Using Intel® Virtualization Technology (Intel® VT) with Intel® QuickAssist Technology

Application Note

October 2020
Intel Confidential
You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted, which includes subject matter disclosed herein.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No product or component can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

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. For more complete information about performance and benchmark results, visit www.intel.com/performance.

The products described 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.

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.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel does not control or audit third-party data. You should review this content, consult other sources, and confirm whether referenced data are accurate.

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.html.

Intel, Atom, Xeon, and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2020, Intel Corporation. All rights reserved.
## Revision History

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Revision Number</th>
<th>Description</th>
<th>Revision Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>330689</td>
<td>010</td>
<td>Updated Appendix A, FAQ</td>
<td>October 2020</td>
</tr>
<tr>
<td>330689</td>
<td>009</td>
<td>Updated relevant resource materials</td>
<td>March 2020</td>
</tr>
<tr>
<td>330689</td>
<td>008</td>
<td>Updated information on enabling virtual functions in Intel® QAT</td>
<td>December 2018</td>
</tr>
<tr>
<td>330689</td>
<td>007</td>
<td>• Added device ID for Intel® Xeon® processor D family.</td>
<td>September 2018</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Made updates to focus on the Intel® QAT v1.7 hardware and software</td>
<td></td>
</tr>
<tr>
<td>330689</td>
<td>006</td>
<td>• Removed section on using QEMU* KVM command line interface</td>
<td>August 2017</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added device ID for Intel® C62x Chipset and Intel Atom® C3000 Processor Product Family</td>
<td></td>
</tr>
<tr>
<td>330689</td>
<td>005</td>
<td>Updated:</td>
<td>July 2016</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 1.4.2, Limitations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 1.5, Software Requirements</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.2, Installing and Configuring the Host Operating System</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.3, Installing Guest OS Image</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.4.1, Using the libvirt* Virtual Machine Manager GUI</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.4.3, Pass-through PCI Device</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.4.2.1, Installing Updated QEMU* KVM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.4.2.2, Pass-through the PCI Device</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.4.2.4, Starting the Guest</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section Error! Reference source not found., Error! Reference source not found.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Appendix A, FAQ</td>
<td></td>
</tr>
<tr>
<td>330689</td>
<td>004</td>
<td>Updated:</td>
<td>February 2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 1.5, Software Requirements</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section 2.4.1, Installing Intel® QuickAssist Technology Software on Host</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section Error! Reference source not found., Error! Reference source not found.</td>
<td></td>
</tr>
<tr>
<td>330689</td>
<td>003</td>
<td>• Updated</td>
<td>November 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section Error! Reference source not found., Error! Reference source not found.</td>
<td></td>
</tr>
</tbody>
</table>
• Added Appendix A, FAQ

330689 002  Updated Section Error! Reference source not found., Error! Reference source not found.  September 2014

330689 001  • First “public” version of the document. Based on “Intel confidential” document number 476488-1.4 with the revision history of that document retained for reference purposes.
  • Updated Section 1.2.
  • Removed Fedora 14 information from Section 1.5, Software Requirements and Section 2.4.1, “Using the libvirt* Virtual Machine Manager GUI” on page 10.
  • Added new step at the end of Section 2.2, “Installing and Configuring the Host Operating System” on page 8.
  • Updated Section Error! Reference source not found., Error! Reference source not found.

476488 1.4  Updates include:
  • Modified step 8 in Section Error! Reference source not found., Error! Reference source not found.

476488 1.3  Updates to make applicable to multiple platforms that use Intel® QuickAssist Technology.

476488 1.2  Added new FAQ items, deleted outdated FAQ items.

476488 1.1  Added Limitations

476488 1.0  Initial release

September 2020
Application Note
Document Number: 330689-010
1.0 Introduction

This document discusses the following topics related to using Intel® Virtualization Technology (Intel® VT) with the Intel® QuickAssist Technology (Intel® QAT) Software:

- Features and limitations
- Build and installation

1.1 About this Document

Users of this document are expected to be familiar with virtualization technologies.

In this document, for convenience:

- Software package is used as a generic term for the Intel® QuickAssist Technology Software package.
- Acceleration drivers is used as a generic term for the software that allows the Intel® QuickAssist Software Library APIs to access the Intel® QuickAssist Accelerator(s) integrated in Intel® QAT.

1.2 Related Documents and References

This section provides references to find current software and documentation.

