PROVISIONING INTEL® OPTANE™ DC PERSISTENT MEMORY

USHA UPADHYAYULA

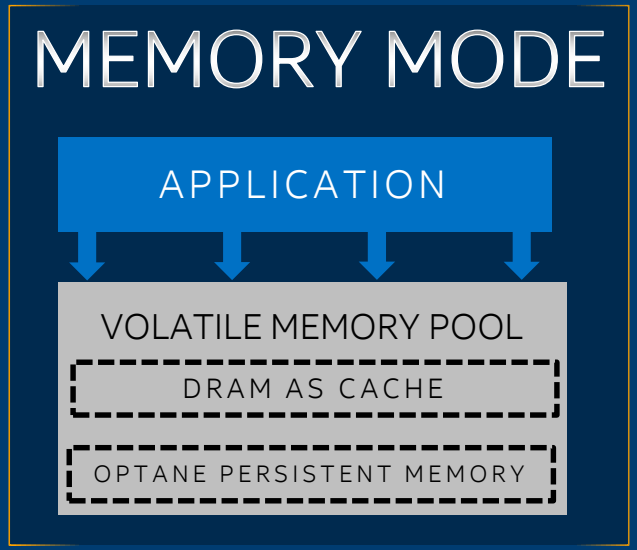
INTEL DATA CENTER GROUP

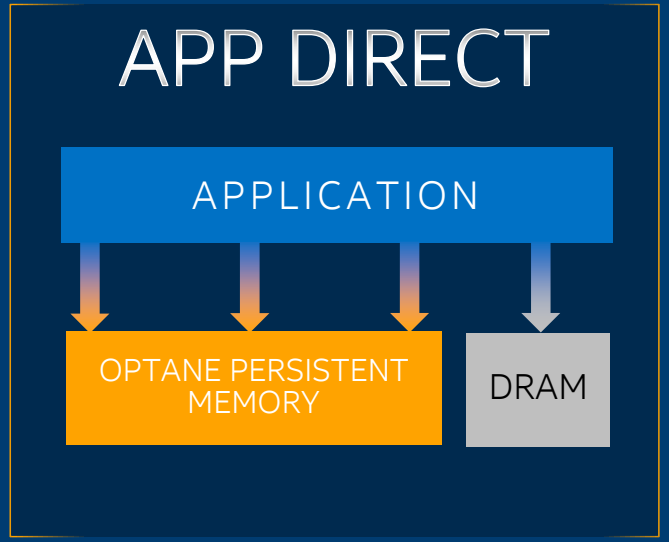
AGENDA

- Introduction
  - Intel® Optane™ DC Persistent Memory Operating Modes
  - Basic Provisioning Concepts
  - Verifying Platform Support
  - Provisioning Utilities
    - Linux & Windows
- Provisioning Persistent Memory
  - Memory Mode
  - App Direct Mode
- Provisioning Examples
- Q&A

INTEL® OPTANE™ DC PERSISTENT MEMORY OPERATING MODES

INTEL® OPTANE™ DC PERSISTENT MEMORY OPERATING MODES



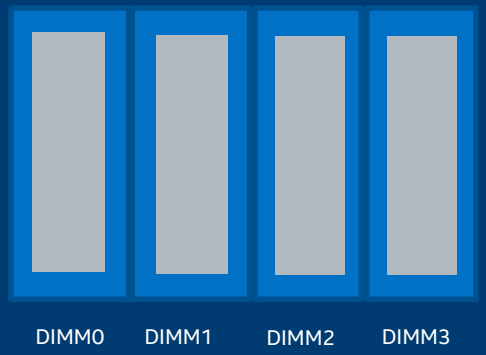


AFFORDABLE & LARGE VOLATILE MEMORY CAPACITY

LARGE CAPACITY PERSISTENT MEMORY

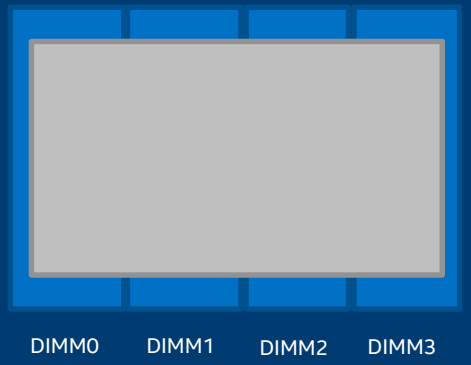
PROVISIONING TERMS & CONCEPTS

Non-Interleaved Regions



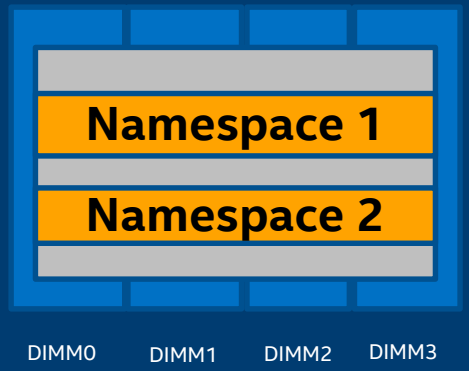
Creates contiguous physical address space within a single DIMM

