target
2. Target reads data from NVMe SSDs
3. Target uses DSA/ISA-L to calculate CRC32C data digest
4. Target sends I/O data + CRC32C data digest to FIO

SPDK NVMe/TCP Test Configurations

1. No Digest: Data digest is disabled. No CRC32C computation.
2. ISA-L Digest: Data digest is enabled. CRC32C calculated with Vector CLMUL AVX512 SIMD instructions
3. DSA Digest: Data Digest is enabled. CRC32C calculation offloaded to Intel® DSA

Test Results

Performance and Efficiency in IOPS/Core Latency

NVMe/TCP Target Single Core Test

Pre-Production 4th Generation Intel® Xeon® Scalable Processor

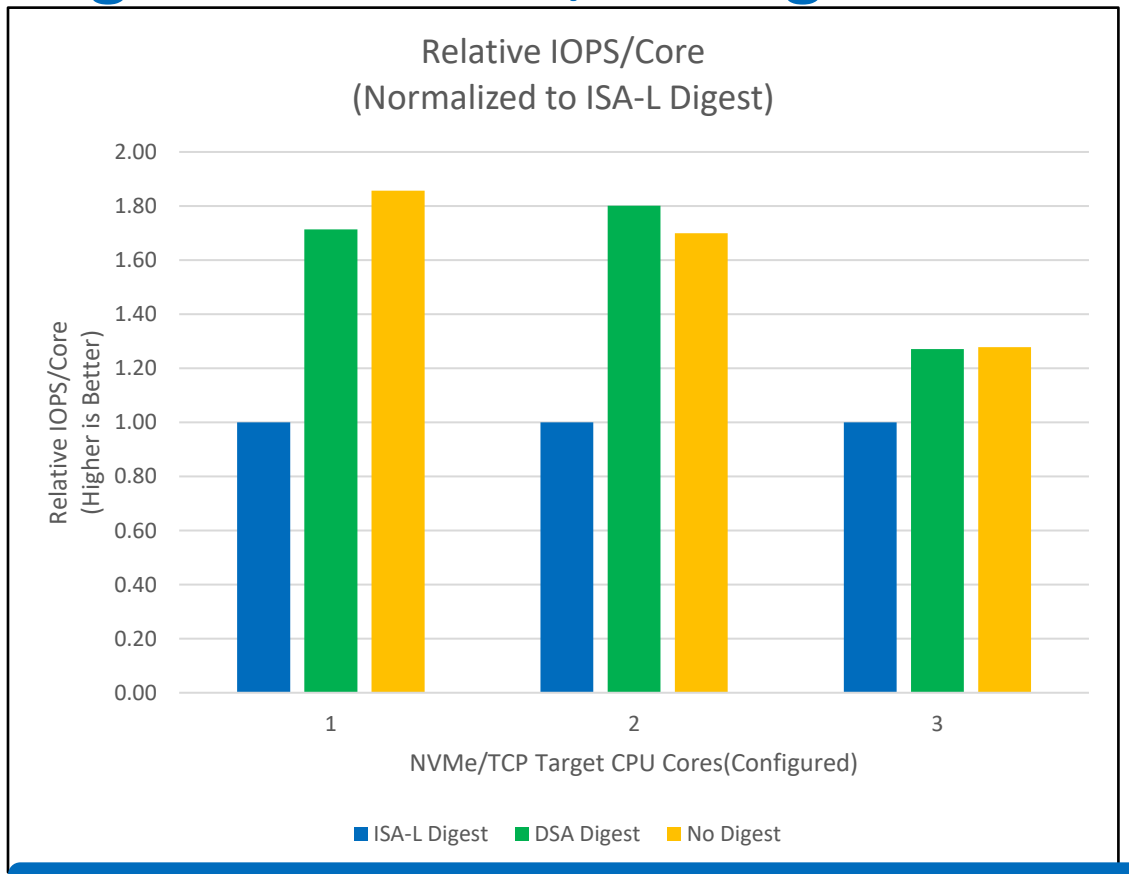
		Relative IOPS/Core
Block Size	QD at Target	DSA vs. ISA-L Digest
4KB (Rand Read)	128	97%
	256	105%
	512	102%
	1024	106%
16KB (Rand Read)	128	107%
	256	106%
	512	117%
	1024	134%
128KB (Seq Read)	128	99%
	256	128%
	512	145%
	1024	171%

