Top-Down Topology-Aware I/O Performance Analysis with Intel® VTune™ Profiler

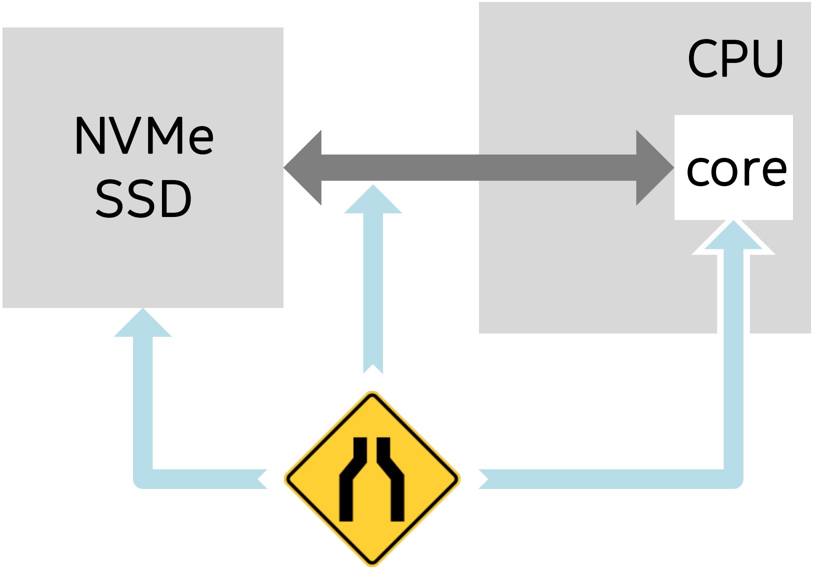
Ilia Kurakin | ilia.kurakin@intel.com

Software Engineer
Intel

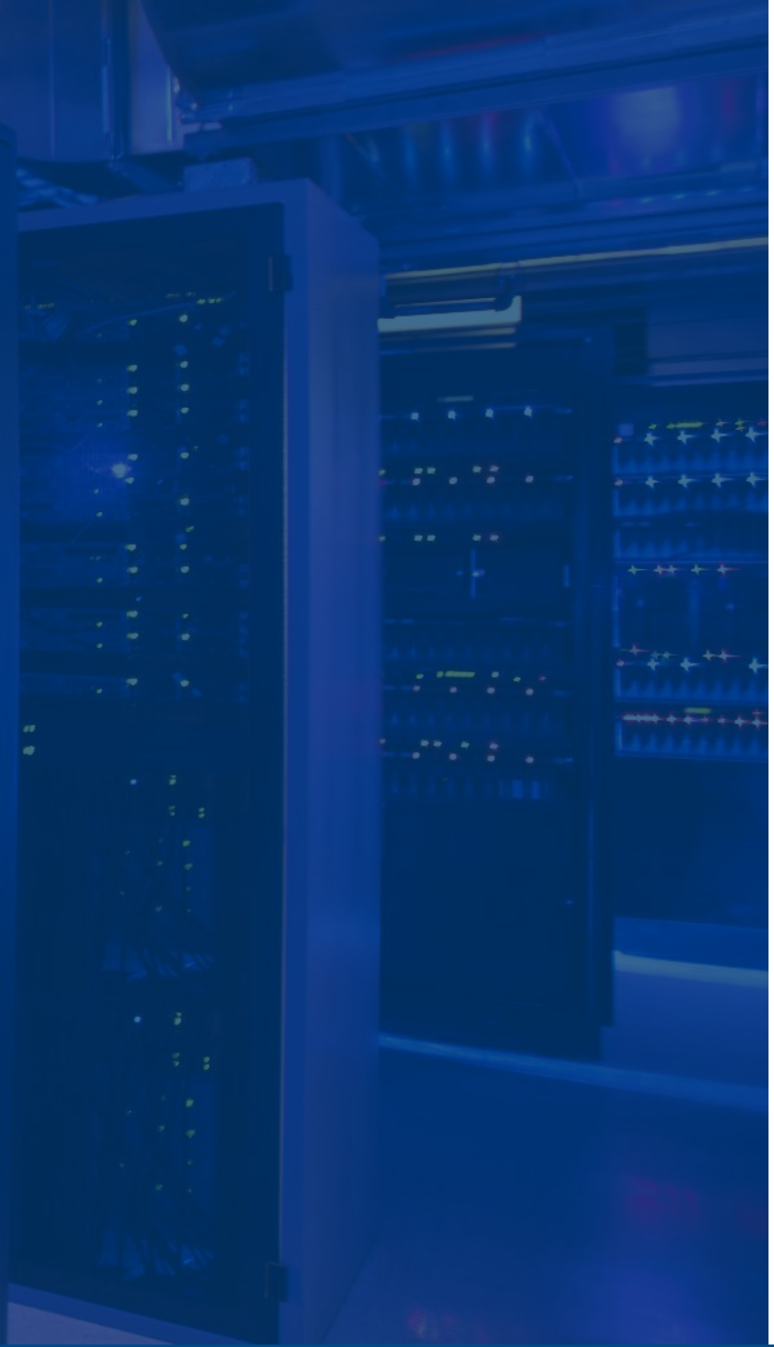
SPDK, PMDK, Intel® Performance Analyzers | **Virtual Forum**

IO-intensive Apps Performance Bottlenecks

Domain	Performance is limited by…	How to detect and address
I/O device bound	… device capabilities	Compare experiment with datasheet
Core bound	… algorithmic or microarchitectural code issues	Core-centric analyses (hotspots, uarch exploration, threading, Intel® Processor Trace -based, …)
Transfer bound	**… non-optimal interactions between devices and CPU**	**Developing “uncore”-centric analyses**



This talk focuses on the latter domain, which introduces the most challenging issues weakly covered with easy-to-follow methodologies



Agenda

1

Architectural Background

Learn more about how I/O traffic is processed on 3rd Generation Intel® Xeon® Scalable Processor

2

Intel® DDIO and MMIO traffic in I/O-intensive Apps

See what microarchitectural issues of Intel® Data Direct I/O and Memory-Mapped I/O traffic may take place in I/O-intensive application

3

Analyzing Platform I/O Performance

Topology-aware top-down platform I/O performance analysis with Input and Output analysis of Intel® VTune™ Profiler

4

SPDK Statistics Collection

Enhance I/O analysis result with SPDK-level application statistics to have a better view on HW resources utilization by SPDK app

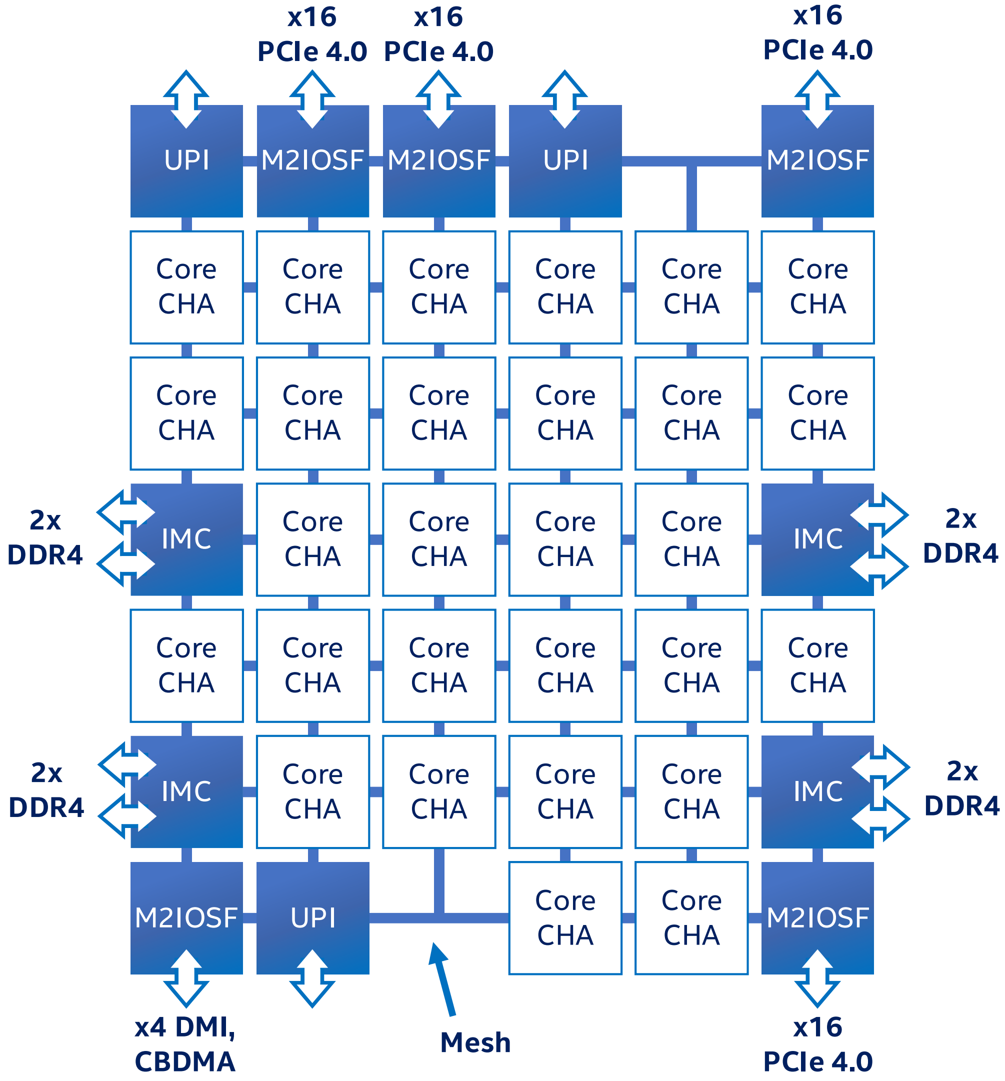
Architectural Background

3rd Generation Intel® Xeon® Scalable Processor

Processor is made by Mesh interconnect and units connected by it:

