APPLICATION NOTE

Intel Corporation

Telco/Cloud Enablement for 2nd Generation Intel® Xeon® Scalable platform - Intel® Resource Director Technology

Authors

Shivapriya Hiremath

Sarita Maini

Andrew Herdrich

1 Introduction

Intel® Resource Director Technology (Intel® RDT) is a key feature set to optimize application performance and enhance the shared resource monitoring and control capabilities of 2nd Generation Intel® Xeon® Scalable processors. This application note shows how Intel® RDT can be used in Intel® Xeon® Scalable processors to improve throughput in deployments with the Data Plane Development Kit (DPDK), Open vSwitch (OvS), and the Kernel-based Virtual Machine (KVM) hypervisor software components.

This document focuses on the Intel® RDT Cache Monitoring Technology (CMT) and Cache Allocation Technology (CAT) features in the 2nd Generation Intel® Xeon® Scalable processors (formerly codenamed Cascade Lake) and showcases their benefits for typical telecom environment deployments, with a particular focus on benefits for virtualized workloads. Additional Intel® RDT technologies may be discussed in future documents.

This document is part of the Network Transformation Experience Kit, which is available at: https://networkbuilders.intel.com/

Table of Contents

Figures

Tables

1.1 Intended Audience

This white paper is intended for communication service providers who are planning and deploying virtualized mobile core infrastructure running on the latest Intel® Xeon® Scalable Processors.

1.2 Terminology

Table 1. Terminology

Abbreviation	Description
CAT	Cache Allocation Technology
CBM	Capacity Bit Mask
CLOS or CoS	Class of Service – used with Intel® RDT Allocation technologies
CMT	Cache Monitoring Technology
DPDK	Data Plane Development Kit – available at dpdk.org
Intel® RDT	Intel® Resource Director Technology – main site is here
KVM	Kernel-based Virtual Machine
L2 cache	Mid-Level Cache (MLC)
L3 cache	Last-Level Cache (LLC)
LLC	Last-Level Cache, also known as L3 cache
MBA	Memory Bandwidth Allocation
MBM	Memory Bandwidth Monitoring
MLC	Mid-Level Cache, also known as L2 cache
MSR	Model Specific Registers
NFV	Network Functions Virtualization
OvS	Open vSwitch
PMD	Poll Mode Driver
PQoS	Platform Quality of Service – an older name for Intel® RDT
RMID	Resource Monitoring ID – used with Intel® RDT Monitoring technologies
SLA	Service Level Agreement – typically a contractual specification of uptime and other operational requirements
SLO	Service Level Objectives – typically a specification of performance or latency requirements.
vCPE	Virtual Customer Premise Equipment
vEPC	Virtual Evolved Packet Core
VM	Virtual Machine
VNF	Virtual Network Function

1.3 Reference Documents

Table 2. Reference Documents

Reference	Source
Intel Architecture Software Developer Manuals, including Intel® RDT enumeration and interface low-level details	www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html
Intel® Open Network Platform Release 2.1 Application Note on Resource Director Technology	https://download.01.org/packet-processing/ONPS2.1/Intel_ONP_Release_2.1_Application_Note_on_RDT_Rev1.0.pdf
Intel® Resource Director Technology (Intel® RDT) Landing Page	http://www.intel.com/content/www/us/en/architecture-and-technology/resource-director-technology.html
Intel® RDT Github repository	https://github.com/intel/intel-cmt-cat
Intel® RDT FAQ, usage examples and useful links.	https://github.com/01org/intel-cmt-cat/wiki
Deterministic Network Functions Virtualization with Intel® Resource Director Technology White Paper	https://builders.intel.com/docs/networkbuilders/deterministic_network_functions_virtualization_with_Intel_Resource_Director_Technology.pdf

2 Intel® Resource Director Technology Overview

Intel® Resource Director Technology (Intel® RDT) is a set of Intel technologies focused on providing improved monitoring and control over shared platform resources, including last-level cache (LLC) and memory bandwidth.

Constituent features of Intel® RDT include:

- Cache Monitoring Technology (CMT) and Cache Allocation Technology (CAT), which provide the hardware framework to monitor and control the use of shared Last Level Cache (LLC), for instance.
- Memory Bandwidth Monitoring (MBM) and Memory Bandwidth Allocation (MBA) which provide the framework to monitor and control memory bandwidth.

As multithreaded and multicore platform architectures continue to evolve, there are workloads running in a single-threaded, multithreaded, or complex virtualized environments with many collaboratively operating virtual machines, such as in Network Function Virtualization (NFV). In such deployments, the LLC and memory bandwidth are key resources to monitor, manage and use optimally to ensure the performance and runtime determinism of the workloads present.

With NFV, meeting Service Level Objectives (SLOs) with predictable performance is a key requirement, for instance to maintain a constant 100Gbps network packet rate. Intel® RDT helps to achieve that predictability in performance with various Virtual Network Function (VNF) deployments, thereby enabling Communications Service Providers, Cloud Service Providers, and Enterprise deployments to meet the SLOs on modern advanced Intel® Xeon® based servers.