Interleaved Regions



Creates contiguous physical address space across multiple DIMMs and provides striped reads/writes for better throughput

Namespaces



Similar to SSD, raw capacity of a region is partitioned into one or more logical devices called namespaces

PROVISIONING STEPS

Memory Mode

1. Create Memory Region (Hardware)
2. Reboot

App Direct Mode

1. Create Persistent Memory Regions (Hardware) & Reboot
2. Create Namespaces (Kernel)
3. Make & Mount File Systems (User Space)

PROVISIONING UTILITIES

PROVISIONING UTILITIES

DIMM Vendor Specific	DIMM Vendor Neutral
ipmctl for Intel® Optane™ Data Center Persistent Memory modules	*ndctl* for Linux Microsoft® *PowerShell Commandlets* for persistent memory

PROVISIONING UTILITIES

IPMCTL

- Intel® Optane® Technology only
- Linux, Windows, & UEFI
- Manage Hardware features
  - Discover & Manage Modules
  - Provision for Operating Modes– Memory or AppDirect
  - View & Update Module Firmware
  - Monitor Health
  - Track Performance
  - Debug & Troubleshoot

NDCTL & PowerShell Commandlets

- Vendor Neutral
- Manage Hardware and OS features
  - Show Module Information
  - Manage Namespaces
  - Monitor Health

	ipmctl	**ndctl**	**PowerShell commandlets**
DIMM Vendor	Intel®	Vendor-neutral	Vendor-neutral
Linux	Yes	Yes	No
Windows*	Yes	No	Yes
Manage Goals/Regions	Yes	No	No
Manage namespaces	No	Yes	Yes
Health/SMART	Yes	Yes	Yes
Track Performance	Yes	No	No
Security	Yes	Yes	No

IPMCTL - UTILITY FOR MANAGING INTEL OPTANE DC PERSISTENT MEMORY MODULES

Provision Memory

```
create -goal
delete -goal
show –goal
dump –goal
load –goal
show -region
```

Device Discovery

```
show –device
show -dimm
show –memory-resources
show –socket
show –system-capabilities
show -topology
```

Health & Performance

```
show -performance
show -sensor
change -sensor
```

Support & Maintenance

```
dump -support
show –events
show –firmware
show –host
acknowledge –event
update -firmware
change –preferences
show -preferences
help
```

Security

```
change –device-passphrase
change –device-security
enable –device-security
erase –device-data
```

Debugging

```
dump –debug-log
inject -error
run -diagnostic
show -acpi
show -pcd
show –nfit
show –error-log
```

OPERATING SYSTEM TOOLS TO MANAGE PERSISTENT MEMORY DEVICES

NDCTL

```
# ndctl
create-namespace    disable-namespace    freeze-security
list                sanitize-dimm        version
check-namespace     disable-region       help
load-keys           setup-passphrase     wait-overwrite
check-labels        enable-dimm          init-labels
monitor             start-scrub          wait-scrub
destroy-namespace   enable-namespace     inject-error
read-labels         update-firmware      write-labels
disable-dimm        enable-region        inject-smart
remove-passphrase   update-passphrase    zero-labels
```

MICROSOFT® POWERSHELL COMMANDS

```
>PS
Get-PmemDisk
Get-PmemPhysicalDevice
New-PmemDisk
Remove-PmemDisk
Get-PmemUnusedRegion
Initialize-PmemPhysicalDevice
```

VERIFYING OPERATING SYSTEM/BIOS SUPPORT

- The BIOS must provide NFIT (Non-Volatile Firmware Interface Table) info

```
# ndctl list --bus all | grep -i nfit
    "provider":"ACPI.NFIT",
```

- Verify the Kernel has PMEM and NVDIMM support enabled

```
# egrep -i "pmem|nvdimm" /boot/config-`uname -r`
CONFIG_X86_PMEM_LEGACY_DEVICE=y
CONFIG_X86_PMEM_LEGACY=m
CONFIG_LIBNVDIMM=y
CONFIG_BLK_DEV_PMEM=m
CONFIG_NVDIMM_PFN=y
CONFIG_NVDIMM_DAX=y
CONFIG_DEV_DAX_PMEM=m
CONFIG_ARCH_HAS_PMEM_API=y
```