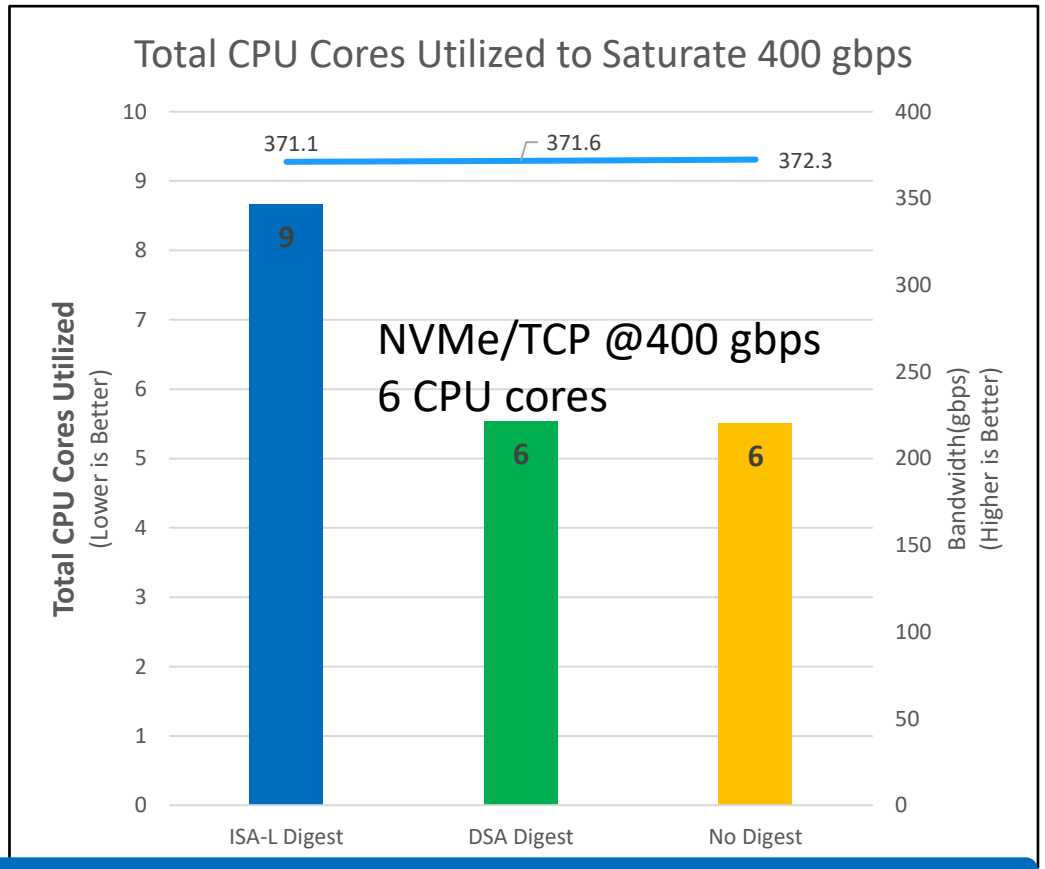
- When CPU-bound, using Intel® DSA improves IOPS/Core by:
  - Up to 1.34x for 16 KB Rand Read
  - Up to 1.71x for 128 KB Seq Reads
- Minimal gains for 4KB I/O
- At low QD workload NOT CPU constrained

Results are based on Intel internal testing on pre-production cpu, systems and software. See backup for workloads and configurations. Results may vary.

128KB Sequential Reads

Significant IOPS/Core gains for 128KB I/O when CPU-bound





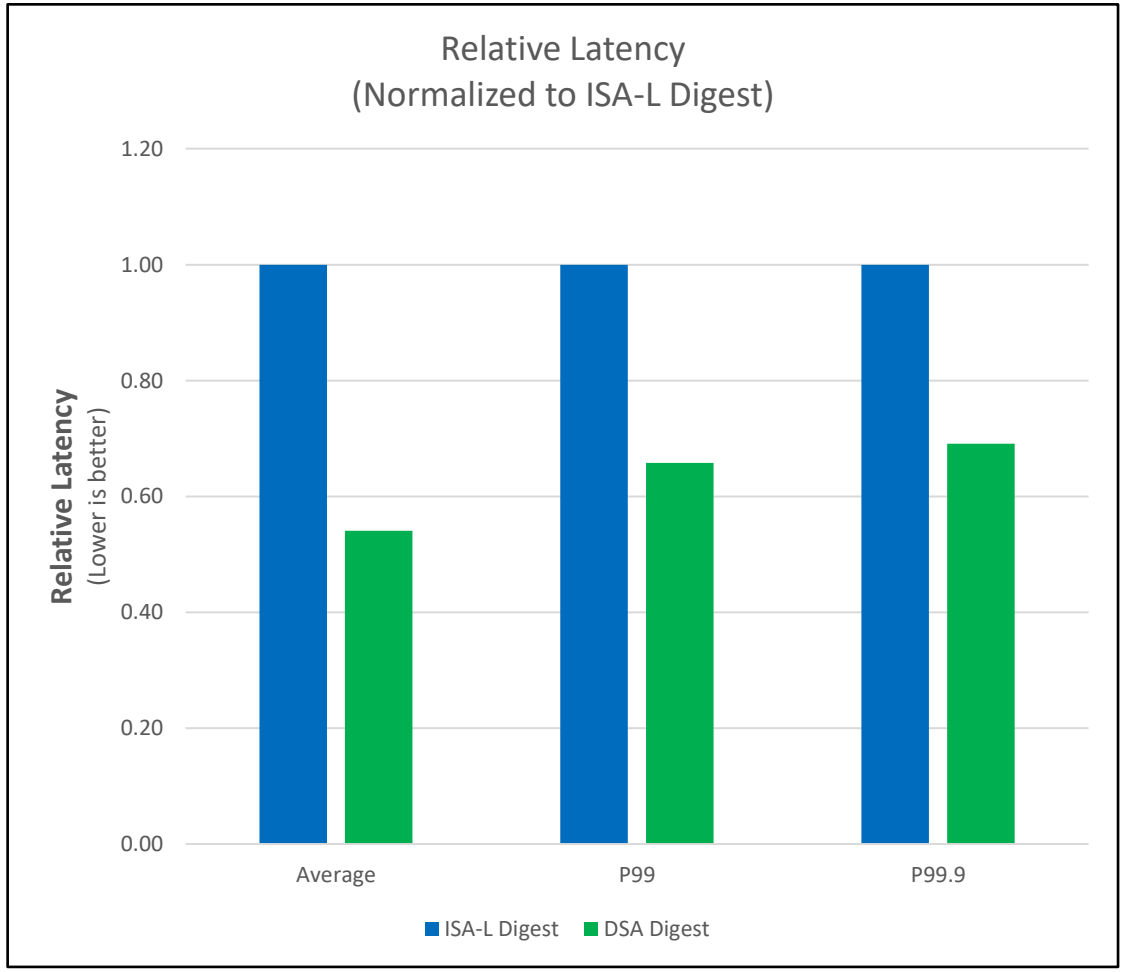
With Intel® DSA 33% reduction in CPU cores used to saturate 400 gbps network, eliminating 3 cores of ISA-L SW overhead

