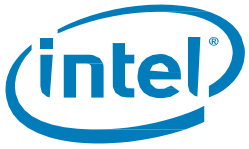
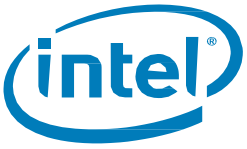


Intel® Virtual RAID on CPU (Intel® VROC) with Intel® Volume Management Device (Intel® VMD) Linux PV Release for RHEL

Release Notes

November 2020
Version 1.0



Legal Notices and Disclaimers

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

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.

Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown". Implementation of these updates may make these results inapplicable to your device or system.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

All documented performance test results are obtained in compliance with JESD218 Standards; refer to individual sub-sections within this document for specific methodologies. See www.jedec.org for detailed definitions of JESD218 Standards. Intel does not control or audit the design or implementation of third party benchmark data or Web sites referenced in this document. Intel encourages all of its customers to visit the referenced Web sites or others where similar performance benchmark data are reported and confirm whether the referenced benchmark data are accurate and reflect performance of systems available for purchase.

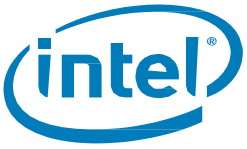
The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

For copies of this document, documents that are referenced within, or other Intel literature please contact you Intel representative.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Driver Revision History

Date	Driver Version	Description
July 2019	• Intel VMD v2.0 Alpha for VROC in RHEL8.0	• Intel VMD-enabled NVMe driver v 2.0 Alpha Release
October 2019	• Intel VROC and VMD v2.0 Beta driver for RHEL8.0	• Intel VROC and VMD-enabled NVMe driver 2.0 Beta Release
Mar 2020	• Intel VROC and VMD v2.5 Beta driver for RHEL8.1	• Intel VROC V7.5 and VMD-enabled NVMe driver 2.5 Beta release
September 2020	• Intel VROC 7.5 and VMD v2.5 PC driver for RHEL8.1 and RHEL7.8	• Intel VROC V7.5 and VMD-enabled NVMe driver 2.5 PC release
November	• Intel VROC 7.5 and VMD v2.5 PC driver for RHEL7.9 and SLES12 SP5	• Update RHEL7.9 and SLES12 SP5 VMD driver



Contents

1 Introduction

1.1 Overview

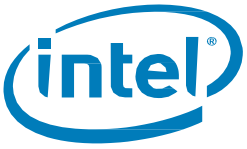
The Intel® Volume Management Device (Intel® VMD) – Enabled NVMe Driver release package contains the 2.5 build release version of the Intel VMD-Enabled NVMe Driver to support Whitley Ice Lake platforms for using VROC.

Intel VMD-enabled NVMe driver assists in the management of CPU and PCH attached PCIe NVMe SSDs.

1.2 New in This Release

Intel VMD NVMe driver version 2.5 support CPU attached NVMe RAID and introduces NVMe RAID 1 feature for PCH attached NVMe. Please see limitation section 3.3.1 below. Mdadm and Ledmon are the tool to manage the RAID volume and LED status. The PV release includes update version on mdadm and ledmon. Intel VROC ledmon support PCIe switch attached Native PCIe Enclosure Management (NPEM) LED management.

Note: When VMD is disabled, the single binary will only load on Intel SSD PCIe NVMe devices (No RAID when VMD is disabled; pass-thru only).



2 Support

2.1 Supported Operating Systems

- RHEL 7.9

2.2 Supported Platforms

- **This release is for operation on Whitley Ice Lake only**

2.3 Supported Configurations

- Up to 1 level deep switch
- Up to 24 PCIe NVMe SSDs

2.4 Supported PCIe NVMe SSD

Intel VMD-enabled NVMe driver supports most shipping enterprise Intel NVMe SSDs as well as 3rd Party NVMe SSDs.* Please see your Intel AE for the most current list of supported / validated 3rd party PCIe NVMe SSDs.

