Intel® Virtual RAID on CPU (Intel® VROC) Integrated Caching (Intel VROC IC)

Release Notes and Quick Start Guide

November 2020

Document Number: 344452-001US

Revision History

Document Version	Description	Date
001	• Initial Release	November 2020

Contents

1 Introduction

1.1 Overview

Intel® Virtual RAID on CPU (Intel® VROC) Integrated Caching (Intel VROC IC) provide an attach point to leverage Intel® Optane™ SSDs to impact three critical server performance and TCO metrics:

- Total Storage Bandwidth
- Application Latency
- Aggregate Storage Subsystem Endurance.

To achieve desired results, recommended caching policies based on Open Cache Acceleration Software are designed to redirect write IO that are at least one of the following:

- Invalidated often (short lifetime)
- Overwritten frequently
- Accessed Often (“Hot Data”)

Intel VROC IC is delivered as Linux OS upgrade to enhances Intel VROC functionality. Intel VROC IC provides SW packages that are validated and supported for Data Center use as part of the enhanced Intel VROC feature set.

1.2 Defect Submission Support

With this release, Intel will accept, and process issues reported by customers through the Intel premier Support (IPS) portal.

To submit an issue, please use the Intel Premier Support (IPS) tool. Your local FAE can provide the necessary requirements to enable submission of an IPS issue.

http://www.intel.com/content/www/us/en/design/support/ips/training/welcome.html

1.3 Supported Operating Systems

- RHEL 7.8

1.4 Supported Platforms

- Intel® Xeon® E5/E7 SP
- Intel® Xeon® Scalable Platform family with Intel® C620 Series chipset

§

2 Release Package Contents

Note: This release of Intel VROC Integrated Caching (Intel VROC IC) comprises both binary and source code tar packages.

The RHEL 7.8 distribution already contains the required Intel VROC packages. There is no extra package to install for Intel VROC. The following files include userspace and kernel modules for Intel VROC IC cache acceleration software.

- open-cas-linux-20.03.1.0292-master.x86_64.rpm
- open-cas-linux-modules_k3.10.0_1127.el7-20.03.1.0292-master.x86_64
- open-cas-linux-20.03.1.0292-release.src.rpm
- quick_start.sh