Results are based on Intel internal testing on pre-production cpu, systems and software. See backup for workloads and configurations. Results may vary.

128KB Sequential Read Latency (NVMe/TCP target configured with 3 CPU cores)

For large seq I/O, offloading data digest to Intel® DSA reduces:

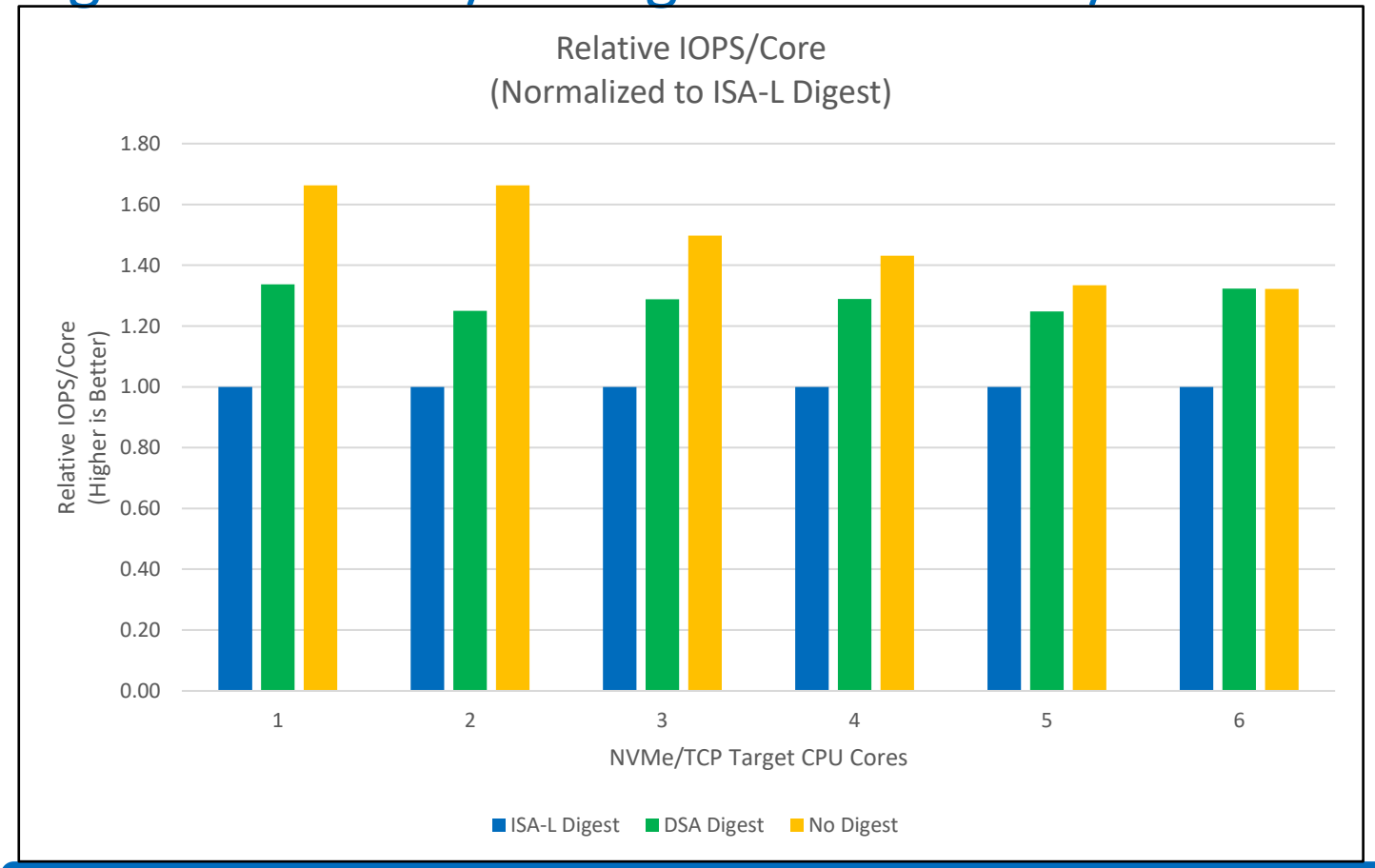
- Average latency by up to 46%
- P99 latency by up to 34%
- P99.9 latency by up to 31%

