



Take Advantage of Intel Optane DCPM in Flink

Ma Yan · Intel / Software
Engineer
Apache Flink Community China



Apache Flink

CONTENT

目录 >>

01 /

What's Intel Optane DC Persistent Memory?

02 /

Modes and Population

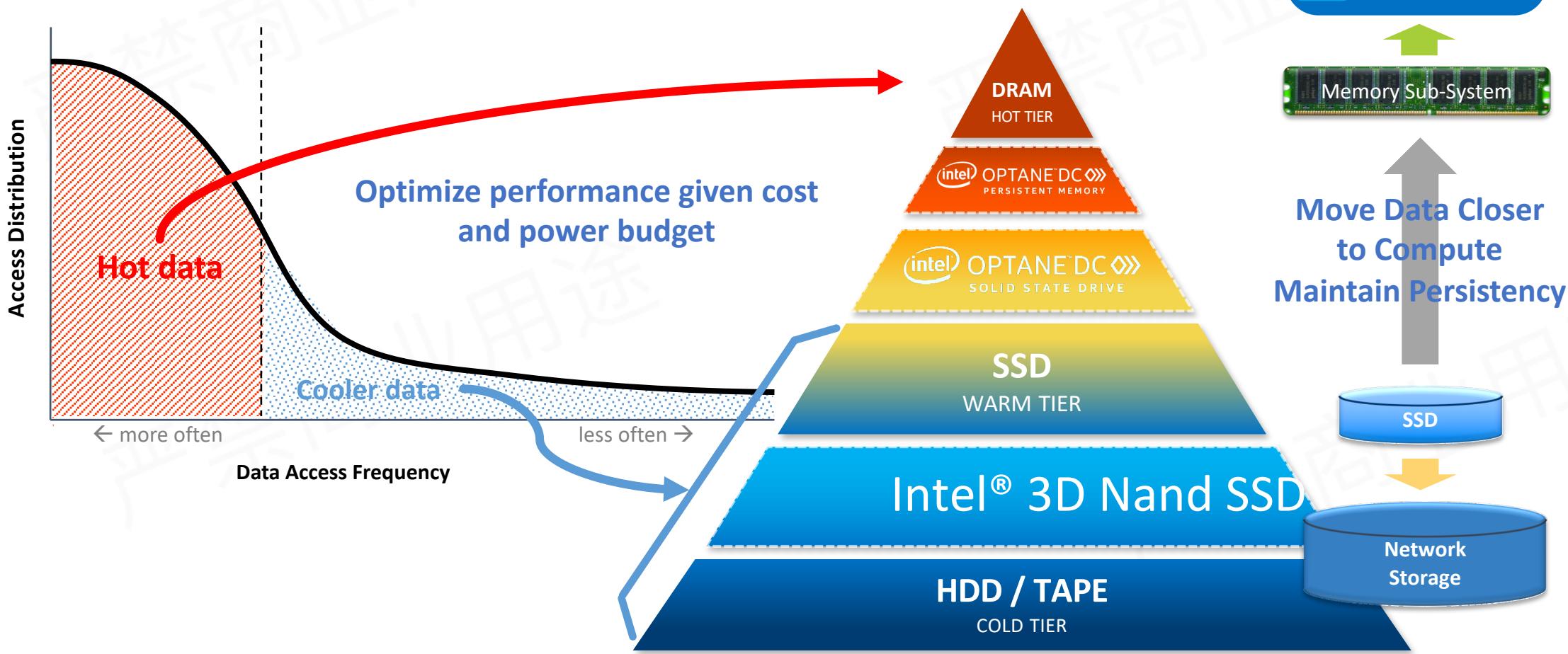
03 /

How to use it in Flink workload?

01

What's Intel Optane DC Persistent Memory?