Associated software and collateral can be found on the open source website: https://01.org/intel-quickassist-technology

The below table includes a list of related documentation.

<table>
<thead>
<tr>
<th>Document Title</th>
<th>Document Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® QuickAssist Technology API Programmer's Guide</td>
<td>330684</td>
</tr>
<tr>
<td>Intel® QuickAssist Technology Cryptographic API Reference Manual</td>
<td>330685</td>
</tr>
<tr>
<td>Intel® QuickAssist Technology Data Compression API Reference Manual</td>
<td>330686</td>
</tr>
<tr>
<td>Intel® QuickAssist Technology Software for Linux® - Getting Started Guide - HW version 1.7</td>
<td>336212</td>
</tr>
<tr>
<td>Intel® QuickAssist Technology Software for Linux® - Release Notes - HW version 1.7</td>
<td>336211</td>
</tr>
<tr>
<td>Intel® QuickAssist Technology Software for Linux® - Programmer's Guide - HW version 1.7</td>
<td>336210</td>
</tr>
<tr>
<td>Intel® QuickAssist Technology Driver for Linux® - HW version 1.7</td>
<td>NA</td>
</tr>
</tbody>
</table>
Note: Sample configuration files are included with the software package.

1.3 Terminology

The below table includes a list of related documentation.

Table 2. Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLI</td>
<td>Command Line Interface</td>
</tr>
<tr>
<td>GigE</td>
<td>Gigabit Ethernet</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>Intel® QAT</td>
<td>Intel® QuickAssist Technology Software</td>
</tr>
<tr>
<td>Intel® VT</td>
<td>Intel® Virtualization Technology</td>
</tr>
<tr>
<td>IOMMU</td>
<td>Input-Output Memory Management Unit</td>
</tr>
<tr>
<td>KVM</td>
<td>Kernel-based Virtual Machine</td>
</tr>
<tr>
<td>PCH</td>
<td>Platform Controller Hub</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>SR-IOV</td>
<td>Single-root Input/Output Virtualization</td>
</tr>
<tr>
<td>PF</td>
<td>Physical Function</td>
</tr>
<tr>
<td>VF</td>
<td>Virtual Function</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
</tr>
</tbody>
</table>

1.4 About Intel® QAT Software

This section lists the features and limitations of the software.

1.4.1 Features

- Peripheral Component Interconnect (PCI) pass-through with Kernel-based Virtual Machine (KVM)
- Single-root Input/Output Virtualization (SR-IOV) with KVM

1.4.2 Limitations

- SR-IOV may not work on GNU*/Linux* kernel versions v2.6.38 or older.
1.5 **Software Requirements**

Software requirements will vary by the particular use case.

**Required:** Intel® QAT Software for Linux*

**Note:** Intel® recommends using the same version of the Intel® QAT driver on both host and guest OS. Consult your Intel® representative if you have a requirement to use different versions of the driver.

These instructions were tested against the following Linux* distribution: CentOS*.

1.6 **Supported Intel® QAT Endpoints and their Device IDs**

The below table includes a list of related documentation.

<table>
<thead>
<tr>
<th>Intel® QAT Endpoint</th>
<th>Physical Function (PF) Device ID</th>
<th>VF Device ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>8925-8955</td>
<td>0435</td>
<td>0443</td>
</tr>
<tr>
<td>Intel® C620 Series Chipsets</td>
<td>37c8</td>
<td>37c9</td>
</tr>
<tr>
<td>Intel® Atom® C3000 Processor Product Family</td>
<td>19e2</td>
<td>19e3</td>
</tr>
<tr>
<td>Intel® Xeon® processor D family</td>
<td>6f54</td>
<td>6f55</td>
</tr>
</tbody>
</table>
2.0 Using Intel® QAT Software with KVM

Intel® Virtualization Technology can use both SR-IOV and PCI pass-through for the acceleration services. SR-IOV enables the creation of VFs from a single Intel® QAT acceleration device to support acceleration for multiple Virtual Machines (VMs). If you do not need to share a single Platform Controller Hub (PCH) device with accelerator capabilities between multiple VMs, PCI pass-through is sufficient. The following sections describe the steps necessary to enable this functionality, with a focus on the SR-IOV use case.