- **Cores**
  = execution units + L1 and L2 caches
- **Uncore units**
  - Slices of shared L3 cache (**LLC/SF**) with L3 cache controller (**CHA**)
  - Integrated memory controllers (**IMC**)
  - Intel® Ultra Path Interconnect (**UPI**) controllers
  - Integrated I/O controllers (**IIO or M2IOSF**)
    – interfaces to integrated (e.g. CBDMA) or external PCIe devices

3rd Gen Intel® Xeon® Processor Scalable Family, Codename Ice Lake, Uncore Performance Monitoring Reference Manual



		x16 PCIe 4.0	x16 PCIe 4.0			x16 PCIe 4.0	
	UPI	M2IOSF	M2IOSF	UPI		M2IOSF	
	Core CHA	Core CHA	Core CHA	Core CHA	Core CHA	Core CHA	
	Core CHA	Core CHA	Core CHA	Core CHA	Core CHA	Core CHA	
2x DDR4	IMC	Core CHA	Core CHA	Core CHA	Core CHA	IMC	2x DDR4
	Core CHA	Core CHA	Core CHA	Core CHA	Core CHA	Core CHA	
2x DDR4	IMC	Core CHA	Core CHA	Core CHA	Core CHA	IMC	2x DDR4
	M2IOSF	UPI		Core CHA	Core CHA	M2IOSF	
	x4 DMI, CBDMA		Mesh			x16 PCIe 4.0	

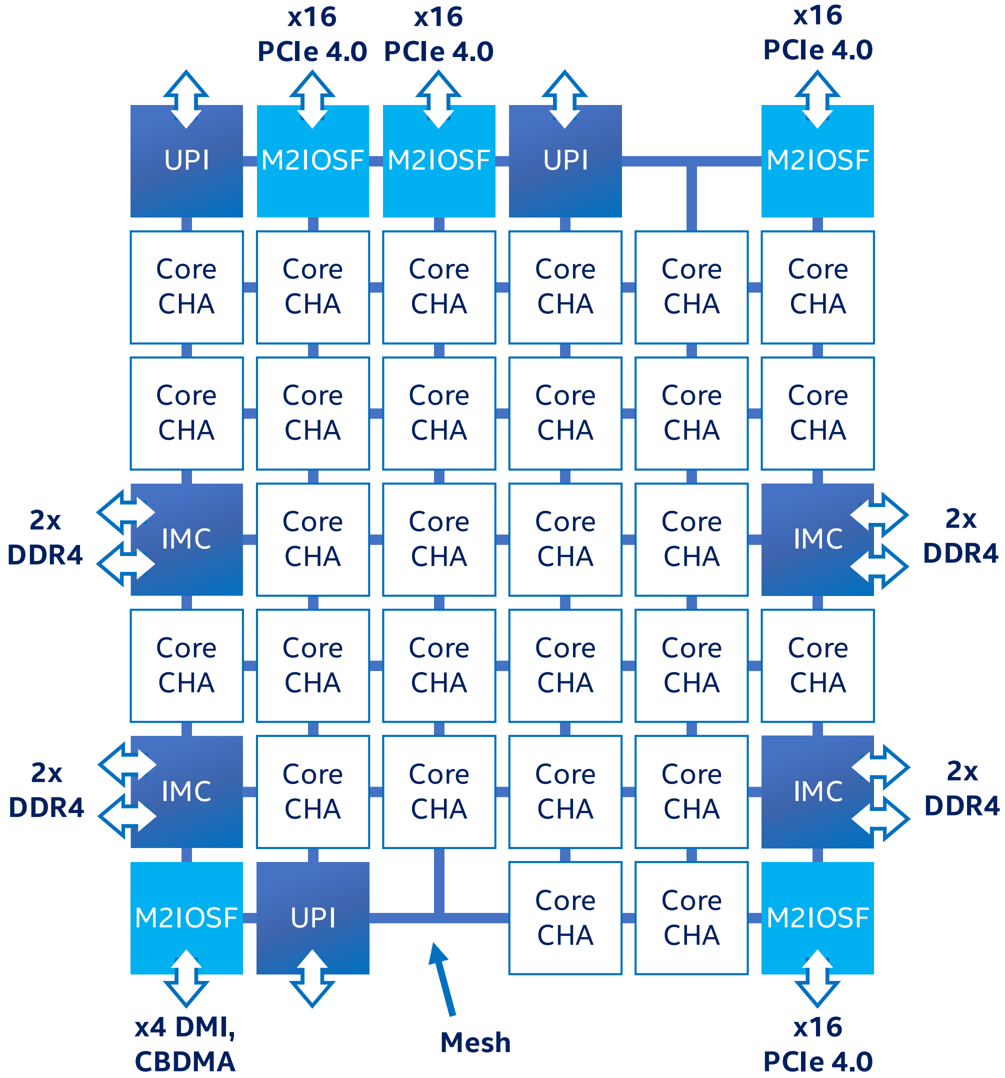
Integrated I/O Controllers

3rd Generation Intel® Xeon® Scalable Processor incorporates up to 5 integrated I/O controllers (M2IOSF units) per socket:

- 4 servicing x16 PCIe gen 4
- 1 servicing CBDMA and DMI

M2IOSF connects ordered **PCIe domain** to the out-of-order **mesh**:

- M2IOSF translates TLPs to cache line (64B) requests and vice versa



Core/Device Communication Compound

Inbound transactions
initiated by I/O device, target system memory

- Inbound read = I/O device reads the system memory
- Inbound write = I/O device writes the system memory

driven by **Intel® Data Direct I/O** hardware technology

Outbound transactions
initiated by cores, target I/O device memory

- Outbound read = core reads the memory of I/O device
- Outbound write = core writes the memory of I/O device

typically done by **Memory-Mapped I/O** address space accesses

Intel® Data Direct I/O (Intel® DDIO) Details [1/2]

The inbound transactions are routed directly to the local L3 cache:

- **Inbound reads** are processed without L3 cache allocation
- **Inbound writes** require a related cache line to be allocated in the L3 and get processed in two phases:

Inbound Write Phase	Details
1. Get cache line ownership for IIO	Cache line location is tracked through L3 line, therefore L3 allocation is required.
2. IIO delivers modified data to the L3, releases ownership	This phase is done in different ways depending on chosen config: a. Allocating – data goes to the LLC b. Non-allocating – data goes to the DRAM

Inbound requests for data lead to L3 cache lookup resulting in L3 hit or miss scenarios.

Intel® Data Direct I/O (Intel® DDIO) Details [2/2]

Following rules apply when platform processes inbound PCIe read and write:

Request	L3 Lookup	Implication
Inbound Read	Hit (good)	The data is read from L3 and sent to the PCIe device
	Miss (bad)	The data is read from the local DRAM or from the remote socket's memory subsystem and sent to the PCIe device
Inbound Write	Hit (good)	The cache line is overwritten with the new data
	Miss (bad)	Some cache line is first evicted. Then, in place of the evicted line, a new cache line is allocated. If the targeted cache line is used remotely, cross-socket accesses are required. Finally, the cache line is updated with the new data.

"DDIO misses" should be avoided for best latency/throughput and not wasting DRAM/UPI traffic and platform power