Intel® RDT provides CMT and CAT to monitor and manage the resource utilization of various workloads across shared resources. This allows end users to monitor the cache and memory bandwidth footprint of their applications, and allocate resource priority to applications as desired, including dynamically at run-time as system conditions change. This monitoring and allocation offers a level of application resource management that is increasingly important in service-assured and increasingly consolidated NFV environments. Workloads with strict jitter or throughput requirements, for instance, can benefit from the application of Intel® RDT.

This application note focuses on the Intel® RDT CAT and CMT features in the 2nd Generation Intel® Xeon® Scalable processors (formerly codenamed Cascade Lake) and showcases their benefits for typical telecom environment deployments, with a particular focus on benefits for virtualized workloads. Additional Intel® RDT technologies may be discussed in future documents.

2.1 Demystifying Intel® Resource Director Technology Components

In the 2nd Generation Intel® Xeon® Scalable processors (formerly codenamed Cascade Lake), the CPU cache architecture has been enhanced relative to previous Intel® Xeon® processors. This has impact on the tuning and usage of the Intel® RDT CAT and CMT technologies in various real-life scenarios, although the software interfaces remain consistent and architectural as defined in the Intel Architecture Software Developer Manuals (refer to Table 2).

An important point to note is that the per-core L2 Cache in the Intel® Xeon® Scalable processors was increased to a generous 1MB on current generation processors, which enabled performance improvements. (Per-core L2 Cache is the same for both first and second Generation Intel® Xeon® Scalable processors.) This will also impact the application performance tuning and improve the ability to maintain predictability of the application or virtual machine execution as more of the working set may fit in the contention-resistant L2 cache, relative to the highly shared L3 cache. Clearly, this is dependent on the size of the working set; however, applications in which the working set now largely fits in the L2 and does not migrate frequently are likely to experience lower levels of contention on the latest Intel® Xeon® platforms and would rely less on CAT to guarantee predictability.

Cache Allocation Technology (CAT) introduces an intermediate construct called a **Class of Service** (CLOS or CoS), which acts as a resource control “tag” into which a thread, app, VM or any combination thereof can be grouped, and the CoS in turn has associated **Resource Capacity Bit Masks** (CBMs) which indicate how much of the cache can be used by a given CoS. Software can associate cores, applications, or Virtual Machines (VMs) with CoS as needed, and settings can be dynamically updated at any time. Typically an OS or VMM is natively enabled to use the Intel® RDT features natively, though in some cases specialized utilities may be helpful for older OS versions or for creating a simplified test environment.

Example software such as the Platform Quality of Service (PQoS) software utility (see section 2.4) provides OS command-line flags to configure the CoS and associate cores/logical threads with a CoS. Administrators or privileged users / user applications can modify CoS associations and CAT bitmasks runtime. One bit in a CoS bitmask corresponds represents a unit of capacity in the last-level cache. Cache mask size can be calculated manually with the total LLC size divided by the length of the CAT bitmask as provided in CPUID (CPUID is used to detect the Intel® RDT functionality, the number of RMIDs available for Intel® RDT monitoring, the number of COS available for allocation, etc., and further detail can be found in the Intel Architecture Software Developer Manuals referenced in Table 2.)

A server equipped with an Intel® Xeon® Scalable Gold 6230 processor features 16 Classes of Service and 11-bit capacity bitmasks (CAT mask length is 11 bits). The size of LLC per core is 1441792 bytes (1.375MiB). For example, on a 20 core processor such as the Intel® Xeon® Gold 6230 processor, the total LLC size is 28835840 bytes, and each of the total 11 CBM bits corresponds to one unit of capacity with a size of 2621440 bytes (2.5MiB) each. Additional details are provided in the upcoming sections which describe a number of uses of Intel® RDT technologies.

2.2 Cache Monitoring Technology (CMT) and Cache Allocation Technology (CAT)

Cache Monitoring Technology (CMT) is a hardware feature that allows an operating system (OS), hypervisor, or virtual machine monitor (VMM) to determine the cache usage by applications running on the platform. CMT can be used to do the following:

- Detect if the platform supports CMT monitoring capabilities via CPUID.
- Have the OS or VMM assign an ID for each application or VM that is scheduled to run on a core. This ID is called the Resource Monitoring ID (RMID).
- Monitor cache occupancy on a per-RMID basis.
- Enable an OS or VMM to read LLC occupancy for a given RMID at any time.
- Through aggregation of the data, the cache sensitivity profile of an app, thread or VM can be determined (e.g., the relationship between cache available and resulting performance).

