README for Intel(R) RDT Software Package

Contents

Overview

This software package provides basic support for
Intel(R) Resource Director Technology (Intel(R) RDT) including:
Cache Monitoring Technology (CMT), Memory Bandwidth Monitoring (MBM),
Cache Allocation Technology (CAT), Code and Data Prioritization (CDP)
and Memory Bandwidth Allocation (MBA).

In principle, the software programs the technologies via
Model Specific Registers (MSR) on a hardware thread basis.
MSR access is arranged via a standard operating system driver:
msr on Linux and cpuctl on FreeBSD. In the most common architectural
implementations, presence of the technologies is detected via the
CPUID instruction.

In a limited number of special cases where CAT is not architecturally
supported on a particular SKU (but instead a non-architectural
(model-specific) implementation exists) it can be detected via brand
string. This brand string is read from CPUID and compared to a table
of known-supported SKUs. If needed, a final check is to probe the
specific MSR's to discover hardware capabilities, however it is
recommended that CPUID enumeration should be used wherever possible.

From software version v1.0.0 the library adds option to use Intel(R) RDT via
available OS interfaces (perf and resctrl on Linux). The library detects
presence of these interfaces and allows to select the preferred one through
a configuration option.
As the result, existing tools like 'pqos' or 'rdtset' can also be used
to manage Intel(R) RDT in an OS compatible way. As of release v4.3.0, OS
interface became the default option. 'pqos' tool wrappers have been added to
help with the interface selection. 'pqos-os' and 'pqos-msr' for OS and MSR
interface operations respectively.

PID API compile time option has been removed and the APIs are always available.
Note that proper operation of these APIs depends on availability and
selection of OS interface.

This software package is maintained, updated and developed on
https://github.com/intel/intel-cmt-cat

https://github.com/intel/intel-cmt-cat/wiki provides FAQ, usage examples and
useful links.

Please refer to INSTALL file for package installation instructions.

Package Content

"lib" directory: \
Includes software library files providing API's for
technology detection, monitoring and allocation.
Please refer to the library README for more details (lib/README).

“lib/perl” directory: \
Includes PQoS library Perl wrapper.
Please refer to the interface README for more details (lib/perl/README).

“lib/python” directory: \
Includes PQoS library Python 3.x wrapper.
Please refer to the interface README for more details (lib/python/README.md).

"pqos" directory: \
Includes source files for a utility that provides command line access to
Intel(R) RDT. The utility links against the library and programs
the technologies via its API's.
Please refer to the utility README for more details "pqos/README".
Manual page of "pqos" utility also provides information about tool usage:
$ man pqos

"rdtset" directory: \
Includes source files for a utility that provides "taskset"-like functionality
for RDT configuration.
The utility links against the library and programs the technologies
via its API's.
Please refer to the utility README for more details "rdtset/README".
Manual page of "rdtset" utility also provides information about tool usage:
$ man rdtset

"appqos" directory:\
Includes source files for an application that allows to group apps into
priority based pools. Each pool is assigned an Intel(R) RDT and Intel(R) SST
configuration that can be set on startup or at runtime through a REST API.
Please refer to the application README for more details "appqos/README".

"appqos_client" directory:\
Includes source files for an App QoS client web application. The app
provides a simple user interface to remotely configure Intel(R) RDT and
Intel(R) SST on systems where App QoS is running.
Please refer to the application README for more details "appqos_client/README".

"examples" directory: \
Includes C and Perl examples of Intel(R) RDT usage via the library API's.
Please refer to README file for more details "examples/README".

"snmp" directory: \
Includes Net-SNMP AgentX subagent written in Perl to demonstrate the use of the PQoS library Perl wrapper API.
Please refer to README file for more details "snmp/README".

"tools" directory: \
Includes membw tool for stressing memory bandwidth with different operations.

"srpm" directory: \
Includes *.src *.rpm and *.spec files for the software package.

"ChangeLog" file: \
Brief description of changes between releases.

"INSTALL" file: \
Installation instructions.

