Intel® Data Streaming Accelerator (DSA) - DPDK-DMA Packet Copying with Intel DSA on 4th Gen Intel® Xeon® Scalable Processors

---

Authors

Chenxi Wang

Ai Bee Lim

Timothy Miskell

Abhijit Sinha

Key Contributors

Jiayu Hu

Bruce Richardson

Sunil Pai G

1 Introduction

Intel® Data Streaming Accelerator (DSA) is a high-performance data copy and transformation accelerator integrated in 4th Gen Xeon scalable processors and future Intel® processors, targeted for optimizing streaming data movement and transformation operations common with applications for high-performance storage, networking, persistent memory, and various data processing applications.

The DPDK DMAdev Library provides a DMA device framework for management and provisioning of hardware and software DMA poll mode drivers, defining generic API which supports several different DMA operations. The packet copying application will copy packets using either Hardware Mode (with Intel® DSA) or Software Mode (without Intel® DSA). The goal of Intel® DSA is to provide higher overall system performance for data mover and transformation operations, while freeing up CPU cycles for higher level functions.

This document describes the deployment and performance comparisons of DPDK-DMA Packet Copying application between Hardware Mode (with Intel® DSA) and Software Mode (without Intel® DSA) on the 4th Gen Intel® Xeon® Scalable Processors (formerly named Sapphire Rapids) with Intel® E810 Network Controllers. Specifically, to saturate the performance of Intel® DSA device, the benchmarking is built on two Intel® Ethernet Network Adapters, which able to obtain maximum 200 Gbps for bandwidth-intensive workloads. We offload packet copies to Intel® DSA in the DPDK DMAdev Library, our benchmarking results show that there is up to 3.5x Throughput improvement of 0.01% Packet Loss performance while copying packets by Intel® DSA acceleration on 4th Gen Intel ® Xeon® Scalable Processor with Intel® E810 Network Controllers. Intel® DSA plays a significant role of DPDK DMA Packet Copying when the packet size greater than or equal to 256 Bytes.

This document is intended for DPDK community to start the evaluation of Intel® DSA.

This document is part of the Network Transformation Experience Kits.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	<March 2023>	Initial release.

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
BIOS	Basic input and output service
DPDK	Data Plane Development Kit
DSA	Intel® Data Streaming Accelerator (Intel® DSA)
DUT	Device Under Test
ENQCMD	An Intel® 64 CPU instruction to enqueue a command to a shared work queue using Deferrable Memory Write (DMWr)
IOMMU	A DMA Remapping Hardware Unit as defined by Intel® Virtualization Technology for Directed I/O
NIC	Network Interface Card
NIC	Network Interface Card/Controller
NUMA	Non-Uniform Memory Access
PCI	Peripheral Component Interconnect
WQ	A queue in the device used to store descriptors submitted by software until they can be dispatched.

1.2 Reference Documentation

Table 2. Reference Documents

Reference	Source
Intel® DSA Architecture Specification	https://software.intel.com/en-us/download/intel-data-streamingaccelerator-preliminary-architecture-specification
Intel® Data Streaming Accelerator User Guide	https://www.intel.com/content/www/us/en/content-details/759709/intel-data-streaming-accelerator-user-guide.html?wapkw=DSA
Intel® Data Streaming Accelerator (DSA) - Packet Copy Offload in OVS with Intel® DSA	https://networkbuilders.intel.com/solutionslibrary/intel-data-streaming-accelerator-packet-copy-offload-ovs-technology-guide
Intel® Data Streaming Accelerator (DSA) - Accelerating DPDK Vhost	https://networkbuilders.intel.com/solutionslibrary/intel-data-streaming-accelerating-dpdk-vhost-technology-guide
Intel® Data Streaming Accelerator (DSA) - Packet Copy Offload in DPDK with Intel® DSA	https://networkbuilders.intel.com/solutionslibrary/intel-data-streaming-accelerator-packet-copy-offload-dpdk-technology-guide
Accel-config	https://github.com/intel/idxd-config
DPDK documentation	https://doc.dpdk.org/guides/index.html
DMA Device Library	https://doc.dpdk.org/guides/prog_guide/dmadev.html

2 Overview

This document describes the deployment and performance comparisons of packet copying application between Hardware Mode (with Intel® DSA) and Software Mode (without Intel® DSA) on the 4th Gen Intel® Xeon® Scalable Processors (formerly named Sapphire Rapids) with Intel® E810 Network Controllers.