Cache Allocation Technology (CAT) is a feature that allows an OS, hypervisor, or VMM to control allocation of a CPU's shared LLC. Once CAT is configured, the processor allows access to portions of the cache according to the established CoS. The processor obeys the CoS rules when it runs an application thread or application process. This can be accomplished by performing the following tasks:

1. Determine if the CPU supports the CAT feature.
2. Configure the CoS to define the amount of resources (cache space) available. This configuration is at the processor level and is common to all logical processors.
3. Associate each logical processor with an available CoS.
4. Run the application on the logical processor that uses the desired CoS.

With the steps above, depending on the goals, CAT can be used to minimize contention, control noisy neighbors, improve determinism, enforce SLOs or meet key performance targets. It is also possible to build software control loops which combine the features for advanced real-time performance management.

While it is recommended that most users rely on OS or VMM support to make use of the CAT feature (which enables easy thread migration, better flexibility and consistent software implementations), strictly speaking, it is possible to use CAT without OS/VMM support. In such cases, CAT can be used without any modifications to the operating system or kernel via certain command-line utilities (such as the Intel PQoS utility posted to https://github.com/01org/intel-cmt-cat) which can be used to apply settings. In such cases, through defining and assigning a class of service to each core, the user can assign portions of the LLC to particular cores by limiting the amount of the LLC into which each core is able to allocate cache lines. Because the core is only able to allocate cache lines into its assigned portion of the cache, it is no longer possible for the core to evict cache lines outside of this region.

CAT is particularly useful in scenarios where a particular "noisy neighbor" application is requesting a large amount of data, putting pressure on other applications, but never reusing the data cached in the LLC (that is, exhibiting poor temporal locality of accesses). File hosting and video streaming or transcoding programs are examples of these types of applications. On an otherwise idle system, these applications can consume the entire LLC, but not necessarily use the LLC space well as the application in question is not reusing most of the data it requests. As the program continues, it thrashes the LLC as it requests new data and evicts old data in a continuous manner, without hitting any of the data in the LLC, thus acting as a "noisy neighbor" to other applications on the platform.

Using CAT the core on which a "noisy neighbor" thread / application / VM is running can be restricted to a small region of the cache. As a result, other applications have a better opportunity to benefit from the LLC, but at the same time, there is a potential for decreasing the noisy neighbor application's performance. In scenarios such as these, a user could also configure the system such that the offending application is bound to a particular core or set of cores. It is worth noting that the CMT and MBM features can be used to track each app / thread / VM's LLC occupancy and memory bandwidth, which can make it easier to detect noisy neighbors (which will have simultaneously high cache and memory bandwidth use).

Refer to Table 2 for more information on CAT.

2.3 Intel® Resource Director Technology Software Package

The `intel-cmt-cat` software package provides basic support for CMT, MBM, CAT and future features, and includes the `pqos` utility. Refer to https://github.com/01org/intel-cmt-cat for details about CMT and CAT software packages.

After compilation, the `pqos` executable can be used to monitor the LLC occupancy, and configure the LLC allocation. The compilation and execution details are provided in the README file of the package.

The `pqos` utility provides the following command-line options:

- `./pqos -h`
  This option displays an extensive help page. Specify the `-h` option for usage details.
- `./pqos -s`
  Shows current CAT, CMT, and MBM configuration.
- `./pqos –T`
  Provides top like monitoring output.
- `./pqos –f FILE`
  Loads commands from the specified file.
- `./pqos -e CLASSDEF, --alloc-class=CLASSDEF`
  Defines allocation classes.
  `CLASSDEF` format is '`TYPE:ID=DEFINITION;`'
  Examples:
  ```
      'llc:0=0xffff;llc:1=0x00ff;llc@0-1:2=0xff00',
      'l2:2d=0xf;l2:2c=0xc',
      'mba:1=30;mba@1:3=80',
      'mba_max:1=4000;mba_max@1:3=6000'.
  ```
- `./pqos -a CLASS2ID, --alloc-assoc=CLASS2ID`
  Associates cores/tasks with an allocation class.
  `CLASS2ID` format is '`TYPE:ID=CORE_LIST/TASK_LIST`'
  Examples: `'llc:0=0,2,4,6-10;llc:1=1'`

3 The Intel-cmt-cat Utility and Example Data

3.1 Application of Technology

Intel® Resource Director Technology (Intel® RDT) has a significant impact on a variety of workloads and has special relevance to virtualized environments. With NFV-based deployments, real-time service assurance is important to ensure that services offered across the network meet the requisite SLOs with predictable performance. Service providers looking to ensure predictable deployments on COTS servers can leverage Intel® RDT with various VNF deployments and achieve the required quality of service. Intel® RDT can provide real time monitoring, control, and tracking of the cache utilization of the entire platform and help shape required quality of service for the desired VNF on the platform.