***Intel NVMe Dual controller devices not supported in this release (DC P3608)**

3 Release Package Contents

3.1 Release Component

VROC 7.5 PV release package includes ISO image:

- iavmd-1.0.0.1451-rhel_79-dud.iso md5: 81289938e81c71ae51180f032ebf49c4

This ISO image has VMD and VROC binary and source packages, here are the contents

Intel VMD-enabled NVMe driver for RHEL 7.9

- kmod-iavmd-1.0.0.1382-rhel_78.x86_64.rpm
- iavmd-1.0.0.1382-rhel_78.src.rpm

3.2 Limitations

3.2.1 Not supported in this PV Release

- For testing VMD behind all NVMe devices (CPU and PCH attached):

  VMD should be enabled only on PCH ports 8 through 12, disable all other PCH ports

- Surprise Hot-plug for PCH NVMe is not supported.

3.2.2 Intel® Virtual RAID on CPU (Intel® VROC) HII Menu in Pass Thru mode

Intel VMD and Intel VROC UEFI drivers are packaged together. Intel VMD UEFI driver enumerates and assigns resources for all NVMe devices under the root port. The Intel VROC UEFI driver exposes those devices to the system.

Due to this packaging, the devices in the UEFI HII Bios menu will be found under the Intel® Virtual RAID on CPU (Intel® VROC) HII menu when Intel VMD is enabled.
Intel VROC in Pass Thru mode is seen so that the user knows that NVMe RAID is not supported when Intel VROC is in pass thru mode.

4 Release Package Installation

For the VROC supported Linux distribution, VMD driver, inbox mdadm and ledmon drivers may not be up to date versions. VMD v2.5 support with minor improvements and bug fixes are in this replacement driver. For this reason, it's recommended to *blacklist* the inbox driver and use the Intel driver instead. And update mdadm and ledmon in the release package.

If you are Installing Linux distribution to NVMe drive behind VMD, please do the below steps manually to blacklist "vmd" blacklisting the inbox driver involves either adding initcall_blacklist=vmd_drv_init to *Boot Options* in first installation menu screen **or** doing the same to an *existing* installation by appending initcall_blacklist=vmd_drv_init to the GRUB_CMDLINE_LINUX_DEFAULT in /etc/default/grub file, updating grub, then rebooting the system.

If you choose to install binary RPM "kmod-iavmd-*.rpm", no need to blacklist manually. RPM's post installation script will automatically blacklist and update the grub with iavmd.

