PROMISE AND PITFALLS OF PERSISTENT MEMORY

Rob Dickinson
Datacenter Solutions Group

September 29, 2017

ABOUT ME

NVM Software Architect at Intel

Specializing in persistent memory and databases

- github.com/RobDickinson
- linkedin.com/in/robfromboulder
- @robfromboulder
- robert.a.dickinson@intel.com

EAT
SLEEP
CODE >>>
REPEAT

BUILD COOL STUFF
VALIDATE ON REAL SAMPLES
SHARE BEST PRACTICES
CONTRIBUTE TO OPEN SOURCE
MAKE THE WORLD A BETTER PLACE

I'M NOT ONE OF THOSE INTEL ENGINEERS IN A BUNNY SUIT

WHAT IS PERSISTENT MEMORY?

aka Storage Class Memory

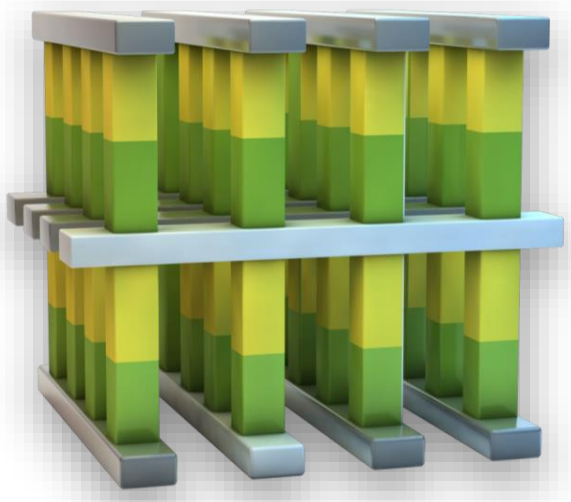
INTEL® PERSISTENT MEMORY

New type of memory:

- Persistent
- 6 TB per two-socket system
- Cheaper than DRAM

Product available:

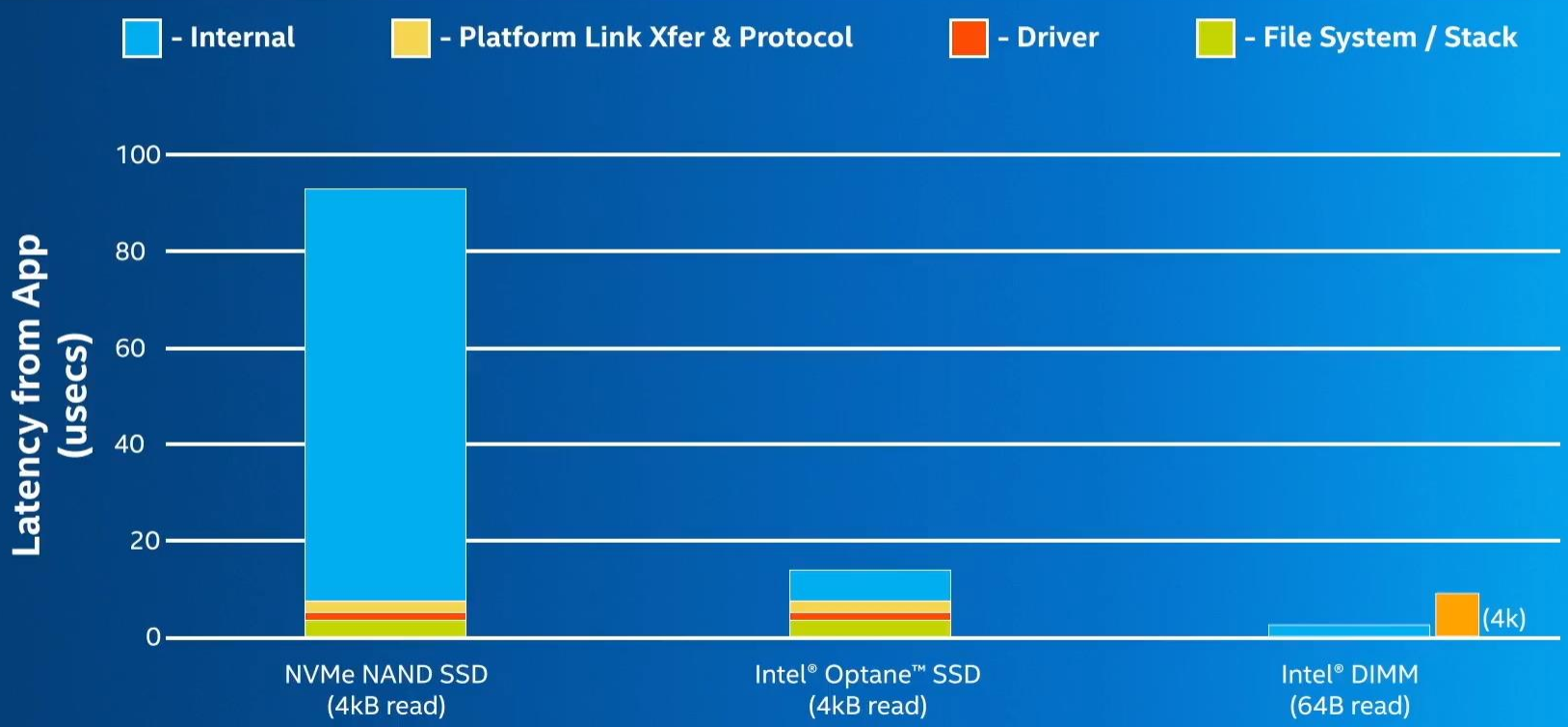
- SSDs in 2017
- Intel DIMMs for next-gen platforms in 2018



