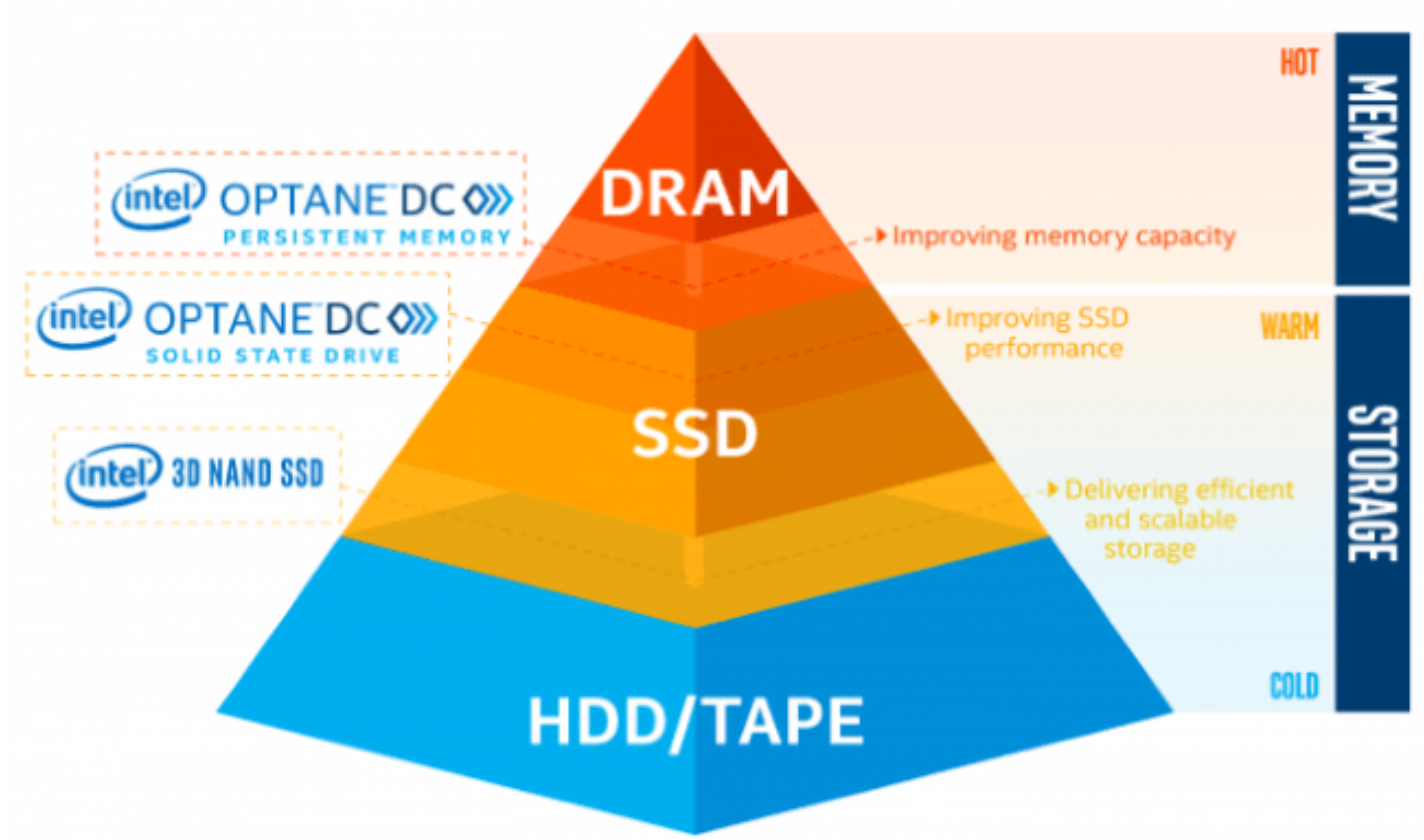
# INTEL® OPTANE™ DC PERSISTENT MEMORY INTRODUCTION

Speaker：Dennis, Wu, dennis.wu@intel.com
Title: software engineer

# Agenda

- What is Persistent Memory
- Hardware Configuration Options
- Persistent Memory Operating Modes
- Linux Utilities
- Demos

# Re-architecting the Memory /



# HARDWARE

CPU, DRAM, and Intel® Optane™ DC Persistent Memory

NEXT GEN INTEL® XEON® SCALABLE PROCESSOR

# Cascade Lake

## With Intel® OPTANE™ DC PERSISTENT MEMORY

Leadership Performance

Optimized Cache Hierarchy

Higher Frequencies


Security Mitigations

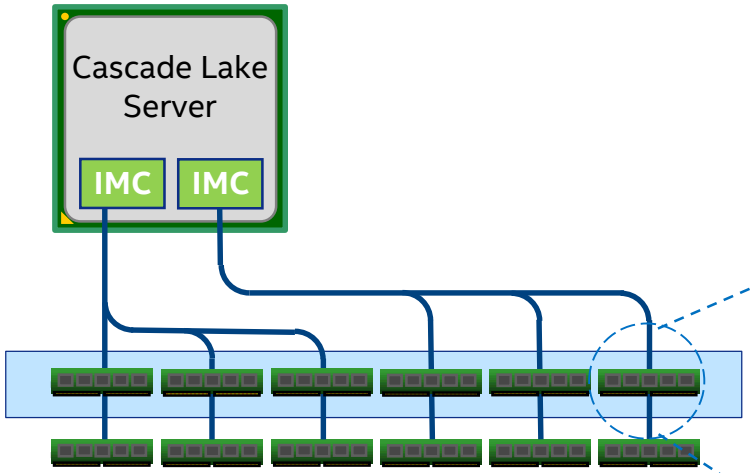
Intel Deep Learning Boost (VNNI)

Optimized Frameworks & Libraries

**BUILDING ON 20 YEARS OF XEON INNOVATION**

# INTEL® OPTANE™ DC PERSISTENT MEMORY - PRODUCT OVERVIEW

(Optane™ based Memory Module for the Data Center)