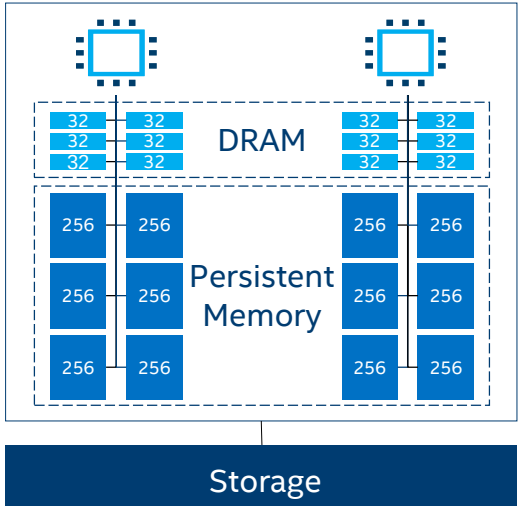
- Check dmesg

```
# dmesg | grep e820 | grep -i persistent
[    0.000000] BIOS-e820: [mem 0x0000003060000000-0x000001aa5fffffff] persistent (type 7)
[    0.000000] BIOS-e820: [mem 0x000001da00000000-0x00000353ffffffff] persistent (type 7)
```

HOW TO VERIFY OPTANE PMEM & DDR TOPOLOGY

```
# ipmctl show -dimm
 DimmID | Capacity   | HealthState | ActionRequired | LockState | FWVersion
==============================================================================
 0x0001 | 252.4 GiB | Healthy     | 0              | Disabled  | 01.02.00.5346
 0x0011 | 252.4 GiB | Healthy     | 0              | Disabled  | 01.02.00.5346
 0x0021 | 252.4 GiB | Healthy     | 0              | Disabled  | 01.02.00.5346
 [….]
 0x1101 | 252.4 GiB | Healthy     | 0              | Disabled  | 01.02.00.5346
 0x1111 | 252.4 GiB | Healthy     | 0              | Disabled  | 01.02.00.5346
 0x1121 | 252.4 GiB | Healthy     | 0              | Disabled  | 01.02.00.5346
```

DRAM & PMEM Solution



384 GiB

3072 GiB

Total: 3456 GiB

```
# ipmctl show -topology

 DimmID | MemoryType| Capacity   | PhysicalID| DeviceLocator
==============================================================
 0x0001 | DCPM      | 252.4 GiB | 0x0028    | CPU1_DIMM_A2
 0x0011 | DCPM      | 252.4 GiB | 0x002c    | CPU1_DIMM_B2
 […]
 N/A    | DDR4      | 32.0 GiB  | 0x004a    | CPU2_DIMM_C1
 N/A    | DDR4      | 32.0 GiB  | 0x0050    | CPU2_DIMM_D1
 N/A    | DDR4      | 32.0 GiB  | 0x0054    | CPU2_DIMM_E1
 N/A    | DDR4      | 32.0 GiB  | 0x0058    | CPU2_DIMM_F1
```

```
# ipmctl show -memoryresources

Capacity=3029.4 GiB
MemoryCapacity=0.0 GiB
AppDirectCapacity=3024.0 GiB
UnconfiguredCapacity=5.2 GiB
InaccessibleCapacity=0.0 GiB
ReservedCapacity=0.2 GiB
```

USING NDCTL TO VIEW PMEM TOPOLOGY

```
$ ndctl list -DFH    # List DIMM, Firmware, & Health
[
  {
    "dev":"nmem1",
    "id":"8089-a2-1837-00000bb3",
    "handle":17,
    "phys_id":44,
    "health":{
      "health_state":"ok",
      "temperature_celsius":29.0,
      "controller_temperature_celsius":28.0,
      "spares_percentage":100,
      "alarm_temperature":false,
      "alarm_controller_temperature":false,
      "alarm_spares":false,
      "alarm_enabled_media_temperature":false,
      "alarm_enabled_ctrl_temperature":false,
      "alarm_enabled_spares":false,
      "shutdown_state":"clean",
      "shutdown_count":1
    },
    "firmware":{
      "current_version":4328543046
    }
  },
  […]
```

```
$ ndctl list -NuRD     # List Modules, Regions, & Namespaces
  "dimms":[
    {
      "dev":"nmem1",
      "id":"8089-a2-1837-00000bb3",
      "handle":"0x11",
      "phys_id":"0x2c"
    },
    […]
"regions":[
    {
      "dev":"region1",
      "size":"1512.00 GiB (1623.50 GB)",
      "available_size":0,
      "max_available_extent":0,
      "type":"pmem",
      "mappings":[
        {
          "dimm":"nmem11",
          "offset":"0x10000000",
          "length":"0x3f00000000",
          "position":5
        },
        […]
"namespaces":[
        {
          "dev":"namespace1.0",
          "mode":"fsdax",
          "map":"dev",
          "size":"1488.37 GiB (1598.13 GB)",
          "uuid":"47cbd389-b1ec-4827-96f1-cbd0c0955c41",
          "sector_size":512,
          "align":2097152,
          "blockdev":"pmem1"
        }
```

