Getting started guide of

Intel Optane* DC persistent memory module in Kubernetes cluster for Alauda

Author: ailin.yang@intel.com

Homework:

Persistent Memory introduction: introduction web page

Intel® Optane™ DC Persistent Memory: Benefit from Greater Capacity, Affordability and Persistence

Concepts of Persistent Memory Provisioning: Region, Label, Namespace, DAX

Configuring Intel® Optane™ DC Persistent Memory for Best Performance: configuration video

Provision Intel® Optane™ DC Persistent Memory in Linux*: video

Configure, Manage, and Profile Intel® Optane™ DC Persistent Memory Modules: docs

The Intel® Optane™ DC Persistent Memory: Programming Model

Intel® Optane™ DC Persistent Memory Modules- Use ipmctl to Debug

Intel® Optane™ DC Persistent Memory Modules – Provision for KVM/QEMU

System preparation:

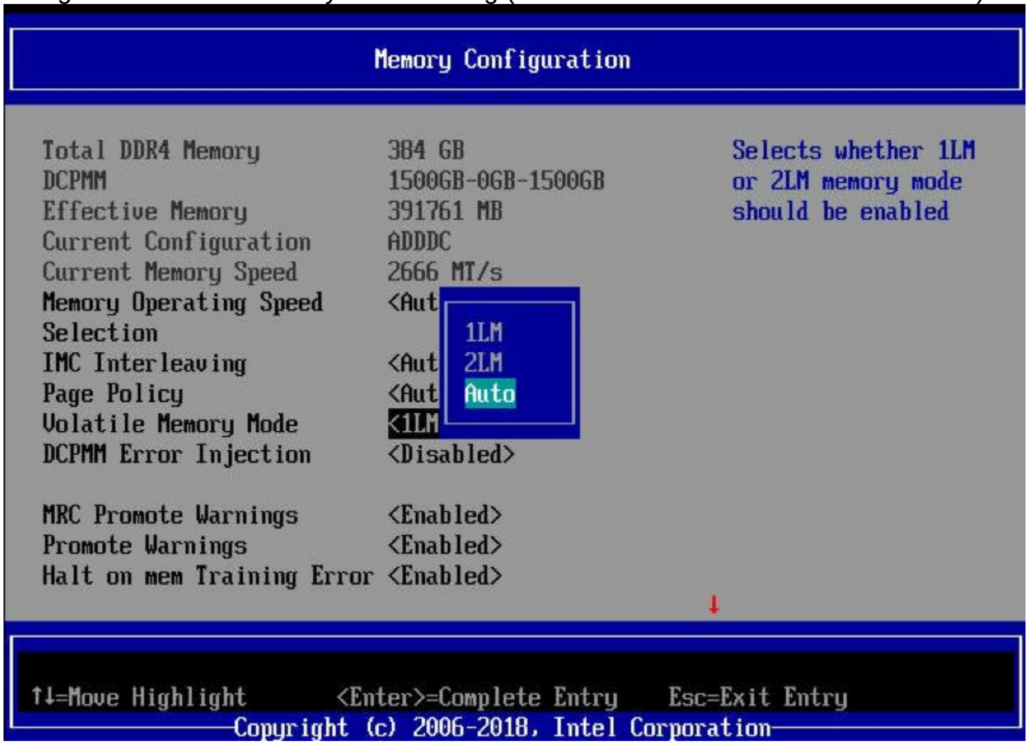
1. 1 Hardware configuration

It depends on the hardware that your server may be different from the one being used in this document. The brief info of the HW is below:

Intel® Server Board S2600WFT, 2x Intel® Xeon® Gold 6252 Processor 2.1GHz 24 cores; 12x 16GB DDR4 DRAM; 12x 128GB Intel Optane DC Persistent Memory

In general, it should be like configure the system to use Intel DCPMM. First it is recommended to confirm that your system has the latest BIOS that support the DCPMM and make sure that Volatile Memory Mode in the Memory Configuration is set to Auto (figure 1).

Figure 1. Volatile memory mode setting (BIOS default recommendation = AUTO).



```
                         Memory Configuration

 Total DDR4 Memory          384 GB                 Selects whether 1LM
 DCPMM                      1500GB-0GB-1500GB      or 2LM memory mode
 Effective Memory           391761 MB              should be enabled
 Current Configuration      ADDDC
 Current Memory Speed       2666 MT/s
 Memory Operating Speed     <Aut
 Selection                       1LM
 IMC Interleaving           <Aut 2LM
 Page Policy                <Aut Auto
 Volatile Memory Mode       <1LM
 DCPMM Error Injection      <Disabled>

 MRC Promote Warnings       <Enabled>
 Promote Warnings           <Enabled>
 Halt on mem Training Error <Enabled>

 ↑↓=Move Highlight        <Enter>=Complete Entry     Esc=Exit Entry
              Copyright (c) 2006-2018, Intel Corporation
```

Note: this is a screen capture of another system with different spec

Figure 2. Topologies per CPU socket (2-2-2, 6 DDR4 DIMMs, 6 DCPMM DIMMs)