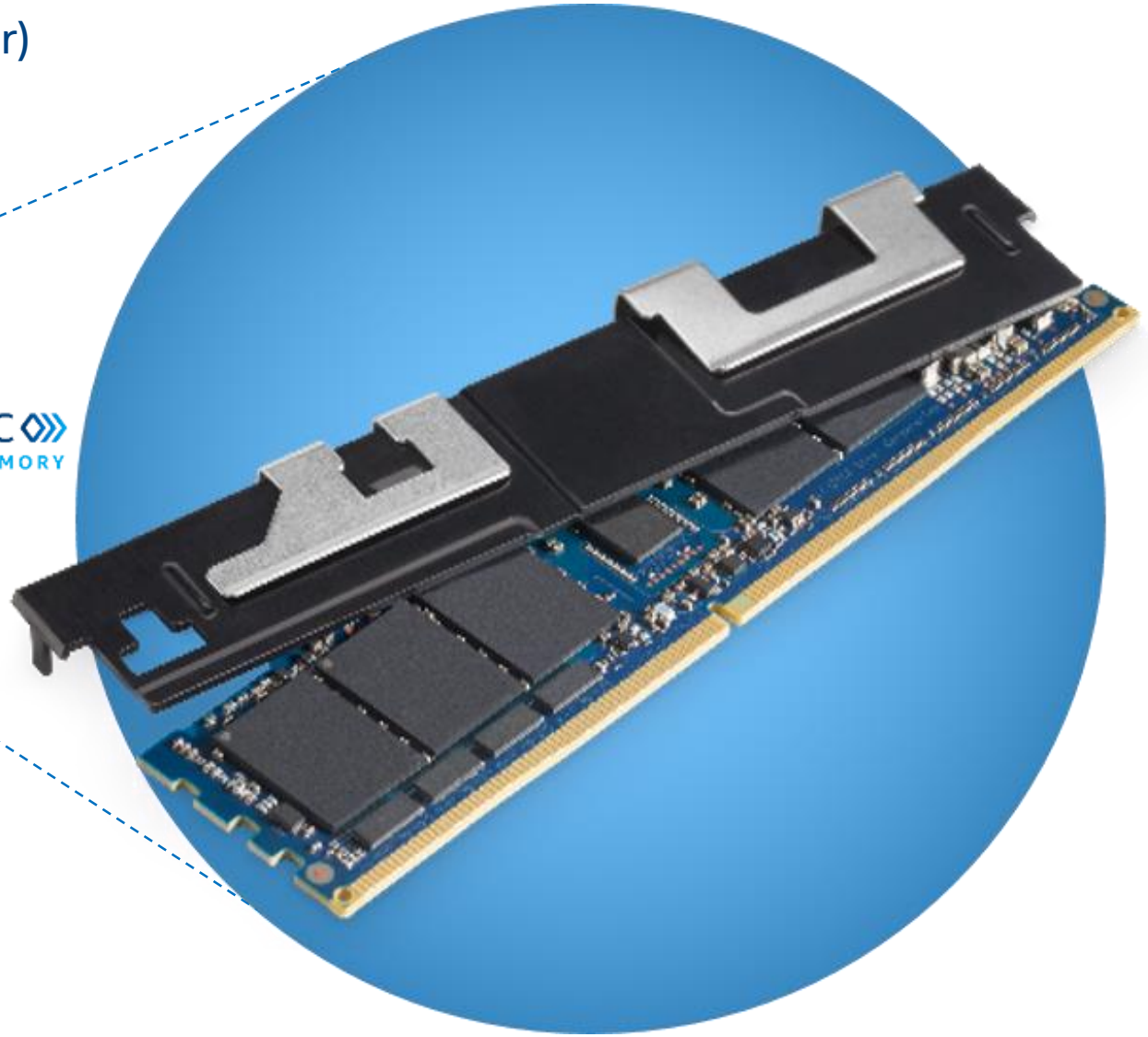



* DIMM population shown as an example only.

**DIMM Capacity**
- 128, 256, 512GB

**Speed**
- 2666 MT/sec

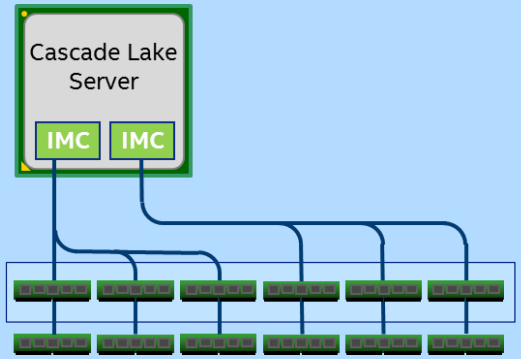
**Capacity per CPU**
- 3TB (not including DRAM)




- DDR4 electrical & physical
- Close to DRAM latency
- Cache line size access

# MEMORY SLOT POPULATION EXAMPLES†

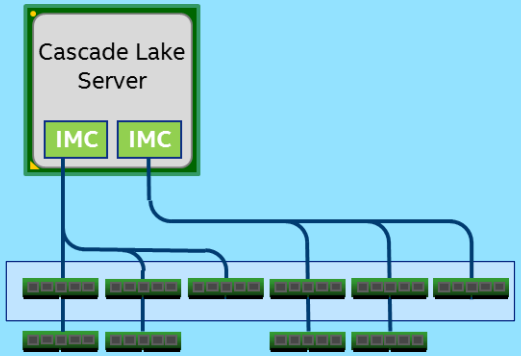
## 2-2-2




**12 slots per CPU**

Max memory capacity and bandwidth

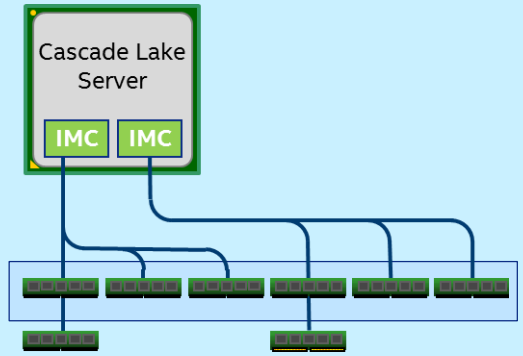
## 2-2-1




**10 slots per CPU**

Trade DDR bandwidth for smaller board real estate on the DIMM slots

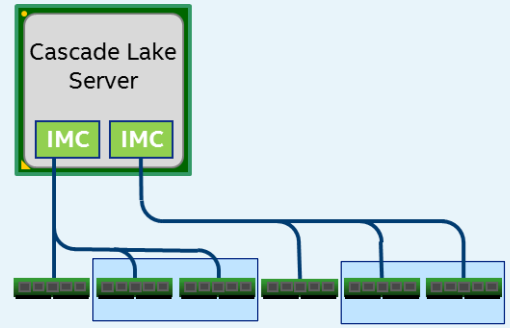
## 2-1-1*




**8 slots per CPU**

Trade DDR bandwidth for smaller board real estate on the DIMM slots

## 1-1-1



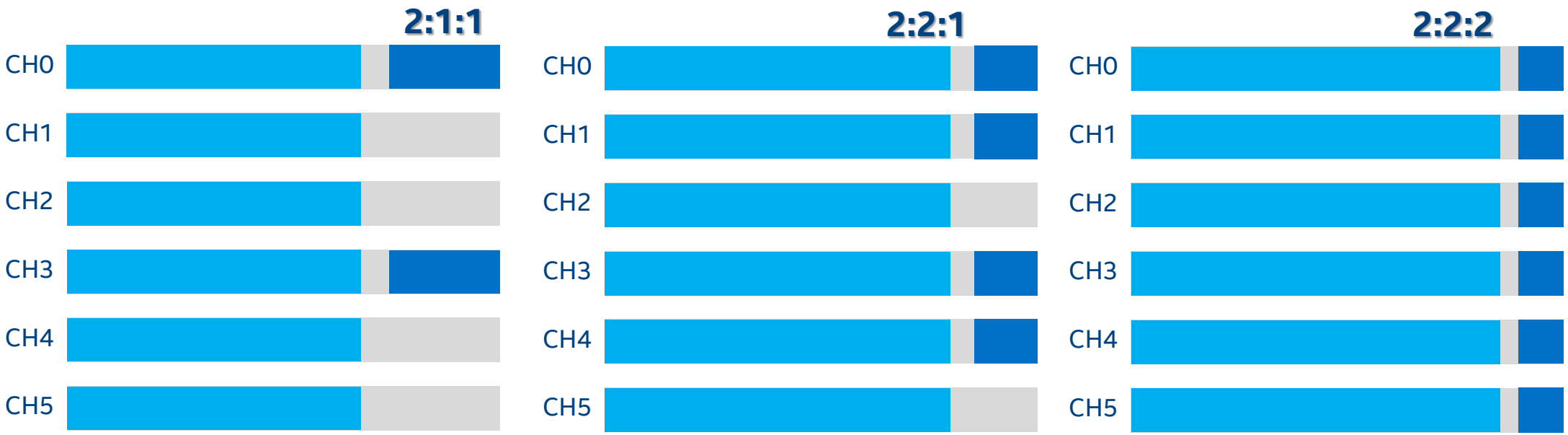

**6 slots per CPU**

Least number of DIMM slots to utilize max memory bandwidth

Refer to the Intel Optane DC Persistent Memory Population Matrix for full list of supported configuration options

* No difference on functionality or performance when 2nd DIMM slot is in channel 0, 1 or 2 for that integrated memory controller (IMC)
† DIMM slots shown. While DRAM DIMMs can populate all slots shown, DCPMM is only populated in slot closest to CPU in each channel.