This application note focuses on the CMT and CAT features to demonstrate how to dynamically monitor the usage of LLC and allocate the shared LLC to achieve better performance. Examples are provided of how CMT and CAT can be used in NFV-based deployments, which typically include a soft switch and a data plane accelerator. The allocation of LLC capacity to CoS definitions and core to CoS association can help to achieve predictable throughput for virtualized workloads, including virtual Firewalls, virtual Load Balancers, virtual Deep Packet Inspection Systems, generic web servers, etc. that are crucial for cloud service providers and communication service providers, as shown in Figure 1. This leads to predictable performance, and helps meet SLO enhancements in key telco use cases, such as virtual Customer Premise Equipment (vCPE), virtual Evolved Packet Core (vEPC), and so forth.

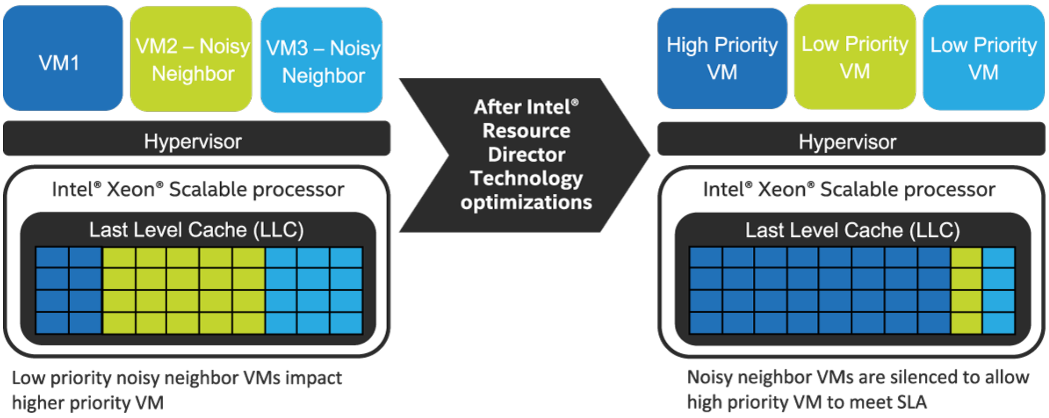
VM1
VM2 – Noisy Neighbor
VM3 – Noisy Neighbor
Hypervisor
Intel® Xeon® Scalable processor
Last Level Cache (LLC)
Low priority noisy neighbor VMs impact higher priority VM
After Intel® Resource Director Technology optimizations
High Priority VM
Low Priority VM
Low Priority VM
Hypervisor
Intel® Xeon® Scalable processor
Last Level Cache (LLC)
Noisy neighbor VMs are silenced to allow high priority VM to meet SLA

Figure 1. Cache Association to High Priority Applications with Intel® RDT

The test case, illustrated in Figure 2, showcases a typical NFV-based deployment with DPDK accelerated Open vSwitch (OvS) and a high-priority Virtual Machine (VM). The test case is based on helping the high-priority VM forwarding traffic (L2 forwarding based on DPDK) achieve predictable performance, even in a noisy environment. CAT and CMT are used to prevent aggressors (two noisy neighbor VMs) from affecting the forwarding VM's performance.

3.2 Test Setup

The test setup used for measuring the performance of the VM is shown in Figure 2 (refer to Appendix A for complete details). This setup shows an OvS with DPDK-based deployment, with the VM under test (VM1) running Layer 2 packet forwarding while the noisy neighbor VMs (VM2 and VM3) used as aggressors, run workloads to stress the memory and caches. An Ixia* traffic generator is used to generate the traffic and measure the performance.

The following workloads were used:

- The high-priority VM, pinned to three cores, was performing L2 forwarding of packets generated by the Ixia* traffic generator, via OpenvSwitch using a DPDK user space Poll Mode Driver (PMD).
- The noisy neighbors used as aggressors, VM2 and VM3 pinned to 3 cores in the same NUMA node as the high-priority VM, were running memtester, which is an effective user space tester for stress-testing the memory subsystem and as a consequence, the cache.