Goal: Efficient Data Centric Architecture



Apache Flink

The best of both worlds with Intel® Optane™ DC Persistent Memory

Memory

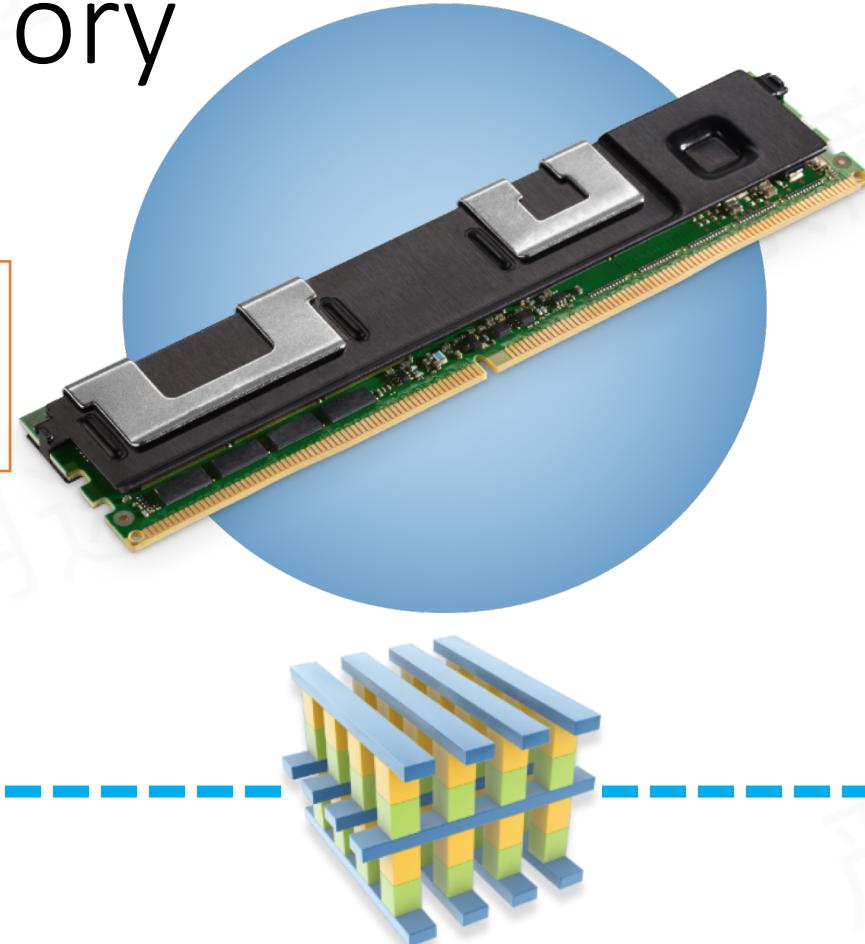
attributes

Performance comparable to DRAM at low latencies¹

Storage

attributes

Data persistence with higher capacity than DRAM²



1. "Fast performance comparable to DRAM" - Intel persistent memory is expected to perform at latencies near DDR4 DRAM. Benchmarks and proof points forthcoming. "low latencies" - Data transferred across the memory bus causes latencies to be orders of magnitude lower when compared to transferring data across PCIe or I/O bus' to NAND/Hard Disk. Benchmarks and proof points forthcoming.