# CHANNEL SHARING: MORE DIMMS IS A GOOD IDEA




Channel BW

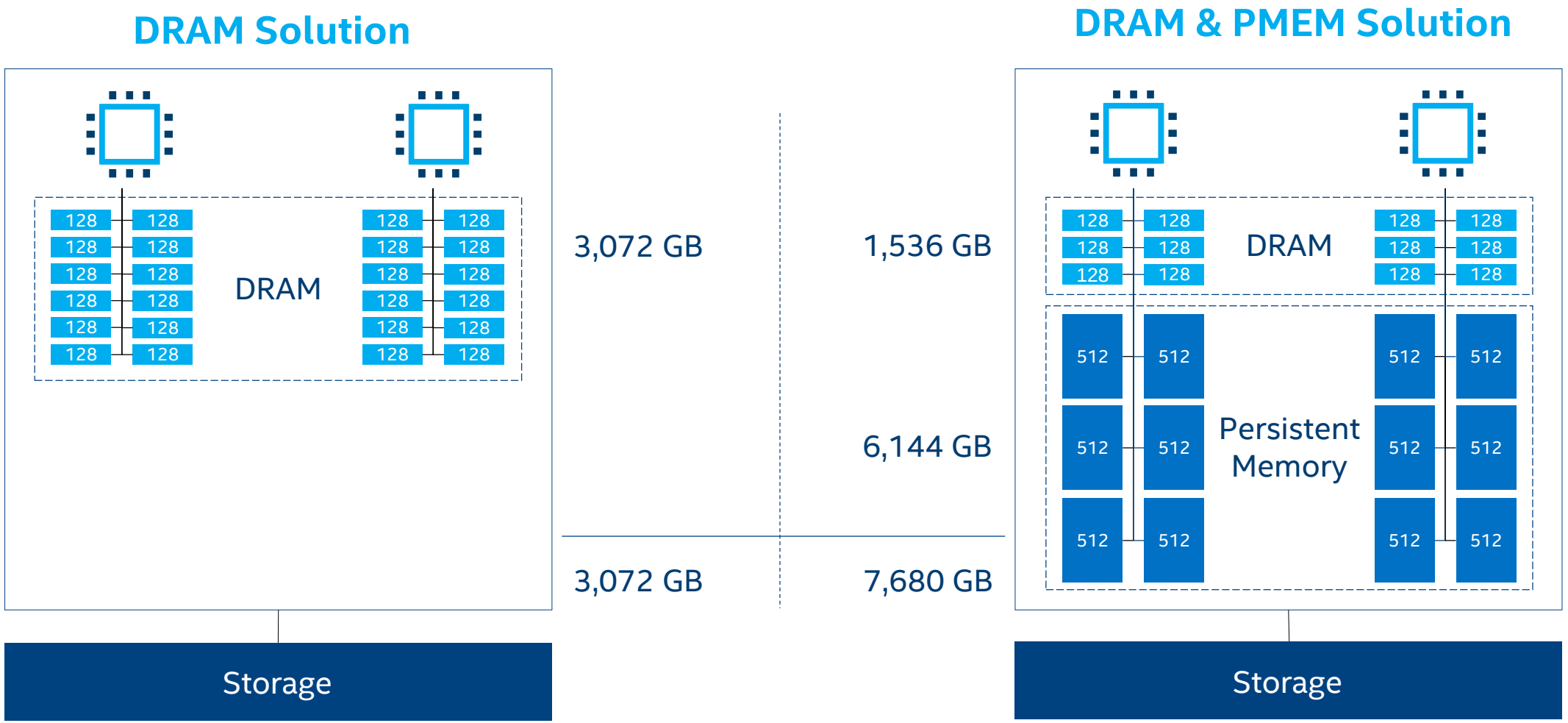
DRAM BW

DCPMM/DRAM (or any other device) BW

Distributed DCPMM BW over more DIMMs help:

- Reduce the pressure on DRAM
- Provide more headroom for DCPMM BW

# INTEL® OPTANE™ DC PERSISTENT MEMORY



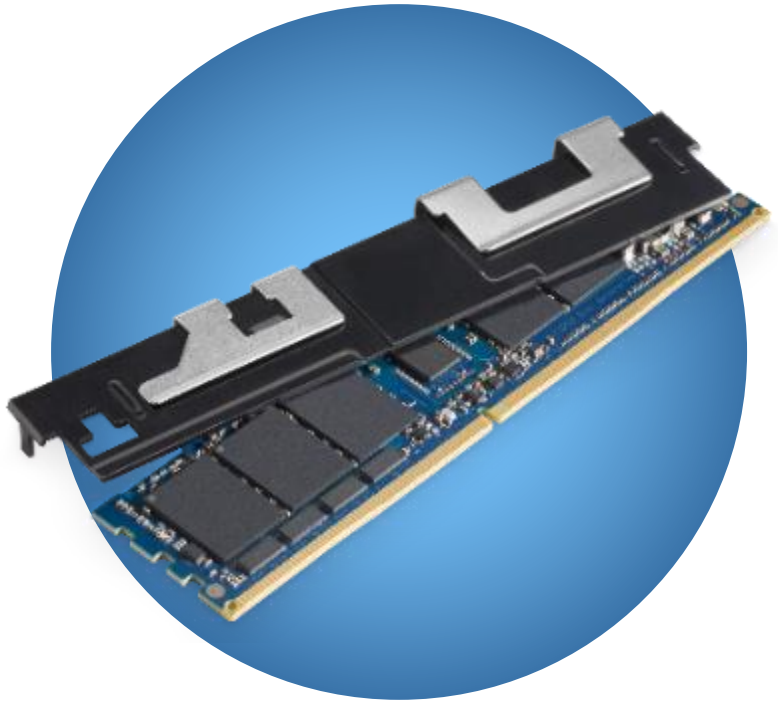
# PERSISTENT MEMORY OPERATING MODES

Memory Mode & AppDirect

# INTEL® OPTANE™ DC PERSISTENT MEMORY - OPERATIONAL MODES

## APP DIRECT MODE

- Persistent
- High availability / less downtime
- Significantly faster storage




## MEMORY MODE

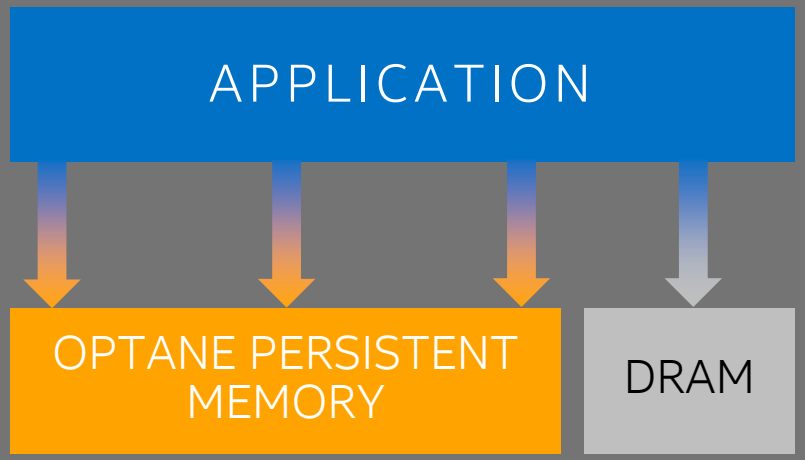
- High capacity
- Affordable
- Ease of adoption†