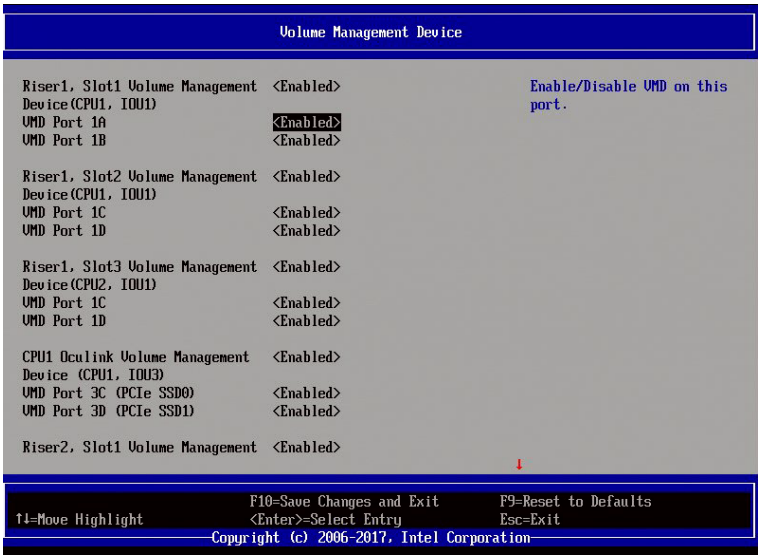
§

3 Quick Start Guide

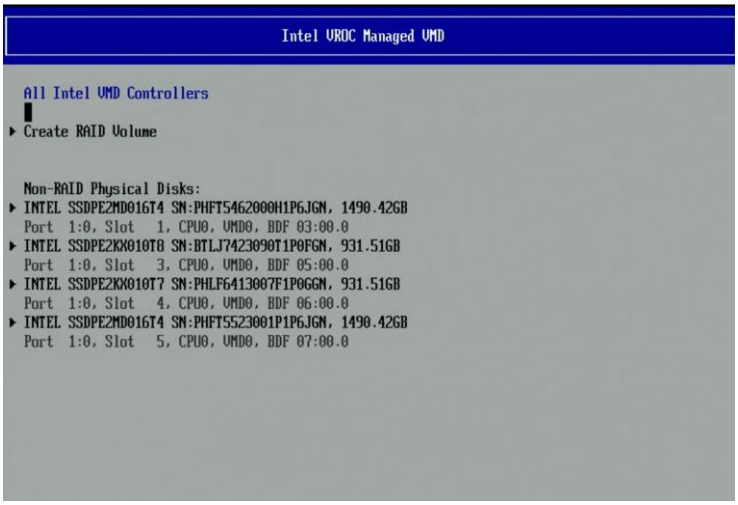
3.1 Platform Configuration – Enable VMD in HII

Enable the VMD ports in the BIOS HII menu for the connected NVMe SSDs that will be used for the Intel VROC RAID and cache devices.

The following screenshots are from an Intel server S2600WF platform BIOS. Different system BIOS may have different HII setup menu. Please contact your OEM vendor if configuration support is needed.



Intel VROC RAID volume can be created in the HII. The RAID volume can also be created in the RHEL 7.8 with the command line or quick start script. Intel VROC KEY is needed to create the Intel VROC RAID volume



3.2 Install RHEL 7.8

RHEL 7.8 can be downloaded from the Redhat website https://access.redhat.com/downloads

Follow the installation guide process to install with either DVD or USB drive

Note: The environment tested in this guide is configured with RHEL installed in a separate single SATA drive.

3.3 Installing Intel VROC IC Packages

The following steps focus on installing Intel VROC IC on a RHEL 7.8 system. Intel VROC packages, the kernel module, ledmon and mdadm userspace packages, are included inbox in the RHEL 7.8 distribution. Install the pre-build binary cache acceleration software by following these steps:

1. Copy “open-cas-linux-20.03.1.0292-master.x86_64.rpm” and “open-cas-linux-modules_k3.10.0_1127.el7-20.03.1.0292-master.x86_64” files to system.

2. Once RHEL 7.8 has been installed, reboot the system. Navigate to the directory where rpm packages are stored and install packages using the following rpm commands:

```
#sudo rpm -i open-cas-linux-modules_k3.10.0_1127.el7-20.03.1.0292-master.x86_64
#sudo modprobe cas_cache
#sudo rpm -i open-cas-linux-20.03.1.0292-master.x86_64.rpm
```

3. Issue the casadm command to confirm that installation was successful:

```
#sudo casadm -V
```

Note: *sudo is not needed if the user has root privilege.*

The followiing output indicates all of the kernel modules and CLI utility are installed with the correct version.

Name	Version
CAS Cache Kernel Module	20.03.01.00000723
CAS Disk Kernel Module	20.03.01.00000723
CAS CLI Utility	20.03.01.00000723

3.4 Configure Intel VROC IC Using Quick Start Script

Intel VROC IC release includes a quick start script (quick_start.sh) that can be used to help setup RAID and caching device configuration.

The quick start script contains several different sections including configuration, setup, sanity check and log output. User can edit the configuration section in the script file with drive devices, RAID level, chunk size and cache policy detail for system configuration setup. The quick start script will create the RAID and setup the cache configuration based on the parameters specified in the file.

Note: User must have root privilege to run the script.