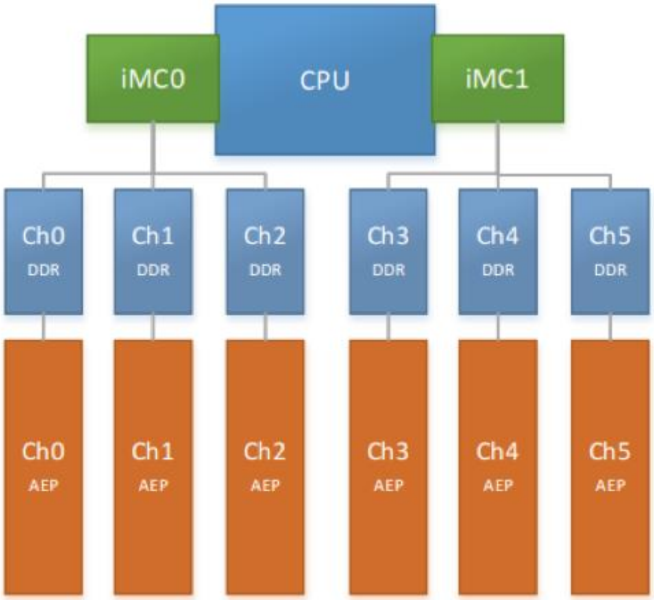


Figure 2 shows the memory topology per CPU socket of the reference system. This can be confirmed by the following command:

```
$ sudo ipmctl show -topology
```

```
 DimmID | MemoryType                  | Capacity  | PhysicalID| DeviceLocator
==============================================================================
 0x0001 | Logical Non-Volatile Device | 126.4 GiB | 0x0028    | CPU1_DIMM_A2
 0x0011 | Logical Non-Volatile Device | 126.4 GiB | 0x002c    | CPU1_DIMM_B2
 0x0021 | Logical Non-Volatile Device | 126.4 GiB | 0x0030    | CPU1_DIMM_C2
 0x0101 | Logical Non-Volatile Device | 126.4 GiB | 0x0036    | CPU1_DIMM_D2
 0x0111 | Logical Non-Volatile Device | 126.4 GiB | 0x003a    | CPU1_DIMM_E2
 0x0121 | Logical Non-Volatile Device | 126.4 GiB | 0x003e    | CPU1_DIMM_F2
 0x1001 | Logical Non-Volatile Device | 126.4 GiB | 0x0044    | CPU2_DIMM_A2
 0x1011 | Logical Non-Volatile Device | 126.4 GiB | 0x0048    | CPU2_DIMM_B2
 0x1021 | Logical Non-Volatile Device | 126.4 GiB | 0x004c    | CPU2_DIMM_C2
 0x1101 | Logical Non-Volatile Device | 126.4 GiB | 0x0052    | CPU2_DIMM_D2
 0x1111 | Logical Non-Volatile Device | 126.4 GiB | 0x0056    | CPU2_DIMM_E2
 0x1121 | Logical Non-Volatile Device | 126.4 GiB | 0x005a    | CPU2_DIMM_F2
 N/A    | DDR4                        | 16.0 GiB  | 0x0026    | CPU1_DIMM_A1
 N/A    | DDR4                        | 16.0 GiB  | 0x002a    | CPU1_DIMM_B1
 N/A    | DDR4                        | 16.0 GiB  | 0x002e    | CPU1_DIMM_C1
 N/A    | DDR4                        | 16.0 GiB  | 0x0034    | CPU1_DIMM_D1
 N/A    | DDR4                        | 16.0 GiB  | 0x0038    | CPU1_DIMM_E1
 N/A    | DDR4                        | 16.0 GiB  | 0x003c    | CPU1_DIMM_F1
 N/A    | DDR4                        | 16.0 GiB  | 0x0042    | CPU2_DIMM_A1
 N/A    | DDR4                        | 16.0 GiB  | 0x0046    | CPU2_DIMM_B1
 N/A    | DDR4                        | 16.0 GiB  | 0x004a    | CPU2_DIMM_C1
 N/A    | DDR4                        | 16.0 GiB  | 0x0050    | CPU2_DIMM_D1
 N/A    | DDR4                        | 16.0 GiB  | 0x0054    | CPU2_DIMM_E1
 N/A    | DDR4                        | 16.0 GiB  | 0x0058    | CPU2_DIMM_F1
```

To install ipmctl tool on Ubuntu:
First install the dependencies packages:

```
sudo apt-get install cmake libndctl-dev doxygen build-essential
sudo add-apt-repository ppa:jhli/libsafec
sudo apt-get update
sudo apt-get install libsafec-dev
sudo apt install ruby-full
sudo gem install asciidoctor-pdf --pre
sudo apt-get --no-install-recommends install asciidoc -y
```

Then build and install ipmctl:

```
Git clone https://github.com/intel/ipmctl.git
Cd ipmctl
mkdir output && cd output
cmake -DRELEASE=ON -DCMAKE_INSTALL_PREFIX=/ ..
make -j all
sudo make install
```

Detail of the tool, please see:
https://github.com/intel/ipmctl
https://github.com/pmem/ndctl