Note that a BIOS update will be required before using Intel persistent memory

# INTEL® OPTANE™ DC PERSISTENT MEMORY
# SUPPORT FOR BREADTH OF APPLICATIONS

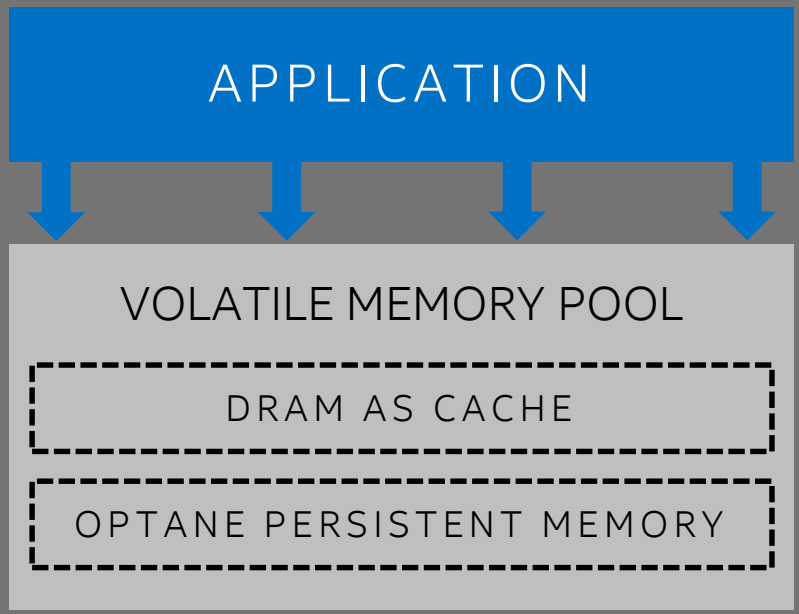
## APP DIRECT MODE

PERSISTENT PERFORMANCE & MAXIMUM CAPACITY



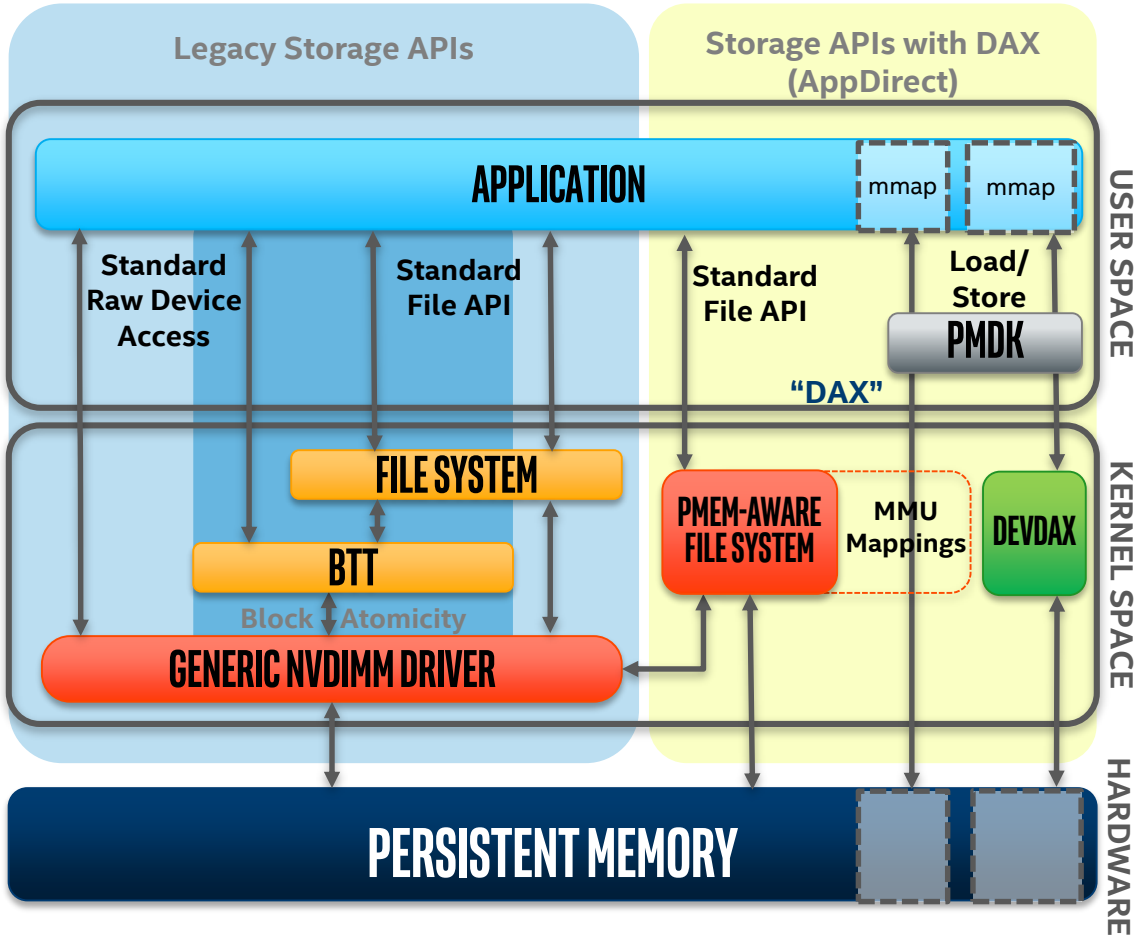

## MEMORY MODE

AFFORDABLE MEMORY CAPACITY FOR MANY APPLICATIONS



# APP DIRECT MODE OPTIONS

- No Code Changes Required
- Operates in Blocks like SSD/HDD
  - Traditional read/write
  - Works with Existing File Systems
  - Atomicity at block level
  - Block size configurable
    - 4K, 512B*
- NVDIMM Driver required
  - Support starting Kernel 4.2
- Configured as Boot Device
- Higher Endurance than Enterprise SSDs
- High Performance Block Storage
  - Low Latency, higher BW, High IOPs




- Code changes may be required*
- Bypasses file system page cache
- Requires DAX enabled file system
  - XFS, EXT4, NTFS
- No Kernel Code or interrupts
- No interrupts
- Fastest IO path possible