The use case is using DPDK DMAdev library, packet copying application able to perform packets copying by Software Mode (without Intel® DSA) and Hardware Mode (with Intel® DSA). In the real-time statistics, the application will print out received/sent packets and packets dropped during the copying process.

According to our benchmarking, we can see a performance improvement of up to 3.5x in Throughput at 0.01% Packet Loss while copying packets by Intel® DSA acceleration on 4th Gen Intel ® Xeon® Scalable Processor with Intel® E810 Network Controllers. Intel® DSA plays a significant role in DPDK DMA Packet Copying when the packet size is greater than or equal to 256 Bytes.

2.1 DPDK DMAdev Library

The DMA library provides a DMA device framework for managing and provisioning hardware and software DMA poll mode drivers, defining generic API that supports several different DMA operations. The DMA framework provides a generic DMA device framework that supports both physical (hardware) and virtual (software) DMA devices, as well as a generic DMA API which allows DMA devices to be managed and configured and supports DMA operations to be provisioned on DMA poll mode driver. Physical DMA controllers are discovered during the PCI probe/enumeration of the EAL function that is executed at DPDK initialization. The DMA devices are dynamically allocated based on the number of hardware DMA channels also known as hardware DMA queues.

Since DPDK version 21.11 LTS, the DPDK DMAdev library can support the new device category of DMA acceleration devices. The Intel® DSA device can be involved to DPDK by the utilizations of DMAdev library.

2.2 Intel® Data Streaming Accelerator (DSA)

The goal of Intel® DSA is to provide higher overall system performance for data mover and transformation operations, while freeing up CPU cycles for higher level functions. Intel® DSA hardware supports high-performance data mover capability to/from volatile memory, persistent memory, memory-mapped I/O, and through a Non-Transparent Bridge (NTB) in the SoC to/from remote volatile and persistent memory on another node in a cluster. It provides a PCI Express compatible programming interface to the Operating System and can be controlled through a device driver.

In addition to performing basic data mover operations, Intel® DSA is designed to perform some number of higher-level transformation operations on memory. For example, it can generate and test CRC checksum or Data Integrity Field (DIF) on the memory region to support usages typical with storage and networking applications. It supports a memory compare operation for equality, generates a delta record, and applies a delta record to a buffer. These are compared and the delta generate/merge functions may be utilized by applications such as VM migration, VM fast check-pointing, and software managed memory deduplication usages.

Intel® DSA may also be used for data movement between different address spaces by using the Inter-Domain capabilities of the device. This may have applications in networking, for example with a virtual switch implementation to efficiently copy data between virtual machines or to speed up inter-process communication (IPC) primitives in the OS or VMM. It may also be used for message and data passing between processes in application domains like HPC and Machine Learning.

3 Deployment and Benchmarking

3.1 DPDK-DMA Packet Copying Application

As shown in the Figure 1, The main function of the DPDK DMAdev library for packet copying application typically follows these steps:

1. Initialize the DPDK environment and the DMAdev library.
2. Parse command line arguments to determine the copy mode of the application.
3. Allocate memory for the receive and transmit buffers and initialize the DMA device (Intel® DSA) if choose hardware copy mode.
4. Start all packet handling lcores and start printing stats in a loop on the main lcore.
5. The main lcore waits for all worker lcores to finish, deallocates resources and exits.

