Lenovo

Enabling Intel Optane DC Persistent Memory in a Linux Virtual Machine

Introduces the process of setting up Persistent Memory

Describes how to configure Guest OS to access virutualized Persistent Memory

Explains the meaning of parameters are used in QEMU command line tool

Provides test cases on the use of virtual Persistent Memory under Linux

Guangzhe Fu

LENOVO PRESS

Abstract

This paper describes how to set up and enable Intel Optane DC Persistent Memory under Linux. It shows how to create a Guest OS with virtual Intel Optane DC Persistent Memory device, step by step. The paper describes how to employ the two kinds of devices as the backend for the virtual Intel Optane DC Persistent Memory and provides test cases about how to enable virtual Intel Optane DC Persistent Memory under Linux.

This paper is intended for IT administrators, who are expected to have basic knowledge of Intel Optane DC Persistent Memory and virtualization deployment.

At Lenovo® Press, we bring together experts to produce technical publications around topics of importance to you, providing information and best practices for using Lenovo products and solutions to solve IT challenges.

See a list of our most recent publications at the Lenovo Press web site:

http://lenovopress.com

Do you have the latest version? We update our papers from time to time, so check whether you have the latest version of this document by clicking the **Check for Updates** button on the front page of the PDF. Pressing this button will take you to a web page that will tell you if you are reading the latest version of the document and give you a link to the latest if needed. While you're there, you can also sign up to get notified via email whenever we make an update.

Contents

Introduction

Intel Optane DC Persistent Memory (DCPMM or PMEM) is a new generation of nonvolatile memory (NVM) technology that is fast enough for processors to access stored data directly, without high latency and without a tremendous reduction in performance. Like flash memory, PMEM is nonvolatile storage; yet, like Dynamic Random Access Memory (DRAM) it is also byte-addressable using regular memory instructions.

Many customers virtualize their applications on Lenovo servers and a key criteria for using persistent memory in such environments is to virtualize the persistent memory devices.

QEMU is an open source emulator that performs hardware virtualization. It has integrated virtual DCPMM devices. Intel confirms that only QEMU supports DCPMM devices for now, starting from QEMU version 2.6.0.

The storage of a virtual PMEM device in QEMU is provided by the memory backend. The memory backend can be device DAX on the real DCPMM device or a file device on the Host. Since the label area is emulated, the Guest OS cannot alter the Host label area. All the virtual DCPMMs are displayed in the Guest as `/dev/pmem*` with a RAW mode namespace. The Guest directly accesses Host PMEM without impacting performance.

The workflow to access virtual NVDIMM is shown in Figure 1.

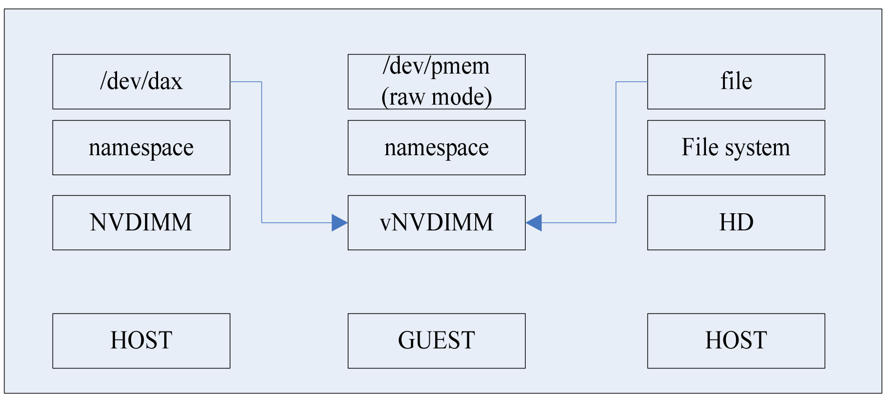

Figure 1   Persistent memory workflow in a virtualized environment

This paper enables Intel Optane DC Persistent Memory in a virtualized environment under SUSE Linux Enterprise Server 12 SP4.

