Enabling Intel Data Streaming Accelerator on Lenovo ThinkSystem Servers

Explains performance bottleneck observation

Introduces the use of Intel DSA to improve performance

Describes how to enable Intel DSA on Lenovo ThinkSystem Servers

Shows how to configure Intel DSA in Linux OS

Adrian Huang

LENOVO PRESS

Abstract

Intel Data Streaming Accelerator (Intel DSA) is a feature of the upcoming Intel Xeon Scalable processors (codename “Sapphire Rapids”). Intel DSA provides a high-performance data copy and data transformation accelerator. The use of the accelerator frees the processor to execute other tasks instead of being busy copying data or transforming data.

This white paper provides a guidance about how to enable Intel DSA in UEFI in ThinkSystem™ servers and guidance on how to use Intel DSA in Linux operating systems. This paper is intended for IT specialists who want to use Intel DSA device for their own applications. Readers should have basic knowledge about Intel DSA and experience in compiling applications in Linux.

At Lenovo® Press, we bring together experts to produce technical publications around topics of importance to you, providing information and best practices for using Lenovo products and solutions to solve IT challenges.

See a list of our most recent publications at the Lenovo Press web site:

http://lenovopress.com

Do you have the latest version? We update our papers from time to time, so check whether you have the latest version of this document by clicking the **Check for Updates** button on the front page of the PDF. Pressing this button will take you to a web page that will tell you if you are reading the latest version of the document and give you a link to the latest if needed. While you’re there, you can also sign up to get notified via email whenever we make an update.

Contents

Introduction

Intel Data Streaming Accelerator (Intel DSA) is a feature of the upcoming Intel Xeon Scalable processors (codename “Sapphire Rapids”). Intel DSA provides a high-performance data copy and data transformation accelerator. The use of the accelerator frees the processor to execute other tasks instead of being busy copying data or transforming data.

Intel DSA supports two categories of functions:

- Data copy

  This function is the same as Intel QuickData technology, which Intel DSA is planned to replace

- Data transformation

  Intel DSA supports the following data transformation features:

  – CRC checksum generation and verification

  – Data Integrity Field (DIF): Protect data integrity for computer data storage

  – Generate and apply delta record: This feature can be applied for guest OS migration that the hypervisor knows modified pages and sends them to the destination machine. The task of generating modified pages (generate delta record) and applying modified pages (apply delta record) can be done by Intel DSA.

Intel DSA is useful in the following scenarios:

- Memory copy functions: Apply data copy (such as memory copy or memory zeroing) to free up CPU cycles.
- Storage: Apply DIF generation to free up CPU cycles.
- Networking: A virtual switch is widely used in virtualization environment. The virtual switch requires data copy in packing processing. Intel DSA can be a virtual switch offloading engine for inter-VM packing switching.
- Guest OS Migration: The hypervisor needs to know modified pages and sends them to the destination machine. Intel DSA can generate the delta record so that the hypervisor can send the delta record to destination machine. This can reduce the network traffic and free up CPU cycles.

Intel DSA supports two types of work queues:

- Dedicated Work Queue (DWQ): The work queue is owned by a single user exclusively. The single user can submit work to it.
- Shared Work Queue (SWQ): In SWQ, works can be submitted by multiple users.

Read more about Intel DSA from these Intel resources:

- Updates on Intel’s Next-Gen Data Center Platform, Sapphire Rapids

  https://www.intel.com/content/www/us/en/newsroom/opinion/updates-next-gen-data-center-platform-sapphire-rapids.html

- Introducing the Intel® Data Streaming Accelerator (Intel® DSA)

  https://01.org/blogs/2019/introducing-intel-data-streaming-accelerator

Comparison with existing offloading technologies

In network data communication, industry has implemented various features to improve CPU overhead for both transmitter and receiver. The technologies include the following:

- TCP Segmentation offload (TSO)
- Zero-copy: CPU does not need to copy user data into kernel space buffer via a `sendfile()` system call.
- Intel I/O Acceleration Technology (I/OAT) features:
  - Split headers
  - Multiple receive queues
  - DMA copy offload engine (Intel QuickData Technology)

A key comparative feature is Intel QuickData Technology. Intel QuickData Technology is a dedicated device to perform data copy that offloads the task of expensive data copies off the CPU. With the use of Intel QuickData Technology, the CPU is free to execute other tasks.