PROVISIONING FOR MEMORY MODE

IPMCTL: CREATING A MEMORY MODE GOAL (100% OF PMEM)

1. Create Memory Region (Hardware)
2. Reboot

```
# ipmctl create -goal MemoryMode=100
The following configuration will be applied:
 SocketID | DimmID | MemorySize | AppDirect1Size | AppDirect2Size
==================================================================
 0x0000   | 0x0001 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0011 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0021 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0101 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0111 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0000   | 0x0121 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1001 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1011 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1021 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1101 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1111 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
 0x0001   | 0x1121 | 252.0 GiB  | 0.0 GiB        | 0.0 GiB
Do you want to continue? [y/n] y
…
A reboot is required to process new memory allocation goals.
# systemctl reboot
```

MEMORY MODE VIEWED FROM THE OPERATING SYSTEM

```
# lsmem
...
Memory block size:          1G
Total online memory:        3T
Total offline memory:       0B
```

```
# free -g
              total        used        free      shared  buff/cache   available
Mem:           2975           5        2969           0           0        2961
Swap:            31           0          31
```

```
# top
top - 16:18:31 up 7 min,  1 user,  load average: 0.00, 0.07, 0.06
Tasks: 839 total,   1 running, 838 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
TiB Mem :      2.9 total,      2.9 free,      0.0 used,      0.0 buff/cache
TiB Swap:      0.0 total,      0.0 free,      0.0 used.      2.9 avail Mem
```

System Config:
2-Socket
DDR: 384GB
PMEM: 3TB

```
# ipmctl show -memoryresources
Capacity=3029.4 GiB
MemoryCapacity=3024.0 GiB
AppDirectCapacity=0.0 GiB
UnconfiguredCapacity=0.0 GiB
InaccessibleCapacity=0.0 GiB
ReservedCapacity=5.4 GiB
```

IPMCTL: CREATING A MIXED MODE GOAL

```
# ipmctl create -goal MemoryMode=50
The following configuration will be applied:
 SocketID | DimmID | MemorySize | AppDirect1Size | AppDirect2Size
==============================================================
 0x0000   | 0x0001 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0000   | 0x0011 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0000   | 0x0021 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0000   | 0x0101 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0000   | 0x0111 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0000   | 0x0121 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0001   | 0x1001 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0001   | 0x1011 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0001   | 0x1021 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0001   | 0x1101 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0001   | 0x1111 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
 0x0001   | 0x1121 | 124.0 GiB  | 128.0 GiB      | 0.0 GiB
Do you want to continue? [y/n] y
…
A reboot is required to process new memory allocation goals.
# systemctl reboot
```

Note: 100% of DRAM is used for the MemoryMode Cache

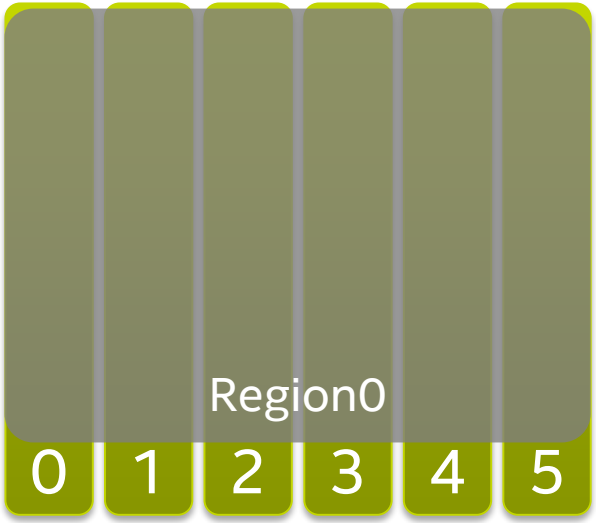
PROVISIONING FOR APP DIRECT

STEP 1 - CREATING REGIONS (VENDOR SPECIFIC)

AN INTEL OPTANE DC PERSISTENT MEMORY EXAMPLE

1. Create Persistent Memory Regions (Hardware) & Reboot
2. Create Namespaces (Kernel)
3. Make & Mount File Systems (User Space)

Interleaved Sets



Note: Cannot Interleave across sockets.