2.1 Updating the BIOS Settings

**Note:** The BIOS settings for your system may differ from the following steps:

1. Power on the development board. Watch closely for the prompt to enter the BIOS setup. Press F2 when prompted.
2. Enable the VT-d parameter in BIOS. The option may be available under:
3. Advanced > System Agent (SA) Configuration > VT-d
4. Enable the SR-IOV parameter in BIOS. The option may be available under:
5. Advanced > System Agent (SA) Configuration > SRIOV

**Note:** Enabling the SR-IOV BIOS parameter is not required if you are not using SR-IOV.

1. Press F4 to Save and Exit. The BIOS changes are saved, and the system will boot.

2.2 Installing and Configuring the Host Operating System

1. Install the CentOS* v7 64-bit version. If necessary, consult the Getting Started Guide section "Installing the OS on a Development Board" (refer to Table 1), taking note that this guide assumes one of those CentOS* v7 64-bit versions as the host OS when SR-IOV is used.

**Note:** CentOS* v7 requires the intel_iommu=on kernel boot parameter to use SR-IOV and VT-d functionality.

2. Install virtualization related packages using the following command (root privileges required):

```
# yum -y install @virtualization
```

**Note:** Alternatively, use yum -y groupinstall Virtualization. This will install qemu-kvm qemu-img virt-manager, libvirt*, libvirt* -python, python-virtinst libvirt* -client, virt-install, virt-viewer, and all of the dependencies that are needed.

3. If the libvirtd service is not running, start it by using the commands:

```
# chkconfig libvirtd on
# service libvirtd start
```

4. Verify SR-IOV hardware capabilities using the command:

```
# lspci -vnc 8086:<Device ID>
```
Refer to Section 1.6 for a list of Intel® QAT supported devices and their device IDs.

It should display one of the capabilities as:
Capabilities: [140] Single Root I/O Virtualization (SR-IOV)

5. Verify BIOS settings using the command:
   # lsmod | grep kvm
   kvm_intel 42122 0
   kvm 257132 1 kvm_intel

6. Ensure that the system supports VT extensions:
   # egrep '^flags.*(vmx|svm)' /proc/cpuinfo

   Note: If nothing is printed out after executing the above command, then the system does not support VT extensions.

7. If kernel boot parameters changed, restart the system:
   # shutdown –r now

8. Power on the system and proceed with the instructions in the following sections.

9. Once the system is restarted, check for DMAR and IOMMU messages, similar to the following:
   # dmesg | grep -e DMAR -e IOMMU

      [ 0.000000] ACPI: DMAR 000000007b79c000 00080 (v01 INTEL INTEL ID 00000001 INTL 20091013)
      [ 0.000000] Intel-IOMMU: enabled
      [ 0.064454] dmar: IOMMU 0: reg_base_addr fbffc000 ver 1:0
   cap d2078c106f0466 ecap f020df
      [ 0.065560] IOAPIC id 8 under DRHD base 0xfbffc000 IOMMU 0 [ 0.065919] IOAPIC id 9 under DRHD base 0xfbffc000 IOMMU 0 [ 2.168898] DMAR: No ATSR found
      [ 2.169358] IOMMU 0 0xfbffc000: using Queued invalidation [ 2.169728] IOMMU: Setting RMRR:
      [ 2.170091] IOMMU: Setting identity map for device 0000:00:1d.0 [0x7a23f000 - 0x7a241fff]
      [ 2.170767] IOMMU: Prepare 0-16MiB unity mapping for LPC
      [ 2.171133] IOMMU: Setting identity map for device 0000:00:1f.0 [0x0 - 0xffffffff]

   Note: If the above command fails, a BIOS update or kernel reconfiguration may be required.

2.3 Installing Guest OS Image

This section describes how to use the libvirt* Virtual Machine Manager Graphical User Interface (GUI) to create the guest OS installation.

Note: The instructions in this section use the GUI approach; information on using the command line interface (CLI) is available at: http://libvirt.org/virshcmdref.html

Using the steps below, enter the root password when prompted.

1. Start the Virtual Machine Manager GUI by selecting it from the top main menu:
Applications > System Tools > Virtual Machine Manager.
2. Open a connection to a Hypervisor by choosing File > Add Connection.
3. Choose QEMU*/KVM for Hypervisor.
4. Make sure Connect to remote host is NOT checked.
5. Make sure Autoconnect is checked.
6. Click Connect.
7. After a connection is opened, select the localhost (QEMU*) and right click to select New.

Create a new VM with the New VM window displayed:

Enter the Name for the Guest VM, selecting Local install media (ISO image or CDROM), and clicking Forward.