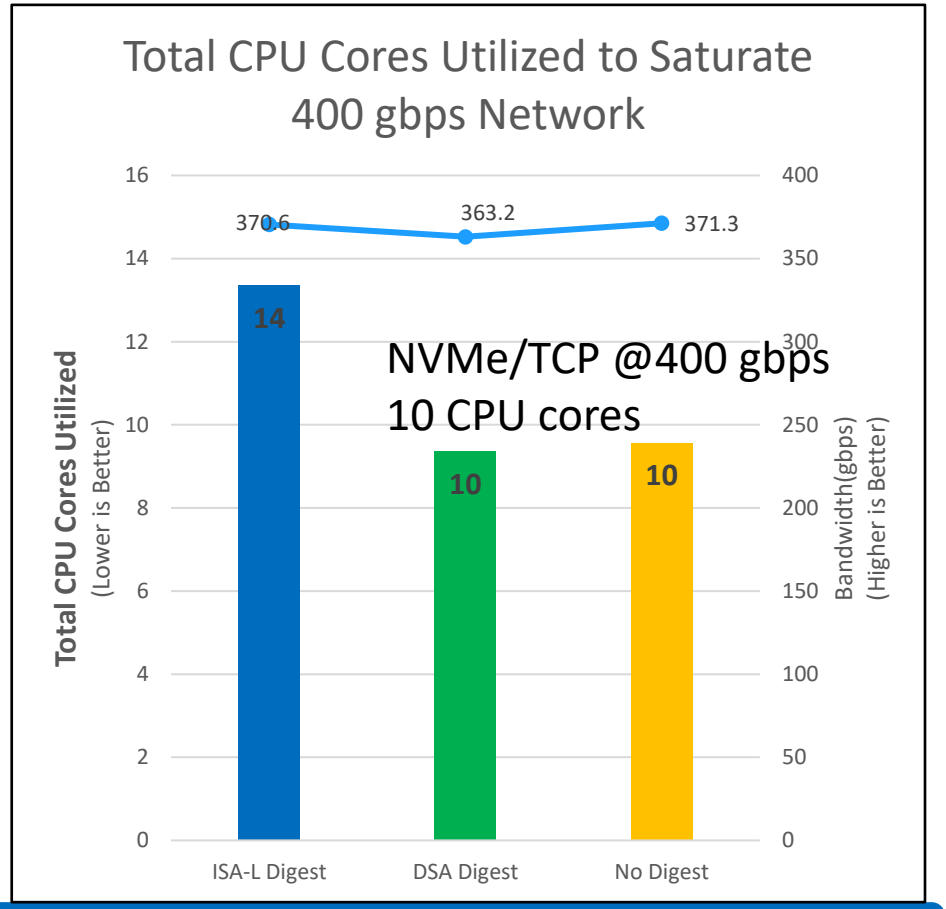


Results are based on Intel internal testing on pre-production cpu, systems and software. See backup for workloads and configurations. Results may vary.