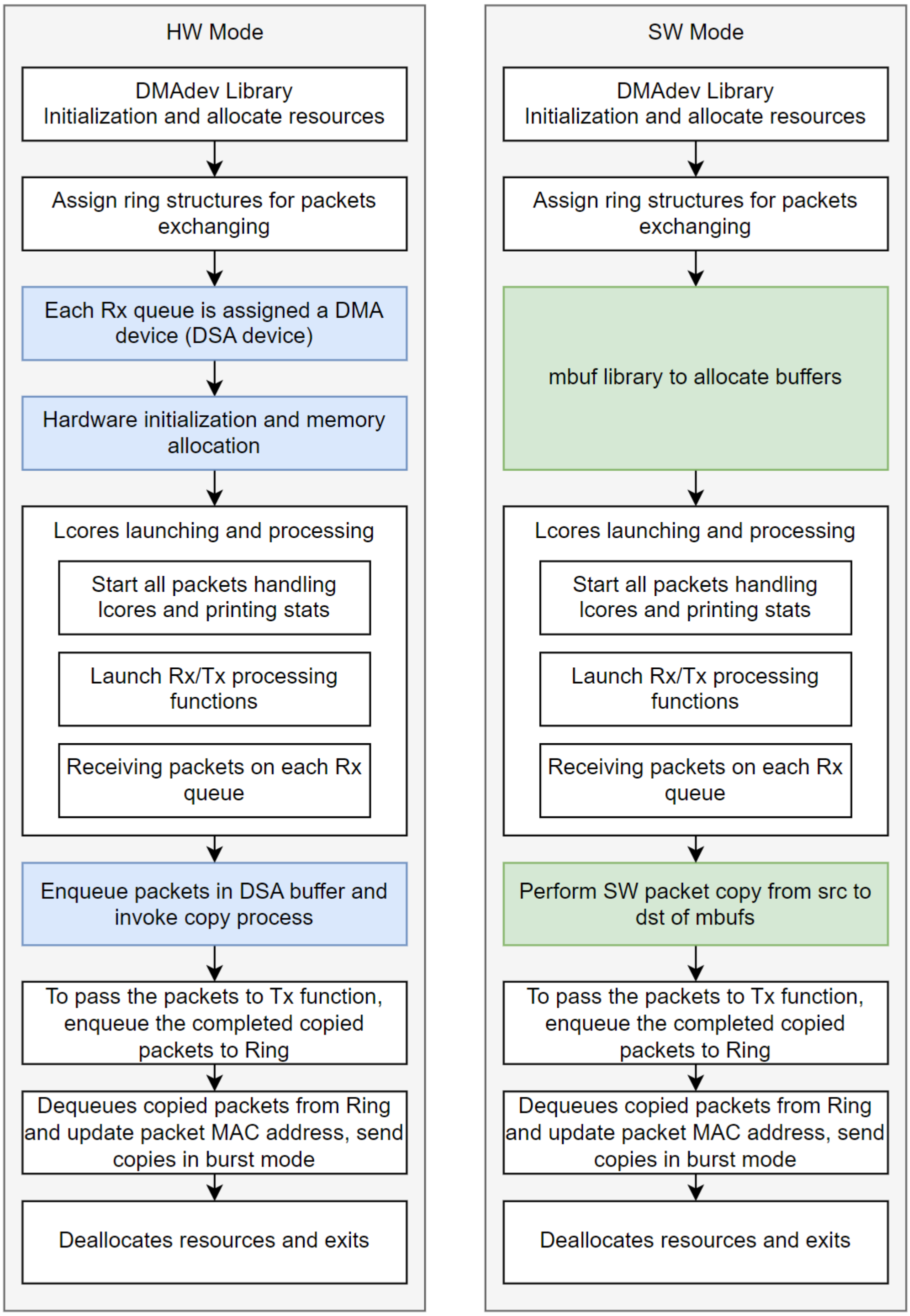

Figure 1. Packet Copying Application using DMAdev Library

(Left: Hardware Mode with Intel® DSA. Right: Software Mode without Intel® DSA.)