PMEM Modules

Non-Interleaved Sets

```
# ipmctl create -goal PersistentMemoryType=AppDirect
# reboot
```

```
# ipmctl create -goal PersistentMemoryType=AppDirect
# reboot
```

IPMCTL: CREATE AN APPDIRECT GOAL

```
# ipmctl create -goal PersistentMemoryType=AppDirect

The following configuration will be applied:
 SocketID | DimmID | MemorySize | AppDirect1Size | AppDirect2Size
==================================================================
 0x0000   | 0x0001 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0000   | 0x0011 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0000   | 0x0021 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0000   | 0x0101 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0000   | 0x0111 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0000   | 0x0121 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0001   | 0x1001 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0001   | 0x1011 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 0x0001   | 0x1021 | 0.0 GiB    | 252.0 GiB      | 0.0 GiB
 …
Do you want to continue? [y/n] y
…
A reboot is required to process new memory allocation goals.
# systemctl reboot
…
# ipmctl show -region

 SocketID | ISetID             | PersistentMemoryType | Capacity   | FreeCapacity | HealthState
=============================================================================================
 0x0000   | 0x2d3c7f48f4e22ccc | AppDirect            | 1512.0 GiB | 1512.0 GiB   | Healthy
 0x0001   | 0xdd387f488ce42ccc | AppDirect            | 1512.0 GiB | 1512.0 GiB   | Healthy
```

NAMESPACE MODES

FSDAX (Default)

- Default namespace mode
- Supports XFS & EXT4
- Removes Page Cache
- Allows Direct Mappings to PMEM
- Block Device (/dev/pmem0)

DEVDAX

- Similar mmap() capabilities to FSDAX
- Useful for RDMA
- Removes Page Cache
- Allows Direct Mappings to PMEM
- Character Device (/dev/dax0.0)

SECTOR

- Supports any filesystem
- Sector/Block Atomicity
- No DAX Support
- OS Bootable
- Block Device (/dev/pmem0s)