Select Use CDROM or DVD, insert the OS installation CDROM/DVD into the CDROM/DVD drive, and make sure that the mounted CDROM appears in box [Media Unknown (dev/sr0)]. Select the OS type and version and then click Forward.

Choose Memory (RAM) in MB and number of CPUs settings (assign a sufficient amount, but it should not affect the Host OS, e.g., for 4 GB RAM and 8 cores, allocate Guest OS < 2 GB RAM and 4 cores CPU). Click Forward.

Note: Many platforms will show twice the actual number of cores due to simultaneous multithreading.

Make sure Enable storage for this VM is checked. Select Create a disk image on the computer's hard drive and specify a sufficient amount of hard drive space in GB (20 GB is recommended, and at least 18 GB may be required). Make sure Allocate entire disk now is checked. Click Forward.

Review the information from Steps 1 through 4. Note the Ready to begin installation of <Name> and the Storage path to the Guest VM image (this will be used if using the QEMU* CLI). Click Finish to begin the installation of the Guest OS.
8. Follow the steps provided in the “Installing CentOS*” section of the appropriate Getting Started Guide (refer to Table 1) to install the Guest OS.
9. Shut down the guest OS.

By default, the guest image is created in the /var/lib/libvirt/images directory. This image can be used by libvirt* APIs (virsh tools) and qemu-kvm to run the guest.

2.4 Installing and Configuring Intel® QuickAssist Technology Software

The following sections detail the steps to use the libvirt* Virtual Machine Manager GUI, though similar steps are possible using the command line interface.
2.4.1 Installing Intel® QuickAssist Technology Software on Host

**Note:** If you are not using SR-IOV and are instead passing through a Physical Function (PF) for acceleration services on one guest only, it is not required to install the Intel® QAT Software package on the host.

**Note:** The configure script included with the software package will automatically take care of certain build environment details, including copying over the correct sample configuration files. If you are not using an included script to build and install the software, you must perform these operations yourself, using the included script as a guide.

**Note:** On more recent kernels, it may be required to have the vfio-pci module inserted with “disable_deny_list=1”. Note that this can be done once with “sudo modprobe vfio-pci disable_deny_list=1” or persistently by adding the option “options vfio-pci disable_deny_list=1” to /etc/modprobe.d/vfio-pci.conf.

1. Enable the SR-IOV build on the host by using:
   ```bash
   # ./configure --enable-icp-sriov=host
   ```
2. Install the QAT software package:
   ```bash
   # make install
   ```
3. Enable the QAT VFs:
   ```bash
   # service qat_service_vfs start
   ```

**Note:** A subset of QAT VFs can be started by altering the qat_service_vfs arguments.

2.4.2 Verifying SR-IOV on Host

**Note:** If you are not using SR-IOV, skip this section.

**Note:** Sample configuration files have been included in the software package.

1. **Optional:** View the sample SR-IOV configuration files that were copied to the /etc directory. Any software instances that are specified in the PF (non-VF) configuration files will not be created. The sample SR-IOV configuration file sets the number of kernel service instances to 0.

2. Verify the VFs by running the following command in the host OS. As an example, with one high-end Intel® C620 Series Chipsets in the system, the output would have 16 or more 37c9 devices, as shown below:
   ```bash
   # lspci | grep 37c9
   01:01.0 Co-processor: Intel Corporation Device 37c9
   01:01.1 Co-processor: Intel Corporation Device 37c9
   ```
Using Intel® QAT Software with KVM

01:01.2 Co-processor: Intel Corporation Device 37c9
...
01:01.7 Co-processor: Intel Corporation Device 37c9
01:02.0 Co-processor: Intel Corporation Device 37c9
01:02.1 Co-processor: Intel Corporation Device 37c9
...
01:02.6 Co-processor: Intel Corporation Device 37c9
01:02.7 Co-processor: Intel Corporation Device 37c9