Memory-Mapped I/O (MMIO) Accesses

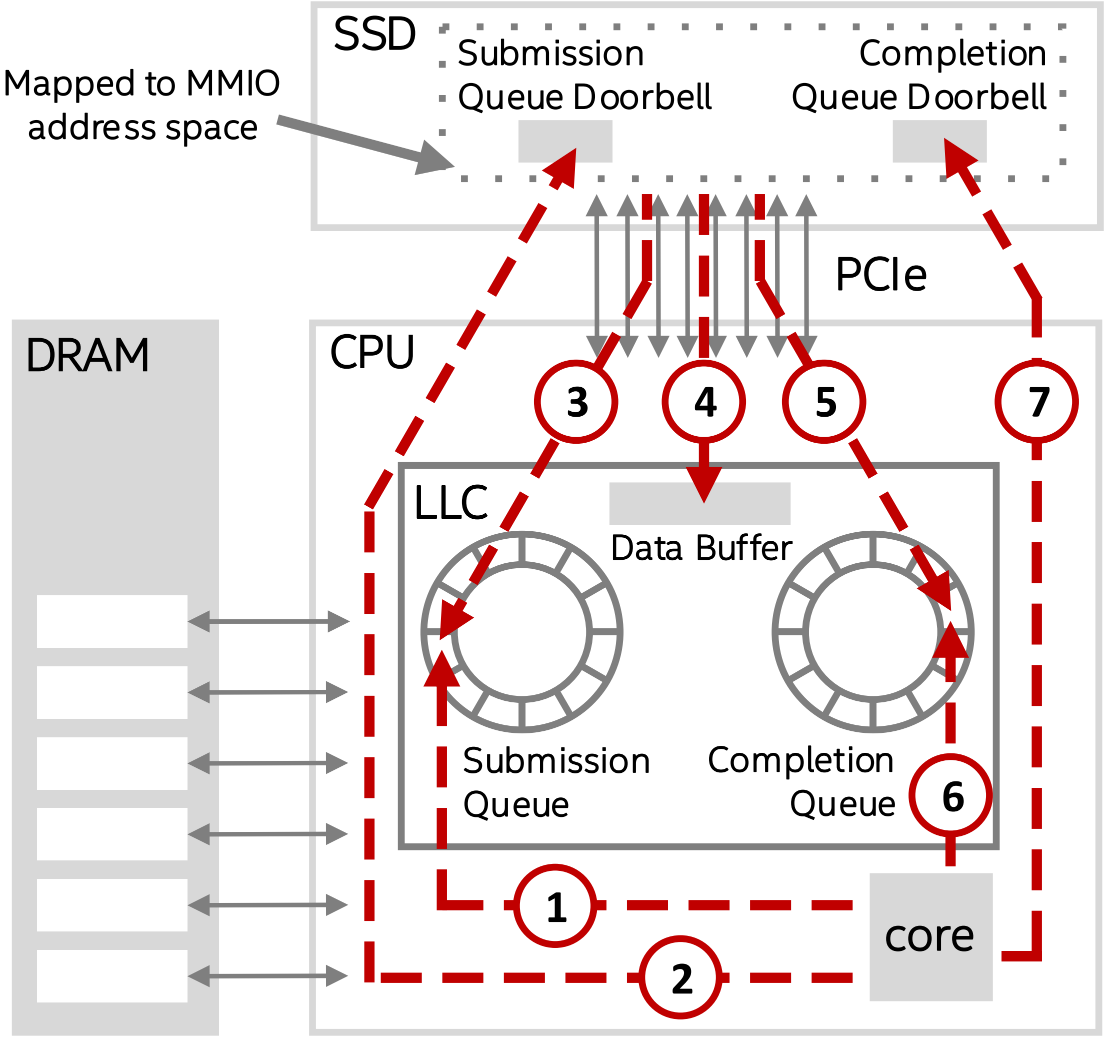
MMIO access is a primary mechanism for accessing device memory.

MMIO accesses are quite expensive and should be limited:

Core Operation →	IIO Transaction	Cost
MMIO Read	Outbound PCIe Read	Most expensive I/O-related transaction from core perspective, since completion requires round trip to device.
MMIO Write	Outbound PCIe Write	Less costly transaction, but core still needs to get an acknowledge (unless done not through MOVDIR)

Avoid MMIO reads and use tricks to minimize MMIO writes on the data path.

DDIO/MMIO Requests in Storage Apps



Example: app reads from SSD

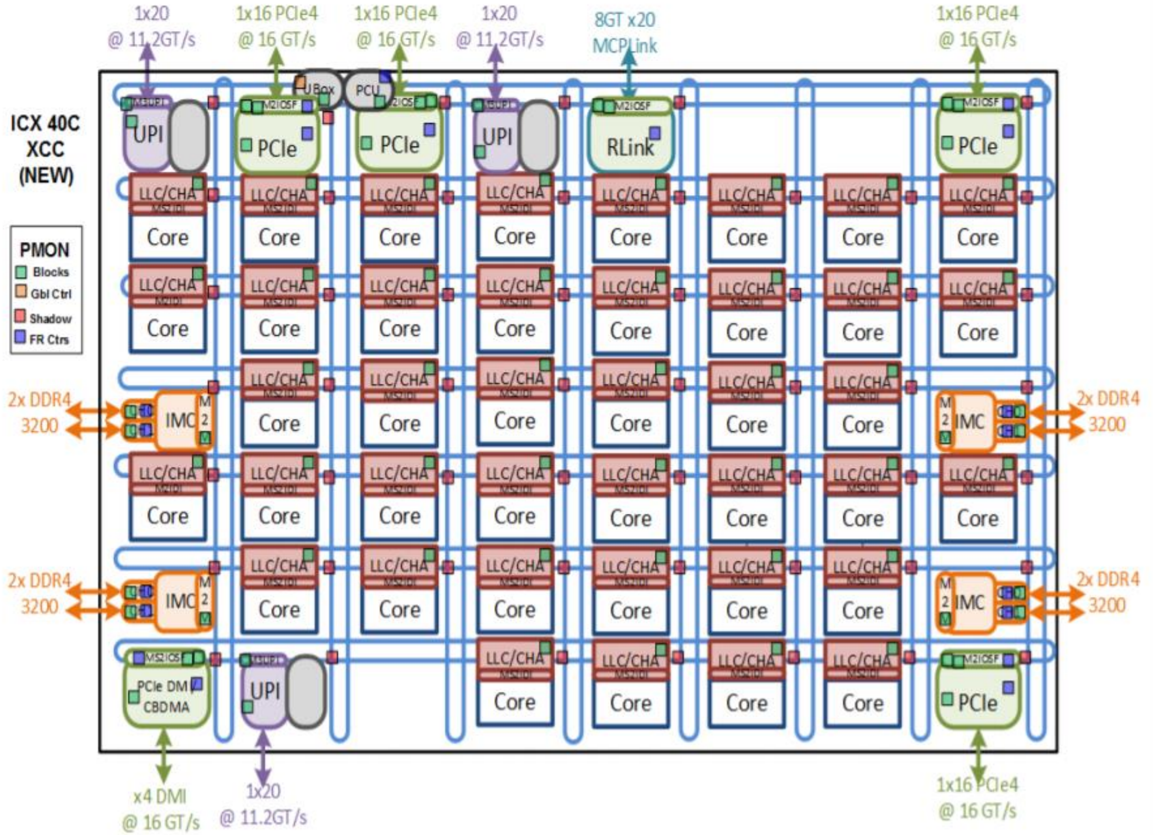
1. Core writes I/O command descriptor and starts polling completion queue element
2. Core notifies SSD that new descriptor is available (**Outbound PCIe Write**)
3. Device reads descriptor to get buffer address (**Inbound PCIe Read**)
4. Device writes I/O data (**Inbound PCIe Write**)
5. Device writes to the completion queue (**Inbound PCIe Write**)
6. Core detects that completion is updated
7. Core moves completion queue tail pointer (**Outbound PCIe Write**)

Performance Analysis

Platform Observability

Thousands of uncore performance monitoring events incorporated in uncore Performance Monitoring Units (PMUs)

- **IIO**: inbound/outbound read/write bandwidth
- **IRP**: coherency-related IIO operations
- **CHA**: mesh and L3 cache controller
- **IMC and M2M**: memory bandwidth, memory directory access
- **UPI**: cross-socket traffic

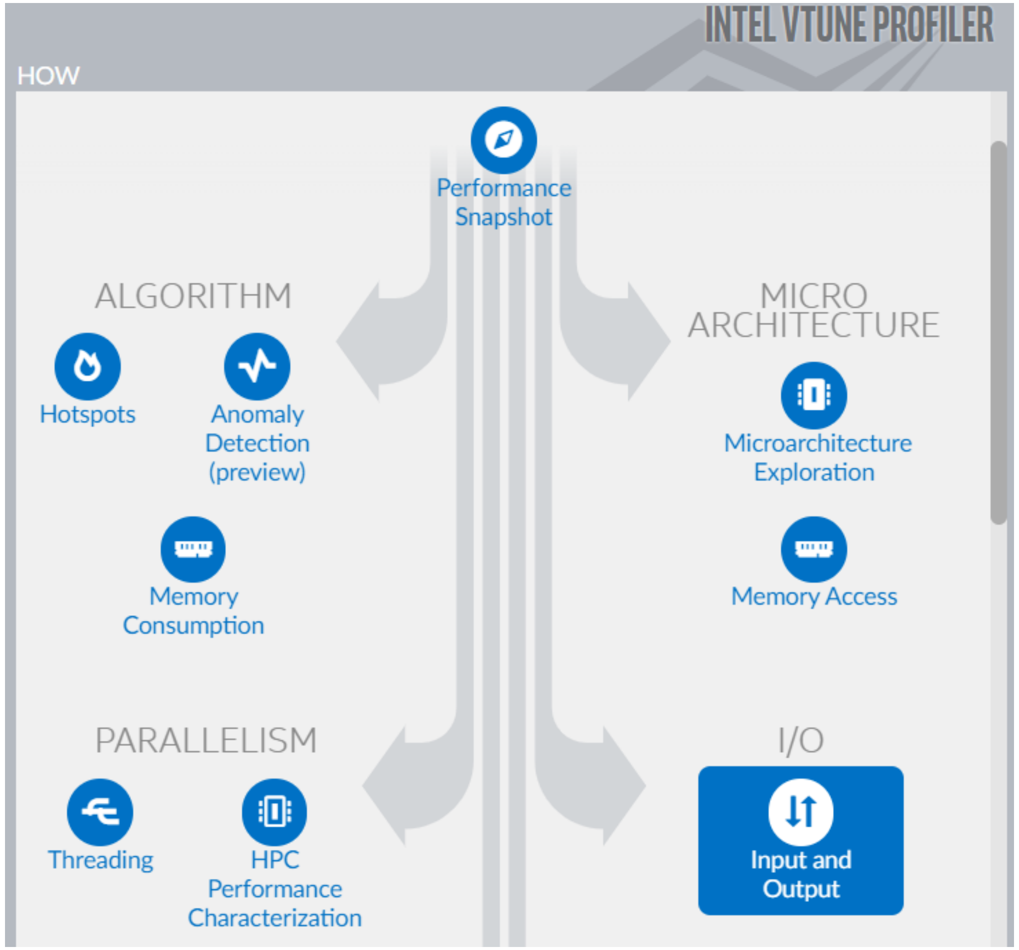


[3rd Gen Intel® Xeon® Processor Scalable Family, Codename Ice Lake, Uncore Performance Monitoring Reference Manual]

