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

Performance Evaluation Guide

November 2020

Document Number: 344691-001US

Revision History

Revision	Description	Revision Date
001	• Initial Release	November 2020

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.

Test and System Configuration information is provided in Section 2.1.

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

Your costs and results may vary.

Contents

1 Introduction

1.1 Purpose

This Intel® VROC Integrated Caching guide is intended to help users set up Intel VROC RAID arrays and caching layers to achieve high performance results with the MongoDB application.

There are two scenarios that this guide addresses:

1. Users looking to reproduce the performance results achieved and documented by Intel
2. Users looking to implement this storage architecture on a production system

To reproduce the performance result in this guide, follow the provided steps exactly.

1.2 Platform Support

Intel® Wolf Pass server board with Intel® Xeon® Scalable Platform family and Intel® C620 series chipset.

https://ark.intel.com/content/www/us/en/ark/products/89015/intel-server-board-s2600wft.html

§

2 Evaluation Setup

This document is intended as a guide for using MongoDB with Yahoo Cloud Services Benchmark (YCSB) workload for systems containing Intel® Xeon® Scalable Processors with Red Hat Enterprise Linux (RHEL) 7.8, and the latest Intel® Virtual RAID On CPU (Intel® VROC) Integrated Caching (Intel® VROC IC) package. It includes testing of the Intel® SSD D3-S4510 and Intel® Optane™ SSD DC P4800X. Specific steps are for testing MongoDB YCSB workload with NVMe SSD RAID arrays on Intel® Volume Management Device (Intel® VMD)-enabled PCIe lane and Intel® VROC Integrated Caching (Intel® VROC IC).

Further details for each command are provided in subsequent sections of this document.

2.1 Hardware Configuration

For maximum use of Intel® VROC with availability to RAID 0, RAID 5, RAID 1, and RAID 10, an Intel VROC hardware key ("Premium" or "Intel-SSD-only") must be installed on the motherboard.

The following info is based on an Intel® Wolf Pass server board S2600WFT.

1. **Small-Capacity Performance**

   System configuration: Intel S2600WFT Platform, 2x Intel® Xeon® Gold 6240M CPU 18cores@2.60GHz, DRAM 192GB, BIOS Version: SE5C620.86B.02.01.0010.010620200716
   OS: RedHat Enterprise Linux v7.8, 3.10.0-1127.el7.x86_64, mdadm - v4.1 - 2018-10-01, THP disabled
   BIOS setting: PackageC-State(C6), HardwareP-States(Native Mode), CPU Power and Performance Policy (Balanced Performance)
   Intel VROC (SATA RAID) 4-Disk RAID5, chunk size=8k, group_thread_cnt=8
   Intel VROC Storage: 4x Intel® SSD D3-S4510 1.92 TB (Model: SSDSC2KB019T8) with Intel VROC RAID5 chunk size=8k, group_thread_cnt=8, Intel VROC Integrated Caching with 2x100 GB Intel® Optane™ SSD DC P4801X Series (Model: SSDPEL1K100GA) or 2x 375 GB Intel® Optane™ SSD DC P4800X Series (Model: SSDPE21K375GA) with Intel VROC RAID1, Write Only Mode, 4k Cache Line Size
   RAID HBA Storgae: 4x Intel® SSD D3-S4610 1.92 TB (Model: PHYG811500101P9DGN) with Intel RAID Adapter RSP3TD160F RAID5
   MySQL Benchmark: Sysbench 1.1.0-bd4b418, oltp_read_write, MySQL 8.0.21, 120 GB Database, 1hr test, 64 threads
   Performance results are based on testing as of 8/12/2020 and may not reflect all publicly available security updates

2. **Mid-Capacity Performance**

   System configuration: Intel S2600WFT Platform, 2x Intel(R) Xeon(R) Gold 6240M CPU 18cores@2.60GHz, DRAM 192 GB, BIOS Version: SE5C620.86B.02.01.0010.010620200716
   OS: RedHat Enterprise Linux v7.8, 3.10.0-1127.el7.x86_64, mdadm - v4.1 - 2018-10-01, THP disabled
   BIOS setting: PackageC-State(C6), HardwareP-States(Native Mode), CPU Power and Performance Policy (Balanced Performance)
   Intel VROC (SATA RAID) 8-Disk RAID5, chunk size=8k, group_thread_cnt=8
   Intel VROC Storage: 8x Intel® SSD D3-S4510 1.92 TB (Model: SSDSC2KB019T8) with Intel VROC RAID5 chunk size=8k, group_thread_cnt=8, Intel VROC Integrated Caching with 2x100 GB Intel® Optane™ SSD DC P4801X Series (Model: SSDPEL1K100GA) or 2x 375GB Intel® Optane™ SSD DC P4800X Series (Model: SSDPE21K375GA) with Intel VROC RAID1, Write Only Mode, 4k Cache Line Size
   RAID HBA Storgae: 8x Intel® SSD D3-S4610 1.92 TB (Model: PHYG811500101P9DGN) with Intel RAID Adapter RSP3TD160F RAID5
   MySQL Benchmark: Sysbench 1.1.0-bd4b418, oltp_read_write, MySQL 8.0.21, 120 GB Database, 1hr test, 64 threads
   Performance results are based on testing as of 8/12/2020 and may not reflect all publicly available security updates