The paper *Accelerating Network Receive Processing - Intel I/O Acceleration Technology*, Linux Symposium (2005) shows CPU is busy at data movement, which is the CPU-intensive task. This CPU intensive task can be offloaded by a dedicated hardware (Intel QuickData Technology or Intel DSA). View the paper at:

https://landley.net/kdocs/ols/2005/ols2005v1-pages-289-296.pdf

Enabling Intel DSA

In this section, we describe how to configure the server to enable Intel DSA. In our lab tests, our ThinkSystem server was running a beta version of RHEL 9.0.

The steps to enable Intel DSA are as follows:

1. Boot the server to UEFI (Press F1 as power on, when prompted)
2. Navigate to **System Information** → **Socket Configuration** → **Uncore Configuration** → **Uncore General Configuration**.

3. Set the option **Limit CPU PA to 46 bits** to Disabled, as shown in Figure 1 on page 5.

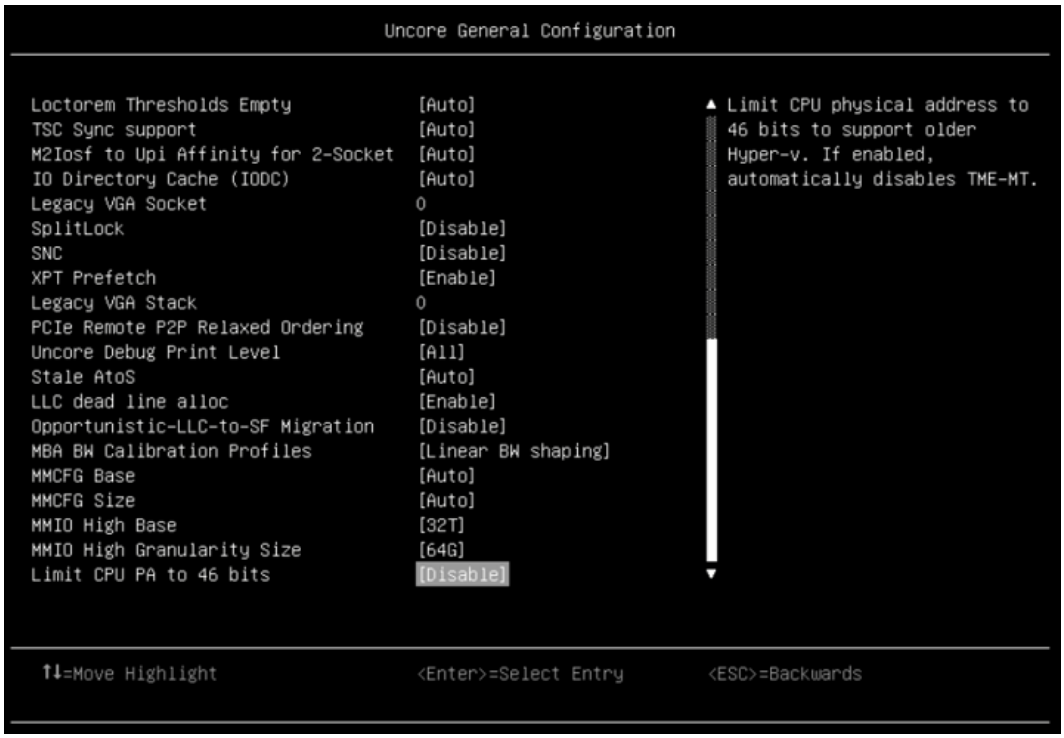

Figure 1 Uncore General Configuration panel in UEFI

4. Boot to Linux and in the grub config file, append the following as a kernel boot parameter:

   `intel_iommu=on,sm_on`

5. Make sure Intel DSA devices (`idxd` devices in Linux kernel) are enabled properly using the **`dmesg`** command, as shown in Figure 2.

```
# dmesg| grep idxd
idxd 0000:6a:02.0: enabling device (0140 -> 0142)
idxd 0000:6a:02.0: Intel(R) Accelerator Device (v100)
idxd 0000:6f:02.0: enabling device (0140 -> 0142)
idxd 0000:6f:02.0: Intel(R) Accelerator Device (v100)
idxd 0000:74:02.0: enabling device (0140 -> 0142)
idxd 0000:74:02.0: Intel(R) Accelerator Device (v100)
idxd 0000:79:02.0: enabling device (0140 -> 0142)
idxd 0000:79:02.0: Intel(R) Accelerator Device (v100)
idxd 0000:e7:01.0: enabling device (0144 -> 0146)
idxd 0000:e7:01.0: Intel(R) Accelerator Device (v100)
idxd 0000:e7:02.0: enabling device (0140 -> 0142)
idxd 0000:e7:02.0: Intel(R) Accelerator Device (v100)
…
```

Figure 2 Output from dmesg command

6. Install accel-config package:

   `# yum install accel-config`

7. Use the **`accel-config list`** command with the **`-i`** (idle) argument to show how many idle DSA devices are in the system as shown in Figure 3. Note that devices `dsa10` and workqueue `wq10.0` are listed; these will be referred to in a later step.

```
# accel-config list -i
[
  {
    "dev":"dsa10",
    "token_limit":0,
    "max_groups":4,
    "max_work_queues":8,
    "max_engines":4,
    "work_queue_size":128,
    "numa_node":1,
    "op_cap":[
      "0x1003f03ff",
      "0",
      "0",
      "0"
    ],
    "gen_cap":"0x40915f010f",
    "version":"0x100",
    "state":"disabled",
    "max_tokens":96,