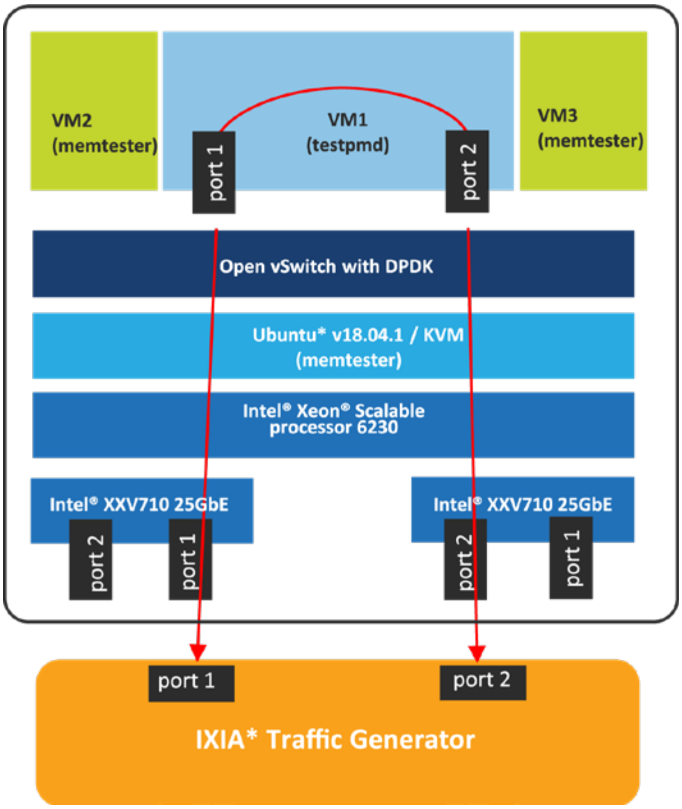

Figure 2. Test Setup with L2 Forwarding VM Workload

3.3 Cache Footprint of VMs

The performance of the VNFs and all other software components on the platform can be significantly affected by the size of the LLC. Logic needs to be applied to determine the conditions as to when a VNF will require a certain cache size, or the maximum cache size, and how much that size should be for optimal performance. In other words, one needs to be able to determine the worst case conditions in order to push the VNF or software component to require as much LLC as possible while not affecting the overall performance. Most certainly it will be required to stress the platform with a networking workload at highest throughput rate possible and with the type of traffic that would be expected to be seen in the production environment. While the platform is under the maximum sustainable stress, without any CAT settings applied, CMT can be used to measure/monitor the footprint of these components. Aggregating this data over a period of time can also help build a history of cache capacity vs. performance (thus guiding CAT settings).

In this use case, 64, 128, 256, 512, 1024, 1280 and 1518 byte size packets were injected at a rate of 25 Gbps. Using RFC2544 methodology, measurements were recorded for the throughput for the acceptable rate loss (0.01%) as well as the maximum LLC occupancy using CMT. Figure 3 shows the graphs of the LLC occupancy vs. each component. The bar colors denote the different packet sizes as per the legend. The maximum LLC occupancy for each packet size was measured using CMT for the software infrastructure components under the test case of PHY-VM-PHY with no noisy neighbor VMs.

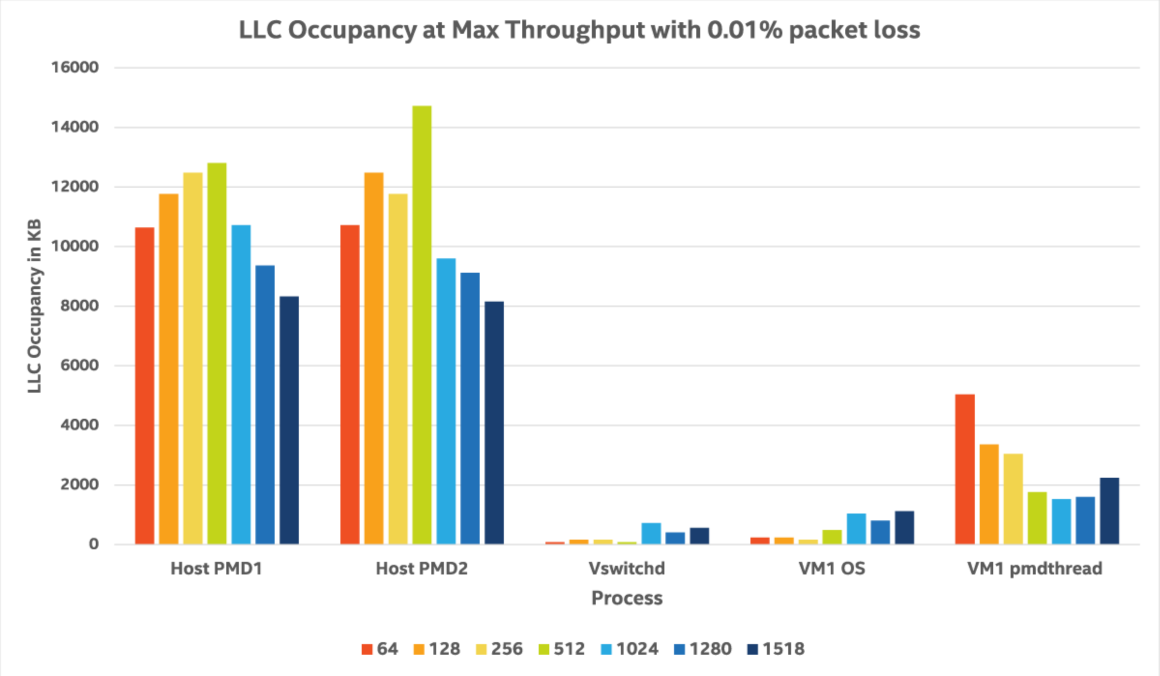

Figure 3. LLC Cache Occupancy by Processes at Maximum Throughput as per RFC2544

Based on the footprints, Table 3 shows the LLC occupancies required for the software infrastructure components for the L2 forwarding workloads shown in Figure 3.