While these steps are based on the Intel® Wolf Pass server board, they may be replicated on other platforms with the directions followed as closely as possible. For more details see Section 1.2, Platform Support.

Note: For best results, Intel recommends populating all DIMM channels with applicable RAM (same manufacturer, same model, same size, same speed, etc.).

2.2 BIOS Settings

The following BIOS settings are based on an Intel® Wolf Pass server board S2600WFT and are recommended for maximum performance. Please refer to the instructions that have been supplied by the user's platform BIOS vendor, as those instructions may differ from the set below.

Enter the BIOS at boot up by pressing <F2>. Load the defaults by pressing <F9>. After loading the defaults in the BIOS, navigate to the following:

Verify… Advanced → Processor Configuration → Intel® Hyper-Threading Tech → Enabled
Change… Advanced → Power & Performance → CPU Power and Performance Policy → Performance
Change… Advanced → Power & Performance → Workload Configuration → I/O Sensitive
Verify… Advanced → Power & Performance → Uncore Power Management → Uncore Frequency Scaling → Enabled
Verify… Advanced → Power & Performance → Uncore Power Management → Performance P-limit → Enabled
Verify… Advanced → Power & Performance → CPU P State Control → Enhanced Intel SpeedStep® Tech → Enabled
Verify… Advanced → Power & Performance → CPU P State Control → Intel Configurable TDP → Disabled
Verify… Advanced → Power & Performance → CPU P State Control → Intel® Turbo Boost Technology → Enabled
Verify… Advanced → Power & Performance → CPU P State Control → Energy Efficient Turbo → Enabled
Verify… Advanced → Power & Performance → Hardware P States → Hardware P-States → Native Mode
Verify… Advanced → Power & Performance → Hardware P States → HardwarePM Interrupt → Disabled
Verify… Advanced → Power & Performance → Hardware P States → EPP Enable → Enabled
Verify… Advanced → Power & Performance → Hardware P States → APS rocketing → Disabled
Verify… Advanced → Power & Performance → Hardware P States → Scalability → Disabled
Verify… Advanced → Power & Performance → Hardware P States → PPO-Budget → Disabled
Change… Advanced → System Acoustic and Performance Configuration → Set Fan Profile → Performance

With Intel VROC, Intel VMD will also need to be enabled on your new platform. The information below is based on a 4-port retimer (x16) installed on Riser 2, PCIe slot 1; your setup may be different. Please enable VMD and their respective ports at the applicable Riser/PCIe Slot in your configuration.

Note: *8-port switches (x8) have only two VMD ports.*

Change… Advanced → PCI Configuration → PCIe Slot Bifurcation Setting → Riser_Slot_2 Bifurcation → x4x4x4x4
Change… Advanced → PCI Configuration → Volume Management Device → Riser2, Slot1 Volume Management → Enabled
Change… Advanced → PCI Configuration → Volume Management Device → VMD Port 2A → Enabled
Change… Advanced → PCI Configuration → Volume Management Device → VMD Port 2B → Enabled
Change… Advanced → PCI Configuration → Volume Management Device → VMD Port 2C → Enabled
Change… Advanced → PCI Configuration → Volume Management Device → VMD Port 2D → Enabled

2.3 Intel® Virtual RAID On CPU (from the BIOS)

The following info is based on an Intel® Wolf Pass server board S2600WFT. For other platforms, accessing Intel Virtual RAID On CPU in the BIOS may differ from below.

After adjusting the BIOS settings as described in Section 2.2 above, reboot the system, and enter the BIOS again by pressing <F2>.

Navigate to Advanced → PCI Configuration → UEFI Option ROM Control → Intel® Virtual RAID on CPU

From here, you can see the Intel VROC hardware key installed, Intel® VROC Pre-OS version number, and any existing Intel VROC RAID volumes. Select "All Intel VMD Controllers" to view the applicable NVMe SSDs.