    "max_batch_size":1024,
    "max_transfer_size":2147483648,
    "configurable":1,
    "pasid_enabled":1,
    "cdev_major":240,
    "clients":0,
    …
    "ungrouped workqueues":[
      {
        "dev":"wq10.0",
        "mode":"shared",
        "size":0,
        "priority":0,
        …
        "state":"disabled",
        "clients":0
      },
      …
    ]
  }
]
```

Figure 3 Output from `accel-config list -i` *command*

8. Use the **`accel-config list`** command without flags to show the active DSA devices in the system. Figure 4 shows no active DSA devices are configured.

```
# accel-config list
[
]
```

Figure 4 Output from `accel-config list` *command*

9. Enable a DSA device and a workqueue (wq) by executing the following commands listed in Figure 5. Note that arguments `dsa10` and `wq10.0` are referred in Figure 3 on page 6.

```
# accel-config config-wq dsa10/wq10.0 --group-id=0
# accel-config config-wq dsa10/wq10.0 --priority=5
# accel-config config-wq dsa10/wq10.0 --wq-size=8
# accel-config config-engine dsa10/engine10.0 --group-id=0
# accel-config config-wq dsa10/wq10.0 --type=user
# accel-config config-wq dsa10/wq10.0 --name="dsa-test"
# accel-config config-wq dsa10/wq10.0 --mode=dedicated
# accel-config enable-device dsa10
enabled 1 device(s) out of 1
# accel-config enable-wq dsa10/wq10.0
enabled 1 wq(s) out of 1
```

Figure 5 Enabling the DSA device and workqueue

10. Rerun the command **`accel-config list`** to make sure the newly enabled DSA device (dsa10) and workqueue device are listed, as shown in Figure 6.

```
# accel-config list
[
  {
    "dev":"dsa10",
    "token_limit":0,
    "max_groups":4,
    "max_work_queues":8,
    "max_engines":4,
    "work_queue_size":128,
    "numa_node":1,
    "op_cap":[
      "0x1003f03ff",
      "0",
      "0",
      "0"
    ],
    "gen_cap":"0x40915f010f",
    "version":"0x100",
    "state":"enabled",
    "max_tokens":96,
    "max_batch_size":1024,
    "max_transfer_size":2147483648,
    "configurable":1,
    "pasid_enabled":1,
    "cdev_major":238,
    "clients":0,
    "groups":[
      {
        "dev":"group10.0",
        …
        "grouped_workqueues":[
          {
            "dev":"wq10.0",
            "mode":"dedicated",
            "size":8,
            "group_id":0,
            "priority":5,
            "block_on_fault":0,
            "max_batch_size":1024,
            "max_transfer_size":2147483648,
            "cdev_minor":0,
            "type":"user",
            "name":"dsa-test",
            "threshold":0,
            "state":"enabled",
            "clients":0
          }
        ],
        …
    …
  }
]
```