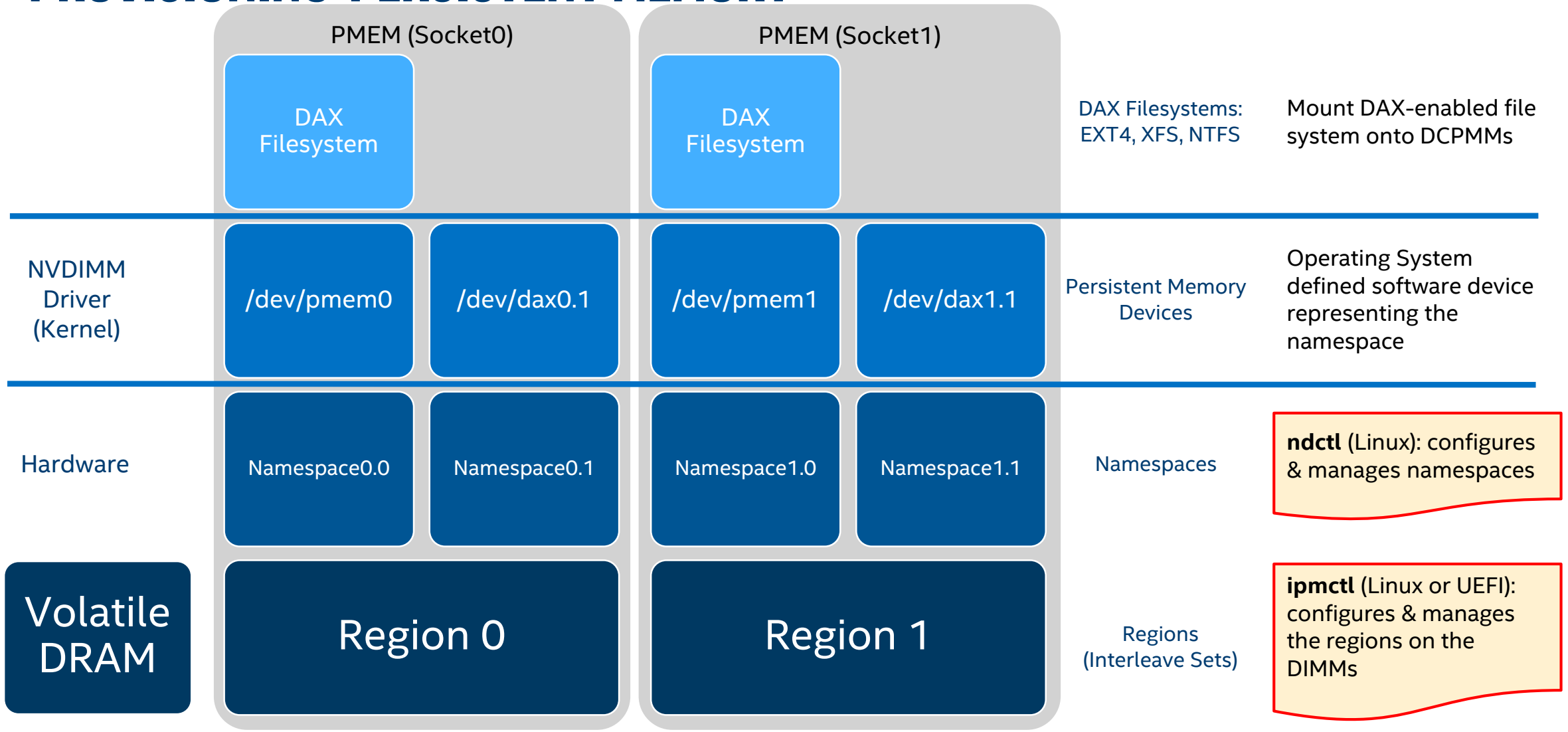
* Code changes required for load/store direct access if the application does not already support this.

*Requires Linux

# PROVISIONING PERSISTENT MEMORY

Memory Mode & AppDirect

# PROVISIONING PERSISTENT MEMORY



# DEMO'S

Memory Mode & AppDirect

# Demo List

- Introduce ipmctl
- Configure Memory Mode
- Configure App Direct
- Introduce ndctl
- Create an FSDAX namespace
- Create a DEVDAX namespace
- Create a SECTOR namespace

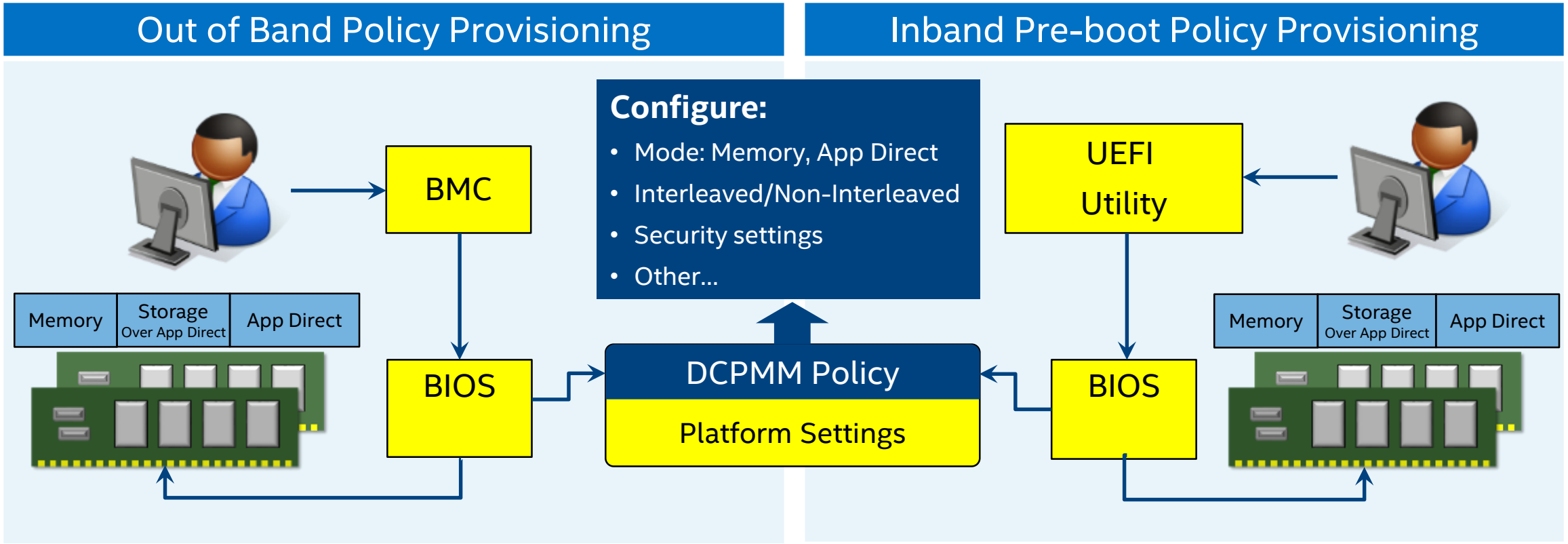
# Resources

- **ipmctl**: https://github.com/intel/ipmctl
- **ndctl**: https://github.com/pmem/ndctl
- **https://docs.pmem.io**
  - Quick Start Guides (Persistent Memory)
  - Getting Started Guides (Persistent Memory)
  - NDCTL User Guide
- **Intel PMEM Developer Zone** – https://software.intel.com/pmem
  - Videos
  - Knowledge Articles
  - PMDK Code Examples