1. Edit the quick_start.sh and input required parameters for configuration:

```
# This is a quick start script for Intel(R) VROC and Intel(R) CAS software
# It is divided into several parts:
# 1) Configuration
#    This section should be edited by the user to configure the script
# 2) Helper funtion - error printing and clean-up
# 3) Sanity check of system and input
# 4) VROC setup
#    4a) IMSM Container setup
#    4b) RAID Volume setup - inside of the IMSM Container
# 5) CAS setup
#    5a) If two devices are specified as "CACHE_DEVICE" - setup of RAID1 for caching device
#    5b) Setup of Cache device
#    5c) Attaching Core device to Cache device
# 6) Config files setup
#    6a) Created RAID volumes will be added to /etc/mdadm.conf (see more: man mdadm.conf)
#    6b) Creating CAS config file for created CAS devices (see more: man opencas.conf)
```

RAID configuration parameters:

- input RAID_DEVICES (identify the Core RAID member drives)
- input RAID_LEVEL
- input STRIPE_SIZE

Example:

```
RAID_DEVICES= “/dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1” or RAID_DEVICES=”/dev/sda
   /dev/sdb /dev/sdc”
RAID_LEVEL=5
RAID_STRIPE_SIZE = 16
```

```
#### RAID configuration ####

##Please specify RAID devices that you would like to use as RAID members
##example: RAID_DEVICES="/dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1"
RAID_DEVICES=""

##Please specify RAID level from 0,1,10 or 5
##example: RAID_LEVEL=5
RAID_LEVEL=

##Please specify the stripe size from 4,8,16,32,64,128
##NOTE: This parameter is ignored for RAID1 (mirroring)
##example: STRIPE_SIZE=128
STRIPE_SIZE=
```

Cache configuration parameters:

- Input CACHE_DEVICE (identify the Cache RAID member drives)
- Input CACHE_POLICY

Example:

```
CACHE_DEVICE= “/dev/nvme4n1 /dev/nvme5n1” (2 Disk RAID 1 will be created if
   identify 2 Optane drives)
CACHE_POLICY=wo
```

```
#### CAS configuration ####

##Please specify Optane SSD that you would like to use as caching device
##If two devices are passed here - they will be combined in RAID1 configuration
##example: CACHE_DEVICE="/dev/nvme4n1" or CACHE_DEVICE="/dev/nvme4n1 /dev/nvme5n1"
CACHE_DEVICE=""

##Please specify Cache policy from write-back, write-through or write-only
##Available options are: wt, wb, wa, pt, wo
## wt - Write-Through
## wb - Write-Back
## wa - Write-Around
## pt - Pass-Through
## wo - Write-Only
##For more details - please see Manual page of casadm - "man casadm"
CACHE_POLICY=
```

2. Run the quick_start.sh to setup system configuration.

```
#chmod +x ./quick_start.sh
#sudo ./quick_start.sh (sudo is not needed if logged in with root privilege)
```

Note: All of the data in the identified drives RAID devices and Cache devices will be destroyed after confirmation.

```
Summary of the configuration:
RAID devices: /dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1
RAID level: 5
RAID Stripe size: 16
Cache device: /dev/nvme6n1 /dev/nvme7n1
Cache device configuration: Two devices in RAID1
Caching policy: wo

WARNING: Proceeding will destroy the data that on above drives!
Do you like to proceed? [Y/N]
```

After running the script, console output will show RAID and cache device configuration has completed correctly.

```
Successfully added cache instance 1
Successfully added core 1 to cache instance 1
Setup Finished correctly!
Summary:
VROC RAID device (CAS core): /dev/md/Volume0
Cache device (CAS cache): /dev/md/Volume1_cache
CAS block device (CAS device): /dev/cas1-1
```

3.5 Configure Intel VROC IC Using Command Line

If the quick start script (quick_start.sh) is not used, mdadm and casadm commands can be used for manual configuration. Following is the list of the commands that the user can run in the RHEL7.8 command line terminal. Please refer to the detailed command sets in Intel VROC IC documentation listed in Section 4 of this guide.

RAID Configuration:

1. Create Core RAID Volume: (Example for 4 disk RAID5 on nvme[0-3]n1 stripe size 16K):

```
#mdadm -C /dev/md/imsm0 /dev/nvme[0-3]n1 -n4 -e imsm
#mdadm -C /dev/md/md1 /dev/md/imsm0 -l5 -n4 -c16
```

2. Create Cache RAID Volume: (Example for 2 disk RAID1 on nvme[5-6]n1):

```
#mdadm -C /dev/md/imsm1 /dev/nvme[5-6]n1 -n2 -e imsm
#mdadm -C /dev/md/md2 /dev/md/imsm1 -l1 -n2
```

Note: The re-sync will be triggered after RAID volume is created. It may take some time to complete re-sync depending on the RAID volume capacity.