"LICENSE" file: \
License of the package.

Hardware Support

Table 1. Intel(R) RDT hardware support

	CMT	MBM	L3 CAT	L3 CDP	L2 CAT	L2 CDP	MBA
Intel(R) Xeon(R) processor E5 v3	Yes	No	Yes (1)	No	No	No	No
Intel(R) Xeon(R) processor D	Yes	Yes	Yes (2)	No	No	No	No
Intel(R) Xeon(R) processor E3 v4	No	No	Yes (3)	No	No	No	No
Intel(R) Xeon(R) processor E5 v4	Yes	Yes	Yes (2)	Yes	No	No	No
Intel(R) Xeon(R) Scalable Processors (6)	Yes	Yes	Yes (2)	Yes	No	No	Yes (5)
Intel(R) Xeon(R) 2nd Generation Scalable Processors (7)	Yes	Yes	Yes (2)	Yes	No	No	Yes (5)
Intel(R) Atom(R) processor for Server C3000	No	No	No	No	Yes (4)	No	No

	CMT	MBM	L3 CAT	L3 CDP	L2 CAT	L2 CDP	MBA
11th Generation Intel(R) Core(TM) i3 Processors (8)	No	No	Yes	No	Yes	Yes	No
11th Generation Intel(R) Core(TM) i5 Processors (8)	No	No	Yes	No	Yes	Yes	No
11th Generation Intel(R) Core(TM) i7 Processors (8)	No	No	Yes	No	Yes	Yes	No
Intel(R) Atom(R) Processor X Series (9)	No	No	Yes	No	Yes	Yes	No
Intel(R) Xeon(R) W Processors (8)	No	No	Yes	No	Yes	Yes	No

References:

1. *Selected SKU's only:*
   - *Intel(R) Xeon(R) processor E5-2658 v3*
   - *Intel(R) Xeon(R) processor E5-2648L v3*
   - *Intel(R) Xeon(R) processor E5-2628L v3*
   - *Intel(R) Xeon(R) processor E5-2618L v3*
   - *Intel(R) Xeon(R) processor E5-2608L v3*
   - *Intel(R) Xeon(R) processor E5-2658A v3*

   Four L3 CAT classes of service (CLOS) and a set of pre-defined classes that should not be changed at run time.
   L3 CAT CLOS to hardware thread association can be changed at run time.
2. *Sixteen L3 CAT classes of service (CLOS). There are no pre-defined classes of service and they can be changed at run time.*
   L3 CAT CLOS to hardware thread association can be changed at run time.
3. *Selected SKU's only:*
   - *Intel(R) Xeon(R) processor E3-1258L v4*
   - *Intel(R) Xeon(R) processor E3-1278L v4*

   Four L3 CAT classes of service (CLOS) and a set of pre-defined classes that should not be changed at run time.
   L3 CAT CLOS to hardware thread association can be changed at run time.
4. *Four L2 CAT classes of service (CLOS) per L2 cluster.*
   L2 CAT CLOS to hardware thread association can be changed at run time.
5. *Eight MBA classes of service (CLOS). There are no pre-defined classes of service and they can be changed at run time.*
   MBA CLOS to hardware thread association can be changed at run time.
6. *Please find errata for Intel(R) Xeon(R) Processor Scalable Family at:*
   https://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/xeon-scalable-spec-update.pdf?asset=14615

7. *Please find errata for Second Generation Intel(R) Xeon(R) Scalable Processors at:* https://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/2nd-gen-xeon-scalable-spec-update.pdf
8. *Selected SKU's only:*
    - *11th Generation Intel(R) Core(TM) i3-1115GRE Processor*
    - *11th Generation Intel(R) Core(TM) i5-1145GRE Processor*
    - *11th Generation Intel(R) Core(TM) i7-1185GRE Processor*
    - *Intel(R) Xeon(R) W-11865MRE*
    - *Intel(R) Xeon(R) W-11865MLE*
    - *Intel(R) Xeon(R) W-11555MRE*
    - *Intel(R) Xeon(R) W-11555MLE*
    - *Intel(R) Xeon(R) W-11155MRE*
    - *Intel(R) Xeon(R) W-11155MLE*