Preparing the server

Before you begin, prepare the server hardware as follows:

1. Verify that the processor and server you are using supports Intel Optane DC Persistent Memory. See the DCPMM product guide:

   `https://lenovopress.com/lp1066-intel-optane-dcpmm#processor-support`

2. Install the DCPMMs and memory DRAMs correctly in the server. For specifics, see the following page in the Information Center:

   `https://thinksystem.lenovofiles.com/help/index.jsp?topic=%2F7X21%2Fmemory_module_installation_order_aep.html`

3. Update the server firmware to the latest version. More information, see the following page in the Information Center:

   https://thinksystem.lenovofiles.com/help/index.jsp?topic=%2F7X21%2Fupdate_the_firmware.html

4. Update the DCPMM firmware to the latest version. For specifics, see the following page:

   https://sysmgt.lenovofiles.com/help/index.jsp?topic=%2Fcom.lenovo.lxca.doc%2Fupdate_fw.html

Lab setup

We set up the test environment as below:

- Hardware:
  - Lenovo ThinkSystem™ SR630
  - 12x 512 GB Intel Optane DC Persistent Memory Modules
  - 12x 32 GB DDR4 DRAM DIMMs
- OS:
  - SLES 12 SP4
  - kernel-default-4.12.14-94.41.1.x86_64
- Firmware:
  - UEFI version 2.11
  - Intel Optane DC persistent memory firmware version: intc-Invgy_fw_pmem_dcpmm-01.02.00.5355b_linux_x86-64.bin

Installing ndctl and ipmctl

`ndctl` (Non-Volatile Device Control) a utility for managing the LIBNVDIMM Linux Kernel subsystem. Download ndctl from the following site:

https://github.com/pmem/ndctlis

Run the following command to install the ndctl tool:

```
# zypper install ndctl
```

`ipmctl` is a utility for configuring and managing Intel Optane DC persistent memory modules (PMM). It offers the following functions:

- Discovering PMMs on the platform
- Provisioning the platform memory configuration
- Viewing and updating the firmware on PMMs
- Configuring data-at-rest security on PMMs
- Monitoring PMM health
- Tracking performance of PMMs
- Debugging and troubleshooting PMMs

When you update the Intel Optane DC persistent memory firmware, ipmctl will be installed automatically. Alternatively, you can download ipmctl from the following site:

https://github.com/intel/ipmctl

QEMU command syntax

Verify that QEMU v2.6.0 or later is installed. Run the following command to verify your QEMU version:

```
# qemu-system-x86_64 -version
QEMU emulator version 2.11.2
```

If needed, install the latest version from the following web site:

https://www.qemu.org/

The storage of a vNVDIMM device (DCPMM device) in QEMU is provided by the memory backend (i.e. `memory-backend-file` and `memory-backend-ram`).

To create a vNVDIMM device at startup time, run **qemu-kvm** command with the following command line options:

```
-machine pc,nvdimm
-m $RAM_SIZE,slots=$N,maxmem=$MAX_SIZE
-object memory-backend-file,id=mem1,share=on,mem-path=$PATH,size=$NVDIMM_SIZE
-device nvdimm,id=nvdimm1,memdev=mem1
```

Where:

- **nvdimm** machine option enables vNVDIMM feature.
- **slots=$N** should be equal to or larger than the total amount of normal RAM devices and vNVDIMM devices. $N should be >= 2.
- **maxmem=$MAX_SIZE** should be equal to or larger than the total size of normal RAM devices and vNVDIMM devices. $MAX_SIZE should be >= $RAM_SIZE + $NVDIMM_SIZE.
- **object memory-backend-file,id=mem1,share=on,mempath=$PATH,size=$NVDIMM_SIZE** creates a backend storage of size $NVDIMM_SIZE on a file $PATH. All accesses to the virtual NVDIMM device go to the file $PATH.
- **share=on/off** controls the visibility of guest writes. If `share=on`, guest writes are applied to the backend file. If another guest uses the same backend file with option `share=on`, the above writes are visible to it as well. If `share=off`, guest writes will not be applied to the backend file and thus are invisible to other guests.
- **device nvdimm,id=nvdimm1,memdev=mem1** creates a virtual NVDIMM device whose storage is provided by the above memory backend device.

If multiple pairs of `-object` and `-device` are provided, multiple vNVDIMM devices can be created.