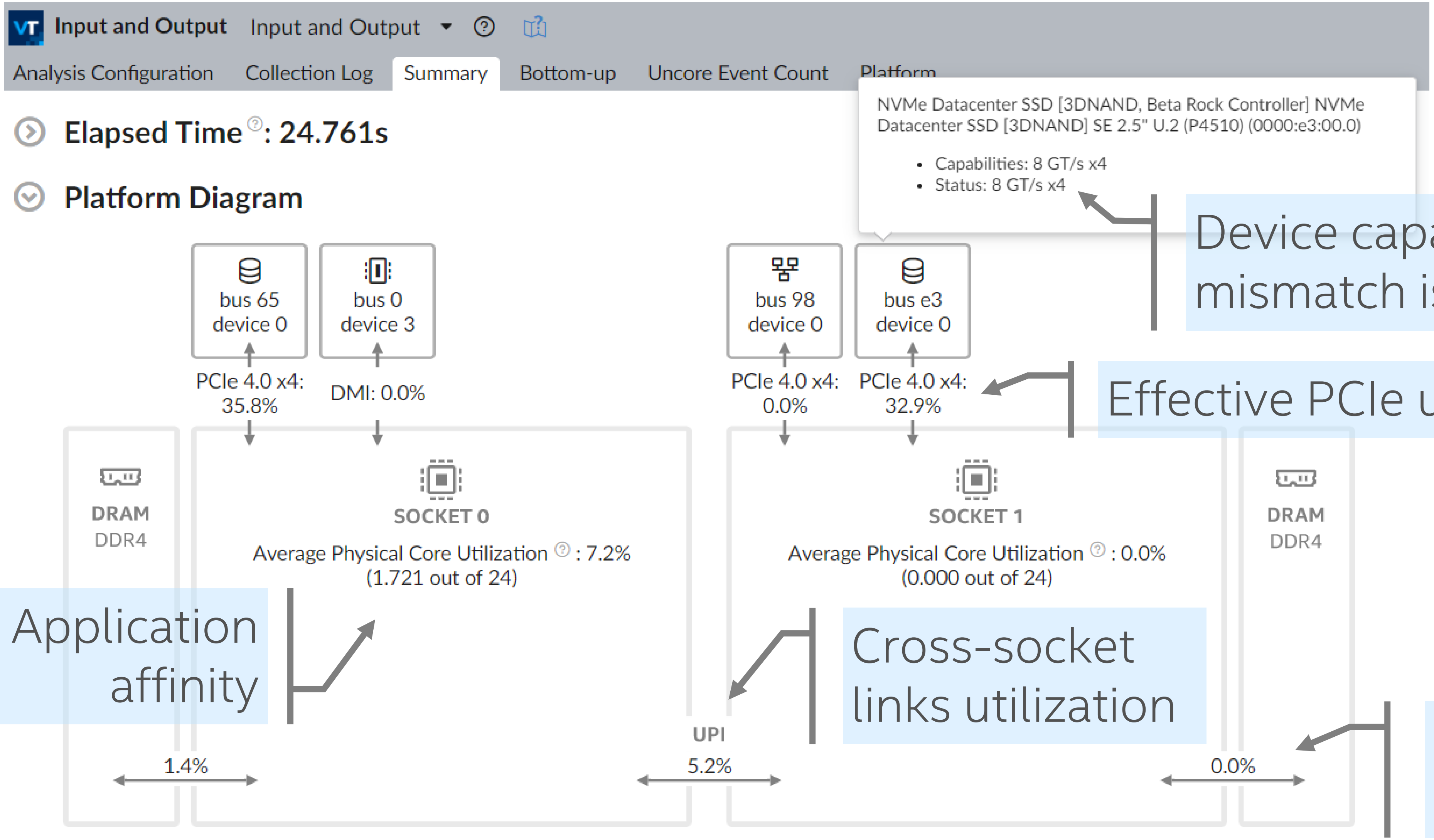
[Intel® VTune™ Profiler] builds performance metrics upon uncore performance monitoring events and gives topology-aware top-down view

VTune Profiler Input and Output Analysis

- Provides uncore- and device-centric view to locate performance bottlenecks in I/O-intensive applications at both HW and SW levels
- Two types of metrics:
  - **Platform**: application-agnostic hardware event-based metrics to analyze DRAM, UPI, PCIe, Intel DDIO, MMIO traffic consumption
  - **Software**: DPDK, SPDK, kernel I/O
- The full set of Input and Output analysis metrics is available on Intel® Xeon® processors
- Linux and FreeBSD are supported



Input and Output Analysis: Platform Diagram



Elapsed Time: 24.761s

Platform Diagram

NVMe Datacenter SSD [3DNAND, Beta Rock Controller] NVMe Datacenter SSD [3DNAND] SE 2.5" U.2 (P4510) (0000:e3:00.0)

- Capabilities: 8 GT/s x4
- Status: 8 GT/s x4

SOCKET 0: Average Physical Core Utilization: 7.2% (1.721 out of 24)

SOCKET 1: Average Physical Core Utilization: 0.0% (0.000 out of 24)

Example 1: SPDK bdevperf

- 2 cores from socket 0
- 2 NVMe SSDs

Device capabilities and status, mismatch is reported as an issue

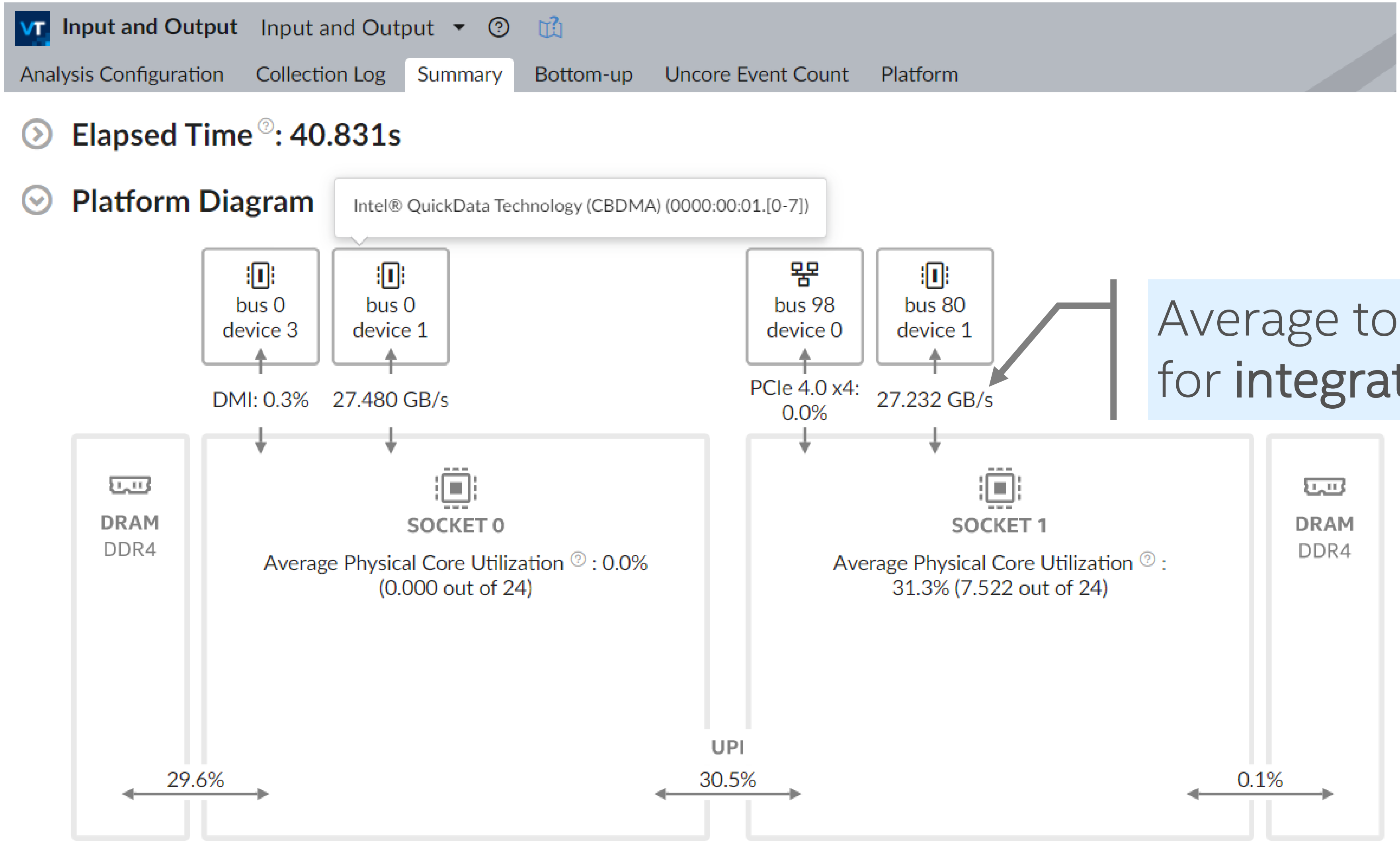
Effective PCIe utilization

Application affinity

Cross-socket links utilization

Memory locality wrt app and devices

Input and Output Analysis: Platform Diagram



Elapsed Time: 40.831s

Platform Diagram

Intel® QuickData Technology (CBDMA) (0000:00:01.[0-7])