1.2 Memory mode and App direct mode

1.2.2 Configuring DCPMM in Memory mode

```
$ # destroy all the possible namespaces existing in AppDirect mode
$ # It is necessary to unmount any partitions that are being mounted.
$ # sudo fdisk /dev/pmemXXX -> select “delete” -> select “w”
$ sudo ndctl destroy-namespace all -f
$ sudo ipmctl delete -goal
$ # create a Memory Mode goal allocation for the entire DCPMM
$ sudo ipmctl create -goal MemoryMode=100
The following configuration will be applied:
 SocketID | DimmID | MemorySize | AppDirect1Size | AppDirect2Size
==================================================================
 0x0000   | 0x0001 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0011 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0021 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0101 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0111 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0121 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1001 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1011 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1021 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1101 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1111 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1121 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
Do you want to continue? [y/n] y
Created following region configuration goal
 SocketID | DimmID | MemorySize | AppDirect1Size | AppDirect2Size
==================================================================
 0x0000   | 0x0001 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0011 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0021 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0101 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0111 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0121 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1001 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1011 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1021 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1101 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1111 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1121 | 126.0 GiB  | 0.0 GiB        | 0.0 GiB
A reboot is required to process new memory allocation goals.
$ sudo reboot
$ sudo ipmctl show -memoryresources
Capacity=1517.1 GiB
MemoryCapacity=1512.0 GiB
AppDirectCapacity=0.0 GiB
UnconfiguredCapacity=0.0 GiB
InaccessibleCapacity=5.1 GiB
ReservedCapacity=0.0 GiB
```

1.2.2 Configuring DCPMM in App Direct mode

```
$ sudo ipmctl delete -goal
$ sudo ipmctl create -goal PersistentMemoryType=AppDirect
The following configuration will be applied:
 SocketID | DimmID | MemorySize | AppDirect1Size | AppDirect2Size
==================================================================
 0x0000   | 0x0011 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0000   | 0x0021 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0000   | 0x0001 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0000   | 0x0111 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0000   | 0x0121 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0000   | 0x0101 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0001   | 0x1011 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0001   | 0x1021 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0001   | 0x1001 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0001   | 0x1111 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0001   | 0x1121 | 0.0 GiB    | 126.0 GiB      | 0.0 GiB
 0x0001   | 0x1101 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
Do you want to continue? [y/n] y
Created following region configuration goal
 SocketID | DimmID | MemorySize | AppDirect1Size | AppDirect2Size
==================================================================
 0x0000   | 0x0011 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0000   | 0x0021 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0000   | 0x0001 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0000   | 0x0111 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0000   | 0x0121 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0000   | 0x0101 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0001   | 0x1011 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0001   | 0x1021 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0001   | 0x1001 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0001   | 0x1111 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0001   | 0x1121 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
 0x0001   | 0x1101 | 0.0 GiB    | 126.0 GiB       | 0.0 GiB
A reboot is required to process new memory allocation goals.
$ sudo reboot
```

Deploy Pmem-csi in Kubernetes cluster

In this documentation, the deployment is based on an exist Kubernetes cluster, so setup a Kubernetes cluster is not covered, and Before start, there is something that needs to be careful here. First please make sure that the system has been configured to use Intel DCPMM in App Direct mode. Second, it is necessary to make sure that the volume groups are clean. It can be checked by using the following command.

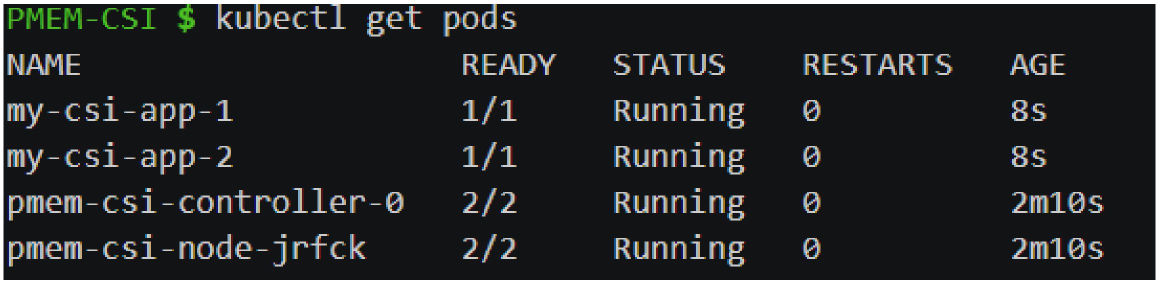
$ sudo vgs

Please confirm that it should return nothing before we move forward.

Also please read readme of https://github.com/intel/pmem-csi to know requirements to deploy it,

Now you can follow up this screencast guide to deploy Pmem-CSI in your Kubernetes cluster

After completing deployment, you should have it run correctly with below status:



```
PMEM-CSI $ kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
my-csi-app-1               1/1     Running   0          8s
my-csi-app-2               1/1     Running   0          8s
pmem-csi-controller-0      2/2     Running   0          2m10s
pmem-csi-node-jrfck        2/2     Running   0          2m10s
```

Congratulations!   You have PEME-CSI ready