16KB Random Read

Significant IOPS/Core gains for 16KB I/O when CPU-bound

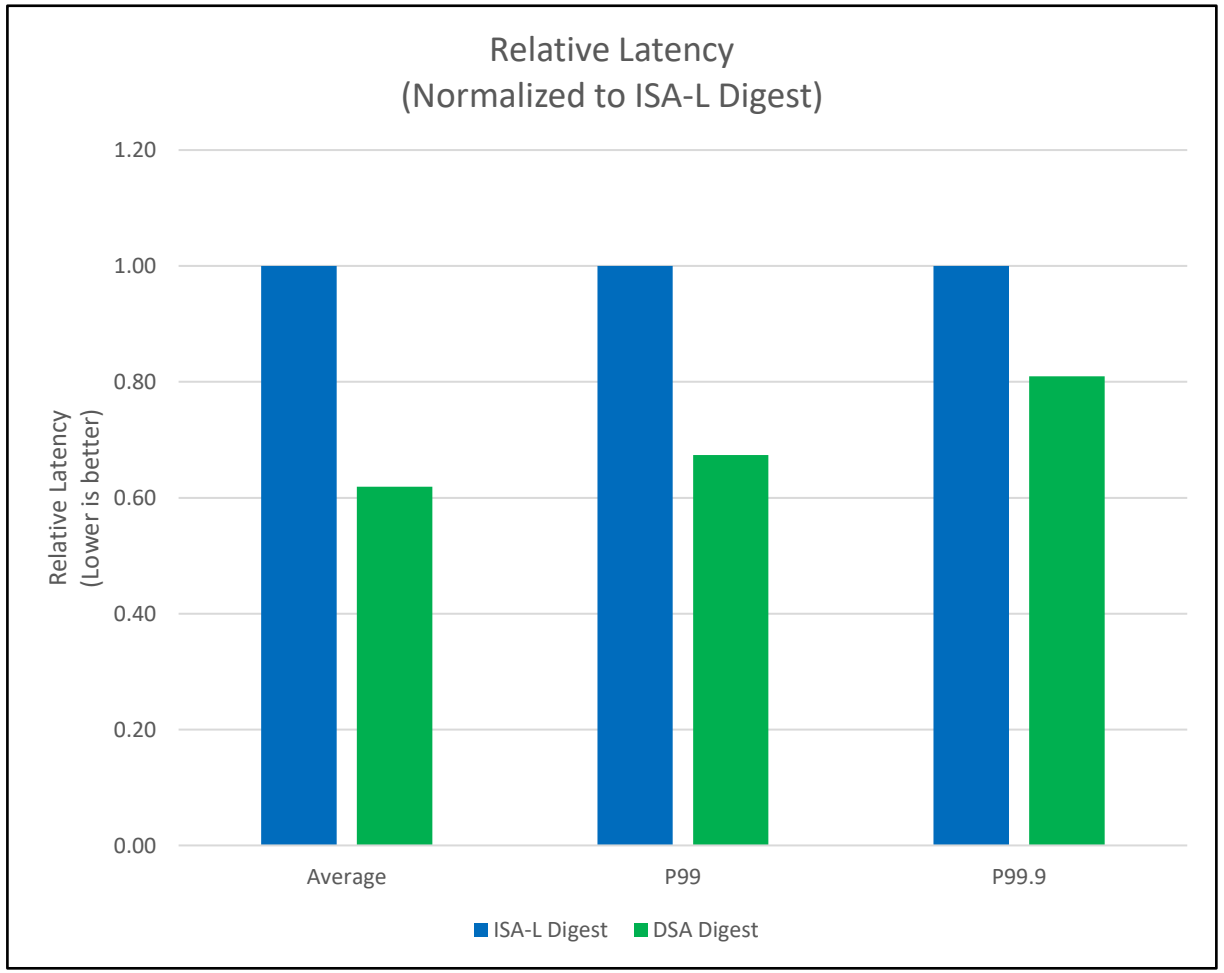




With Intel® DSA 28% reduction in CPU cores to saturate 400 gbps network, eliminating 4 cores of SW overhead

Results are based on Intel internal testing on pre-production cpu, systems and software. See backup for workloads and configurations. Results may vary.

16KB Rand Read Latency (NVMe/TCP target configured with 6 CPU cores)