Average Physical Core Utilization : 0.0% (0.000 out of 24)

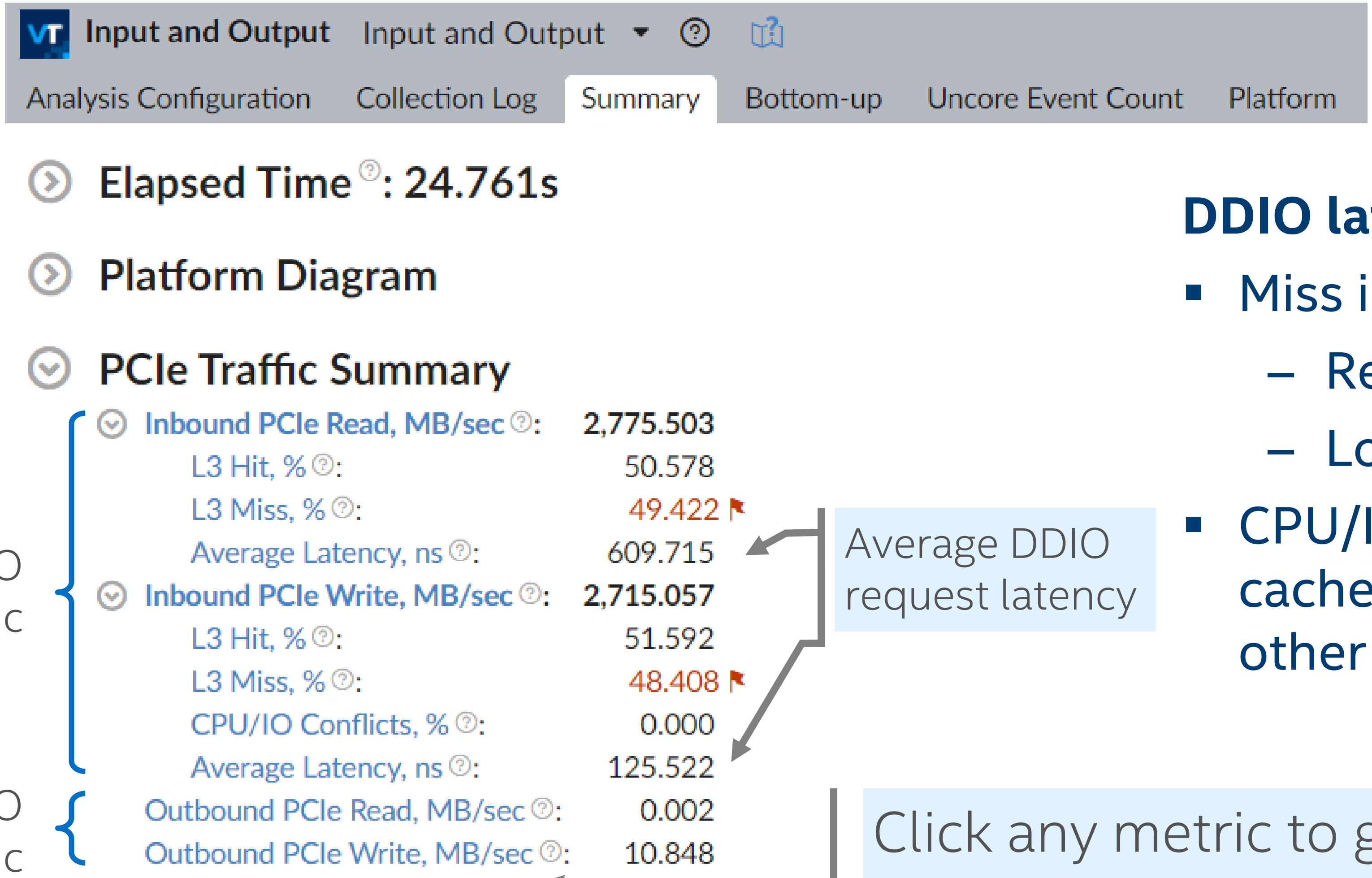
Average Physical Core Utilization : 31.3% (7.522 out of 24)

Example 2: SPDK ioat_perf

- 8 cores from socket 1
- 16 CBDMA channels

Average total I/O bandwidth for **integrated devices**

Input and Output Analysis: Detailed I/O Metrics



Elapsed Time: 24.761s

Platform Diagram

PCIe Traffic Summary

Inbound PCIe Read, MB/sec:	2,775.503
L3 Hit, %:	50.578
L3 Miss, %:	49.422
Average Latency, ns:	609.715
Inbound PCIe Write, MB/sec:	2,715.057
L3 Hit, %:	51.592
L3 Miss, %:	48.408
CPU/IO Conflicts, %:	0.000
Average Latency, ns:	125.522
Outbound PCIe Read, MB/sec:	0.002
Outbound PCIe Write, MB/sec:	10.848

DDIO traffic

MMIO traffic

Average DDIO request latency

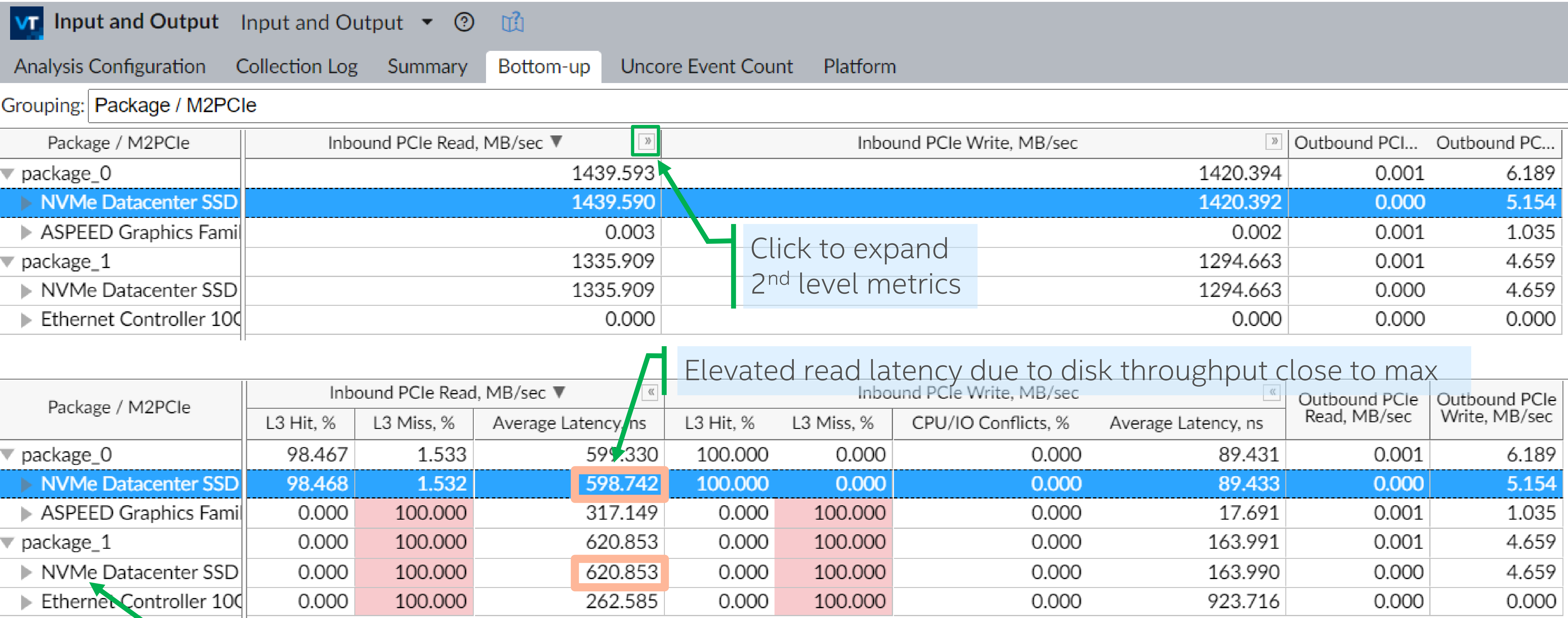
Click any metric to get navigated to the per-device view

Result is captured for SPDK bdevperf app (2 cores, 2 SSDs)

DDIO latency might be elevated due to:

- Miss in the L3 cache resolved by
  - Remote memory/cache accesses
  - Local memory accesses
- CPU/IO conflicts – contentions for cache lines between IIO and some other agent (core or another IIO)