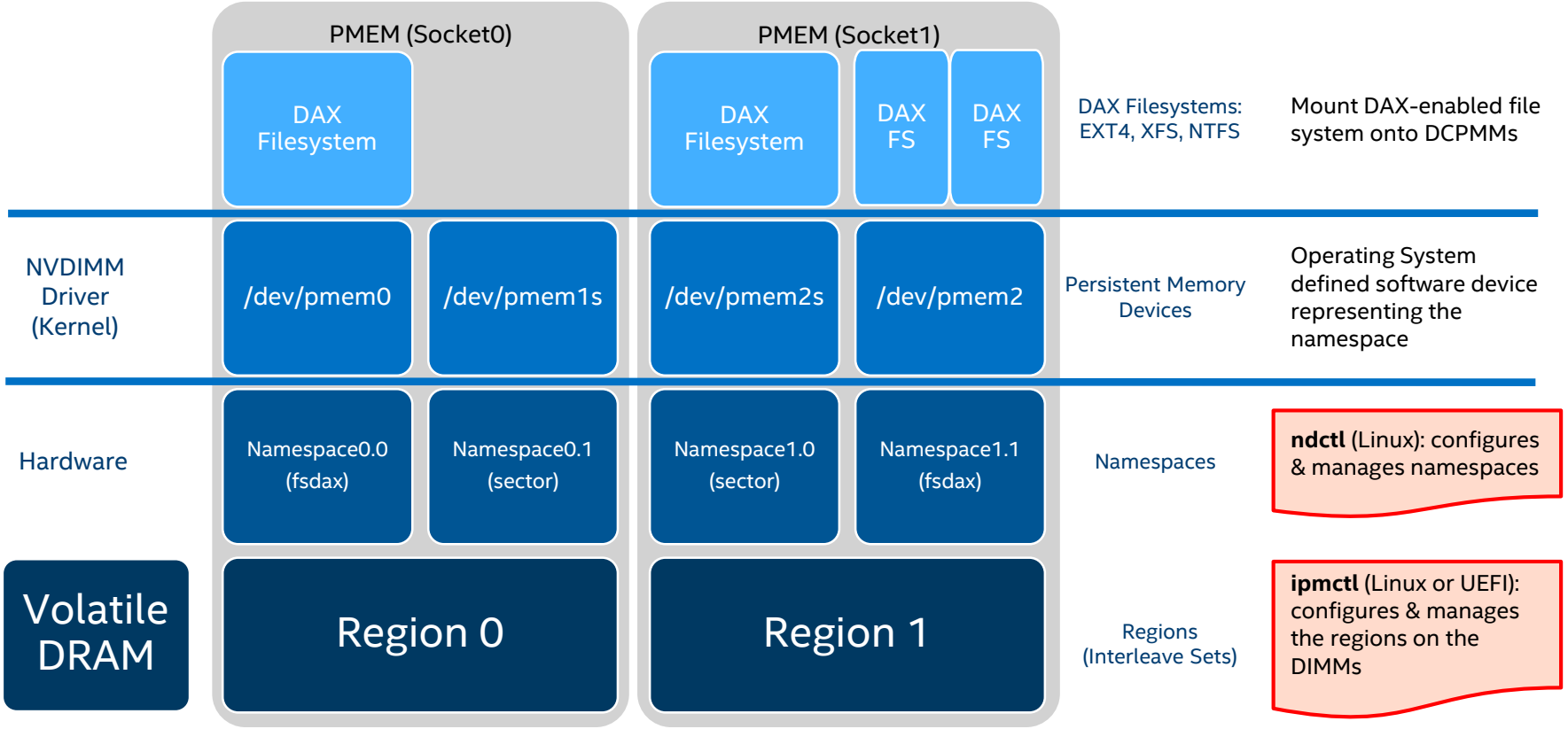
RAW

- Acts like a Memory Disk
- No DAX Support

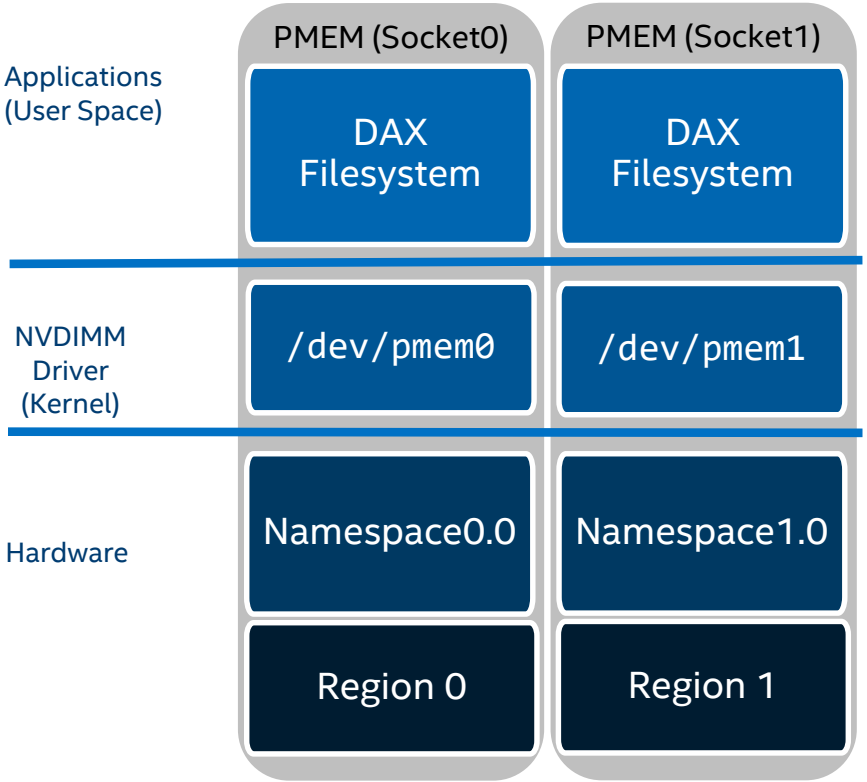
See:

- https://docs.pmem.io/ndctl-users-guide/managing-namespaces
- # man ndctl-create-namespace

STEP 2 & 3 – CREATING NAMESPACES & FILESYSTEMS



HOW TO CREATE DAX FILESYSTEMS (FSDAX)




```
# mkfs.ext4 /dev/pmem0
# mkfs.ext4 /dev/pmem1
# mount -o dax /dev/pmem0 /pmemfs0
# mount -o dax /dev/pmem1 /pmemfs1
```