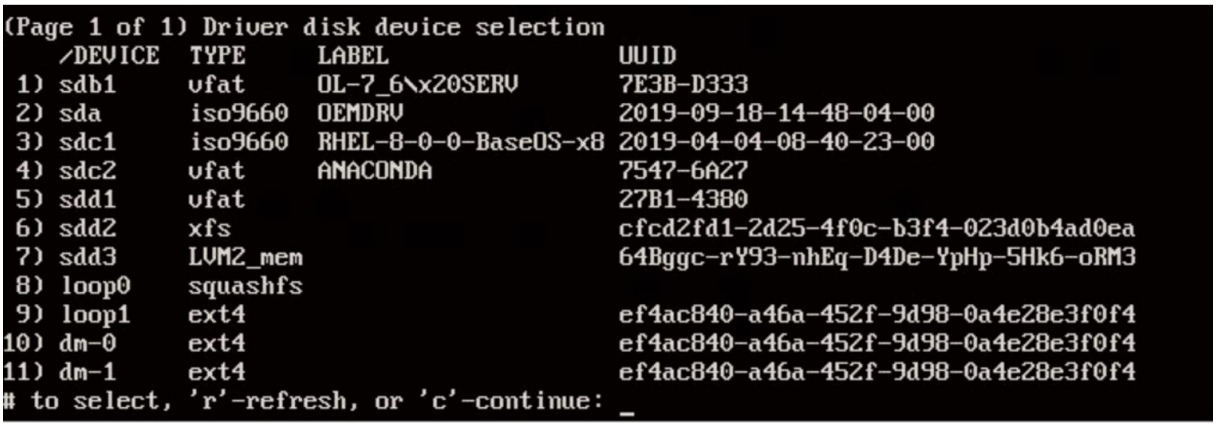
4.1.1 Installation

1. Download and mount the iso, copy binary RPMpackages to your target folder
2. Open a *terminal* and cd to the directory containing the RPM installer
3. Issue the following command: sudo rpm -i kmod-iavmd-*.x86_64.rpm. No output is expected from this
4. Run rpm -q kmod-iavmd. This will show whether the prior step was successful
5. Upon installing "kmod-iavmd-*.rpm",must reboot the system, so that "iavmd" will automatically load and bind to any active VMD controllers.
6. Install mdadm and ledmon(RHEL8.1 as example)
   rpm -i ledmon-0.94-RHEL_8.1_INT_015.01.1.1.1359.el8.x86_64.rpm
   rpm -i mdadm-4.1-RedHat_8.0_64_SMP_2020_WW28.1_06_13_50.x86_64.rpm
7. If you haven't already enabled VMD hardware for the PCIe ports you're targeting, run sudo reboot and enable VMD from the BIOS. Once Linux reboots, the VMD driver will automatically load and bind to any active VMD controllers.
8. Run lsmod | grep iavmd to show if the previous step had the intended effect.

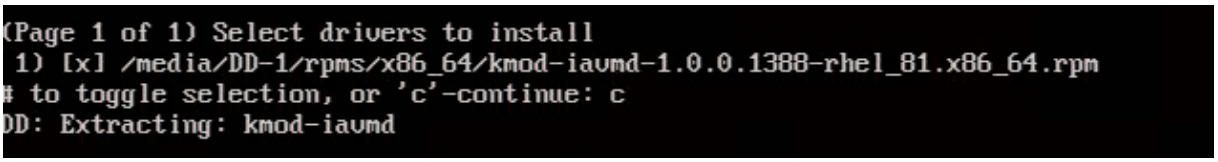
4.1.2 Installation with anaconda installer

1. Download/copy the ISO package to your target system
2. Use dd dump the ISO file to USB drive eg: dd if= vroc_7.5_RHEL_8.1_PC.iso of=/dev/sdX (RHEL8.1 as example)
3. Plug in the USB drive to the system and start the RHEL installation
4. Edit the boot command line “inst.dd” , “initcall_blacklist=vmd_drv_init”
5. Select OEMDRV in the installation menu
6. Complete the RHEL installation by following installer steps
7. After installation complete, reboot the system and add boot command line “initcall_blacklist=vmd_drv_init to grub configure file
8. Mount the ISO and Install mdadm and ledmon(RHEL8.1 as example)
   rpm -i ledmon-0.94-RHEL_8.1_INT_015.01.1.1.1359.el8.x86_64.rpm
   rpm -i mdadm-4.1-RedHat_8.0_64_SMP_2020_WW28.1_06_13_50.x86_64.rpm

Installer will pop-out for user to select the kmod-iavmd driver in OEMDRV



Select the kmod-iavmd-1.0.0.1388-rhel1_81.x86_64.rpm (For RHEL8.1as example)

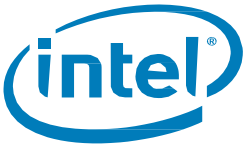


4.1.3 Removal

1. (Optional, but recommended) Reboot the system and disable any VMD hardware from the BIOS
2. Open a *terminal* and run `rpm -q kmod-iavmd` to find the name of the previously installed binary RPM package. E.g., `kmod-iavmd`
3. Call `sudo rpm -e kmod-iavmd` using the result of the last command.
4. Remove ledmon and mdadm (RHEL8.1 as example)

   `rpm -e ledmon-0.94-RHEL_8.1_INT_015.01.1.1.1359.el8.x86_64`

   `rpm -e mdadm-4.1-RedHat_8.0_64_SMP_2020_WW28.1_06_13_50.x86_64`
5. reboot the system.
6. Run `lsmod | grep iavmd` to show if the VMD driver is correctly *not* running



5 Known issues

id	title	Priority
18011200447	Migration hangs after reboot - Ubuntu 20.04 / RHEL 7.8 / RHEL 8.1 - Purley-R / Cooper City / Wilson City Workaround: use #mdadm --grow --continue /dev/mdX{container} to manually continue migration	P3
18011560190	(Olce Chkp) Olce reshape hangs after reboot - Ubuntu 20.04/RHEL8.1/RHEL7.8 - Wilson City / Cooper City / Purley-R Workaround: use #mdadm --grow --continue /dev/mdX{container} to manually continue migration	P3