COMPARED TO MEMORY AND STORAGE

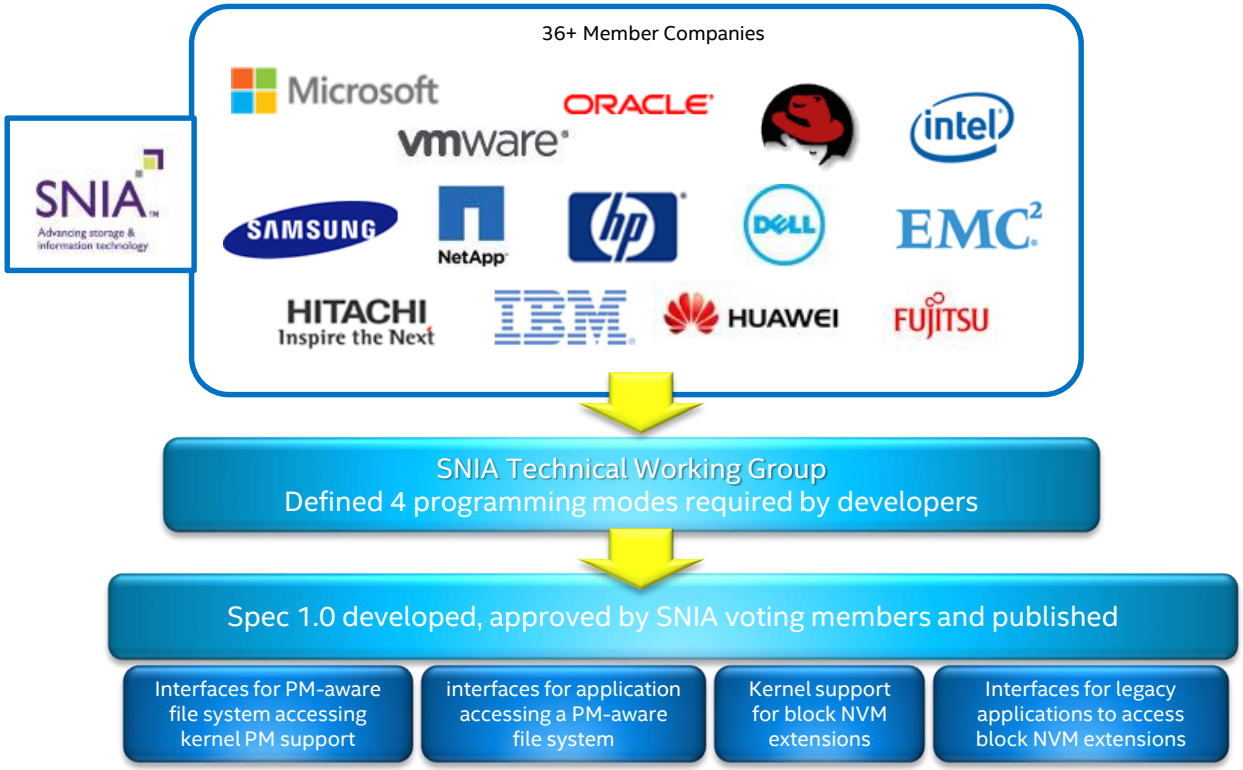
Aspect of Comparison	Memory (DRAM)	Persistent Memory	Storage (SSD)
Capacity	GB	TB	TB
Power-Safe Durability	No	Yes	Yes
Access Model (native units)	Byte Addressable (CL)	Byte Addressable (CL/ECC)	Blocks (4K)
Latency	Nanoseconds	Nanoseconds	Microseconds
Wear Leveling	No	No	Yes
Cost	$$$	$$	$