Table 3. Measured LLC Footprint Required per Component (Maximum)

Component	LLC Footprint
vSwitchd	720 KB
PMDs in the hypervisor	15000 KB
VM (L2 forwarding)	5000 KB

3.4 Configurations

Three different configurations will be explored in our testing.

- "No CAT, without Aggressors"

This is effectively the baseline scenario without the use of CAT or any aggressor noisy neighbor virtual machines. It is intended to demonstrate the maximum performance under an optimal environment.

- "No CAT, with Aggressors"

In this scenario, two aggressor virtual machines running the memtester application that stress the LLC are added along with running memtester in the host, causing some reduction in performance. CAT is not used in this case.

- "CAT with Aggressors"

In this scenario, CAT is used to manage the cache to isolate the packet processor workloads from the aggressors in an attempt to recover performance lost due to the aggressors.

Each of these scenarios uses the same packet processing workload as shown in Figure 3. The aggressor scenarios add two additional VM's (each with 3 cores) that are intended to thrash the L3 cache by streaming to memory, acting as "noisy neighbors".

Table 4 shows the cache allocations for the "CAT with Aggressors" case, based on the LLC footprint shown in Table 3.

The 2nd Generation Intel® Xeon® Scalable processor (formerly codenamed Cascade Lake) features 16 Classes of Service and 11-bit capacity bitmasks (CAT mask length is 11 bits). For a processor with 20 cores, the cache size is 28835840 bytes and each CBM bit represents 2621440 bytes.

The `pqos` commands to define the allocation class for llc and associate the cores to them as given in Table 4. Refer to Section 2.3 for command options.

```
pqos -e "llc:0=0x7f0;llc:1=0x2;llc:2=0xc;"
pqos -a "llc:0=23-28;llc:1=29-37;llc:2=0,1,20,21"
```

Table 4. Class of Service Association for Software Components

Physical Core	Process/VM	"CAT with Aggressors" Case				Cache Allocation Scheme										
		CoS	Capacity Bit Mask (CBM)	LLC CBM Bits	LLC Size (KB)	11 bit CBM representation										
23	PMD1	0	0x7F0	7	17920	🟩 10	🟩 9	🟩 8	🟩 7	🟩 6	🟩 5	🟩 4	3	2	1	0
24	PMD2	0	0x7F0	7	17920	🟩 10	🟩 9	🟩 8	🟩 7	🟩 6	🟩 5	🟩 4	3	2	1	0
25	ovs-vswitchd	0	0x7F0	7	17920	🟩 10	🟩 9	🟩 8	🟩 7	🟩 6	🟩 5	🟩 4	3	2	1	0
26,27,28	VM1 - SUT	0	0x7F0	7	2560	🟩 10	🟩 9	🟩 8	🟩 7	🟩 6	🟩 5	🟩 4	3	2	1	0
29,30,31	VM2 - Noisy Neighbor	1	0x2	1	2560	10	9	8	7	6	5	4	3	2	🟩 1	0
32,33,34	VM3 - Noisy Neighbor	1	0x2	1	2560	10	9	8	7	6	5	4	3	2	🟩 1	0
35,36,37	Host - 3 memtesters in Hypervisor	1	0x2	1	2560	10	9	8	7	6	5	4	3	2	🟩 1	0
0,1,20,21	OS	1	0xC	2	5120	10	9	8	7	6	5	4	🟩 3	🟩 2	1	0

3.5 Relevant Benchmarks

For profiling as well as test runs, using the RFC2544 methodology, the maximum throughput was measured for the acceptable rate loss (0.01%).

Details:

- Executed the following test case with the methodology described, based on the findings in profiling:
  - PHY-VM-PHY with two VMs as noisy neighbors and running memtester in the host
  - L2 Forwarding within the VM under test
- RFC2544 benchmark methodology
- 100,000 bidirectional flows
- Three trials for each test case
- Packet sizes of 64, 128, 256, 512, 1024, 1280 and 1518 bytes
- 60-second iterations for each run, until the packet loss was less than 0.01%

The Intel PQoS tool was used to allocate LLC capacity to each CoS and associate each CoS to specific cores for the "CAT with Aggressors" case. The LLC size allocated to noisy neighbor VMs was limited to a small portion of total LLC, using CAT. This prevented noisy neighbor VMs from overusing the shared resource (LLC in this case), and enabled the high-priority VM to achieve the best performance.

- The Rx Throughput comparison of "No CAT without Aggressors", "No CAT with Aggressors", and the "CAT with Aggressors" cases for different packet sizes is shown in Figure 4.
- In the "No CAT with Aggressors" case, the throughput decreased by up to 31%, when the Open vSwitch, PMD threads, and the Virtual Machine under test were deprived of adequate LLC capacity when aggressor noisy neighbor VMs were added, as shown in Figure 4.
- In the "CAT with Aggressors" case, the throughput increased up to about 13% by using CAT to limit the LLC size for the aggressors, as shown in Figure 4.
- The selected LLC allocation and CoS to core association in the "CAT with Aggressors" CoS allocation was constructed using profiling data, as detailed in Table 3 and Table 4.

