Intel Optane DC and Java

Lessons learned in practice

Jiří Holuša
Hazelcast

@jholusa

About me

- Currently working at Hazelcast as Product Manager
- But I’m really a developer! (PM for 2 months)
- I’m also not Intel advocate

About Hazelcast

HAZELCAST IMDG

HAZELCAST JET

[1] https://hazelcast.org/

› Motivation

Agenda

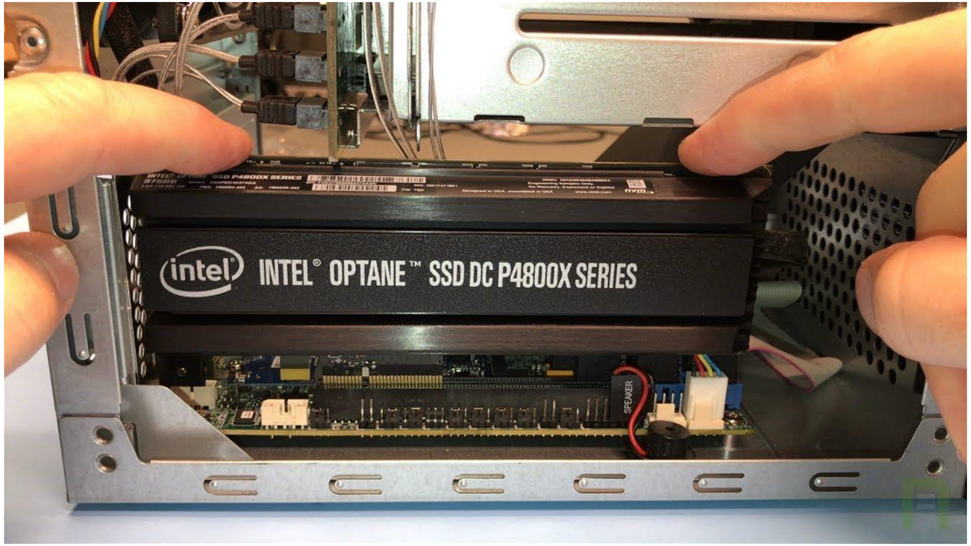
- Introduction to Intel Optane DC
- Use cases
- How to work with it in Java in JDK 14+ (demo)
- How to work with in pre-JDK 14 (demo)
- Discussion and conclusions

What is Intel Optane DC (PMM)?

- PMM (Persistent Memory Module) - it’s DIMM module
- Sits on the memory bus
- Orders of magnitude faster than SSD and persistent

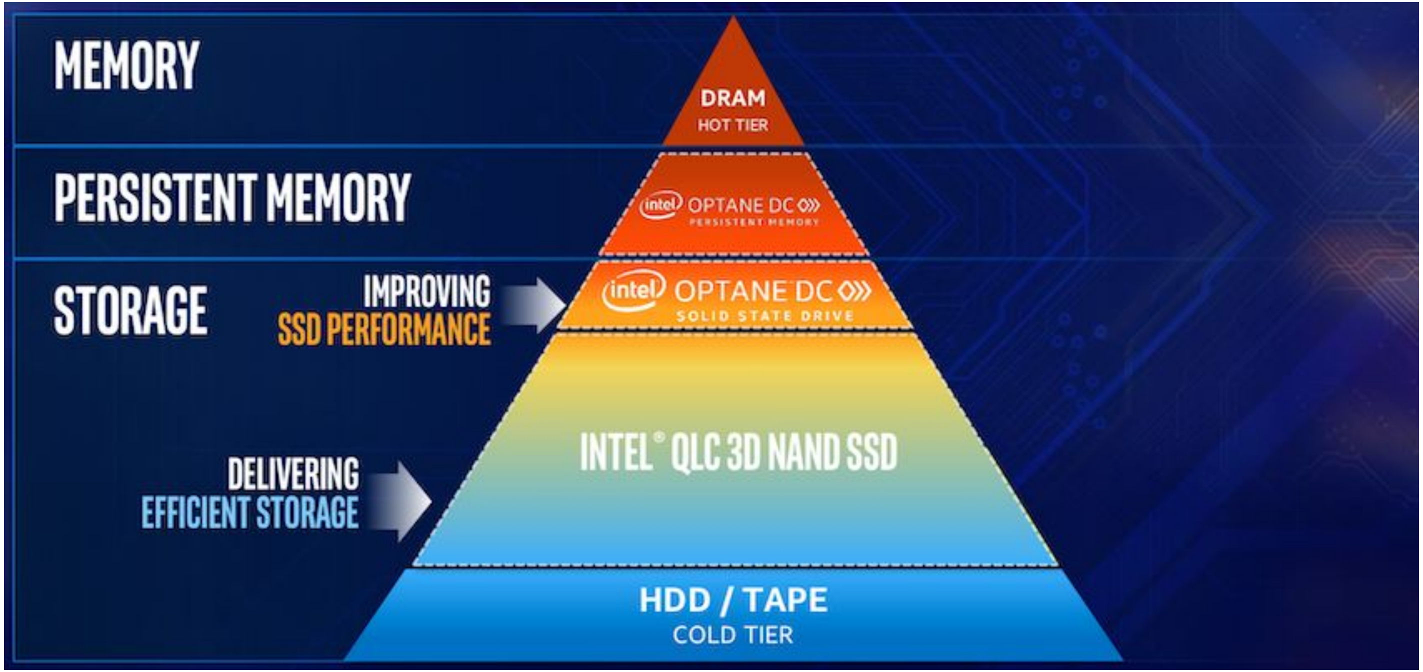
But I heard about something in PCI-E slot!