new

Defaults

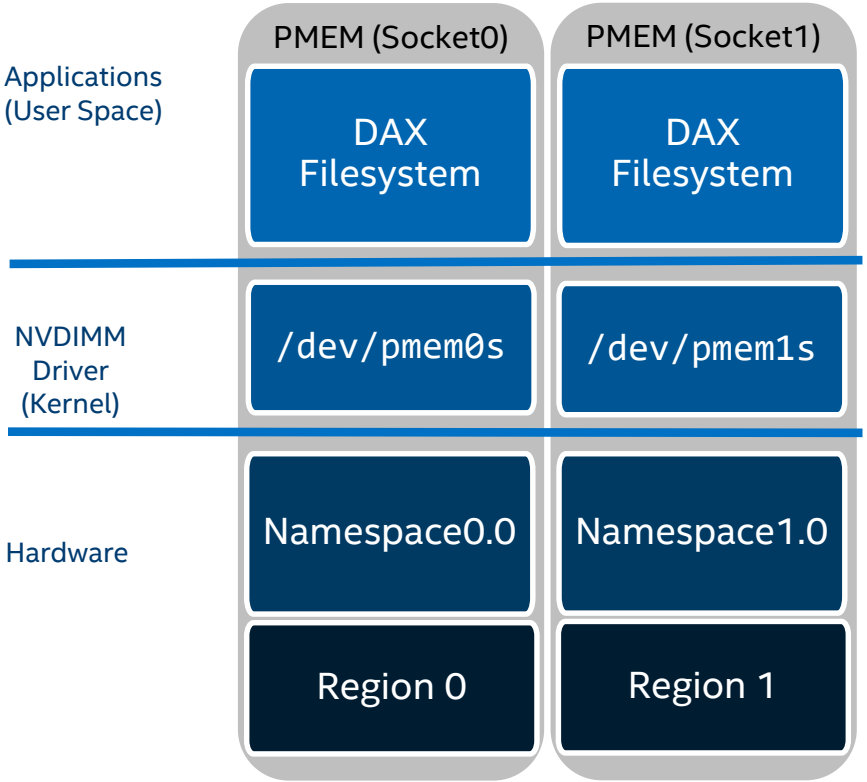
```
# ndctl create-namespace --region region0 --mode fsdax
# ndctl create-namespace --region region1 --mode fsdax
```

Vendor Neutral

```
# ipmctl create -goal PersistentMemoryType=AppDirect
# reboot
```

Vendor Specific

HOW TO CREATE BLOCK ATOMIC FILESYSTEMS (SECTOR)




```
# mkfs.ext4 /dev/pmem0s
# mkfs.ext4 /dev/pmem1s
# mount /dev/pmem0s /pmemfs0
# mount /dev/pmem1s /pmemfs1
```

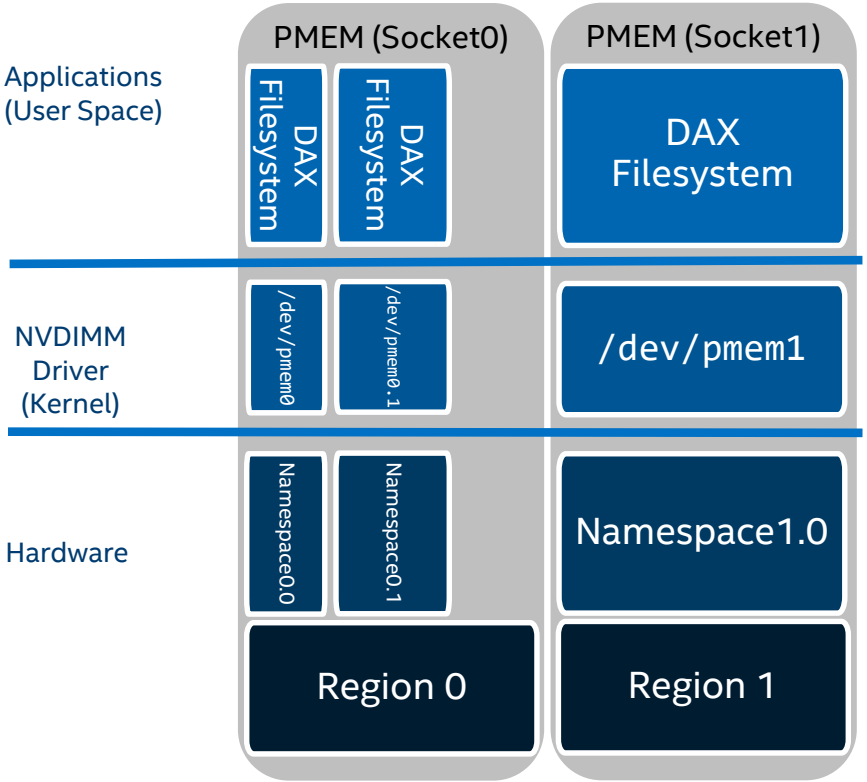
```
# ndctl create-namespace --region region0 --mode sector
# ndctl create-namespace --region region1 --mode sector
```

Vendor Neutral

```
# ipmctl create -goal PersistentMemoryType=AppDirect
# reboot
```