2.4 Red Hat Enterprise Linux Installation/Dependencies

All Red Hat Enterprise Linux installations must be done in UEFI mode.

After entering the "INSTALLATION SUMMARY" screen, the "Minimal Install" option under "SOFTWARE SELECTION" is sufficient for this performance testing.

Once installation is complete, set up a repository containing all the necessary rpm packages, and install the following dependencies by running the following command:

```
#yum install gcc libaio-devel zlib-devel unzip sysstat libreport-filesystem numactl
    redhat-lsb-core sg3_utils nvme-cli iotop hdparm wget htop java maven –y
```

Note: Please refer to the Redhat website for more details on configuring the repository: *https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/sec-configuring_yum_and_yum_repositories*

Note: User must have root privilege to install necessary rpm packages.

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

2.5 Installing Intel® VROC Integrated Caching (Intel® VROC IC)

Intel VROC packages, the kernel module, ledmon and mdadm userspace packages, are included inbox in the RHEL 7.8. Follow the steps below to install the cache acceleration software in RHEL 7.8 for Intel VROC IC.

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. Use **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 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

2.6 Downloading and Installing Intel® Memory and Storage Tool

Intel® Memory and Storage Tool (Intel® MAS) is used to update firmware on Intel SSDs, and is used to perform a low-level format on Intel NVMe SSDs.

1. Download Intel® MAS at:
   https://downloadcenter.intel.com/download/29821?v=t

   ```
   #unzip Intel®_MAS_CLI_Tool_1.2_Linux.zip
   #rpm -i intelmas-1.2.79-0.x86_64.rpm
   ```

2. Verify Intel® MAS installation by running the intelmas command:

   ```
   #intelmas version
   ```

3. To list Intel SSD devices in Intel® MAS:

   ```
   #intelmas show -intelssd
   ```

2.7 Setting System to Performance Mode

Setup the system in performance mode with the systemd service.

1. First create and update the rc.local script into /etc/rc.d then restart the systemd rc-local service.

   ```
   #vi /etc/rc.d/rc.local
   ```

2. The content of the rc.local:

   ```
   #!/bin/bash
   cpupower frequency-set –g performance
   ```

3. Run/restart the systemd service using systemctl:

   ```
   #chmod 755 /etc/rc.d/rc.local
   #systemctl enable rc-local.service
   #systemctl start rc-local.service
   #systemctl status rc-local.service
   ```

2.8 Test Drive Configuration and Preparation

2.8.1 Formatting the NVMe Intel SSDs

Drives with previous RAID metadata, or that were previously used for other applications, could have diminished performance. Clearing the metadata and erasing or performing a low-level format can stabilize your drives providing more consistent results.

1. To remove the RAID metadata:

   ```
   #mdadm –-zero-superblock /dev/nvme*n1
   ```

2. To show Intel SSD in Intel MAS indexes:

   ```
   #intelmas show -intelssd
   ```

3. To perform a low-level format:

   ```
   #intelmas start -force -nvmeformat -intelssd <intelmas index number>
   ```

Note: Repeat these steps for all SSDs to be used in new array.

2.8.2 Setup RAID Volume and Cache Device

1. Create the Core 4 Disk RAID5 Volume:

```
#mdadm -C /dev/md/imsm0 /dev/nvme[0-3]n1  -n4 -e imsm  (example to create container
    with 4 disk nvme[0-3]n1)

#mdadm -C /dev/md1  /dev/md/imsm0 -l5 -n4 -c8  (example to create 4Disk RAID5 with
    chunk size 8k)
```

2. Create the cache 2 Disk RAID1 volume:

```
#mdadm -C /dev/md/imsm1  /dev/nvme[4-5]n1 -n2 -e imsm  (example to create container
    with 2 disk nvme[4-5]n1)

#mdadm -C /dev/md2 /dev/md/imsm1 -l1 -n2  (example to create 2Disk RAID1 for
    caching volume)
```

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. Change the group_thread_cnt for the performance.

   After creating a RAID5 volume, there is a setting that has demonstrated slightly improved RAID5 performance. This setting is not found with RAID 0, RAID 1, or RAID 10. Apply the following setting to a RAID 5 volume:

```
#echo 8 > /sys/block/md1/md/group_thread_cnt   (assume md1 is the RAID5 volume)
```

Note: A group thread count of 0 is the default and limited testing has been conducted with other group thread counts. A group thread count of 4 was determined to be the most efficient, boosting performance ~10%. More testing is planned to determine optimal performance and latency impacts. Larger group thread counts appear to impact read performance.