For the above command line options, if the Guest OS has the proper NVDIMM driver, it should be able to detect a NVDIMM device which is in the persistent memory mode and whose size is $NVDIMM_SIZE.

Using QEMU to enable PMEM virtualization

The following are the two kind of memory backends we used:

- Device DAX
  a. Run the following command to create the device DAX in the Host:

```
Shell>ipmctl create –goal persistentmemorytype=appdirectnotinterleaved
Shell>reboot
Shell>ndctl create-namespace --type=pmem  --mode=devdax
```

  b. Run the following command to create the Guest:

```
#qemu-kvm -M pc,accel=kvm,nvdimm \
-m 10G,slots=20,maxmem=40G \
-object memory-backend-file,id=mem1,share=on, \
mem-path=/dev/dax3.0,size=4G,align=2M \
-device nvdimm,id=nvdimm1,memdev=mem1,label-size=128K \
-boot order=cd \
-hda /home/sles12-4.qcow2  \
-cdrom /home/SLE-12-SP4-Server-DVD-x86_64-Beta4-DVD1.iso
-vnc 0.0.0.0:9
```

- File in the Host
  a. Run the following command to create an nvdimm file on the Host:

```
truncate -s 4G /tmp/nvdimm
```

  b. Run the following command to create the Guest:

```
# qemu-kvm -M pc,nvdimm \
-m 10G,slots=20,maxmem=40G \
-objectmemory-backend-file,id=mem1,share=on,mem-path=/tmp/nvdimm,size=4G \
-device nvdimm,unarmed=on,id=nvdimm1,memdev=mem1,label-size=128K \
-boot order=cd \
-hda /home/sles12-4.qcow2  \
-cdrom /home/SLE-12-SP4-Server-DVD-x86_64-Beta4-DVD1.iso
-vnc 0.0.0.0:9
```

After installing the system, you can restart the Guest without option **-boot order=cd**.

Verifying the result

After you start the Guest, you can see /dev/pmem device in Guest OS. Do the following to check the results:

1. Run the following command to check /dev/pmem device by ndctl:

```
# ndctl list -N
{
"dev":"namespace0.0",
"mode":"raw",
"map":"mem",
"size":4294967296,
"sector_size":512,
"blockdev":"pmem0",
"numa_node":0
}
```

2. Run the following command to format and mount the `/dev/pmem` device:

```
# mkfs.ext4  -F /dev/pmem0
# mount –t ext4  /dev/pmem0 /test
```

3. Create a file on /dev/pmem device and then restart the guest OS to check whether the file is persistent or not. Even you restart host system, the file is still there.

```
# echo “12345” >/test/file1
# reboot
# mount –t ext4  /dev/pmem0 /test
# cat /test/file1
12345
```

Important considerations:

- If you use device DAX as the vNVDIMM device, after you restart the Host, you need to wait for about 10 minutes before device DAX is ready for using. Then, you can start the Guest with it.
- Be careful that backend file size is not equal to the size given by `size` option. QEMU will truncate the backend file by ftruncate(2), which will corrupt the existing data in the backend file, especially for the shrink case.

References

For more information, see these resources:

- QEMU Virtual NVDIMM

  https://github.com/qemu/qemu/blob/master/docs/nvdimm.txt
- LIBNVDIMM: Non-Volatile Devices

  https://www.kernel.org/doc/Documentation/nvdimm/nvdimm.txt
- Utility library for managing the libnvdimm sub-system in the Linux kernel

  https://github.com/pmem/ndctl
- ipmctl

  https://github.com/intel/ipmctl

Author

Guangzhe Fu is a Linux Engineer in the Lenovo Data Center Group in Beijing, China. He joined the OS team in 2017. His major focus is the Network and Virtualization feature of Linux kernel development in Lenovo. He has two years experience as a Software Architecture engineer, six years experience as a Linux Kernel Development engineer.

Thanks to the following people for their contributions to this project:

- Linux OS team in Lenovo
- David Watts, Lenovo Press

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document was created or updated on September 4, 2019.

Send us your comments via the **Rate & Provide Feedback** form found at
http://lenovopress.com/lp1224

Trademarks