- That's Intel Optane (DC) SSD
- Faster SSDs



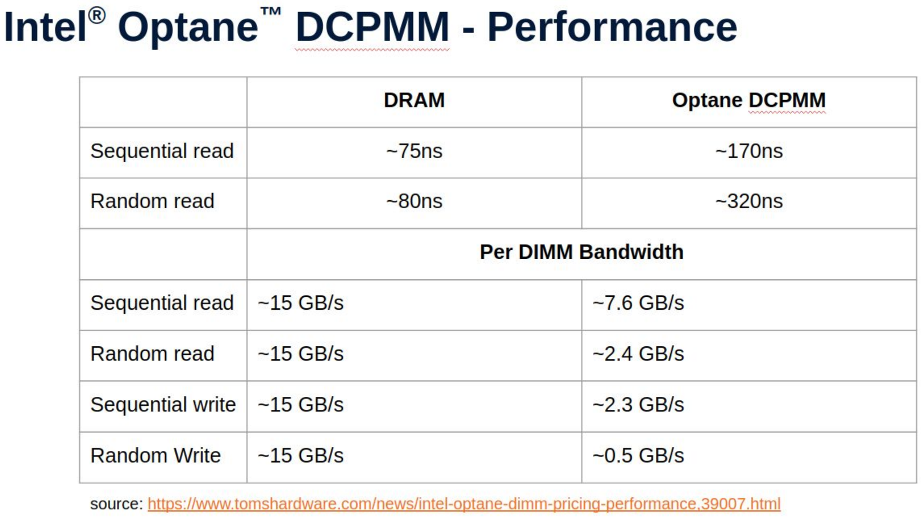
hazelcast

Big picture


MEMORY
PERSISTENT MEMORY
STORAGE
IMPROVING
SSD PERFORMANCE
DELIVERING
EFFICIENT STORAGE
DRAM
HOT TIER
intel OPTANE DC
PERSISTENT MEMORY
intel OPTANE DC
SOLID STATE DRIVE
INTEL® QLC 3D NAND SSD
HDD / TAPE
COLD TIER

Intel Optane DC PMM

- Byte addressable
- Persistent
- “DRAM-like” speed
- Half the price of DRAM



Intel® Optane™ DCPMM - Performance

	DRAM	Optane DCPMM
Sequential read	~75ns	~170ns
Random read	~80ns	~320ns
	Per DIMM Bandwidth	
Sequential read	~15 GB/s	~7.6 GB/s
Random read	~15 GB/s	~2.4 GB/s
Sequential write	~15 GB/s	~2.3 GB/s
Random Write	~15 GB/s	~0.5 GB/s

source: https://www.tomshardware.com/news/intel-optane-dimm-pricing-performance,39007.html

› Use cases

- Cheap DRAM
  - Providing twice that much memory for a memory intensive server for the same price keeping the speed (~)
  - ~$2000 for 128 GB DRAM module, ~$700 for an Optane DC PMM
- Super fast storage
  - Useful for loading a lot of data quickly, imagine restart of a DB server with TBs of data in a few minutes
- Tiered storage
  - Keep the hottest data in DRAM for speed
  - Frequently used data in Optane
  - Storage data on a slow disk

I love it, but I want to see the code!