3.2 System Configuration

To perform the packet copying benchmarking on the 4th Gen Intel® Xeon® Scalable Processors, refer the system configuration of DUT as detailed in Table 3.

For Intel® IOMMU driver, it needs to be enabled in the kernel configuration. The Intel® IOMMU driver can also be compiled into the kernel, but not be enabled by default. In both cases, the Linux kernel boot parameters must include:

```
intel_iommu=on,sm_on
```

Table 3. System Configurations

System Configuration	
Baseboard	Jabil 2U8 Midplane
Chassis	Rack Mount Chassis
CPU Model	Intel(R) Xeon(R) Gold 6438N
Microarchitecture	SPR
Sockets	1
Cores per Socket	32
Hyperthreading	Enabled
CPUs	64
Intel Turbo Boost	Disabled
Base Frequency	2.0GHz
All-core Maximum Frequency	2.7GHz
Maximum Frequency	2.0GHz
NUMA Nodes	1
Prefetchers	L2 HW, L2 Adj., DCU HW, DCU IP
PPINs	b964bc27550ac01c
Accelerators	DLB:2, DSA:1, IAX:0, QAT (on CPU):2, QAT (on chipset):0
Installed Memory	128GB (8x16GB DDR5 4800 MT/s [4800 MT/s])
Hugepagesize	1048576 kB
Transparent Huge Pages	madvise
Automatic NUMA Balancing	Disabled
NIC	3x Ethernet Controller E810-C for QSFP, 1x Ethernet interface, 1x I210 Gigabit Network Connection
NIC Firmware	4.20 0x80017784 1.3346.0
Disk	1x 894.3G WUS3BA196C7P3E3, 1x 931.5G INTEL SSDPELKX010T8
BIOS	a2101g
Microcode	0x2b000161
OS	Ubuntu-Server 22.04.1 2022.09.21 (Cubic 2022-09-21 16:19)
Kernel	5.15.0-43-generic
TDP	205 watts
Power & Perf Policy	Performance
Frequency Governor	powersave
Frequency Driver	intel_pstate
Max C-State	9

3.3 BIOS Setting

To enable Intel® DSA and tuning the DUT configuration, refer to the BIOS setting as indicated in Table 4.

Table 4. BIOS Setting

BIOS Menu	Sub Menu	Sub Menu 2	Setting
IIO Configuration	Port configuration		**x8x8 for all ports**
	Intel VT for Directed IO		**Enable**
	PCIe ENQCMD/ENQCMDS		**Yes**
Advanced Power Management Configuration	CPU P state Control	Speed Step	**Enable**
		AVX Licence Pre-Grant	**Disable**
		AVX ICCP pre-grant level	**N/A**
	Hardware PM State Control	Hardware P-States	**Native Mode w/o Legacy Support**
	CPU C state Control	CPU C1 auto demotion	**Disable**
		CPU C1 auto undemotion	**Disable**
	Package C State Control	Package C State	**C0/C1 State**
	CPU Advanced PM Tuning	Uncore Freq Scaling	**Disable**
		Uncore Freq RAPL	**Disable**
	Energy perf bias	Power Performance Tuning	**BIOS Controls EPB**
		ENERGY_PERF_BIAS_CFG Mode	**Performance**
		Workload Configuration	**I/O Sensitive**

3.4 Testing Setup

As shown in Figure 2, there are two Intel® Ethernet Network Adapters installed in DUT. The first network adapter is 100GbE E810-2CQDA2 (code named Chapman Beach) with dual ports, up to 200 Gbps for bandwidth-intensive workloads. The second network adapter is E810-CQDA2 for OCP (code named Tacoma Rapids) with dual ports, up to 100 Gbps for bandwidth-intensive workloads.

More importantly, to saturate the performance of Intel® DSA device, we are adopting one port of each NIC mentioned above, obtaining a total 200 Gbps workload for packet copying application with only one Intel® DSA device.