Vendor Specific

HOW TO PARTITION REGIONS



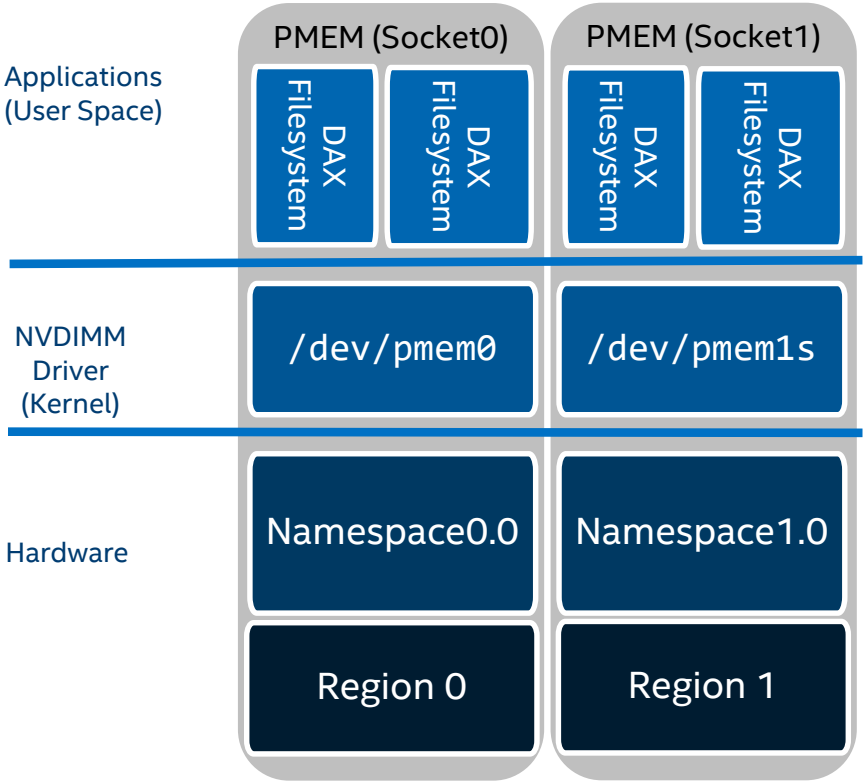

```
# mkfs.ext4 /dev/pmem0
# mkfs.xfs  /dev/pmem0.1
# mkfs.ext4 /dev/pmem1
# mount -o dax /dev/pmem0   /pmemfs0-0
# mount -o dax /dev/pmem0.1 /pmemfs0-1
# mount -o dax /dev/pmem1   /pmemfs1-0



# ndctl create-namespace --region region0 --mode fsdax\
    --size 50g
# ndctl create-namespace --region region0 --mode fsdax\
    --size 200g

# ndctl create-namespace --region region1 --mode fsdax
```

Vendor Neutral

```
# ipmctl create -goal PersistentMemoryType=AppDirect
# reboot
```

Vendor Specific

HOW TO PARTITION NAMESPACES (FSDAX & SECTOR)




```
# fdisk /dev/pmem0
<Create partition table>
# fdisk /dev/pmem1s
<Create partition table>
# mkfs.ext4 /dev/pmem0p0
# mkfs.xfs  /dev/pmem0p1
# mkfs.ext4 /dev/pmem1sp0
# mkfs.ext4 /dev/pmem1sp1
# mount -o dax /dev/pmem0p0 /pmemfs0-0
# mount -o dax /dev/pmem0p1 /pmemfs0-1
# mount /dev/pmem1sp0 /pmemfs1-0
# mount /dev/pmem1sp1 /pmemfs1-1

# ndctl create-namespace --region region0 --mode fsdax
# ndctl create-namespace --region region1 --mode sector
```

Vendor Neutral

```
# ipmctl create -goal PersistentMemoryType=AppDirect
# reboot
```

Vendor Specific

RESOURCES

- **Intel PMEM Developer Zone** - https://software.intel.com/pmem
- **Webinar - Provisioning Intel® Optane Data Center Persistent Memory**
- Quick Start Guides
  - ndctl
  - Get Started with Intel® Optane™ DC Persistent memory - Short Video
  - Quick Start Guide – Tutorial
- ipmctl on Github
- ndctl on Github

Q&A

BACKUP

VIRTUALIZED ENVIRONMENTS

CONFIGURE INTEL® OPTANE™ DC PERSISTENT MEMORY IN VIRTUALIZED ENVIRONMENTS

- VMware* vSphere
  - https://vspherecentral.vmware.com/t/hardware-acceleration/persistent-memory-pmem/
- KVM/QEMU
  - https://docs.pmem.io/getting-started-guide/creating-development-environments/virtualization/qemu
- Hyper-V
  - https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/manage/persistent-memory-cmdlets
- Kubernetes
  - Persistent Memory Container Storage Interface (PMEM-CSI) is a storage driver for container orchestrators like Kubernetes.