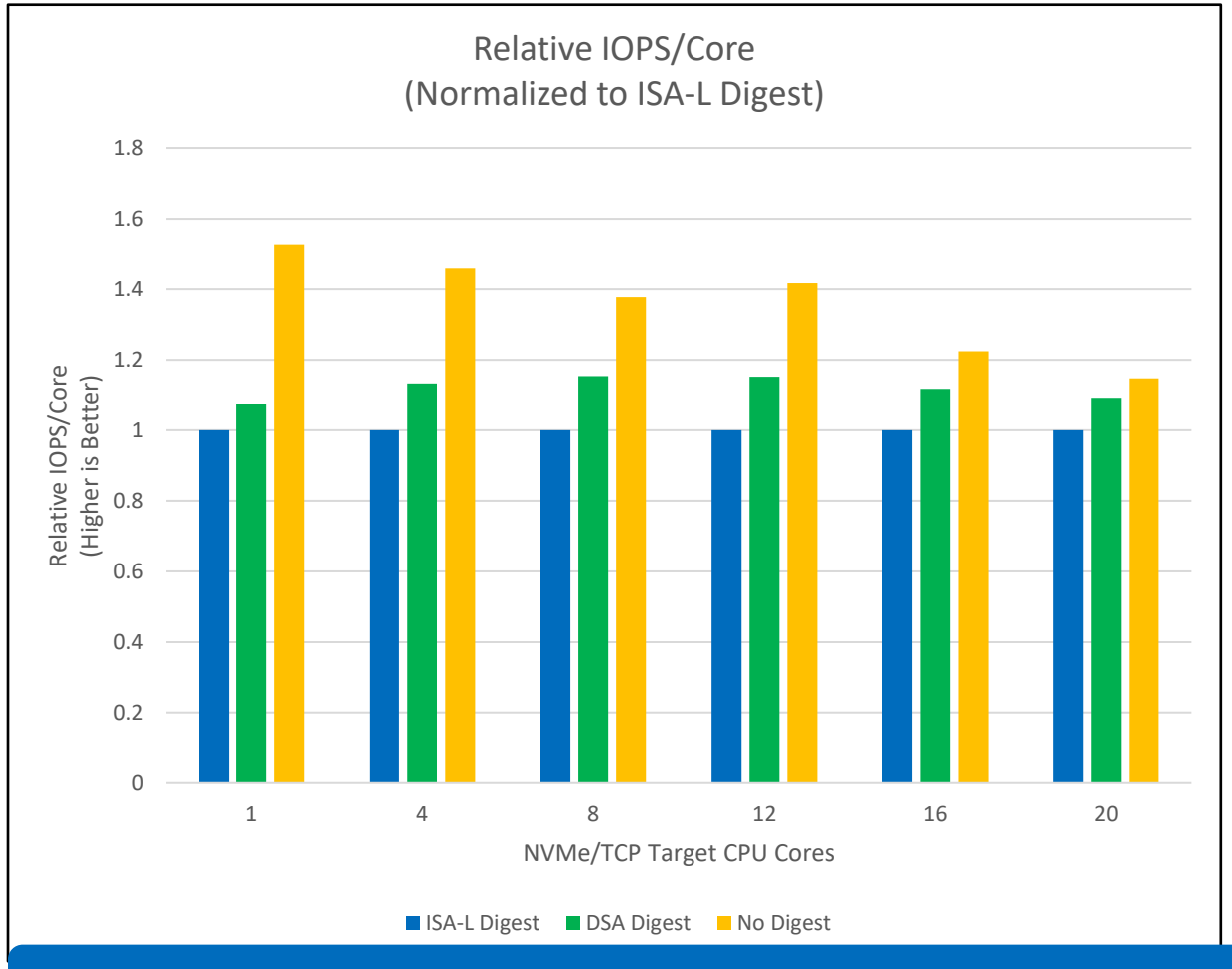


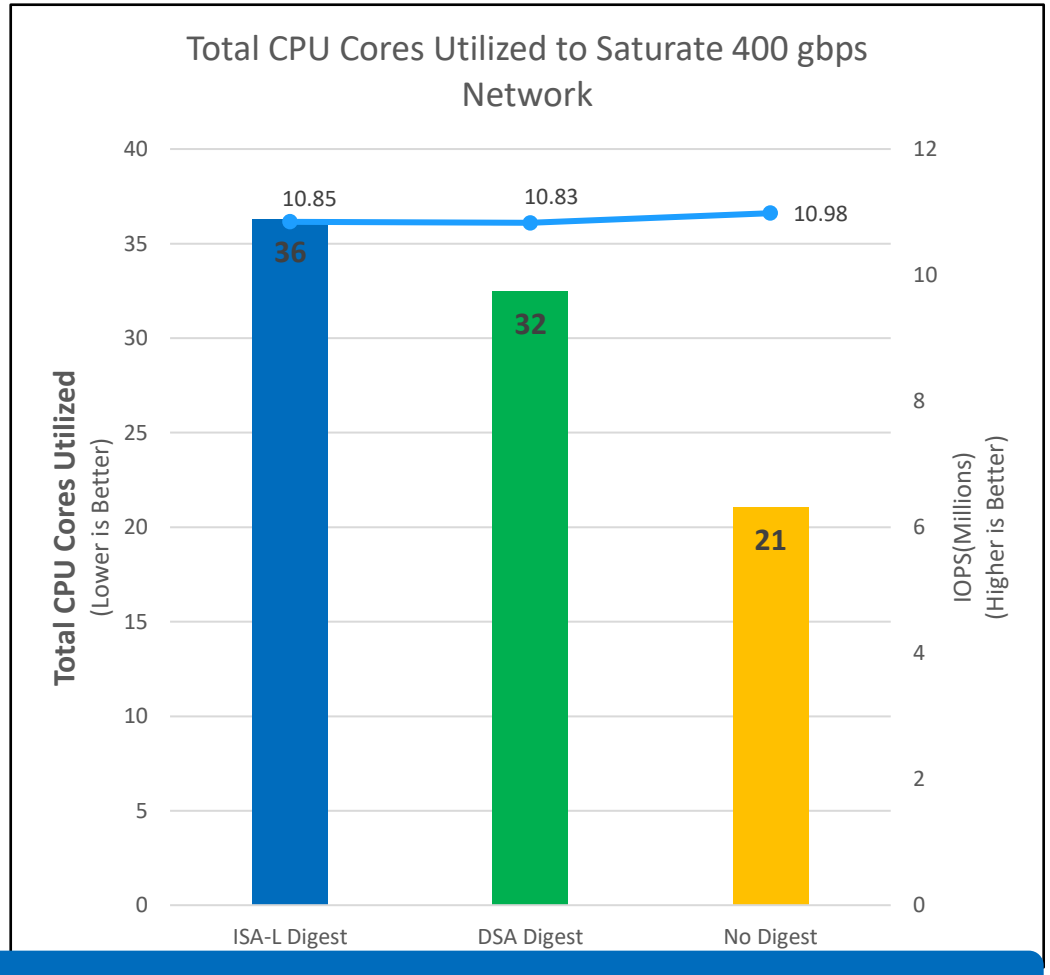
For 16KB Random Read, offloading data digest to Intel® DSA reduces:

- Average Lat by up to 38%
- P99 latency by up to 33%
- P99.9 latency by up to 19%

Results are based on Intel internal testing on pre-production cpu, systems and software. See backup for workloads and configurations. Results may vary.