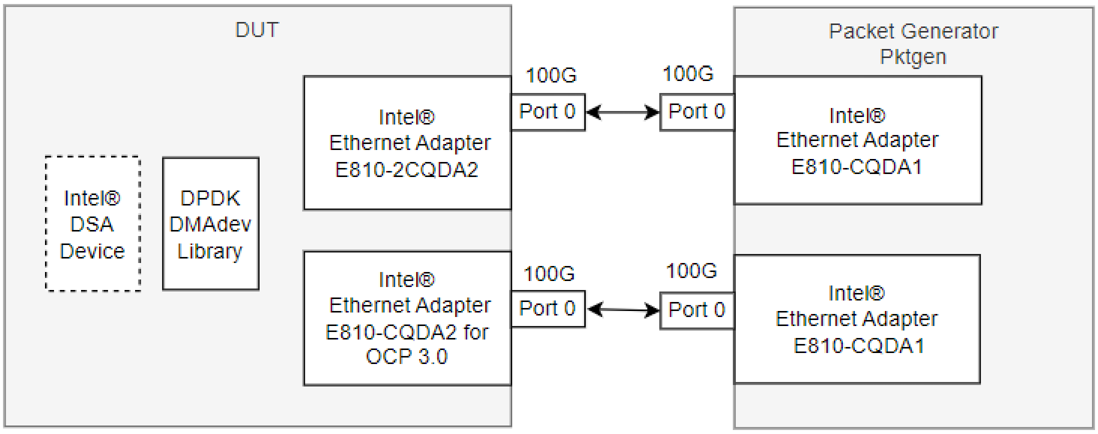

Figure 2. Setup of DUT and Packet Generator

3.5 Application Configuration

Packet copying can be launched in various configurations, depending on the provided parameters.

For the lcore configuration in the packet copying application, except a main lcore to manage the application statistics print out and devices configuration, the application can use up to 2 lcores for packets transmission work. For example, the first lcore receives incoming traffic and makes a copy of each packet, the second lcore then updates MAC address and sends the copy. If single lcore per port is used, both operations are done sequentially. For our benchmarking, we have adopted the configuration of a single lcore for packets transmission as baseline.

For the Intel® DSA device configuration in DPDK, there are two kernel drivers available. First, if the DSA device was bound to the Linux Kernel Driver 'idxd', it configures specific DSA resources for DPDK utilization, a DSA instance could be shared by multiple DPDK processes. Otherwise, DPDK will automatically configure all the Intel® DSA resources after the Linux kernel driver 'vfio-pci' bound to the Intel® DSA device.

In our case, instead of using accel-config to configure DSA WQs and Engines via 'idxd' Linux kernel driver, the DPDK-DMA packet copying application will take over the provisioning work of DSA device due to the DSA device was completely dedicated for this application by binding it to the Linux kernel driver 'vfio-pci'.

3.6 Testing Deployment

1. Check the info of NIC ports, bind the two ports with kernel driver 'vfio-pci'.
   ```
   # ./dpdk-devbind.py -s
   # ./dpdk-devbind.py -b vfio-pci 43:00.0 17:00.0
   ```
2. Identify the NUMA node, local cpu list, and all DSA devices with their NUMA node info.
   ```
   # lscpu | grep NUMA
   # lspci | grep -i 0b25
   ```
3. Choose one DSA devices which is on the same NUMA node of the NIC and bind them with kernel driver 'vfio-pci'.
   ```
   # ./dpdk-devbind.py -b vfio-pci f2:01.0
   ```
4. Start a new terminal for Pktgen on the end of packet generator.
5. Start the DPDK DMA packet copying application to perform the testing.
   a. Hardware Mode (with Intel® DSA)
   ```
   # ./dpdk-dma -l 10-11 --main-lcore=10 -n 8 -a 0000:43:00.0 -a 0000:17:00.0 -a 0000:f2:01.0 -- -
   p 0x3 -q 1 --mac-updating -c hw
   ```
   b. Software Mode (without Intel® DSA)
   ```
   # ./dpdk-dma -l 10-11 --main-lcore=10 -n 8 -a 0000:43:00.0 -a 0000:17:00.0 -- -p 0x3 -q 1 --
   mac-updating -c sw
   ```
6. Send traffic with various packet sizes from Pktgen. Then check statistics on the output of dpdk-dma packet copying application:
   a. The copy ops on "Total submitted ops" and "Total completed ops" were counted by every DMA channel. "Total completed ops" should be almost equal to "Total submitted ops"
   b. "Total failed ops" should be 0