Input and Output Analysis: Per-Device I/O Metrics



Grouping: Package / M2PCIe

Package / M2PCIe	Inbound PCIe Read, MB/sec ▼	Inbound PCIe Write, MB/sec	Outbound PCI...	Outbound PC...
package_0	1439.593	1420.394	0.001	6.189
NVMe Datacenter SSD	1439.590	1420.392	0.000	5.154
ASPEED Graphics Famil	0.003	0.002	0.001	1.035
package_1	1335.909	1294.663	0.001	4.659
NVMe Datacenter SSD	1335.909	1294.663	0.000	4.659
Ethernet Controller 10(	0.000	0.000	0.000	0.000

Click to expand 2nd level metrics

Elevated read latency due to disk throughput close to max

Package / M2PCIe	Inbound PCIe Read, MB/sec ▼			Inbound PCIe Write, MB/sec				Outbound PCIe Read, MB/sec	Outbound PCIe Write, MB/sec
	L3 Hit, %	L3 Miss, %	Average Latency, ns	L3 Hit, %	L3 Miss, %	CPU/IO Conflicts, %	Average Latency, ns		
package_0	98.467	1.533	599.330	100.000	0.000	0.000	89.431	0.001	6.189
NVMe Datacenter SSD	98.468	1.532	598.742	100.000	0.000	0.000	89.433	0.000	5.154
ASPEED Graphics Famil	0.000	100.000	317.149	0.000	100.000	0.000	17.691	0.001	1.035
package_1	0.000	100.000	620.853	0.000	100.000	0.000	163.991	0.001	4.659
NVMe Datacenter SSD	0.000	100.000	620.853	0.000	100.000	0.000	163.990	0.000	4.659
Ethernet Controller 10(	0.000	100.000	262.585	0.000	100.000	0.000	923.716	0.000	0.000

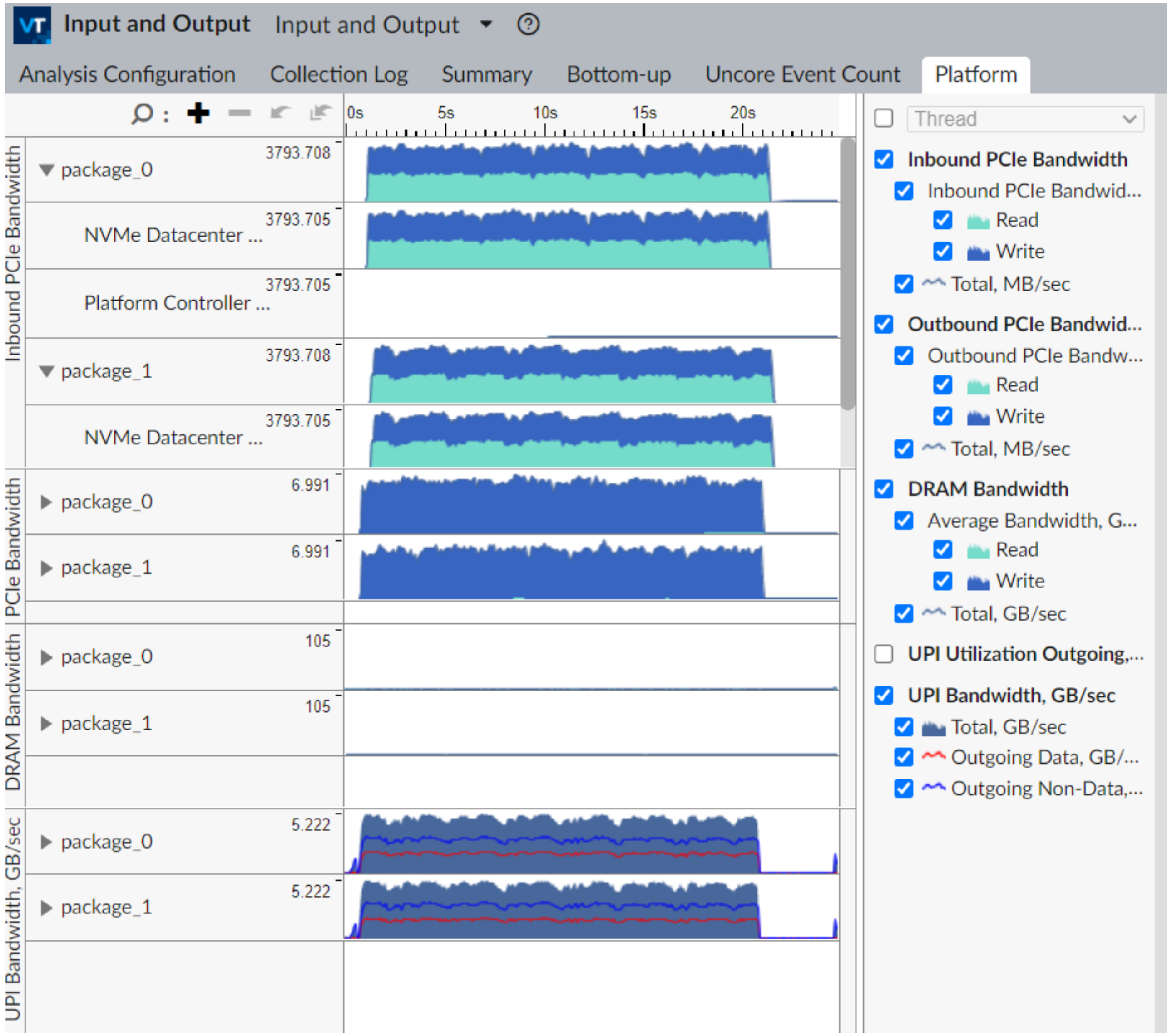
In this example, SSD on the socket 1 has 100% DDIO requests missing L3 and higher latency

Input and Output Analysis: Overtime Bandwidth

Per-device Inbound (DDIO) and Outbound (MMIO) PCIe bandwidth

Per-channel DRAM bandwidth

Per-link UPI Tx traffic with data/non-data itemization



□ Thread

☑ Inbound PCIe Bandwidth
- ☑ Inbound PCIe Bandwid...
  - ☑ Read
  - ☑ Write
- ☑ Total, MB/sec

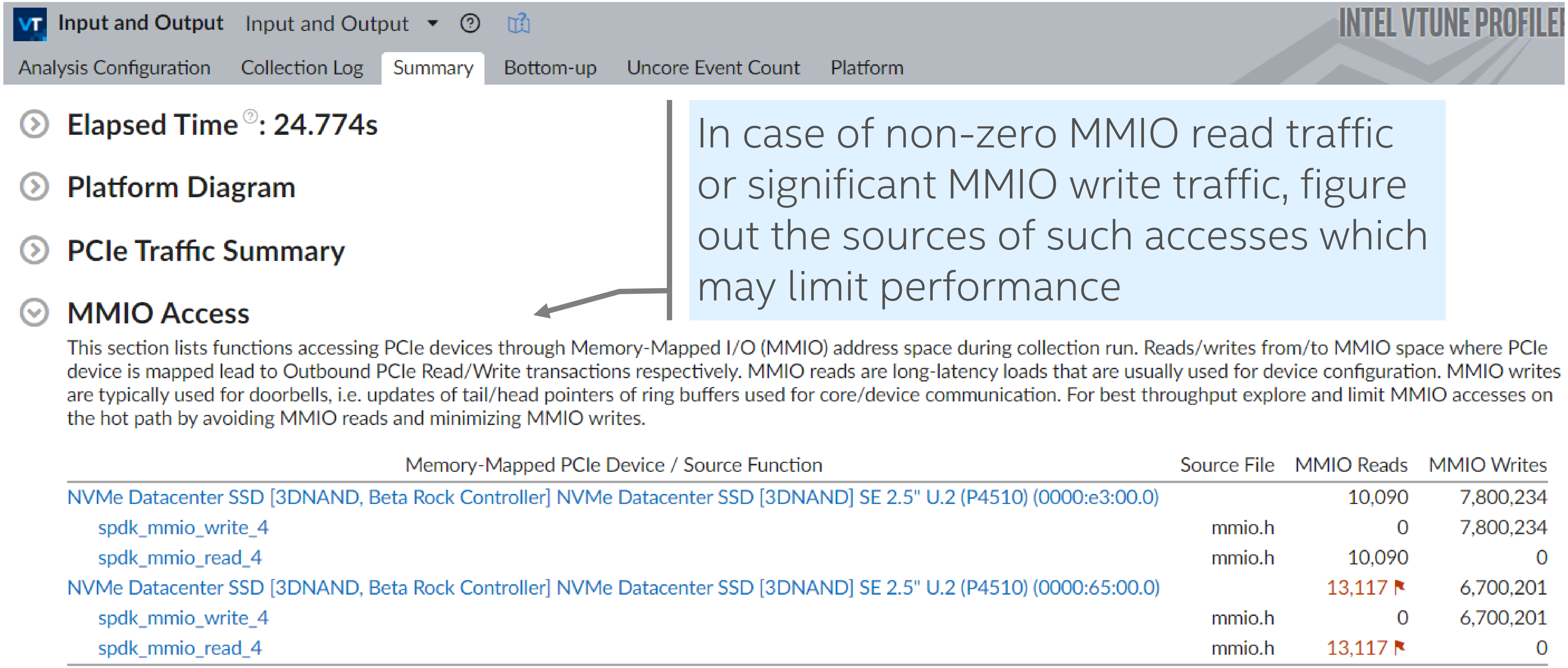
☑ Outbound PCIe Bandwid...
- ☑ Outbound PCIe Bandw...
  - ☑ Read
  - ☑ Write
- ☑ Total, MB/sec

☑ DRAM Bandwidth
- ☑ Average Bandwidth, G...
  - ☑ Read
  - ☑ Write
- ☑ Total, GB/sec

□ UPI Utilization Outgoing,...