# Q&A

# BACKUP

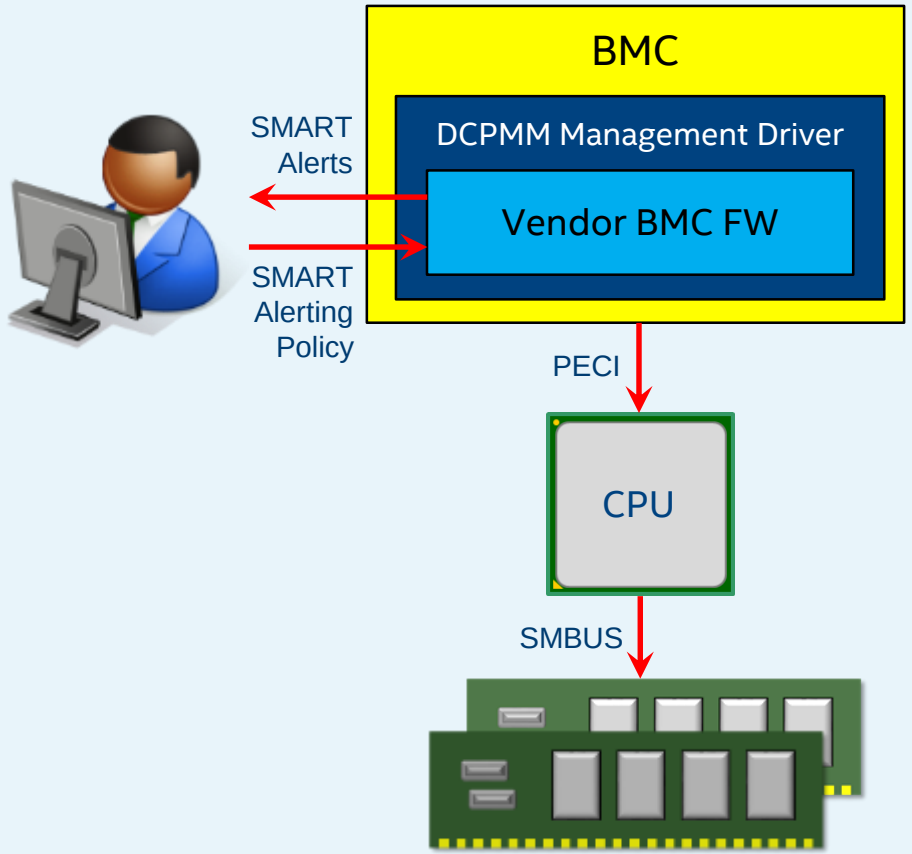
# OUT-OF-BAND & IN-BAND PROVISIONING



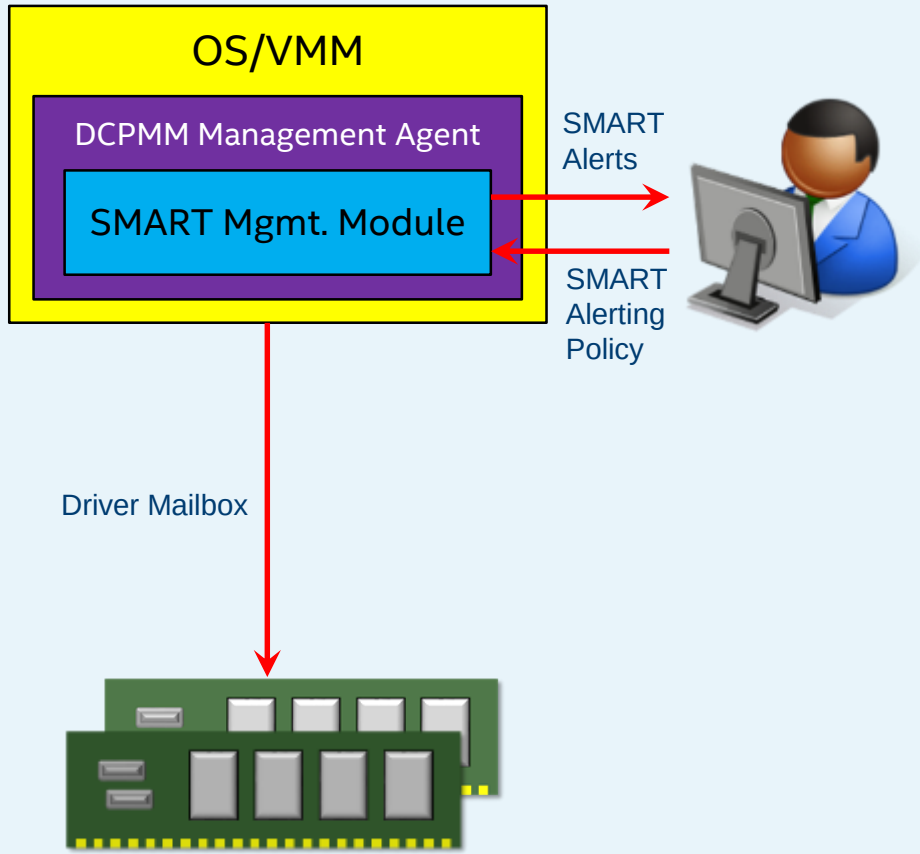

- DCPMM can simultaneously support Memory and App Direct partitions, partitioning done at boot time
- Datacenter manager can communicate partitioning policy (in response to workload needs) to the platform agent
- Allocation within a partition under VMM/OS/CRNL controls
- BIOS to initializes DCPMM and sets up partitioning based on policy