4 Testing Results

As shown below, both Software Mode and Hardware Mode follow the same metrics and DMAdev settings, except for the different copy modes (i.e. sw/hw).

- lcore amount for DMA forwarding: 1
- DSA device: 1
- Ring size 2048
- RX queues: 1
- Updating MAC: enabled
- Packet loss: 0.01%
- Hugepagesize: 1G
- Copy Mode: SW/HW

As shown in Table 5, the Hardware Mode (with Intel® DSA) takes advantages of the throughput when the packet size equal to or greater than 256 Bytes. Based on the benchmarking results, the packet copying throughput in Hardware Mode best outperforms the Software Mode with packet size of 1024 Bytes.

Note: DPDK DMAdev library packet copying in Software Mode (without Intel® DSA) takes the advantage of small size packets copying, such as 64 Bytes and 128 Bytes. For small size packets, the core operation outperforms Intel® DSA offloading because of its efficiency.

Table 5. Throughput Comparison between DPDK DMA Packing Copying in Hardware Mode and Software Mode

Packet Size (Bytes)	Software Mode (Gbits/s)	Hardware Mode (Gbits/s)
64	18	12
128	30	30
256	44	56
512	40	114
1024	44	200
1280	48	200

According to the comparisons in Figure 3 and Figure 4, we can see that when the packet size gradually increases, the throughput of the Hardware Mode increases sharply compared with that of the Software Mode, and gradually widens the huge gap. When the packet size increased to 1024 Bytes, up to 3.5x throughput improvement while copying packets by Intel® DSA acceleration on 4th Gen Intel ® Xeon® Scalable Processor with Intel® E810 Network Controllers.