4. Change the sync_speed_max for initialization speed.

   After creation of RAID 5, RAID 1, or RAID 10 volumes, initialization will start automatically.
   To speed up initialization or rebuild times, the following setting can be applied.

```
#echo 1000000 > /sys/block/md1/md/sync_speed_max   (assume md1 is the RAID5 volume)
```

Note: The default setting for "sync_speed_max" is 200,000.

2.8.3 Create Filesystem and Mount the Storage Node

1. Create the filesystem with xfs on created RAID5 volume and mount for core RAID volume:

```
#mkfs.xfs /dev/md1    (assume md1 is the RAID5 volume)
#mkdir -p /mnt/raid
#mount /dev/md1 /mnt/raid
```

2. Update the /etc/fstab with UUID to mount the drive in the system reboot to list all the UUID in the sysfs:

```
#lsblk -f
```

3. Edit the /etc/fstab as in the following example:

```
UUID=c346151a-dedc-41c4-b362-97033c69def6 /mnt/raid     xfs    defaults    1 2
```

§

3 Preparing MongoDB

3.1 Download and Install MongoDB

Download and install MongoDB by following the steps in this link:
https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/

3.2 Download YCSB

The YCSB project is to develop a framework and common set of workloads for evaluating the performance of different "key-value" and "cloud" serving stores. Download YCSB benchmark tool and set up database for testing by following the readme in the link https://github.com/brianfrankcooper/YCSB

Note: User can skip the YCSB readme guide which contain the MongoDB setup in the section 3.1 above.

3.3 Disable Transparent Huge Pages (THP)

Transparent Huge Pages (THP) is a Linux memory management system that reduces the overhead of Translation Lookaside Buffer (TLB) lookups on machines with large amounts of memory by using larger memory pages. However, database workloads often perform poorly with THP enabled because they tend to have sparse rather than contiguous memory access patterns. When running MongoDB on Linux, THP should be disabled for best performance.

Follow the steps in this link to disable the THP:
https://docs.mongodb.com/manual/tutorial/transparent-huge-pages/

3.4 Configure SELinux

The current SELinux Policy does not allow the MongoDB process to access /sys/fs/cgroup, which is required to determine the available memory on your system. If you intend to run SELinux in enforcing mode, you will need to make the following adjustment to your SELinux policy https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/

Another option is for the user to disable the SELinux (only for testing) by following these steps:

1. Check and disable SELinux
   Check if SELinux is disabled in the system

   ```
   #cat /etc/sysconfig/selinux
   ```

   The following output shows SELinux is disabled:

   ```
   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   #     enforcing - SELinux security policy is enforced.
   #     permissive - SELinux prints warnings instead of enforcing.
   #     disabled - No SELinux policy is loaded.
   SELINUX=disabled
   # SELINUXTYPE= can take one of three two values:
   #     targeted - Targeted processes are protected,
   #     minimum - Modification of targeted policy. Only selected processes are protected.
   #     mls - Multi Level Security protection.
   SELINUXTYPE=targeted
   ```

If SELinux is not disabled, follow these steps to disable the SELinux:

2. Edit the sysconfig file and add “SELINUX=disabled” in the file

   ```
   #vi /etc/sysconfig/config
   ```

3. Edit grub configure file, In the GRUB_CMDLINE_LINUX parameter add “selinux=0”

   ```
   #vi /etc/default/grub
   ```

4. Rebuild the grub configure by run grub2-mkconfig

   ```
   #grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
   ```

5. Reboot system and check again if SELinux is disabled in the system.

3.5 Setup MongoDB Configuration

1. Create mount point for MongoDB:

   ```
   #mkdir -p /mnt/raid/mongo/
   #chown mongod:mongod /mnt/raid/mongo
   ```

2. Modify the MongoDB configuration file mongod.conf:

   ```
   #vi /etc/mongod.conf  (modify the changes in bold)
   ```

3. Change the content of the bold text in the mongod.conf:

   ```
   # where to write logging data.
      systemLog:
      destination: file
      logAppend: true
      quiet: true
      path: /var/log/mongodb/mongod.log

   # Where and how to store data.
   storage:
      dbPath: /mnt/raid/mongo
      journal:
        enabled: true
   # engine:
      wiredTiger:
        engineConfig:
          journalCompressor: none
          directoryForIndexes: true
   ```