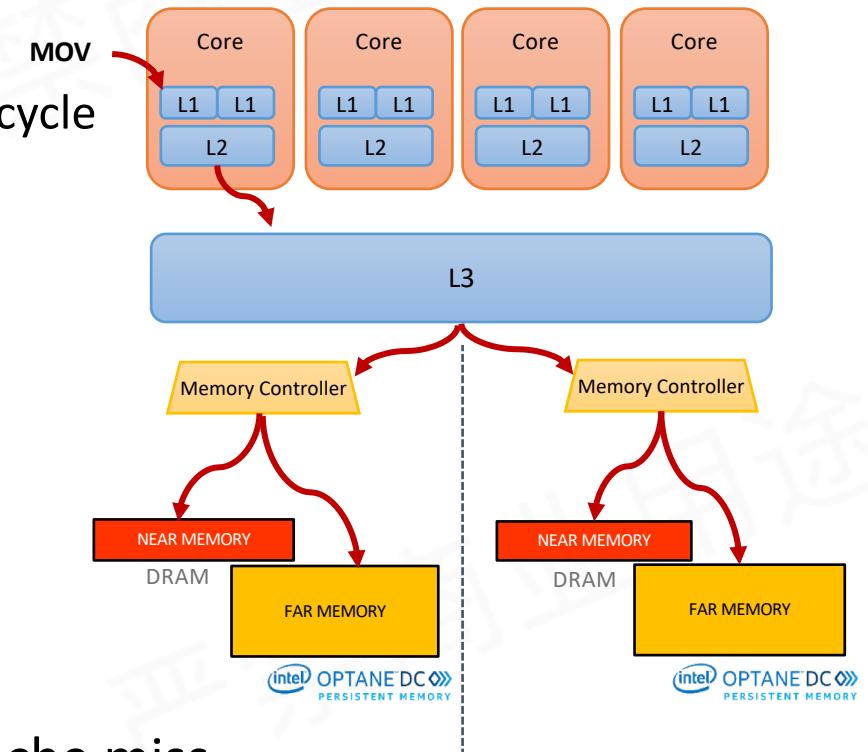
2. Intel persistent memory offers 3 different capacities – 128GB, 256GB, 512GB. Individual DIMMs of DDR4 DRAM max out at 256GB.

02

Modes and Population

Memory Mode

- No software/application changes required
- To mimic traditional memory, data is “volatile”
 - Volatile mode key cleared and regenerated every power cycle
- DRAM is “near memory”
- Used as a write-back cache
- Managed by host memory controller
- Within the same host memory controller, not across
- Ratio of far/near memory (PMEM/DRAM) can vary
- Overall latency
- Same as DRAM for cache hit
- Intel® Optane™ DC persistent memory + DRAM for cache miss