- How do I implement the use cases in Java?
- This is where it gets ugly
- All the libraries are primarily written in C 😒 [1]
- But … there are ways to make use of the persistent memory in Java already 😏

[1] https://pmem.io/pmdk/

Cheap RAM use case

- Not using persistency
- Two options:
  - DRAM serving as L4 cache for Optane PMM - system sees total memory equals to size of Optane PMMs (Memory Mode)
  - Intelligent concatenation of DRAM and Optane PMM - system sees total memory equals to DRAM + Optane (App Direct mode)
- Still pretty useful for memory demanding applications

Super fast storage use case

- The easy way using memory mapped ByteBuffer is newly available in JDK14 [1]
- On older JDKs, you'll have to go for a combination of C library and JNI calls 😥

[1] https://openjdk.java.net/jeps/352

Tiered storage use case

- The easy way using memory mapped ByteBuffer is newly available in JDK 14 [1]
- On older JDKs, again, you have to go for a combination of C library and JNI calls 😰

[1] https://openjdk.java.net/jeps/352

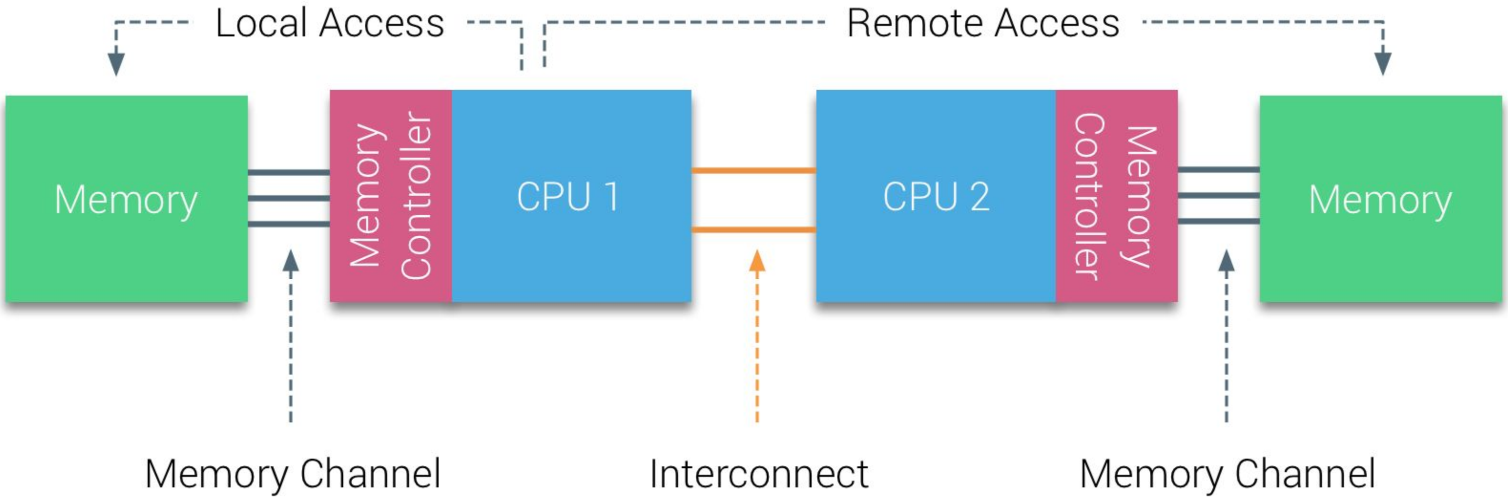
› Access through ByteBuffer with JDK 14

- Access through JNI calls

There's more … but that's for the next time

- Other libraries are being developed but not yet in production quality, e. g. https://github.com/pmem/pcj
- There are some limitations:
  - Need for high-end Xeon Platinum processors
- Issues to be addressed to make the best use out of it:
  - Implement NUMA awareness for maximal performance
  - DIMMs are mounted based on the socket that they are assigned to making creation of "one big chunk" of Optane memory more difficult
  - and more …

Insight: NUMA-awareness affects performance a lot with Optane PMM



When is Intel Optane DC PMM for me?

- If your application needs top level performance with large amount of data that doesn’t fit into DRAM memory,
- If fast storage access is critical to you,
- If you’re trying to be cost effective with your memory and performance demands,

- … then Intel Optane DC PMM is worth looking into.

What about Intel Optane DC PMM and Java?

- With JDK14+ pretty nice and easy
- With older JDKs, you have to be not afraid of JNI and Unsafe
- More persistent memory enhancements on the road map [1]

[1] https://openjdk.java.net/jeps/383

Conclusion

- The Intel Optane DC PMM technology itself is showing amazing results and we'll hear about it more soon
- Java support in recent versions is nice, before it's a bit clumsy
- However, it's possible to do some real magic stuff already providing massive performance gains for certain workloads

- And you can try it yourself on Google Cloud! [1]

[1] https://cloud.google.com/blog/topics/partners/available-first-on-google-cloud-intel-optane-dc-persistent-memory

hazelcast

› Thank you. Questions?

Jiří Holuša
Hazelcast

@jholusa