☑ UPI Bandwidth, GB/sec
- ☑ Total, GB/sec
- ☑ Outgoing Data, GB/...
- ☑ Outgoing Non-Data,...

Input and Output Analysis: Locating MMIO Accesses



In case of non-zero MMIO read traffic or significant MMIO write traffic, figure out the sources of such accesses which may limit performance

Elapsed Time: 24.774s

Platform Diagram

PCIe Traffic Summary

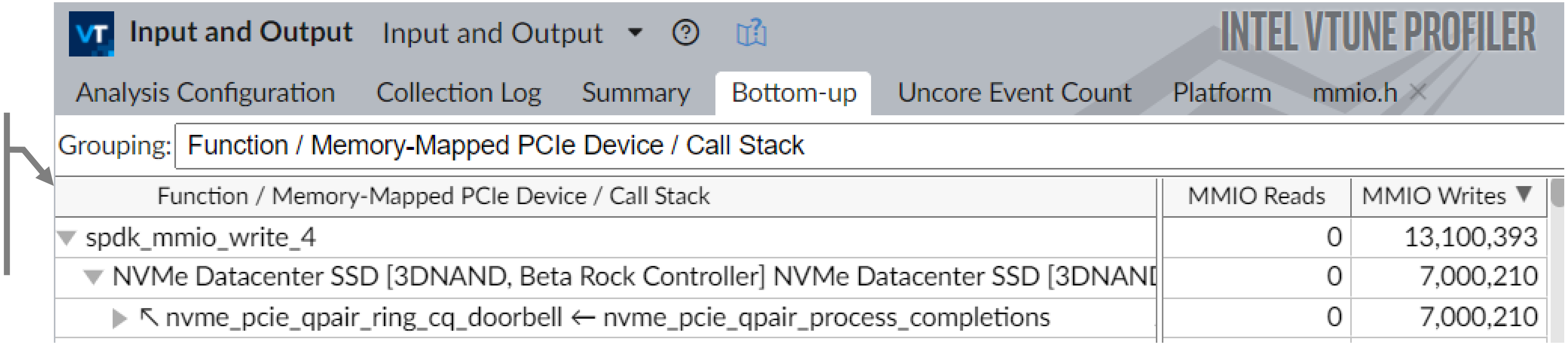
MMIO Access

This section lists functions accessing PCIe devices through Memory-Mapped I/O (MMIO) address space during collection run. Reads/writes from/to MMIO space where PCIe device is mapped lead to Outbound PCIe Read/Write transactions respectively. MMIO reads are long-latency loads that are usually used for device configuration. MMIO writes are typically used for doorbells, i.e. updates of tail/head pointers of ring buffers used for core/device communication. For best throughput explore and limit MMIO accesses on the hot path by avoiding MMIO reads and minimizing MMIO writes.

Memory-Mapped PCIe Device / Source Function	Source File	MMIO Reads	MMIO Writes
NVMe Datacenter SSD [3DNAND, Beta Rock Controller] NVMe Datacenter SSD [3DNAND] SE 2.5" U.2 (P4510) (0000:e3:00.0)		10,090	7,800,234
spdk_mmio_write_4	mmio.h	0	7,800,234
spdk_mmio_read_4	mmio.h	10,090	0
NVMe Datacenter SSD [3DNAND, Beta Rock Controller] NVMe Datacenter SSD [3DNAND] SE 2.5" U.2 (P4510) (0000:65:00.0)		13,117 ⚑	6,700,201
spdk_mmio_write_4	mmio.h	0	6,700,201
spdk_mmio_read_4	mmio.h	13,117 ⚑	0

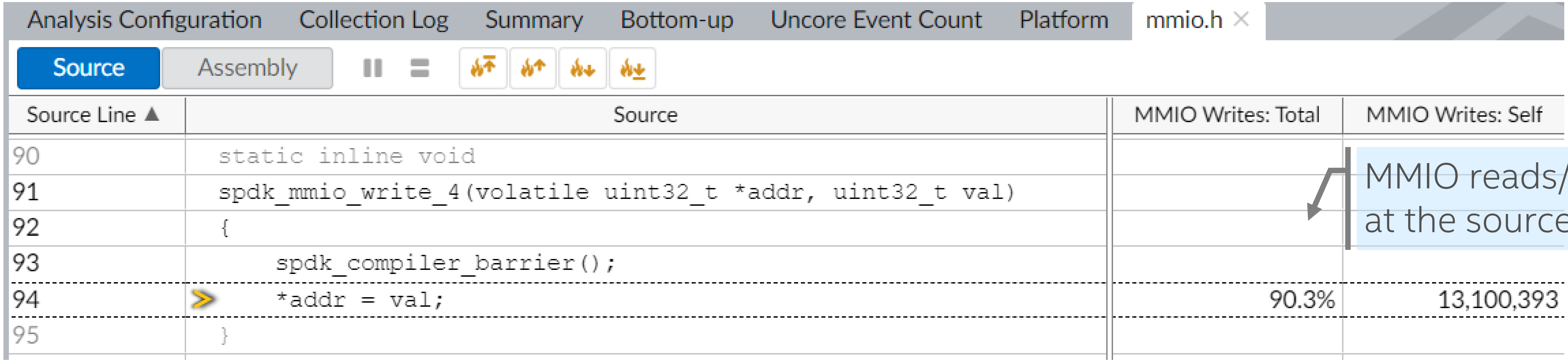
Input and Output Analysis: Locating MMIO Accesses

Call stacks leading to MMIO reads/writes



Grouping: Function / Memory-Mapped PCIe Device / Call Stack

Function / Memory-Mapped PCIe Device / Call Stack	MMIO Reads	MMIO Writes ▼
▼ spdk_mmio_write_4	0	13,100,393
▼ NVMe Datacenter SSD [3DNAND, Beta Rock Controller] NVMe Datacenter SSD [3DNANI	0	7,000,210
▶ ↖ nvme_pcie_qpair_ring_cq_doorbell ← nvme_pcie_qpair_process_completions	0	7,000,210

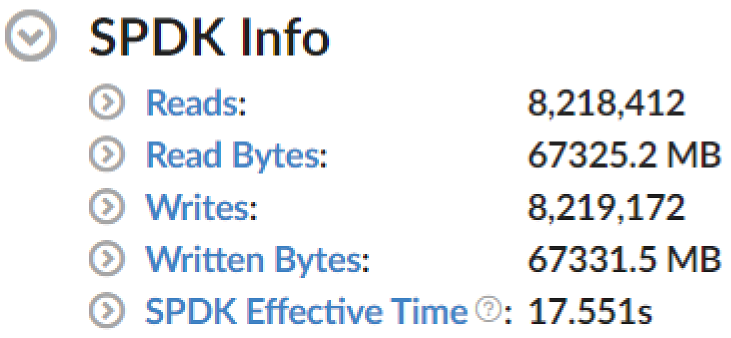


Source Line ▲	Source	MMIO Writes: Total	MMIO Writes: Self
90	`static inline void`		
91	`spdk_mmio_write_4(volatile uint32_t *addr, uint32_t val)`		
92	`{`		
93	`    spdk_compiler_barrier();`		
94	`    *addr = val;`	90.3%	13,100,393
95	`}`		

MMIO reads/writes at the source level

Input and Output Analysis: SPDK Metrics

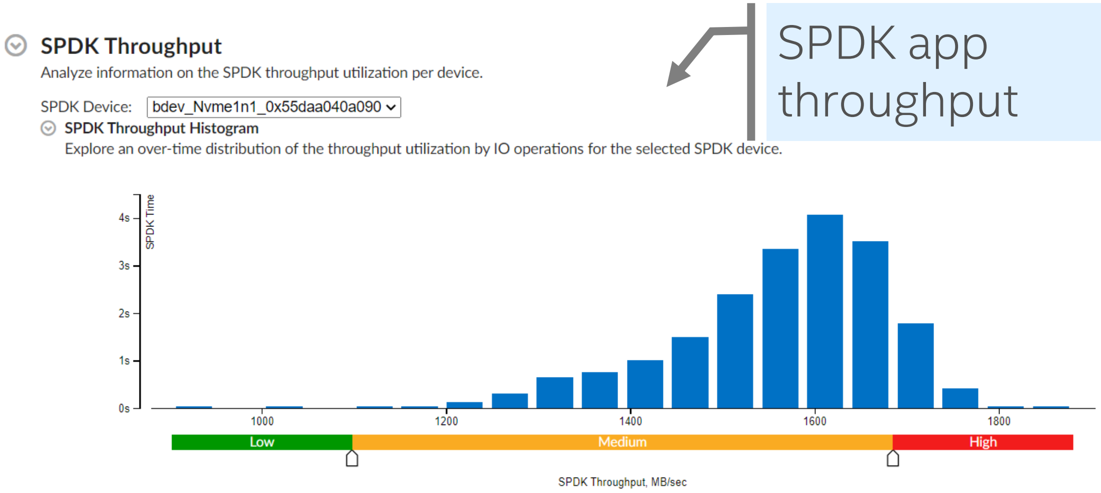
- VTune statistics collection is integrated into SPDK
- Enable it by adding `--with-vtune=<VTUNE_DIR>` configuration option