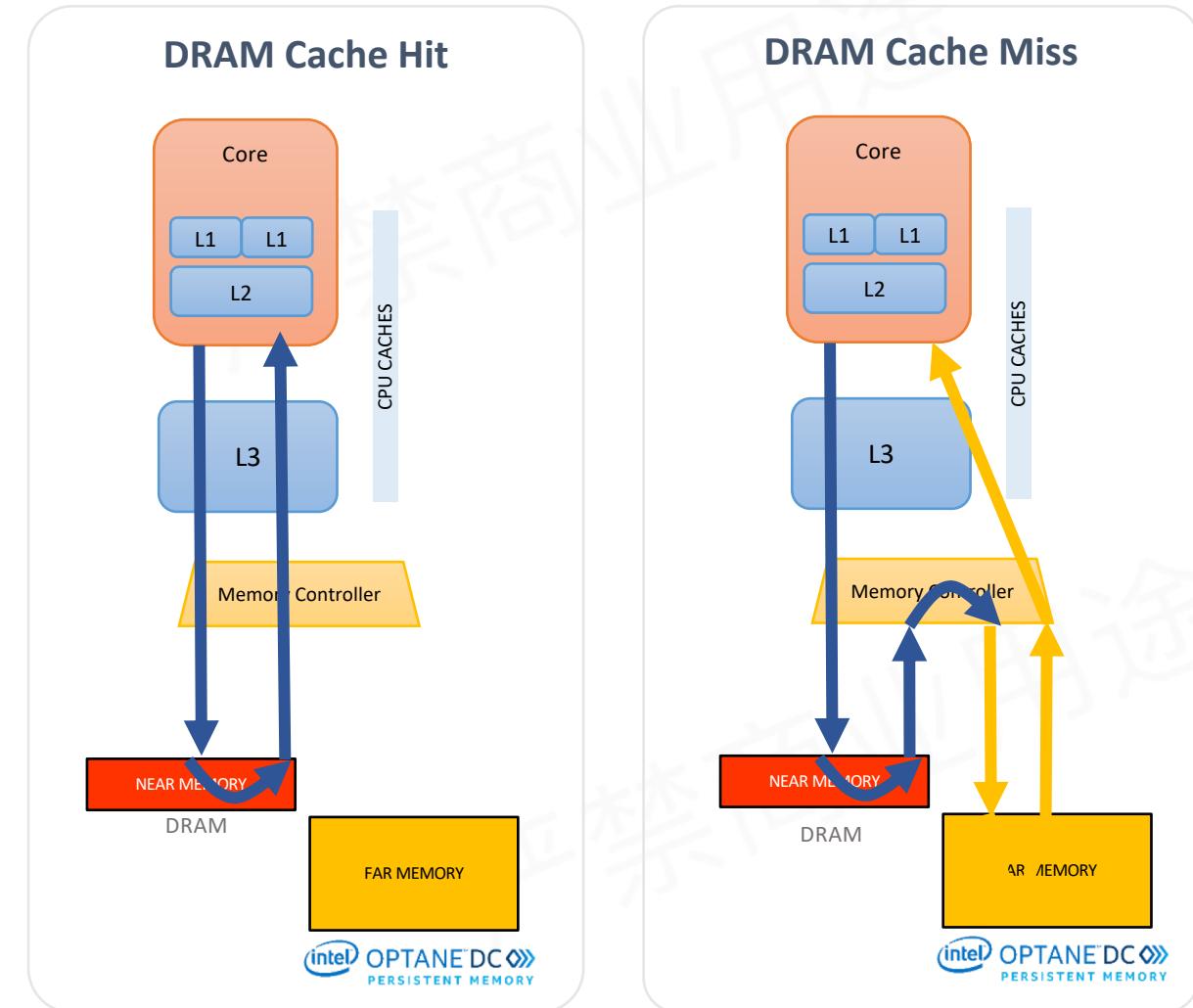


<https://itpeernetwork.intel.com/intel-optane-dc-persistent-memory-operating-modes/#gs.qwufar>

Memory Mode Transaction Flow

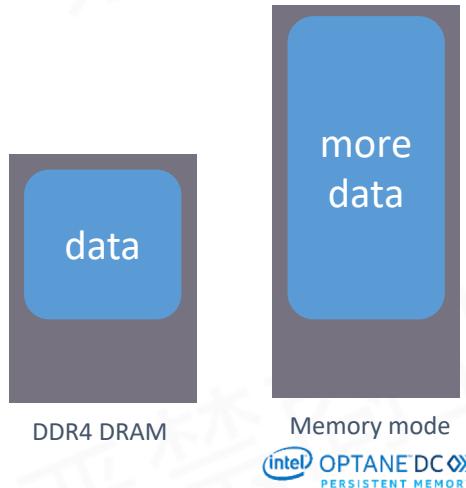


- Good locality means near-DRAM performance
 - Cache hit: latency same as DRAM
 - Cache miss: latency DRAM + Intel® Optane™ DC persistent memory
- Performance varies by workload
 - Best workloads have the following traits:
 - Good locality for high DRAM cache hit rate
 - Low memory bandwidth demand
 - Other factors:
 - #reads > #writes
 - Config vs. Workload size