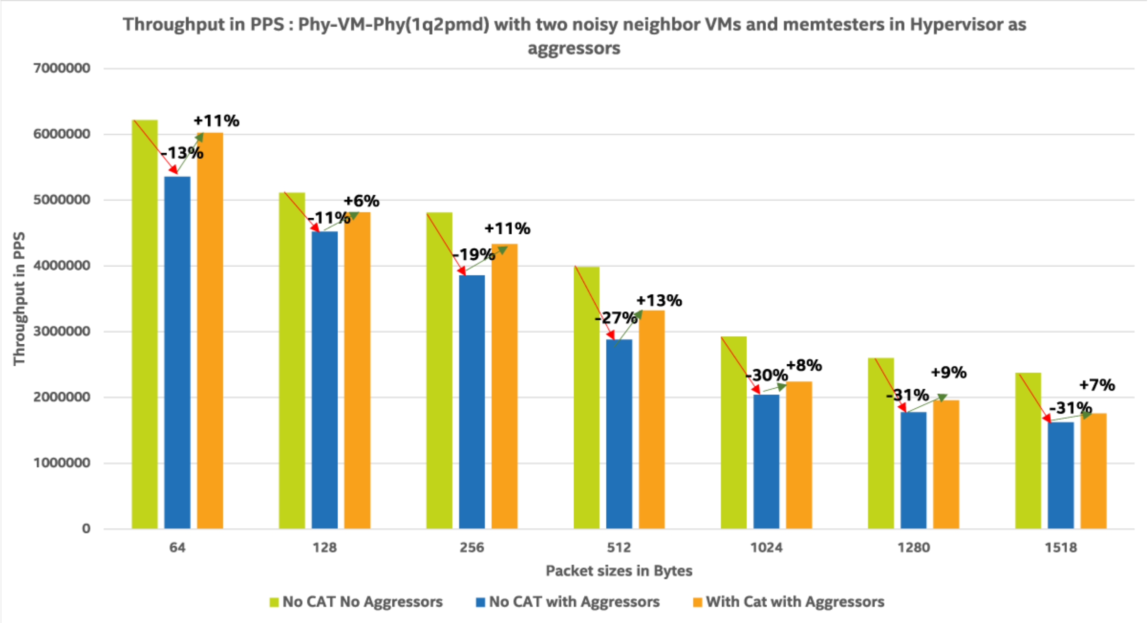

Figure 4. Rx Throughput with Different Cache Allocation Configurations

Note: 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 3/20/2019 and may not reflect all publicly available security updates. See configuration disclosure in Appendix A for details. No product or component can be absolutely secure.

3.6 Conclusion from Results

Through the application of CAT, it is possible to restore 7-11% performance that would otherwise be lost due to thread contention on the system.

It is also notable that the significant recent changes in the 2nd Generation Intel® Xeon® Scalable processor cache architecture have implications on the usage of Cache Allocation Technology, and software tuning may be helpful. Since the L2 cache size is much larger in Intel® Xeon® Gold 6230 and other 2nd Generation Intel® Xeon® Scalable processor SKUs than in previous generations of Intel® Xeon® processors and due to other optimizations, the number of L2 cache hits are generally higher, and L2 cache misses are lower for typical applications; therefore, LLC occupancy per VM, app, or thread is lower. As a result, the performance impact with Intel® RDT CAT is more significant as you increase the number of virtual machines and workloads running in the hypervisor and the virtual machines (adding more pressure on the LLC).

In such cases, CAT may provide a significant improvement in performance and help achieve required predictability for the VNFs on the platform. In addition the virtual switch and virtual machine that are involved in packet processing are primarily forwarding, there is no additional packet inspection involved. The time the packet has to remain in cache is limited and therefore the performance reduction incurred as a result of aggressors is limited. Communication and Service Provider applications will be more complex and therefore requiring the packet data to remain in cache for a longer period of time, this will increase the need for CAT. Thus, service providers can ensure predictable performance through utilizing Intel® RDT on the latest Intel® Xeon® servers when considering Virtual Network Function (VNF) deployments to achieve the required quality of service to meet the required SLOs.

This document describes one approach to maximize performance by taking advantage of Cache Allocation Technology. Further tuning may be possible with other Capacity Bit Mask schemes. There can be additional benefits when combined with other Intel® RDT features like Memory Bandwidth Allocation (MBA) in conjunction with Memory Bandwidth Monitoring (MBM).

A next step is to investigate providing support for Intel® RDT in the higher level of the software stack, such as Orchestrators and Schedulers, to automate the process of intelligently allocating shared resources like cache and memory bandwidth using the Intel® RDT features under dynamic runtime conditions.