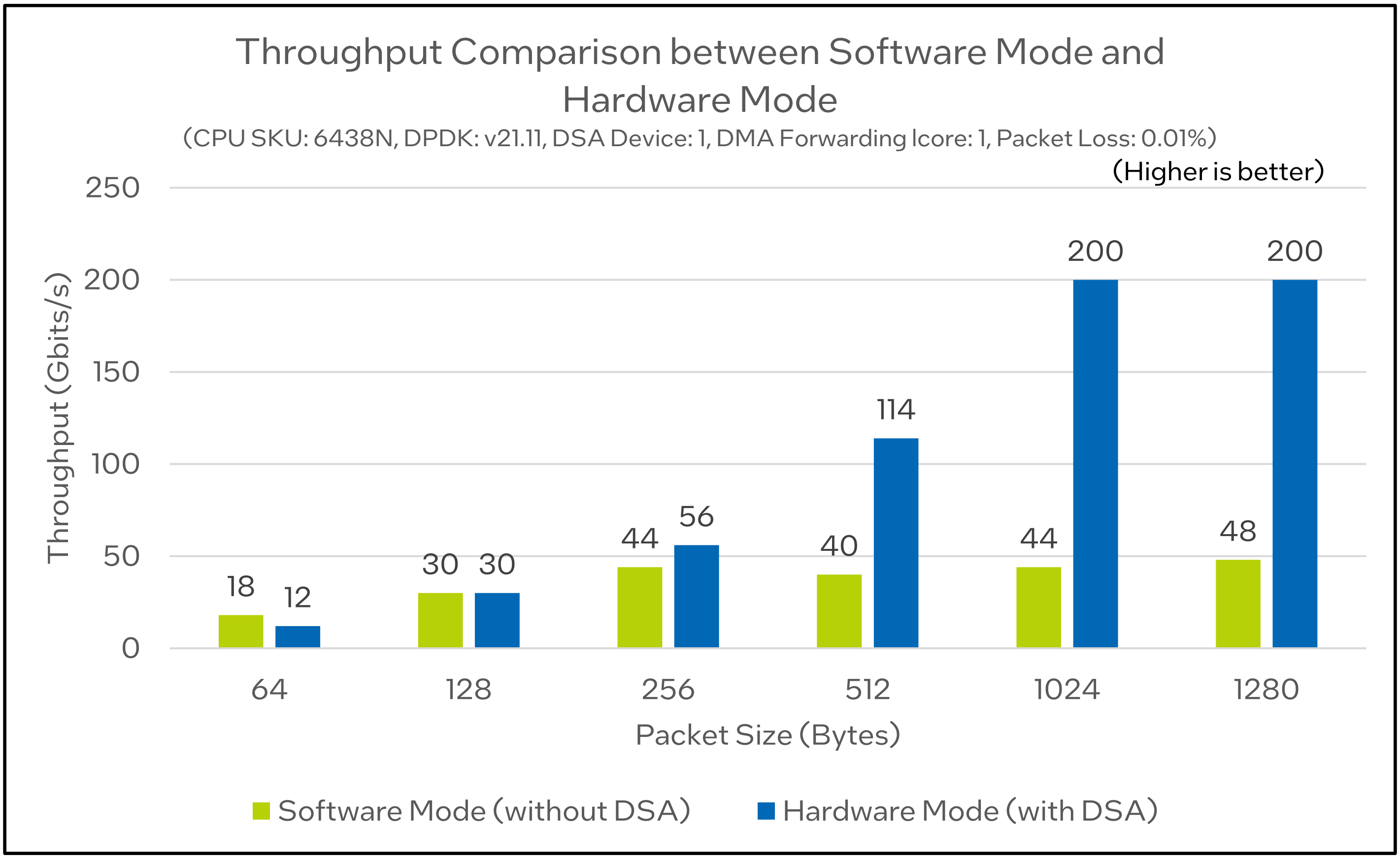

Figure 3. Throughput Comparison between Software Mode and Hardware Mode[1]

---

[1] As measured on Sapphire Rapids MCC: Test by Intel as of 03/03/23. 1-node, 1x Intel(R) Xeon(R) Gold 6438N CPU @ 2.0GHz, 32 cores, HT On, Turbo Off, Total Memory 128GB (8x16GB DDR5 4800 MT/s [4800 MT/s]), BIOS a2101g, Microcode 0x2b000161, 1x I210 Gigabit Network Connection, 3x Ethernet Controller E810-C for QSFP, 1x Ethernet interface, 1x I210 Gigabit Network Connection, Disk 1x 931.5G INTEL SSDPELKX010T8, 1x 894.3G WUS3BA196C7P3E3, Ubuntu-Server 22.04.1, 5.15.0-43-generic, GCC 11.3.0, DPDK 21.11.