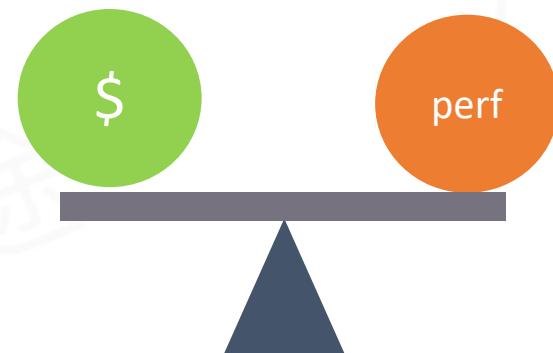


Larger Memory Capacity enables new usages

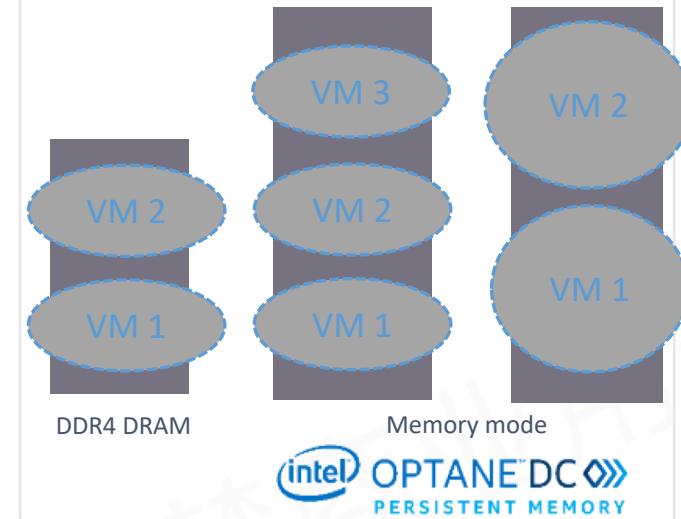
Larger databases and data sets



More options for TCO balancing



Larger vms, More VMs¹



```
[root@sr128 ~]# free -g
total        used        free      shared  buff/cache available
Mem:       187          5       181          0          0       181
Swap:        3          0          3          0          0          0
```

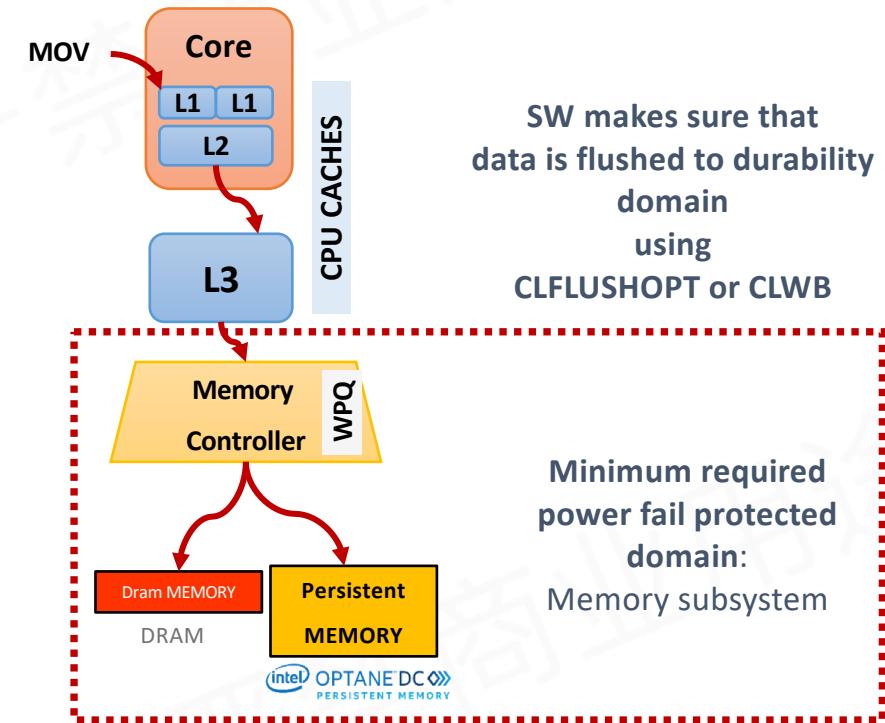


```
[root@sr128 ~]# free -g
total        used        free      shared  buff/cache available
Mem:       991         93       869          0          28       893
Swap:        3          0          3          0          0          0
```