# MONITORING

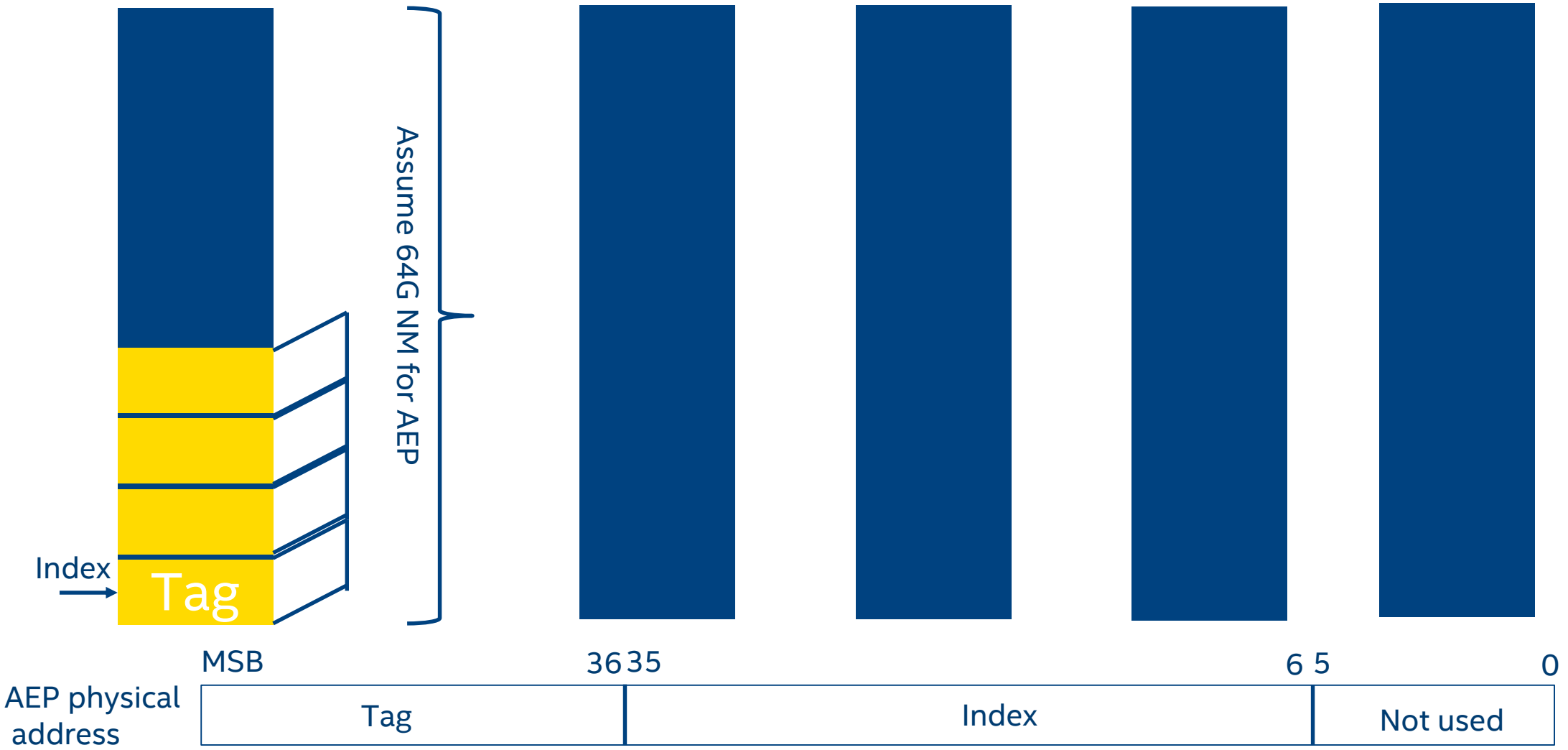
## Out of Band Monitoring



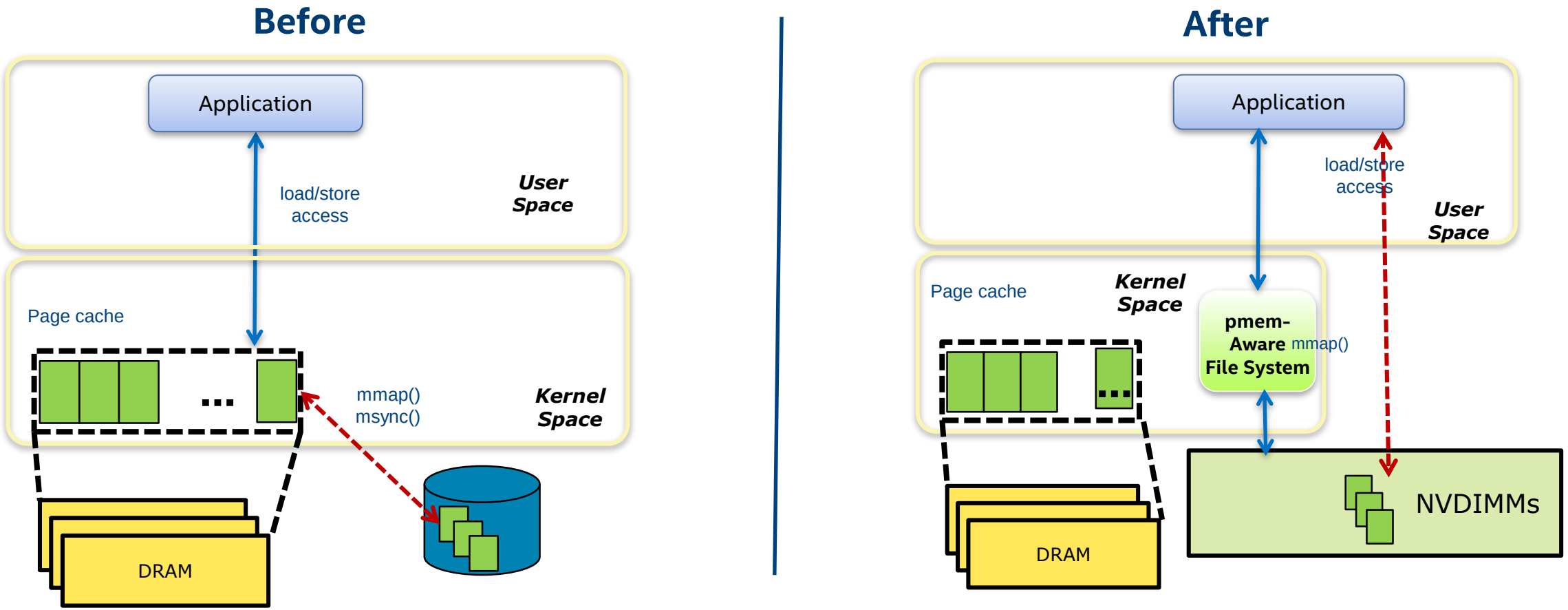

## Inband Monitoring



# DRAM as direct mapped cache

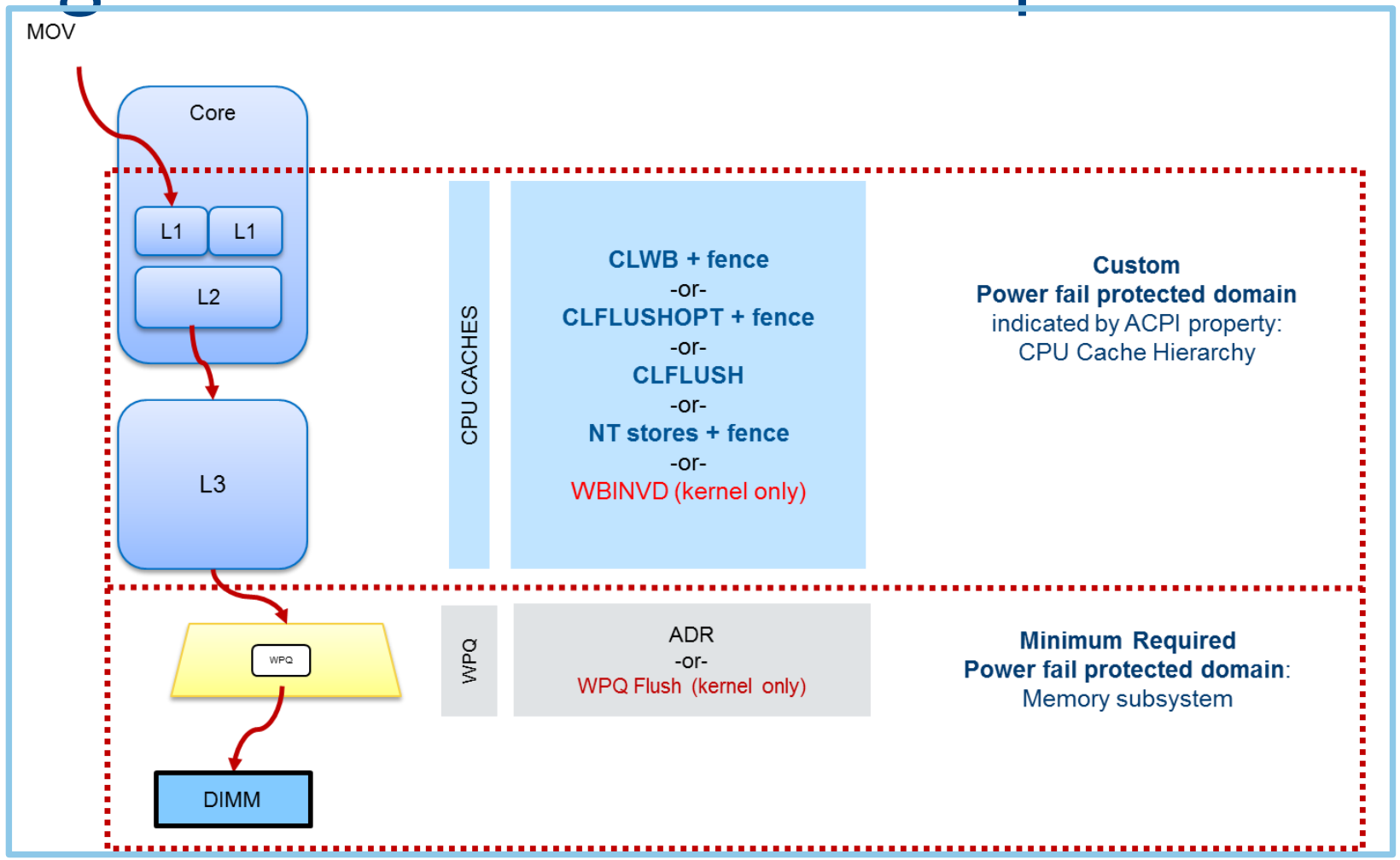


# BYTE ADDRESSABLE STORAGE WITH MEMORY MAPPED FILES



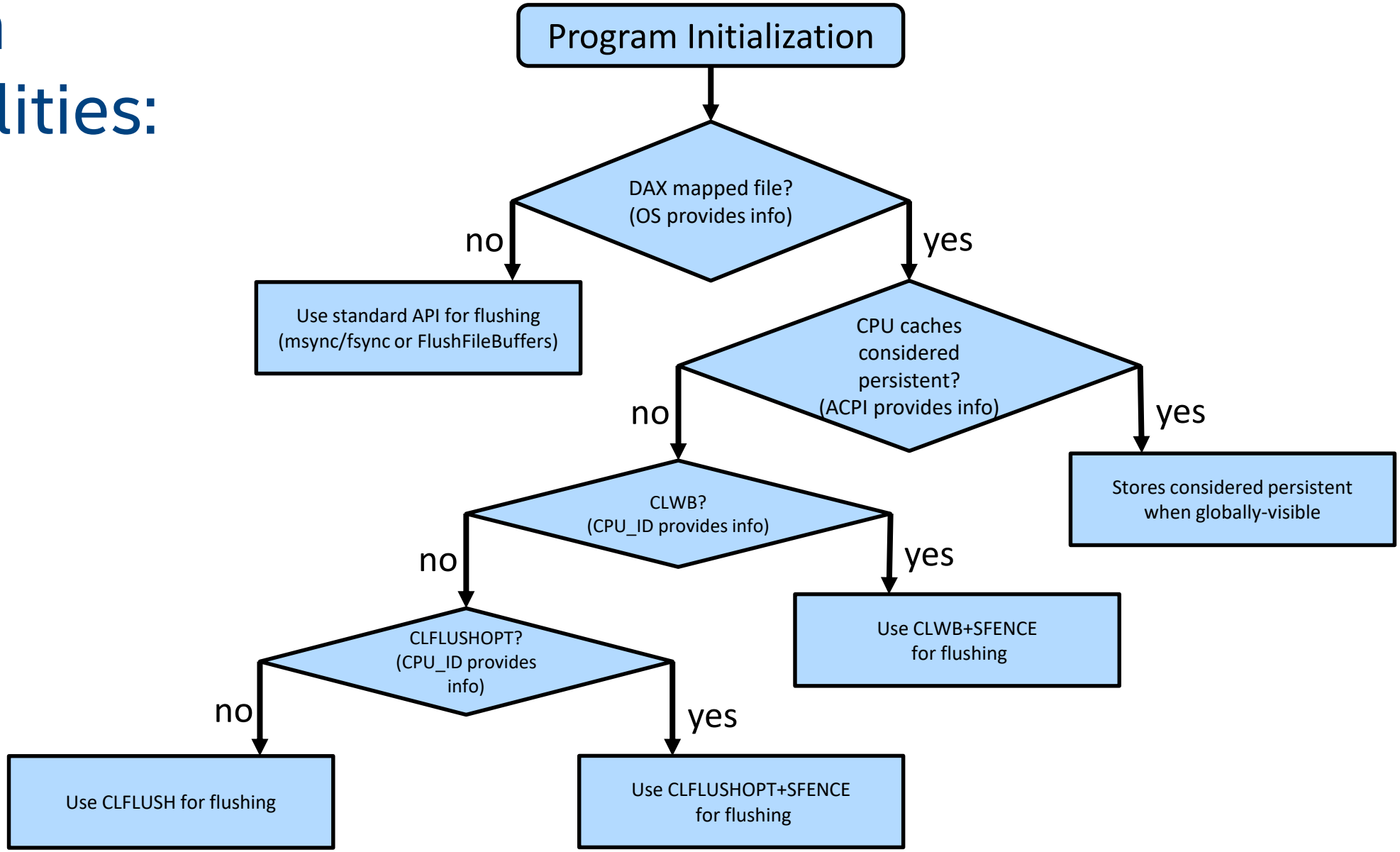

Avoid Overhead of Paging/Context Switching into Kernel

# Leverage clflush for Cached Sequential Writes*




- LLC WB evictions lead to near random behavior (lower BW). SW recommendation: Do CLFLUSH often enough to avoid LLC evictions.