Figure 6 Verifying that the device and workqueue are now enabled

11. Optionally, if the user does not want to use the active DSA device anymore, the following commands can be used to disable the work queue and the DSA device.

```
# accel-config disable-wq dsa10/wq10.0
# accel-config disable-device dsa10
```

Figure 7 Disabling the workqueue and device

Executing DSA operations using idxd-config

The Intel DSA specification defines a list of operations (For example: memory move, compare, create delta record, CRC generation and so on). For details, see the Intel DSA Architecture Specification, available at the following location:

https://software.intel.com/en-us/download/intel-data-streaming-accelerator-preliminary-architecture-specification

The following example shows how to execute “memory move” operation and “memory fill” operation via idxd-config tool.

1. Download the idxd-config source code from:

   https://github.com/intel/idxd-config

2. Install the required packages:

   ```
   # yum install xmlto uuid libuuid-devel json-c-devel
   ```

3. Compile idxd-config

```
# ./configure --enable-test
# make
```

Figure 8 Compile idxd-config

4. Configure one dedicated workqueue and one DSA device, using the commands listed in Figure 9.

```
# accel-config config-wq dsa10/wq10.0 --group-id=0
# accel-config config-wq dsa10/wq10.0 --priority=5
# accel-config config-wq dsa10/wq10.0 --wq-size=8
# accel-config config-engine dsa10/engine10.0 --group-id=0
# accel-config config-wq dsa10/wq10.0 --type=user
# accel-config config-wq dsa10/wq10.0 --name="dsa-test"
# accel-config config-wq dsa10/wq10.0 --mode=dedicated
# accel-config enable-device dsa10
enabled 1 device(s) out of 1
# accel-config enable-wq dsa10/wq10.0
enabled 1 wq(s) out of 1
```

Figure 9 Configuring the DSA device and workqueue

5. Execute memory move operation via idxd-config tool as shown in Figure 10 on page 10. The syntax of the command is as follows:
   - Dedicated work queue (**-w 0**)
   - Buffer size = 2MB (**-l 2097152**)
   - Operation: memory move (**-o 0x3**)
   - Flag: block on fault (**-f 0x1**)
   - Timeout: 200ms (**t200**)
   - Verbose mode: (**-v**)

   Note: Make sure you are under idxd-config folder when you run the command

```
# ./test/dsa_test -w 0 -l 2097152 -o 0x3 -f 0x1 t200 -v
[debug] umwait supported
[ info] alloc wq 0 shared 1 size 8 addr 0x7f73b166f000 batch sz 0x400 xfer sz 0x80000000
[ info] testmemory: opcode 3 len 0x200000 tflags 0x1 num_desc 1
[debug] initilizing task 0x5448a0
[debug] Mem allocated: s1 0x7f73b124c040 s2 0 d1 0x7f73b104b040 d2 0
[ info] preparing descriptor for memcpy
[ info] Submitted all memcpy jobs
[debug] desc addr: 0x54b6a0
[debug] desc[0]: 0x0300000c00000000
[debug] desc[1]: 0x000000000054b8a0
[debug] desc[2]: 0x00007f73b124c040
[debug] desc[3]: 0x00007f73b104b040
[debug] desc[4]: 0x0000000000200000
[debug] desc[5]: 0x0000000000000000
[debug] desc[6]: 0x0000000000000000
[debug] desc[7]: 0x0000000000000000
[debug] completion record addr: 0x54b8a0
[debug] compl[0]: 0x0000000000000001
[debug] compl[1]: 0x0000000000000000
[debug] compl[2]: 0x0000000000000000
[debug] compl[3]: 0x0000000000000000
[ info] verifying task result for 0x5448a0
```

Figure 10 Using idxd-config to execute a memory move operation

6. Execute memory fill operation via idxd-config tool as shown in Figure 11. The syntax of the command is as follows:
   - Dedicated work queue (**-w 0**)
   - Buffer size = 2MB (**-l 2097152**)
   - Operation: memory fill (**-o 0x4**)
   - Flag: block on fault (**-f 0x1**)
   - Timeout: 200ms (**t200**)
   - Verbose mode: (**-v**)