App Direct Mode



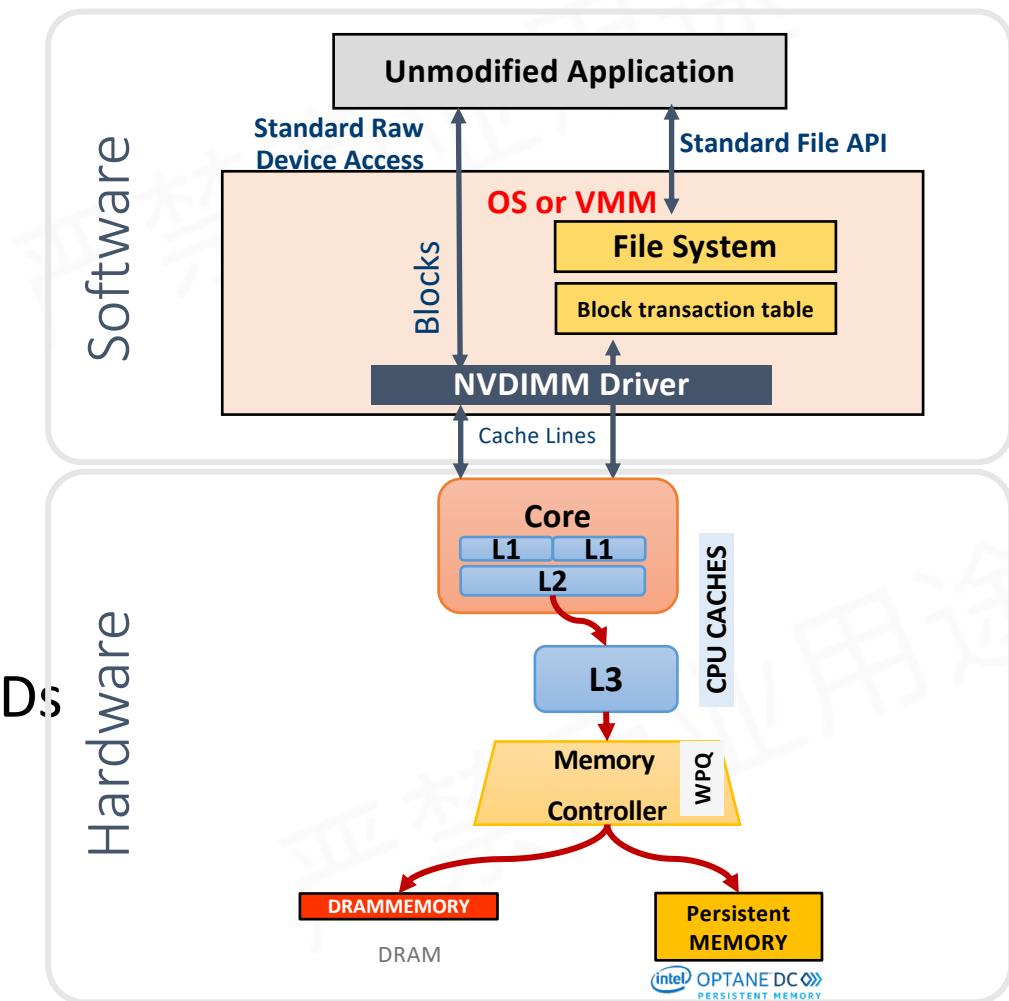
- PMEM-aware software/application required
 - Adds a new tier between DRAM and block storage (SSD/HDD)
 - Industry open standard programming model and Intel PMDK
- In-place persistence
 - No paging, context switching, interrupts, nor kernel code executes
- Byte addressable like memory
 - Load/store access, no page caching
- Cache Coherent
- Ability to do DMA & RDMA



Storage Over App Direct



- Operates in blocks like SSD/HDD
 - Traditional read/write instructions
 - Works with existing file systems
 - Atomicity at block level
 - Block size configurable (4K, 512B)
- NVDIMM driver required
 - Support starting kernel 4.2
- Scalable capacity
- Higher endurance than enterprise class SSDs
- High performance block storage
 - Low latency, higher bandwidth, high IOPs