OPTIMIZED SYSTEM INTERCONNECT



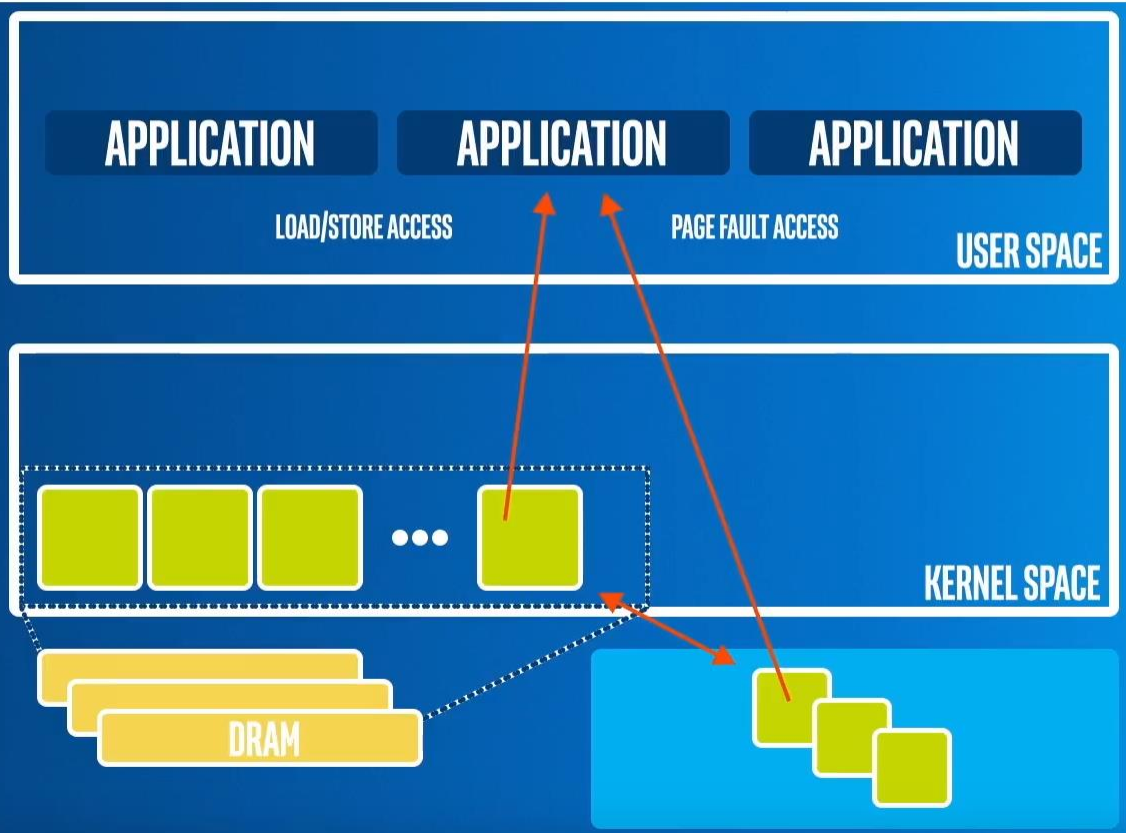
PROGRAMMING FOR PERSISTENT MEMORY

STANDARD FOR PERSISTENT MEMORY PROGRAMMING

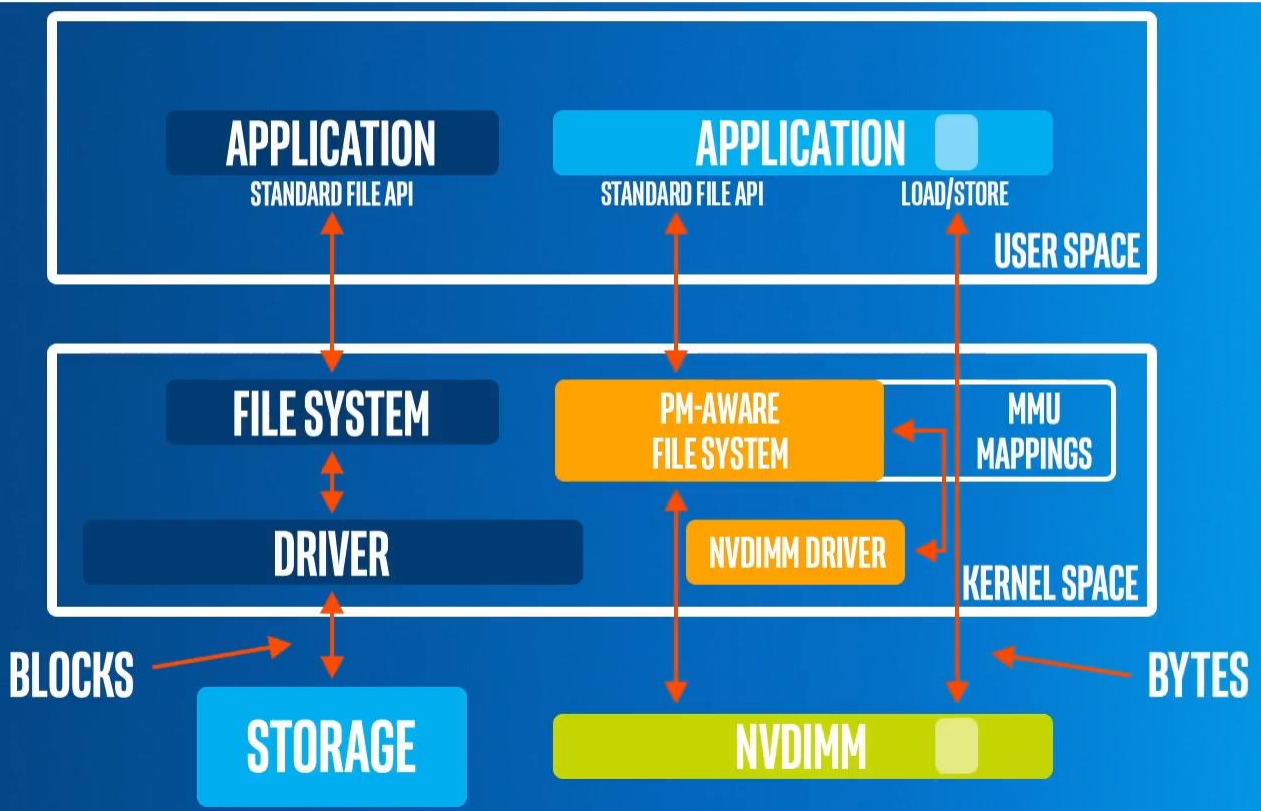


http://snia.org/sites/default/files/NVMProgrammingModel_v1.pdf

MEMORY-MAPPED FILES



PERSISTENT MEMORY PROGRAMMING MODEL

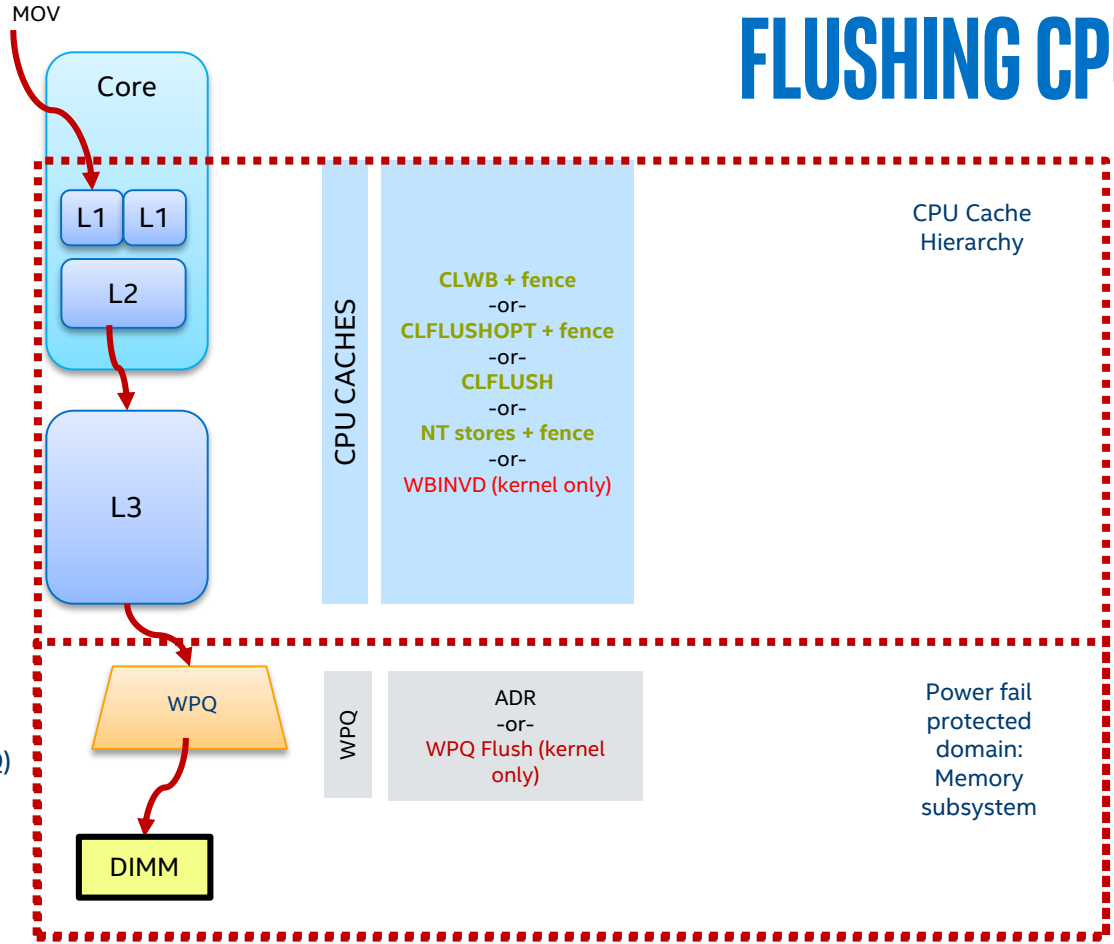


FLUSHING CPU CACHES

Stores to pmem from application

CPU Caches

WPQ is the **Write Pending Queue** in the memory controller (sometimes called the TPQ)



THE BIG CHALLENGE FOR PROGRAMMERS

Software must ensure transactional consistency of persistent data structures!

ADAPTING SOFTWARE FOR PERSISTENT MEMORY

GUIDELINES FOR ADAPTING SOFTWARE