    Four L3 CAT classes of service (CLOS).
    Eight L2 CAT classes of service (CLOS) per L2 cluster.
    CLOS to hardware thread association can be changed at run time.
9. *Selected SKU's only:*
    - *Intel(R) Atom(R) x6200FE Processor*
    - *Intel(R) Atom(R) x6212RE Processor*
    - *Intel(R) Atom(R) x6414RE Processor*
    - *Intel(R) Atom(R) x6425RE Processor*
    - *Intel(R) Atom(R) x6427FE Processor*

    Four L3 CAT classes of service (CLOS).
    Sixteen L2 CAT classes of service (CLOS) per L2 cluster.
    CLOS to hardware thread association can be changed at run time.

For additional Intel(R) RDT details please refer to the Intel(R) Architecture Software Development Manuals available at: https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4.html
Specific information can be found in volume 3a, Chapters 17.18 and 17.19.

OS Support

Overview

Linux is the primary supported operating system at the moment. There is a FreeBSD port of the software but due to limited validation scope it is rather experimental at this stage. Although most modern Linux kernels include support for Intel(R) RDT, the Intel(R) RDT software package predates these extensions and can operate with and without kernel support. The Intel(R) RDT software can detect and leverage these kernel extensions when available to add functionality, but is also compatible with legacy kernels.

OS Frameworks

Linux kernel support for Intel(R) RDT was originally introduced with Linux perf system call extensions for CMT and MBM. More recently, the Resctrl interface added support for CAT, CDP and MBA. On modern Linux kernels, it is advised to use the kernel/OS interface when available. Details about these interfaces can be found in resctrl_ui.txt. This software package, Intel(R) RDT, remains to work seamlessly in all Linux kernel versions.

Interfaces

The Intel(R) RDT software library and utilities offer two interfaces to program Intel(R) RDT technologies, these are the MSR & OS interfaces.

The MSR interface is used to configure the platform by programming the hardware (MSR's) directly. This is the legacy interface and requires no kernel support for Intel(R) RDT but is limited to monitoring and managing resources on a per core basis.

The OS interface was later added to the package and when selected, the library will leverage Linux kernel extensions to program these technologies. This allows monitoring and managing resources on a per core/process basis and should be used when available.

Please see the tables below for more information on when Intel(R) RDT feature (MSR & OS) support was added to the package.

Table 2. MSR interface feature support

Intel(R) RDT version	RDT feature enabled	Kernel version required
0.1.3	L3 CAT, CMT, MBM	Any
0.1.4	L3 CDP	Any
0.1.5	L2 CAT	Any
1.2.0	MBA	Any
2.0.0	L2 CDP	Any

Table 3. OS interface feature support

Intel(R) RDT version	RDT feature enabled	Kernel version required	Recommended interface
0.1.4	CMT (Perf)	4.1	MSR (1)
1.0.0	MBM (Perf)	4.7	MSR (1)
1.1.0	L3 CAT, L3 CDP, L2 CAT (Resctrl)	4.10	OS for allocation only (with the exception of MBA) MSR for allocation + monitoring (2)
1.2.0	MBA (Resctrl)	4.12	OS for allocation only MSR for allocation +

Intel(R) RDT version	RDT feature enabled	Kernel version required	Recommended interface OS
1.2.0	MBA (Resctrl)	4.12	monitoring (2)
2.0.0	CMT, MBM (Resctrl)	4.14	OS
2.0.0	L2 CDP	4.16	OS
3.0.0	MBA CTRL (Resctrl)	4.18	OS

References:

1. *Monitoring with Perf on a per core basis is not supported and returns invalid results.*
2. *The MSR and OS interfaces are not compatible. MSR interface is recommended if monitoring and allocation is to be used.*

Software dependencies

The only dependencies of Intel(R) RDT is access to C and pthreads libraries and:

- without kernel extensions - 'msr' kernel module
- with kernel extensions - Intel(R) RDT extended Perf system call and Resctrl filesystem

Enable Intel(R) RDT support in:

- kernel v4.10 - v4.13 with kernel configuration option CONFIG_INTEL_RDT_A
- kernel v4.14+ with kernel configuration option CONFIG_INTEL_RDT
- kernel v5.0+ with kernel configuration option CONFIG_X86_RESCTRL

Note: No kernel configuration options required before v4.10.