4KB Random Reads

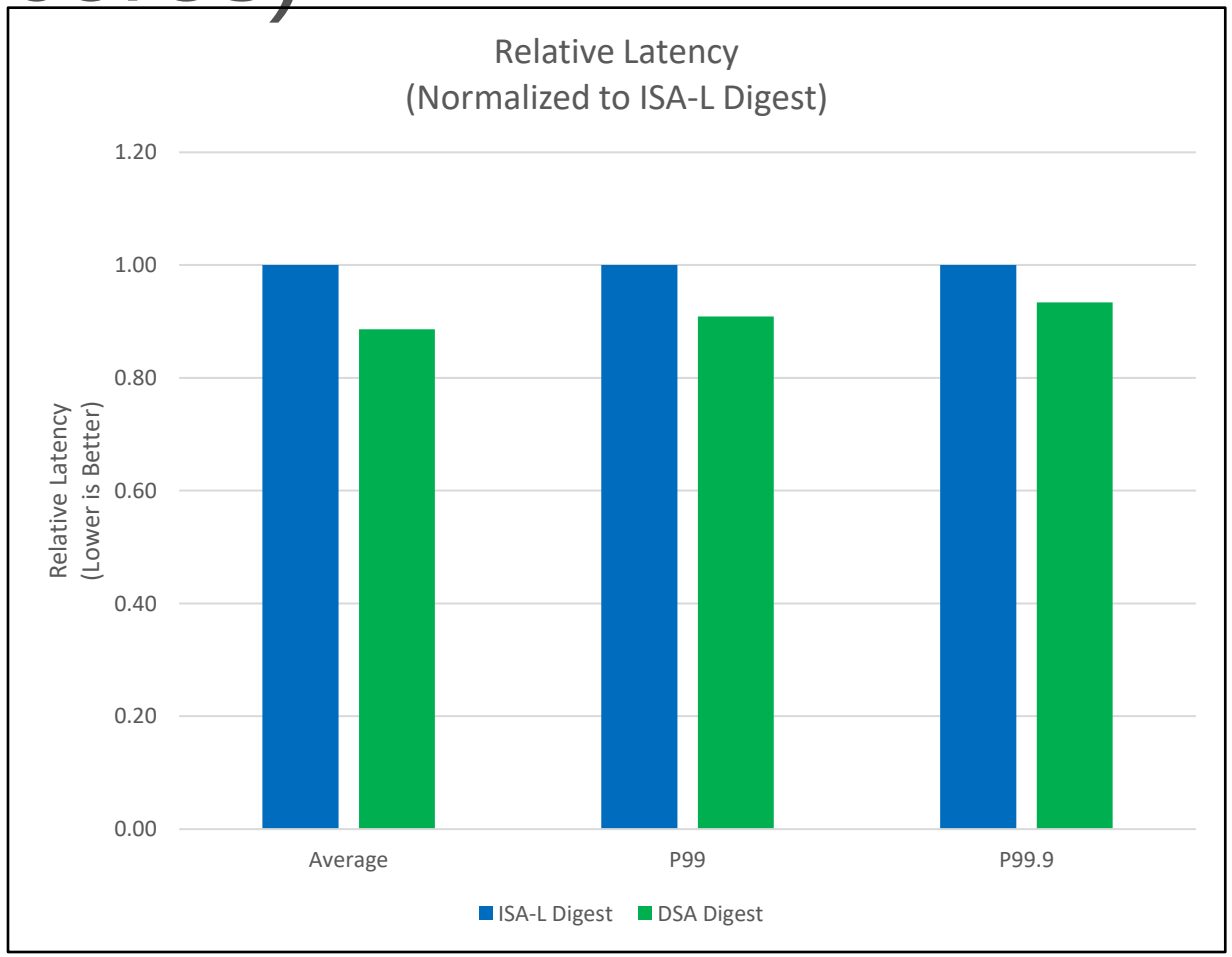




With Intel® DSA 10% reduction in CPU cores to saturate 400 gbps network

Results are based on Intel internal testing on pre-production cpu, systems and software. See backup for workloads and configurations. Results may vary.

4KB Random Reads Latency (NVMe/TCP target configured with 20 CPU cores)



For small block workloads, offloading data digest to Intel® DSA reduces:

- Average Lat by up to 11%
- P99 latency by up to 9%
- P99.9 latency by up to 7%

Results are based on Intel internal testing on pre-production cpu, systems and software. See backup for workloads and configurations. Results may vary.