3. Cache configuration: (Example for "wo" caching policy with 4K cacheline size)

```
#casadm -S -d /dev/md/md2 -c wo -x 4
#casadm -A -i 1 -d /dev/md/md1
```

4. After the configuration successfully runs, use the casadm command to list all cache instances and core devices:

```
#casadm -L
```

The output will look similar to the following screenshot:

```
type    id   disk          status    write policy   device
cache   1    /dev/md124    Running   wo             -
└core   1    /dev/md126    Active    -              /dev/cas1-1
```

3.6 Run the Simple Workload with Intel VROC IC

Following is the simple short-term flexible I/O (FIO) workload that you can use to generate 4K random write to the core and cache device, in order to verify the configuration setup. Please see the Intel VROC IC Performance Evaluation Guide for workload performance testing details.

1. List the cache status before running the FIO workload.

```
#casadm -P -i 1
```

Block statistics	Count	%	Units
Reads from core(s)	0	0.0	4KiB Blocks
Writes to core(s)	0	0.0	4KiB Blocks
Total to/from core(s)	0	0.0	4KiB Blocks
Reads from cache	0	0.0	4KiB Blocks
Writes to cache	0	0.0	4KiB Blocks
Total to/from cache	0	0.0	4KiB Blocks
Reads from exported object(s)	0	0.0	4KiB Blocks
Writes to exported object(s)	0	0.0	4KiB Blocks
Total to/from exported object(s)	0	0.0	4KiB Blocks

2. Run the simple FIO workload and check the cache status again:

```
#fio --time_based --ioengine=libaio --direct=1 --name=test --runtime=30 --
   filename=/dev/cas1-1 --rw=randwrite –random_generator=tausworthe64 --bs=4k --
   iodepth=32 --numjobs=8
#casadm -P -i 1
```

After running the FIO workload the cache block statistics will show increased count values.

Block statistics	Count	%	Units
Reads from core(s)	781	0.1	4KiB Blocks
Writes to core(s)	1311100	99.9	4KiB Blocks
Total to/from core(s)	1311881	100.0	4KiB Blocks
Reads from cache	1311100	45.0	4KiB Blocks
Writes to cache	1600188	55.0	4KiB Blocks
Total to/from cache	2911288	100.0	4KiB Blocks
Reads from exported object(s)	781	0.0	4KiB Blocks
Writes to exported object(s)	1600188	100.0	4KiB Blocks
Total to/from exported object(s)	1600969	100.0	4KiB Blocks

3.7 Cleaning the Intel VROC IC Configuration

The following steps can be used to clean the metadata, caching, Intel VROC IC setup, and to remove configuration files that are no longer needed. Refer to the detail command set in Intel VROC IC documentation and user guide referenced in Section 4 of this guide.

Stop Cache Instances:

1. To stop all configured cache instances using the opencas.conf file, execute the following command:

```
# casctl stop
```

If the operating Open CAS Linux is in write-back mode dirty data may exist within the caching environment.

2. To stop open CAS Linux, issue the following command:

```
# casctl stop –flush
```

3. Stop RAID volume/container and remove the metadata:

```
#cat /proc/mdstat (list all the RAID volumes and containers)
#mdadm -S /dev/mdXXX (stop RAID volumes and containers)
#mdadm --zero-superblock /dev/nvmeXnY (clean the imsm metadata in each drive)
```

Note: The /etc/mdadm.conf and /etc/opencas/opencas.conf configuration files are updated during the quick start script configuration. After the cleanup is complete, you may need to update according to your configuration.

§

4 Intel VROC IC Documentation and User Guide

4.1 Intel VROC User Guide

https://www.intel.com/content/dam/support/us/en/documents/memory-and-storage/ssd-software/Linux_VROC_6-0_User_Guide.pdf

4.2 Open Cache Acceleration Software User Manual

https://open-cas.github.io/guide_introduction.html

§