- Persistent memory is not exactly like DRAM nor SSD
- Persistent memory will be used in concert with memory and storage
- Persistent memory will be used in different ways by different applications
- Persistent memory exposes applications to some classic problems

INTERESTING CASES FOR PERSISTENT MEMORY

- WAL acceleration
- Persistent caches
- In-memory DBs
- Hybrid (multi-tier) data structures
- Converged compute and storage

CLASSIC PROBLEMS RE-EXPOSED

WHY?

Having direct access to persistent memory, bypassing the kernel and filesystem, brings classic problems into userspace (where programmers aren't used to seeing them)

1. POSITION INDEPENDENCE

Virtual memory ranges change all the time
Persistent memory ranges do not

2. MEMORY ALLOCATION & GC

Regular memory allocators aren't appropriate
A leak in persistent memory remains leaked

3. TRANSACTIONS & LOCKING

Preventing torn updates using transactions
Resetting abandoned locks

4. ERROR DETECTION & HANDLING

Coping with media failures or corruption

CALL TO ACTION

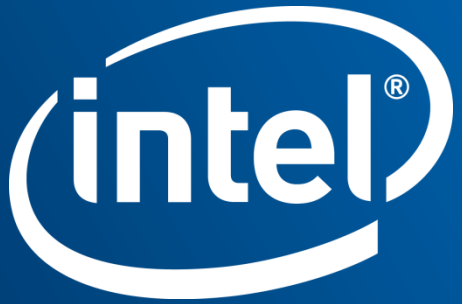
GETTING STARTED WITH PERSISTENT MEMORY

- Consider persistent memory as its own unique tier
- Look for break-out opportunities to hybridize your architecture
- Use libraries to assist in safely using persistent memory

  pmem.io

  github.com/pmem/nvml/

  software.intel.com/en-us/persistent-memory
- Start with emulation, but validate & tune on a real platform


intel®