SPDK Info

Reads:	8,218,412
Read Bytes:	67325.2 MB
Writes:	8,219,172
Written Bytes:	67331.5 MB
SPDK Effective Time:	17.551s

Operations with the break down by block device / thread, overtime charts are available



SPDK Throughput

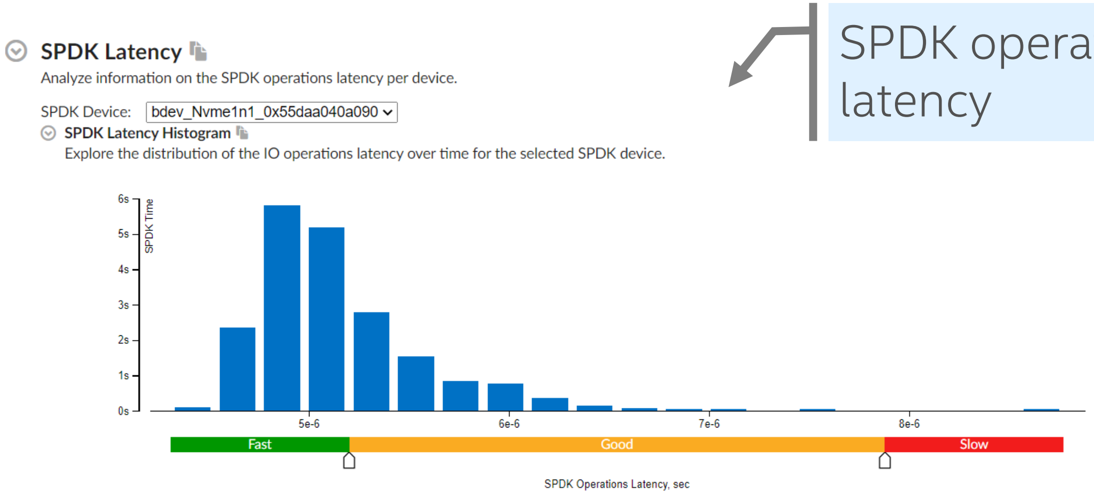
Analyze information on the SPDK throughput utilization per device.

SPDK Device: bdev_Nvme1n1_0x55daa040a090

SPDK Throughput Histogram

Explore an over-time distribution of the throughput utilization by IO operations for the selected SPDK device.

SPDK app throughput



SPDK Latency

Analyze information on the SPDK operations latency per device.

SPDK Device: bdev_Nvme1n1_0x55daa040a090

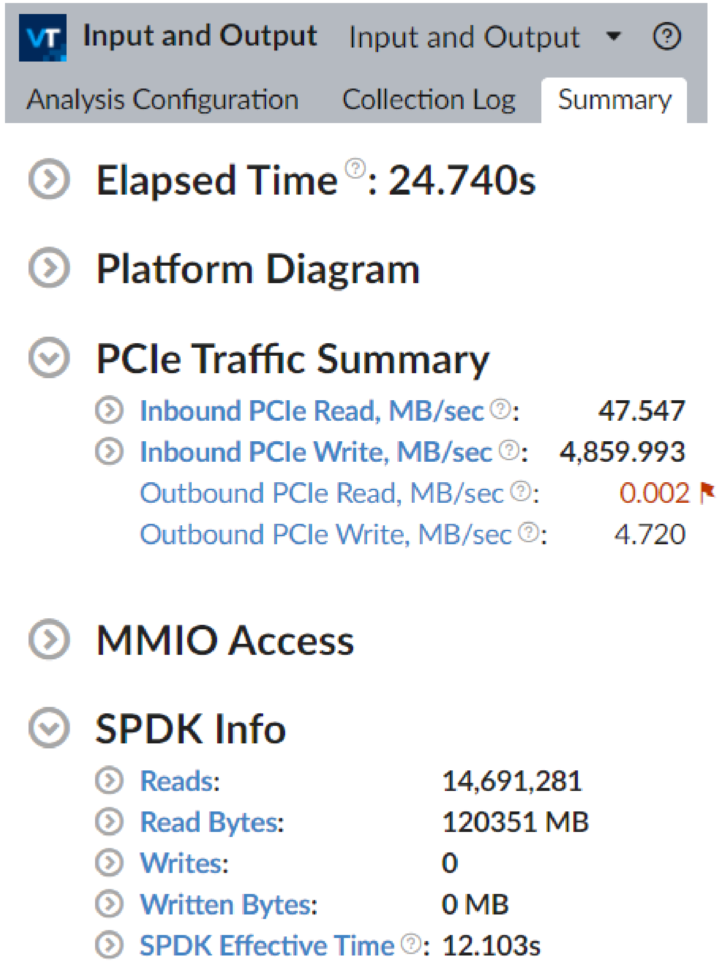
SPDK Latency Histogram

Explore the distribution of the IO operations latency over time for the selected SPDK device.

SPDK operation latency

Joint exploration of API and platform level metrics gives a better view on how workload utilizes hardware resources

SPDK and Platform Metrics Correlation



Elapsed Time: 24.740s

Platform Diagram

PCIe Traffic Summary

Inbound PCIe Read, MB/sec:	47.547
Inbound PCIe Write, MB/sec:	4,859.993
Outbound PCIe Read, MB/sec:	0.002
Outbound PCIe Write, MB/sec:	4.720

MMIO Access

SPDK Info

Reads:	14,691,281
Read Bytes:	120351 MB
Writes:	0
Written Bytes:	0 MB
SPDK Effective Time:	12.103s

Example: MMIO Accesses per IOP

App makes pure disk reads from random addresses

$$\text{MMIO Write Bytes per SPDK Read} = \frac{\text{Outbound PCIe Write [MB/sec]} \cdot 10^6 \cdot \text{Elapsed Time [sec]}}{\text{SPDK Reads}}$$

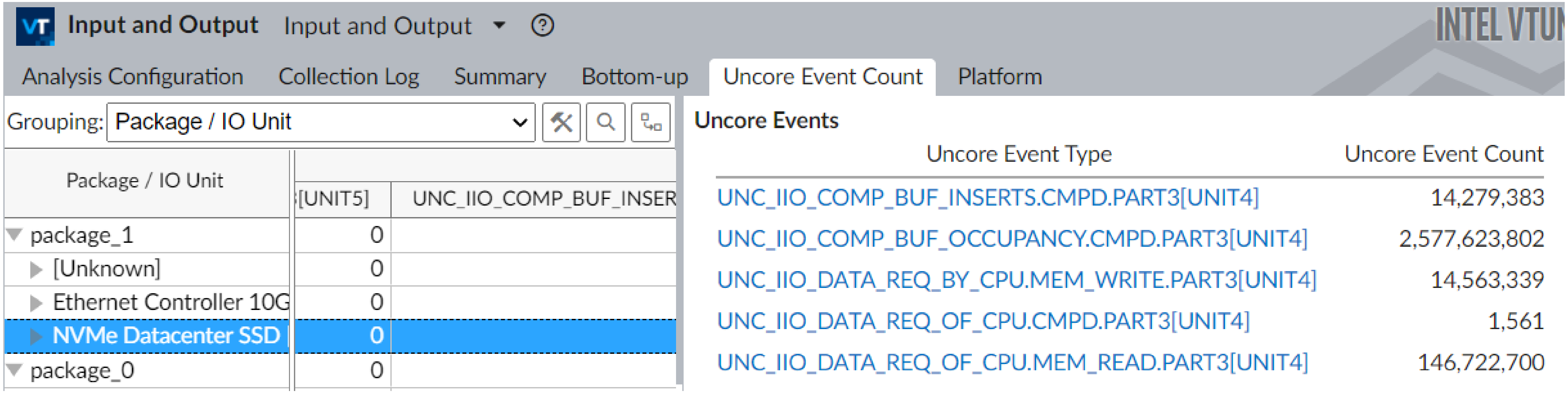
≈ **8B for this example**

Does this match core/device communication model of the workload under analysis? In this case – yes:

- For each IOP app makes two doorbells to Submission and Completion queues
- The doorbell register size is 4B

Advanced Analysis

- See a detailed view on analyzing raw events
- Customize Input and Output analysis by adding more uncore performance monitoring events
- Get a per-device view for events when applicable



Grouping: Package / IO Unit

Package / IO Unit	[UNIT5]	UNC_IIO_COMP_BUF_INSER
▼ package_1	0	
▶ [Unknown]	0	
▶ Ethernet Controller 10G	0	
▶ NVMe Datacenter SSD	0	
▼ package_0	0	

Uncore Events

Uncore Event Type	Uncore Event Count
UNC_IIO_COMP_BUF_INSERTS.CMPD.PART3[UNIT4]	14,279,383
UNC_IIO_COMP_BUF_OCCUPANCY.CMPD.PART3[UNIT4]	2,577,623,802
UNC_IIO_DATA_REQ_BY_CPU.MEM_WRITE.PART3[UNIT4]	14,563,339
UNC_IIO_DATA_REQ_OF_CPU.CMPD.PART3[UNIT4]	1,561
UNC_IIO_DATA_REQ_OF_CPU.MEM_READ.PART3[UNIT4]	146,722,700

SPDK, PMDK, Intel® Performance Analyzers | **Virtual Forum**