Linux kernel and driver changes: https://www.youtube.com/watch?v=owmN_lcMK2M

App Direct Flexibility for developer to optimize

1. DRAM data and App Direct (Intel® Optane™ DC persistent memory) are separate regions in memory space
 - App Direct region can be used as persistent memory

2. Intel Optane DC persistent memory to enable larger memory data structures.
Some example partitioning:
 - Move all in-memory data to persistent memory (AD)
 - Move some in-memory data to persistent memory (AD), leaving most actively accessed data in DRAM

<https://software.intel.com/en-us/persistent-memory>

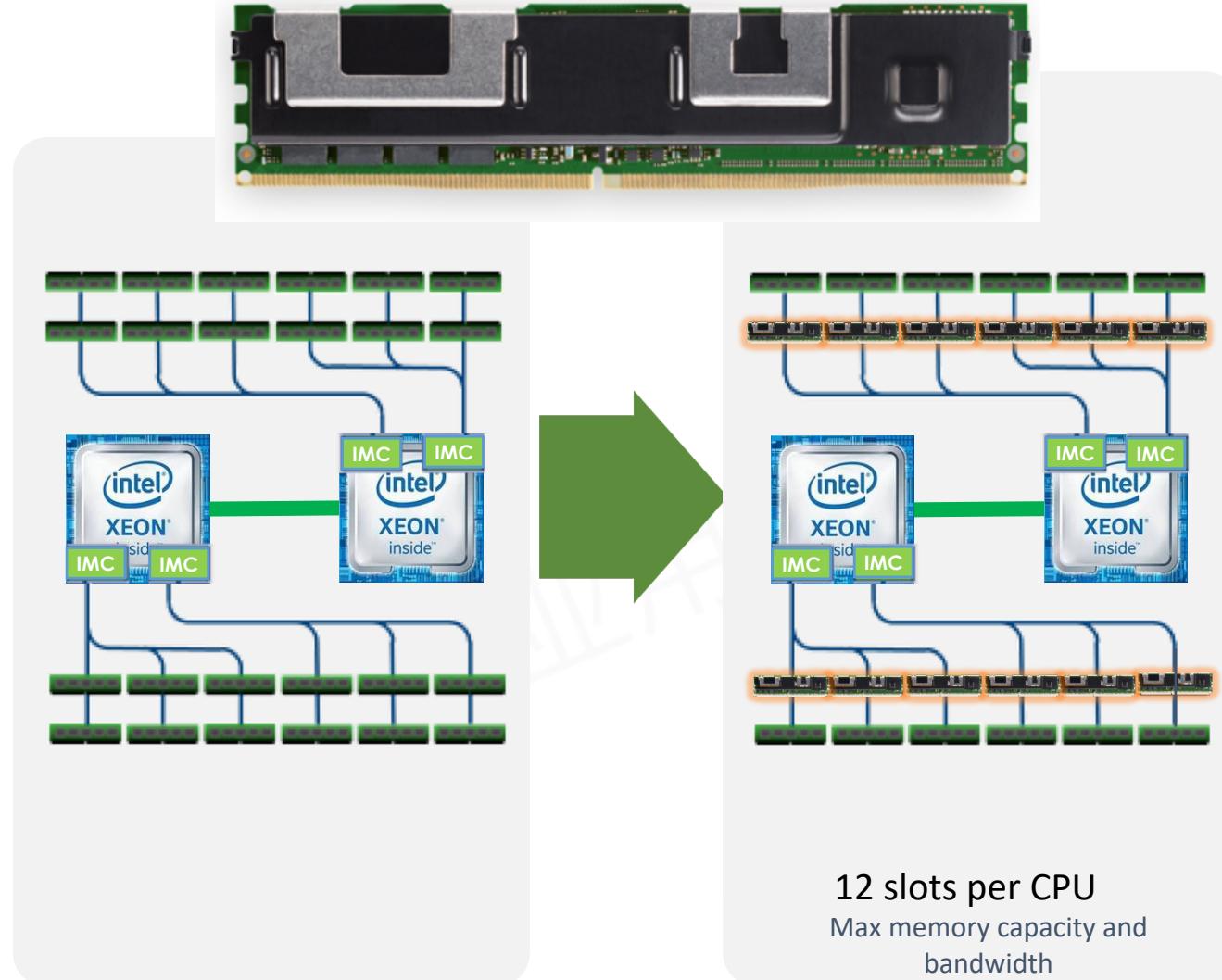
App Direct Flexibility for developer to optimize