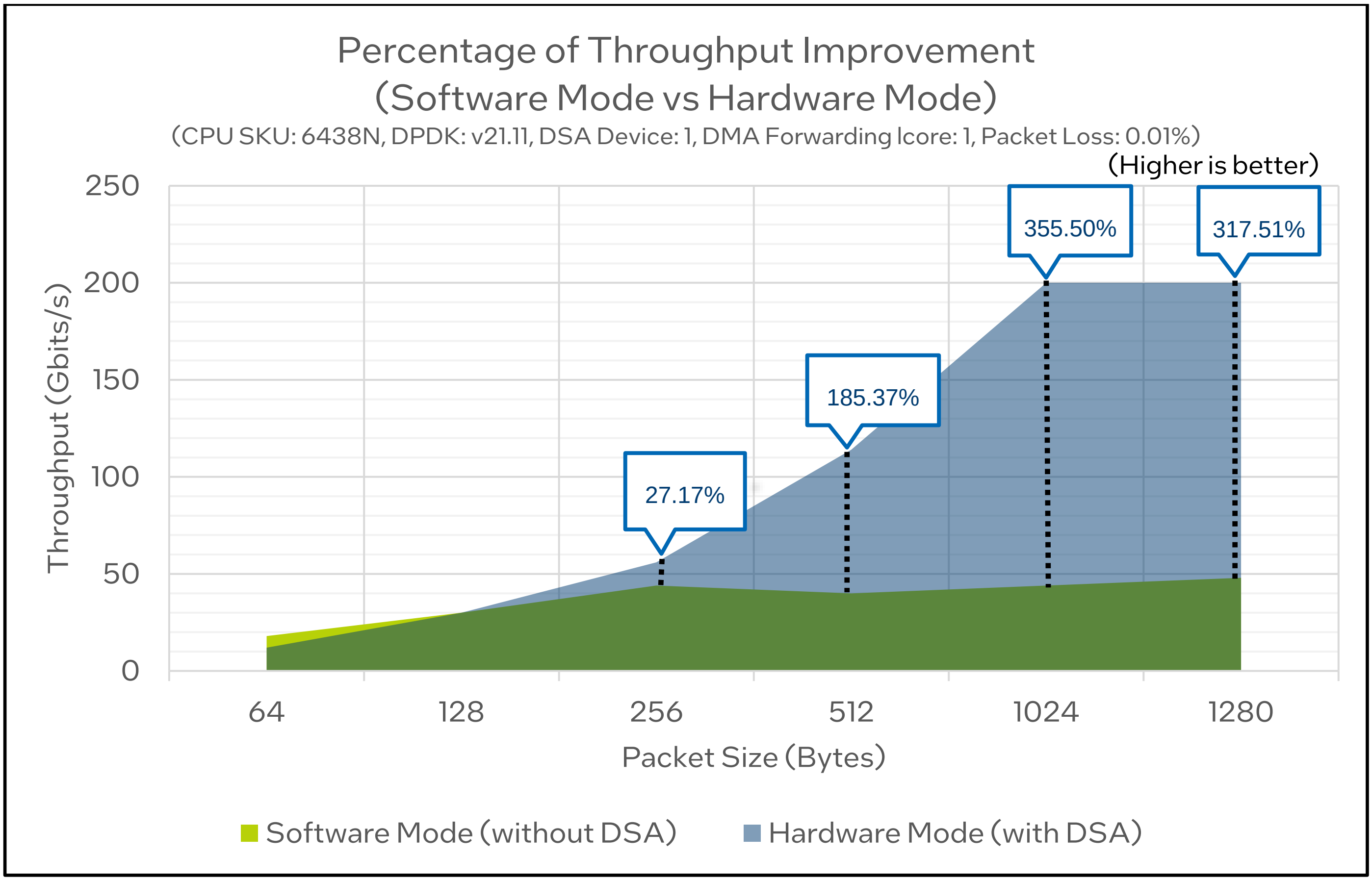

Figure 4. Percentage of Throughput Improvement between Software Mode and Hardware Mode

5 Summary

In conclusion, through the packet copying application of the DPDK DMAdev library, we demonstrated the value proposition of Intel® DSA on 4th Gen Intel® Xeon® Scalable Processors. Based on the benchmarking results, we compared the performance of Hardware Mode (with Intel® DSA) or Software Mode (without Intel® DSA) with various packet sizes. Specifically, to saturate the performance of the Intel® DSA device, benchmarking is built on two Intel® Ethernet Network Adapters, which can achieve a maximum throughput of 200 Gbps for bandwidth-intensive workloads.

According to the benchmarking results, we found that Hardware Mode (with Intel® DSA) consistently outperformed Software Mode (without Intel® DSA) in terms of throughput when the packet size is equal to or greater than 256 Bytes. As the packet size increases, we can find that the throughput of Intel DSA accelerated copying also increases sharply, the throughput improvement reaching Up to 3.5x over software packet copying.

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.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

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

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

0423/CW/Intel/PDF 776814-001US