   Note: Make sure you are under idxd-config folder when you run the command

```
# ./test/dsa_test -w 0 -l 2097152 -o 0x4 -f 0x1 t200 -v
[debug] umwait supported
[ info] alloc wq 0 shared 1 size 8 addr 0x7fe315e9d000 batch sz 0x400 xfer sz 0x80000000
[ info] testmemory: opcode 4 len 0x200000 tflags 0x1 num_desc 1
[debug] initilizing task 0x9968a0
[debug] Mem allocated: s1 0 s2 0 d1 0x7fe315a7a040 d2 0
[ info] preparing descriptor for memfill
[ info] Submitted all memcpy jobs
[debug] desc addr: 0x99d6a0
[debug] desc[0]: 0x0400000c00000000
[debug] desc[1]: 0x000000000099d8a0
[debug] desc[2]: 0x0123456789abcdef
[debug] desc[3]: 0x00007fe315a7a040
[debug] desc[4]: 0x0000000000200000
[debug] desc[5]: 0x0000000000000000
[debug] desc[6]: 0x0000000000000000
[debug] desc[7]: 0x0000000000000000
[debug] completion record addr: 0x99d8a0
[debug] compl[0]: 0x0000000000000001
[debug] compl[1]: 0x0000000000000000
[debug] compl[2]: 0x0000000000000000
[debug] compl[3]: 0x0000000000000000
[ info] verifying task result for 0x9968a0
```

Figure 11 Using idxd-config to execute a memory fill operation

Considerations in using Intel DSA

The current upstream kernel only supports DWQ configuration because SWQ kernel code introduces the side effect. For details, see

https://lore.kernel.org/all/87mtsd6gr9.ffs@nanos.tec.linutronix.de/

This has the affect that only RHEL 9.0, RHEL 8.6 and SLES 15 SP4 support DWQ configuration.

The kernel community plans to support SWQ in kernel version v5.18 (which was not released at the time this paper was written)

The lab work in this paper only focuses on DWQ configuration.

Resources

- Updates on Intel's Next-Gen Data Center Platform, Sapphire Rapids

  https://www.intel.com/content/www/us/en/newsroom/opinion/updates-next-gen-data-center-platform-sapphire-rapids.html
- Introducing the Intel® Data Streaming Accelerator (Intel® DSA)

  https://01.org/blogs/2019/introducing-intel-data-streaming-accelerator
- Intel I/O Acceleration Technology (Intel I/OAT)

  https://www.intel.com/content/www/us/en/wireless-network/accel-technology.html

- Accelerating Network Receive Processing - Intel I/O Acceleration Technology, Linux Symposium (2005)

  https://landley.net/kdocs/ols/2005/ols2005v1-pages-289-296.pdf
- Intel Data Streaming Accelerator Architecture Specification

  https://software.intel.com/en-us/download/intel-data-streaming-accelerator-preliminary-architecture-specification
- Intel Scalable I/O Virtualization Technical Specification

  https://www.intel.com/content/www/us/en/develop/download/intel-scalable-io-virtualization-technical-specification.html
- Intel Data Accelerator Control Utility and Library

  https://github.com/intel/idxd-config
- Accelerating High-Speed Networking with Intel I/O Acceleration Technology

  https://www.intel.com/content/www/us/en/io/i-o-acceleration-technology-paper.html
- Pedal To The Metal: Accelerator Configuration and Control for Open Source:

  https://01.org/blogs/2020/pedal-metal-accelerator-configuration-and-control-open-source

Author

Adrian Huang is a Senior Linux Engineer at the Lenovo Data Center Group in Taipei, Taiwan. He has experience with Linux kernel IOMMU subsystem, block device layer and memory management. He also contributes some kernel patches to the kernel community.

Special thanks to the following people for their contributions and suggestions:

- Xiaochun Li, Lenovo Linux Engineer
- Song Shang, Lenovo Linux Engineer
- Gary Cudak, Lenovo OS Architect
- David Watts, Lenovo Press

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document was created or updated on April 18, 2022.

Send us your comments via the **Rate & Provide Feedback** form found at
http://lenovopress.com/lp1582

Trademarks