- Intel® Optane™ DC persistent memory to accelerate data previously located on disk with significantly lower latency. Some examples:
 - Move data from disk to persistent memory, re-architect data structures from blocks to byte addressable removing software overhead
 - Used as a cache
 - Multiple modules can be interleaved for higher bandwidth
 - Storage Over AD - Mount file system to App Direct (DAX mode to avoid copies) for initial testing

Population examples: 2 socket system



Apache Flink



- To ensure system configuration flexibility for different population, Intel® Optane™ DC persistent memory can be populated:
 - On the same channel as DRAM
 - On the slot closest to the CPU on each channel
 - Up to 6 modules per CPU
- BIOS can recognize which DDR slot(s) have Intel Optane DC persistent memory and in which mode it is running

* No difference on functionality or performance when 2nd DIMM slot is in channel 0, 1 or 2 for that integrated memory controller (IMC)

03

How to use Intel Optane Persistent Memory in Flink?



Cluster Configurations

		Master	Slave
Hardware	Memory	128GB (8x 16GB DDR4)	192GB (16x 12GB DDR4) +1TB DCPM (8 x 128GB)
	DCPMM Mode	N/A	Memory mode/ SoAD mode
	Storage	1TB SSD*5	1.8TB SSD*5
	CPU	Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz	Intel(R) Xeon(R) Platinum 8280L CPU @ 2.70GHz
Software	Hadoop	hadoop-2.8.5	
	Flink	Flink 1.5.1	
	OS	Fedora release 29 with kernel 5.0.5-200.fc29.x86_64	Fedora release 29 with kernel 4.19.35-600.nvdimm.fc29.x86_64
Workload(TPC-DS)	Data Scale	1TB	
	SQL Queries	Simple query against TPC-DS table web_sales	



Demo

- Set App Direct mode
 - ipmctl create -goal PersistentMemoryType=AppDirect
 - reboot
 - ndctl list -R
 - ndctl create-namespace -m fsdax -r region0
 - ndctl create-namespace -m fsdax -r region1
 - fdisk -l
 - mount -o dax /dev/pmem0 /mnt/pmem0
 - mount -o dax /dev/pmem1 /mnt/pmem1
- Memkind library for AD mode
 - https://github.com/memkind/memkind/blob/master/examples/pmem_malloc.c
- Memcache library for AD mode
 - <https://github.com/pmem/vmemcache/blob/master/tests/example.c>
- Use DCPM SoAD mode in Flink
 - refer [SoAD.cast](#)
- Switch DCPM from App Direct mode to Memory mode
 - umount /mnt/pmem0
 - umount /mnt/pmem1
 - ndctl disable-namespace namespace0.0
 - ndctl destroy-namespace namespace0.0
 - ndctl disable-namespace namespace1.0
 - ndctl destroy-namespace namespace1.0
 - ndctl disable-region region0
 - ndctl disable-region region1
 - ipmctl create -goal MemoryMode=100
 - reboot
- Use DCPM memory mode in Flink
 - refer [memory-mode.cast](#)

Reference Link

<https://docs.pmem.io/ndctl-users-guide/concepts/libnvdimm-pmem-and-blk-modes>

<https://software.intel.com/en-us/articles/introduction-to-programming-with-persistent-memory-from-intel>

<https://software.intel.com/en-us/articles/intel-optane-dc-persistent-memory-a-major-advance-in-memory-and-storage-architecture>

ndctl: <https://github.com/pmem/ndctl>

ipmctl: <https://github.com/intel/ipmctl/releases>



Apache Flink

THANKS