Summary

- **Huge gains in IOPS/Core for large block workloads**.
  - Observed clear gains starting at 16K block sizes.
  - At 4K, gains limited to high QD. ISA-L SW performed better at low QD.
- As QD at target increases, more work can be submitted to Intel® DSA Engine concurrently which increases the value of using DSA. **Queue depth is key**!
- Comparing against hyper-optimized software implementation (ISA-L) that uses Vector CLMUL AVX512 instructions.

Call to Action

- SPDK community is building a framework for using HW accelerators like Intel® DSA, QAT, etc.
- Designed for SDS
- Accelerated functions: CRC32C, Copy, fill, compare, dual cast, copy+CRC, de/compress and more coming soon.
- Join the SPDK community at spdk.io/community

Please take a moment to rate this session.

Your feedback is important to us.

Back up

Section Subtitle

Configuration Details

Storage Target: 1-node, 2x Pre-Production Intel Xeon Scalable Processor(Sapphire Rapids > 40 cores) on Archer City with 1TB (16 slot/ 64GB/4800 [run @ 4800 MHz]) total DDR5 memory, ucode 0x89000080, HT On, Turbo ON, Ubuntu 22.04, Linux Kernel 5.15.0-41-generic, gcc 11.2.0 compiler, fio 3.30, SPDK 22.05, 2x Intel® DSA(1 per socket), DSA Driver in SPDK 22.05, ISA-L 2.30, Storage: 16x Kioxia® SSD KCM61VUL3T20 3.2TB, Network: 2x Dual port Intel® E810-C. Test by Intel as of 8/22/2022.

Host 1: 1-node, 2x Intel Xeon Platinum 8380 processor on CYP with 1TB (16 slots/ 64GB/ 3200 [run @ 3200 MHz]) total DDR4 memory, ucode 0xd000363, HT on, Turbo on, Ubuntu 22.04, Linux Kernel 5.15.0-41-generic, gcc 11.2.0 compiler, fio 3.30, SPDK 22.05, Network: 2x Dual port Intel® E810-C. Test by Intel as of 8/22/2022.

Host 2: 1-node, 2x Intel Xeon Platinum 8380 processor on CYP with 1TB (16 slots/ 64GB/ 3200 [run @ 3200 MHz]) total DDR4 memory, ucode 0xd000363, HT on, Turbo on, Ubuntu 22.04, 5.15.0.41, gcc 11.2.0 compiler, fio 3.30, SPDK 22.05, Network: 2x 100 GbE NVIDIA® Mellanox® ConnectX®-5 Ex

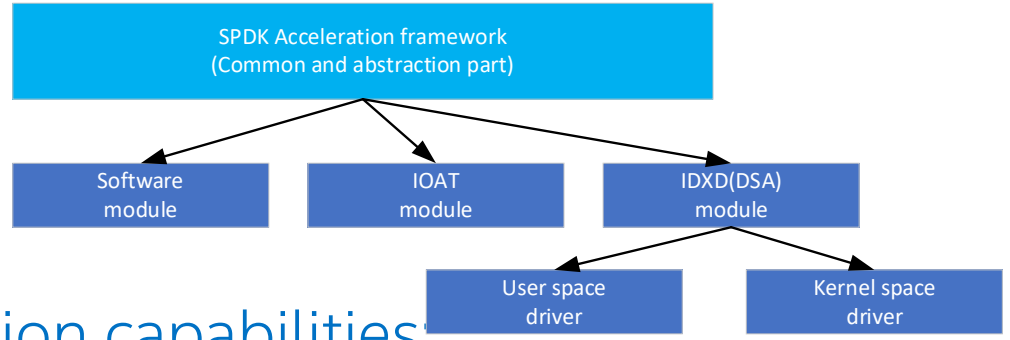
FIO QD at Target for IOPS Scalability test

128KB Seq Read	
Target CPU cores	QD at Target
1	1024
2	1024
3	1024

16KB Rand Read	
Target CPU cores	QD at Target
1	1024
2	1024
3	2048
4	2048
5	2048
6	4096

4KB Rand Read	
Target CPU cores	QD at Target
1	2048
4	2048
8	4096
12	6144
16	8192
20	8192

SPDK Acceleration Framework



- A framework for abstracting general acceleration capabilities.
  - With HW engines like Intel® DSA, IOAT, QAT, etc.
  - Designed for SW defined infra/storage: SW plug-in modules for environments without HW accelerators.
  - Asynchronous workflow: application uses CPU for other work while HW accelerator is moving/transforming data.
- Accelerated Functions: CRC32CC, copy, fill, compare, dual cast, copy_crc32c, de/compress
- SPDK provides libs/apps for organization building enterprise, SDS, object store solutions.
- NVMe-oF target: User-space storage target, presenting block devices over ethernet fabrics uses the accel FW to offload CRC32C digest calculation to DSA.
- Documentation