Software Compatibility

In short, using Intel(R) RDT or PCM software together with
Linux perf and cgroup frameworks is not allowed at the moment.

As disappointing as it is, use of Linux perf for CMT & MBM and
Intel(R) RDT for CAT & CDP is not allowed. This is because
Linux perf overrides existing CAT configuration during its operations.

There are a number of options to choose from in order to make use of CAT:

- Intel(R) RDT software for CMT/MBM/CAT and CDP (core granularity only)
- use Linux resctrl for CAT and Linux perf for monitoring (kernel 4.10+)
- patch kernel with an out of tree cgroup patch (CAT) and
  only use perf for monitoring (CMT kernels 4.1+, MBM kernels 4.6+)

Table 4. Software interoperability matrix

	Intel(R) RDT	PCM	Linux perf	Linux cgroup	Linux resctrl
Intel(R) RDT	Yes(1)	Yes(2)	Yes(5)	No	Yes(5)
PCM	Yes(2)	Yes	No	No	No
Linux perf	Yes(5)	No	Yes	Yes(3)	Yes

	Intel(R) RDT	PCM	Linux perf	Linux cgroup	Linux resctrl
Linux cgroup	No	No	Yes	Yes(3)	No
Linux resctrl (4)	Yes(5)	No	Yes	No	Yes

References:

1. *pqos monitoring from Intel(R) RDT can detect other pqos monitoring processes in the system. rdtset from Intel(R) RDT detects other processes started with rdtset and it will not use their CAT/CDP resources.*
2. *pqos from Intel(R) RDT can detect that PCM monitors cores and it will not attempt to hijack the cores unless forced. However, if pqos monitoring is started first and then PCM is started then the latter one will hijack monitoring infrastructure from pqos for its use.*
3. *Linux cgroup kernel patch https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt*
4. *Linux kernel version 4.10 and newer. A wiki for Intel resctrl is available at: https://github.com/intel/intel-cmt-cat/wiki/resctrl*
5. *Only with Linux kernel version 4.10 (and newer), Intel(R) RDT version 1.0.0 (and newer) with selected OS interface See '-I' option in 'man pqos' or 'pqos-os'.*

PCM is available at:
https://github.com/opcm/pcm

Table 5. Intel(R) RDT software enabling status.

	Core	Task	CMT	MBM	L3 CAT	L3 CDP	L2 CAT	MBA
Intel(R) RDT	Yes	Yes(7)	Yes	Yes	Yes	Yes	Yes	Yes
Linux perf	Yes(6)	Yes	Yes(1)	Yes(2)	No(3)	No(3)	No(3)	No
Linux cgroup	No	Yes	No	No	Yes(4)	No	No	No
Linux resctrl (5)	Yes	Yes	Yes(8)	Yes(8)	Yes	Yes	Yes	Yes(9)

Legend:

- **Core** - use of technology with core granularity
- **Task** - use of technology per task or group of tasks

References:

1. *Linux kernel version 4.1 and newer*
2. *Linux kernel version 4.6 and newer*

3. *Linux perf corrupts CAT and CDP configuration even though it doesn't enable it*
4. *This is patch and relies on Linux perf enabling*
5. *Linux kernel version 4.10 and newer*
6. *perf API allows for CMT/MBM core monitoring but returned values are incorrect*
7. *Intel(R) RDT version 1.0.0 monitoring only and depends on kernel support*
8. *Linux kernel version 4.14 and newer*
9. *Linux kernel version 4.12 and newer*

Legal Disclaimer