4 Summary

Intel® Resource Director Technology (Intel® RDT) has many benefits. The key values include:

- Intel® RDT brings new levels of visibility and control over how a shared resource, such as LLC, is used by applications and virtual machines VMs.
- The Cache Monitoring Technology (CMT) feature provides improved visibility into how the last-level cache is used by apps, threads and VMs.
- The Cache Allocation Technology (CAT) feature:
  - Enables allocation of dedicated cache resources to priority applications, enabling the provider to achieve specified throughput requirements.
  - Allocates cache to VMs, threads, and applications to significantly improve throughput.
  - Protects and isolates VMs and VNFs from each other, avoiding the negative effects of intentionally malicious noisy neighbors trying to cause resource starvation or low-priority neighbors consuming large portions of shared resources, such as LLC.
- Intel® RDT, including the CMT, CAT, Memory Bandwidth Monitoring (MBM), and Memory Bandwidth Allocation (MBA), features discussed in this document can be used to achieve improved performance in a virtualized environment.

This document explained the Intel® RDT CAT and CMT features in the 2nd Generation Intel® Xeon® Scalable processors (formerly codenamed Cascade Lake) and described one approach to maximize performance by taking advantage of Cache Allocation Technology.

Appendix A Test Information

A.1 System Configuration

The tests presented in this document were conducted using an Intel® Server Board S2600WF (formerly codenamed Wolf Pass). This appendix outlines the details.

Table 5. System Configuration

Hardware Components		
Platform		**Intel® Server Board S2600WF System**
CPU	Product	Intel® Xeon® Gold 6230 CPU
	Speed	2.10 GHz Actual frequency is 2.8GHz after enabling Intel® Turbo Boost in BIOS
	Number of CPUs	2 CPUs, 20 Cores/CPU
	Last Level Cache	27.5 MB
	Max TDP (W)	125W
Memory	Vendor	Micron*
	Type	DIMM, DDR4
	Configured Speed	2666MT/s
	Part Number	18ASF2G72PDZ-2G6B1
	Size per DIMM	16 GB
NIC	Type	Intel® Ethernet Network Adapter XXV710
	Vendor	Intel
	Speed	25000 Mbps

A.1.1 Software Components

Table 6. Software Components

Host Operating System	Ubuntu* 18.04.1 LTS (Bionic Beaver) Kernel version: 4.19.0-041900-generic x86_64
VM Operating System	Ubuntu* 18.04.1 LTS Kernel version: 4.15.0-45-generic
KVM	QEMU* 2.11.1
Open vSwitch	Open vSwitch* 2.10.1 release
DPDK	DPDK version: 17.11.4-stable [same for host and guest]
NIC Driver	i40e version 2.3.2-k
NIC Firmware	6.80 0x80003d05 1.1568.0
Memtester	4.3.0

A.1.2 BIOS Settings

Table 7. BIOS Settings

Menu (Advanced)	Path to BIOS Setting	BIOS Settings	Required Setting for Deterministic Performance
Advanced	Processor Configuration	Hyperthreading	Disabled
Power Configuration	Power and Performance	CPU Power and Performance Policy	Performance
		Workload Configuration	I/O Sensitive
	Power and Performance → CPU P-State Control	Enhanced Intel® SpeedStep Technology	Enabled
	Power and Performance → Hardware P States	Hardware P States	Disabled
	Power and Performance -→ CPU C State Control	Package C-State	C0/C1 state
		C1E	Disabled
		Processor C6	Disabled
	Power and Performance →Uncore Power Management	Uncore Frequency Scaling	Disabled
		Performance P-limit	Disabled
Memory Configuration	Advanced → Memory Configuration	IMC Interleaving	2-Way Interleaving
Virtualization Configuration	Processor Configuration	Intel® Virtualization Technology (VT)	Enabled
	Processor Configuration	Intel® VT for Directed I/O	Enabled
Thermal Configuration	Advanced → System Acoustic and Performance Configuration	Set Fan Profile	Performance

A.1.3 Kernel Grub Command

The following was the Linux grub command in the host:

```
cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-4.19.0-041900-generic root=UUID=0a6acda1-a8b3-4040-9138-e59d2c67b50f ro
iommu=pt intel_iommu=on default_hugepagesz=1GB hugepagesz=1G hugepages=128 hugepagesz=2M
hugepages=2048 nohz_full=2-19,22-39 rcu_nocbs=2-19,22-39 rcu_nocb_poll=1 tsc=reliable idle=poll
irqaffinity=0 selinux=0 enforcing=0 rhgb quiet text nomodeset isolcpus=2-19,22-39
```

A.1.4 Tools

The main tool used in the testing was the PQoS utility. This software package provides basic support for Intel® RDT, including CMT, MBM, CAT/CDP, and future features.

The software programs the technologies via Model Specific Registers (MSRs) on a per core or hardware thread basis. The presence of the technologies is detected using the CPUID instruction. This software package is maintained, updated, and developed on https://github.com/01org/intel-cmt-cat.

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 3/20/2019 and may not reflect all publicly available security updates.

For more information go to www.intel.com/benchmarks

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.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

0619/DN/PTI/PDF

338932-002US