As another example, with one Intel® Communications Chipset 8925 to 8955 Series device in the system, the output would have 32 0443 devices, as shown below:

```
# lspci | grep 0443
01:01.0 Co-processor: Intel Corporation Device 0443
01:01.1 Co-processor: Intel Corporation Device 0443
01:01.2 Co-processor: Intel Corporation Device 0443
...
01:01.7 Co-processor: Intel Corporation Device 0443
01:02.0 Co-processor: Intel Corporation Device 0443
01:02.1 Co-processor: Intel Corporation Device 0443
...
01:02.6 Co-processor: Intel Corporation Device 0443
01:02.7 Co-processor: Intel Corporation Device 0443
01:03.0 Co-processor: Intel Corporation Device 0443
01:03.1 Co-processor: Intel Corporation Device 0443
...
01:03.6 Co-processor: Intel Corporation Device 0443
01:03.7 Co-processor: Intel Corporation Device 0443
01:04.0 Co-processor: Intel Corporation Device 0443
01:04.1 Co-processor: Intel Corporation Device 0443
...
01:04.6 Co-processor: Intel Corporation Device 0443
01:04.7 Co-processor: Intel Corporation Device 0443
```

Refer to 
Table 3 for supported devices and their device IDs.
2.4.3 Pass-through PCI Device


Figure 1. Virtual Machine Manager
2. Right-click on the guest and click Open (Do not run the guest).
A new window for the VM is displayed. Go to View > Details.

**Figure 2. View VM Details**

![VM Details](image)

Configure the processor, memory, boot options, and virtual hardware for the guest.
3. To add co-processor VFs (refer to Table 3 for supported devices and their device IDs) or GigE ports, select Add Hardware in the bottom-left corner and click PCI Host Device.

Figure 3. Add New Virtual Hardware

Select the appropriate PCI device (for instance, in Figure 3, 02:01:1 is one of the 0443 devices) to attach to Guest and click Finish. This newly added device should appear in the left column of details for the Guest.

Note: This action will internally unbind the PCI device from the Host driver currently being used and bind it to vfio-pci (CentOS* v7.1). If using a CLI, a similar sequence is: virsh detach <pci_func> and virsh-attach <domain> <pci_func>.

4. Optional: To detach a PCI device from the guest, click the PCI device to be detached from the details page left column and click Remove (bottom row).

Note: You can add and remove some PCI devices while the guest is running.

5. To run the guest, go to Virtual Machine > Run or click Play Radio on the Menu bar.
7. To view the guest console, go to View > Console.

2.4.4 Installing Intel® QuickAssist Technology Software on Guest

1. In the Guest OS, verify that the appropriate device has been passed through (see Section 2.4.3), as evidenced by the `lspci` command. Refer to Table 3 for the VF device IDs.

2. Install the Intel® QAT Software package on the Guest.

3. Enable the SR-IOV build on the host by using:
   ```
   # ./configure --enable-icp-sriov=guest
   ```

3. Install the QAT software:
   ```
   # make install
   ```

*Note:* This configuration file supports a limited number of service instances. Specifically, the limitations is a budget of 16 rings per VF. Refer to the relevant Programmer's Guide (Table 1) for more information on the configuration file formats. More devices can be passed through if more service instances are required.
A.1 Q: How can I pass through the Intel® QAT PF to a guest?

Intel® QAT Hardware 1.6 and 1.7 devices are not fully compliant to PCI specs. For this reason, when a FLR is done on the device by a driver different than the Intel® QAT driver (e.g. vfio-pci), the value of MPS is restored to the reset value and not the previous value. This has an impact on full direct pass-through. Follow this procedure to assign a device (using 8086:37c8 as an example) to a guest using full direct passthrough:

**On the host:**
1. Load the vfio-pci driver.
   ```bash
   modprobe vfio-pci
   ```
2. Bind the vfio driver to Intel® QAT devices.
   ```bash
   echo 8086 37c8 > /sys/bus/pci/drivers/vfio-pci/new_id
   ```
3. Read the device MPS.
   ```bash
   lspci -vvvnd 8086:37c8 | grep "MaxPayload [1-9]* bytes, Max"
   ```

**On the guest:**

*Note:* Make sure Intel® QAT driver is not installed in the guest!

Enter a command of the form:
```bash
qemu-system-x86_64 -enable-kvm -hda <path to your HD image> -m <memory in MB>M -device vfio-pci,host=<BDF of your QAT device>
```

For example:
```bash
qemu-system-x86_64 -enable-kvm -hda /var/lib/libvirt/images/f24.qcow2 -m 2048M -smp 16,cores=8,threads=1,sockets=2,maxcpus=16 -device vfio-pci,host=03:00.0
```

**On the host:**

Set the MPS to its original value. For example, if the MPS in the upstream bridge is equal to 256, enter the command:
```bash
setpci -d 8086:37c8 0x7c.b=0x37
```

**On the guest:**

Install and use the driver.

*Note:* If the guest is rebooted, the MPS will be changed.