# Application Responsibilities: Flushing



# Application Responsibilities: Consistency

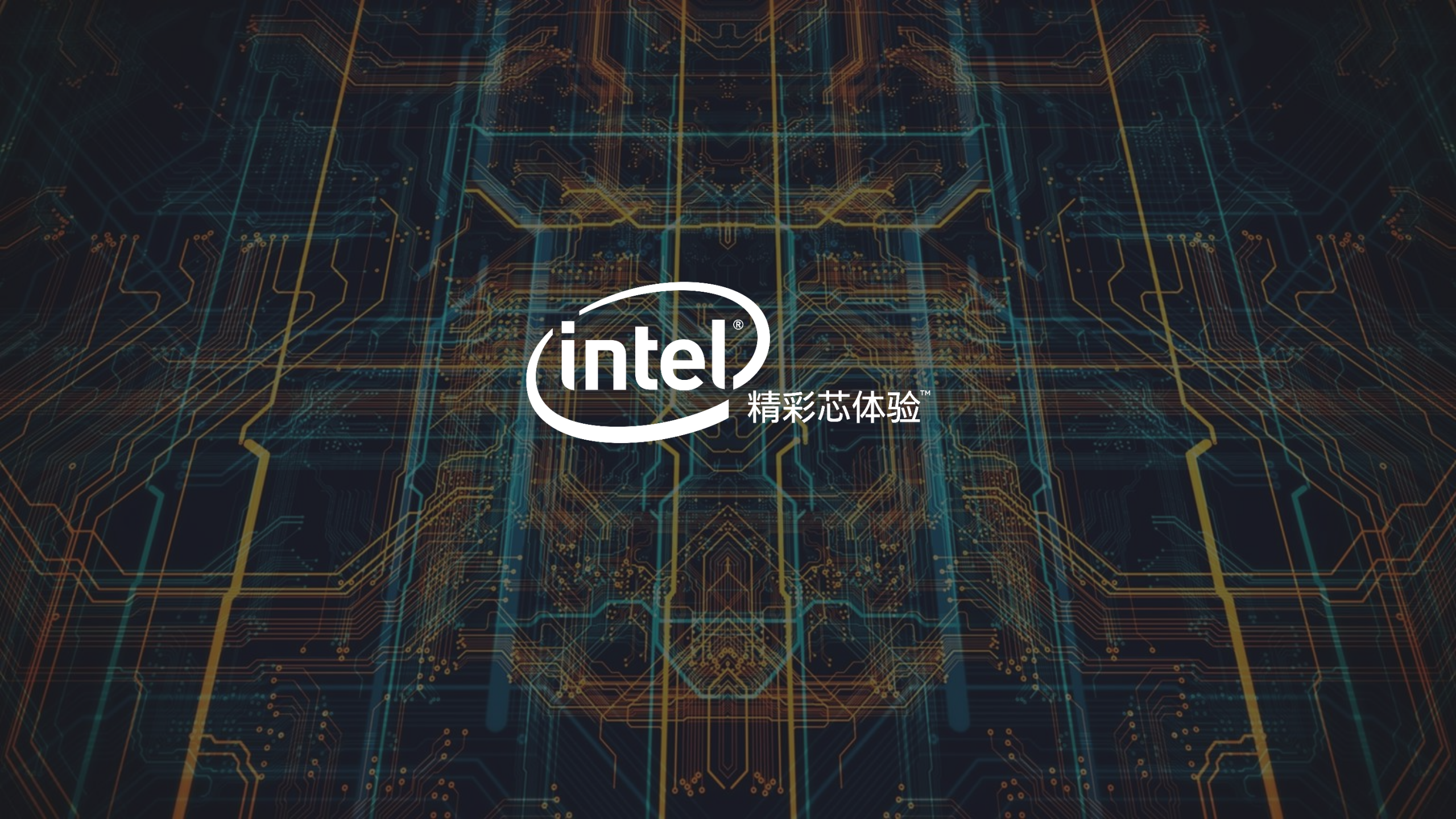
```
open(…);

mmap(…);

strcpy(pmem, "Hello, World!");

pmem_persist(pmem, 14);
```

Crash

`pmem_persist()` may be faster,
but is still **not** transactional

Result

1. "\0\0\0\0\0\0\0\0\0\0..."
2. "Hello, W\0\0\0\0\0\0..."
3. "\0\0\0\0\0\0\0\0orld!\0"
4. "Hello, \0\0\0\0\0\0\0\0"
5. "Hello, World!\0"


intel®
精彩芯体验™


谢谢