Note: It is important to follow exact indent. Indent may affect how mangod.conf can be parsed

4. Configure mongod service in systemd:

   ```
   #vi /usr/lib/systemd/system/mongod.service
   ```

5. Minor change with the numctl and timeout setting:

   ```
   ExecStart=/usr/bin/numactl --interleave=all /usr/bin/mongod $OPTIONS
   TimeoutSec=900
   ```

6. Reload the mongodb service:

   ```
   #systemctl daemon-reload
   #systemctl start mongod
   ```

7. Create a user for benchmarking in the MongoDB command console.
   Run the MongoDB command shell:

   ```
   #mongo
   ```

8. In the MongoDB command shell, run the commands shown in bold characters below:

   ```
   MongoDB shell version v4.2.5
   ….
   > use admin
   switched to db admin
   > db.createUser(
   ... {user: "mongoAdmin",
   ... pwd: "MYPASSWORD",
   ... roles:[{role: "userAdminAnyDatabase" , db:"admin"}]})
   ```

   The output after creating user in the MongoDB command shell:

   ```
   Successfully added user: {
           "user" : "mongoAdmin",
           "roles" : [
                   {
                           "role" : "userAdminAnyDatabase",
                           "db" : "admin"
                   }
           ]
   }
   ```

9. Create database.

   The following example is to create a 2 TB database. The database will be created in the /mnt/raid on core volume device. Run the ycsb command in the ycsb folder to create the database:

   ```
   #./bin/ycsb load mongodb -s -P workloads/workloada -p recordcount=1600000000 -threads
      32 -p mongodb.url="mongodb://mongoAdmin:MYPASSWORD@localhost:27017/admin"
   ```

Note: The newer ycsb version may use the ycsb.sh instead of ycsb

3.6 Configure and Run the Caching

1. Unmount /mnt/raid:

   ```
   #systemctl stop mongod
   #umount /mnt/raid
   ```

2. Start caching:

   ```
   #casadm -S -d /dev/md126 -c wo -x 4   (md126 is the cache optane 2 disk RAID1 volume)
   #casadm -A -i 1 -d /dev/md124   (md124 is the core 4 disk RAID5 volume)
   #mount -o discard /dev/cas1-1 /mnt/raid
   #systemctl start mongod
   ```

Note: The MongoDB testing was performed using the 4K cacheline size with 4DR5 on 8K chunk size. Performance may vary with different numbers of RAID member drives, different cacheline size and chunk size configurations.

3. Load the io classfication file.
   Create the ioclass.csv as follow content in /etc/opencas/ioclass_mongo.csv:

   ```
   IO class id,IO class name,Eviction priority,Allocation
   0,unclassified,1,1
   1,directory:/mnt/raid/mongo/journal,,1
   ```

4. Load the ioclass file:

   ```
   #casadm -C -C -i 1 -f /etc/opencas/ioclass_mongo.csv
   ```

3.7 Run MongoDB Benchmark

Use the “ycsb run” command to run the benchmark workload. It will take time to complete the benchmark depending on the operationcount parameter in the command:

```
#./bin/ycsb run mongodb -s -P workloads/workloada -p recordcount=1600000000 -threads
   32  -p mongodb.url="mongodb://mongoAdmin:MYPASSWORD@localhost:27017/admin?j=true" -
   p operationcount=200000000
```

§

4 Test Results

Intel VROC IC for MongoDB with Small-Capacity Performance (SATA SSD)

MongoDB					
Metric	Legacy HBA	Intel VROC IC (100GB)	%	Intel VROC IC (375 GB)	%
Ops/s	9,892 ops	11,912 ops	↑20%	14,112 ops	↑43%
Avg. Update Latency	5,701 us	4,425 us	↓22%	3,676 us	↓36%
Storage Lifetime Ops.	2,389B	3,443B	↑44%	5,554B	↑132%

Intel VROC IC for MongoDB with Mid-Capacity Performance (SATA SSD)

MongoDB					
Metric	Legacy HBA	Intel VROC IC (100GB)	%	Intel VROC IC (375 GB)	%
Ops/s	5,664 ops	13,340 ops	↑136%	15,099 ops	↑167%
Avg. Update Latency	10,673 us	3,998 us	↓63%	3,460 us	↓68%
Storage Lifetime Ops.	4,211 B	3,408B	↓19%	5,542B	↑32%

The achieved results, presented above, are based on the documented steps, and the configuration detail provided in Section 2.1, Hardware Configuration.

Results may vary based on, among other things, different hardware and software configuration.

§