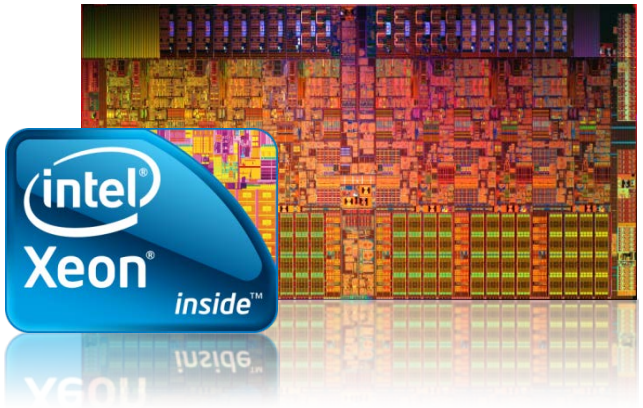


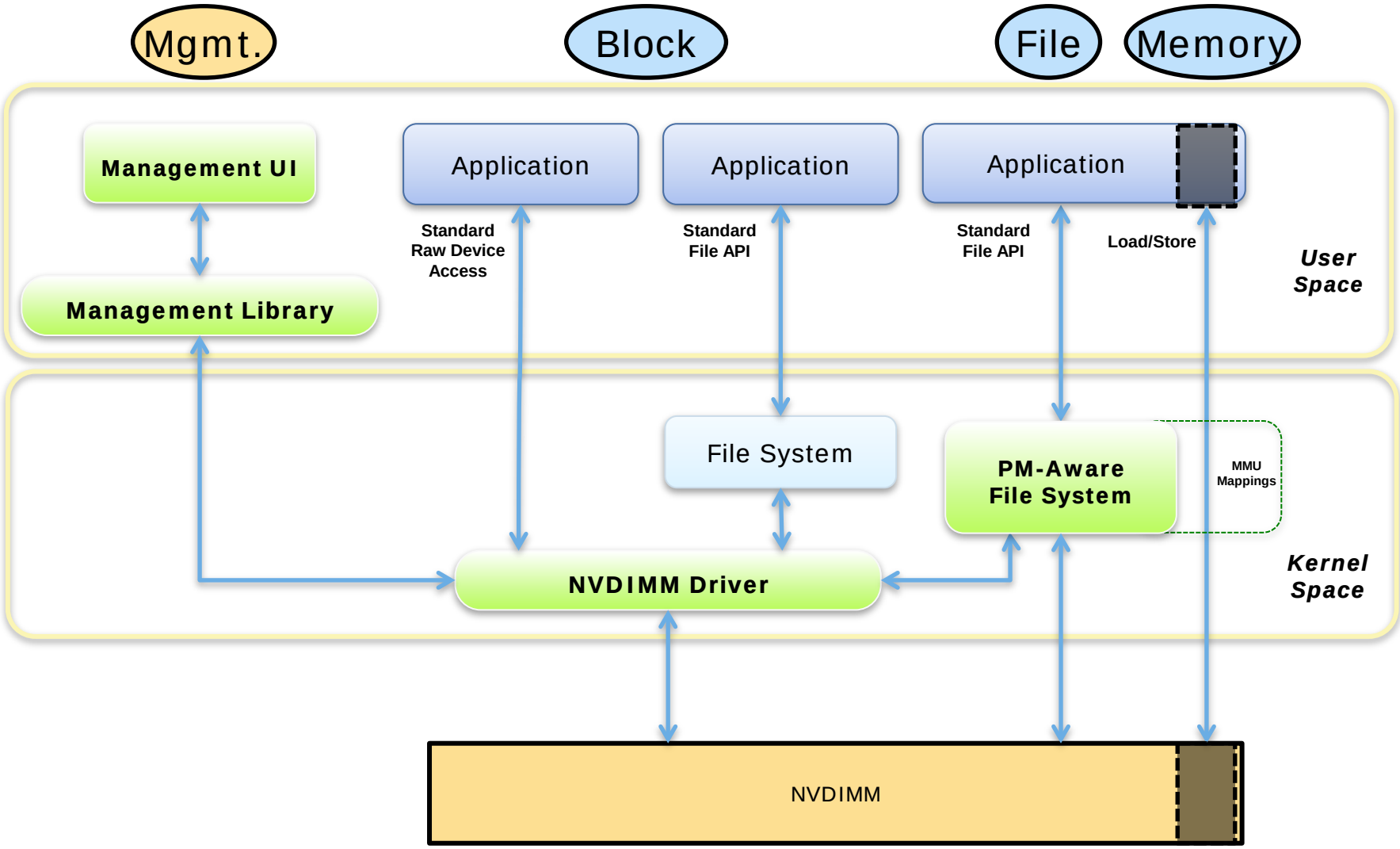
Processor Support for NVM Programming

Andy Rudoff
Intel Corporation

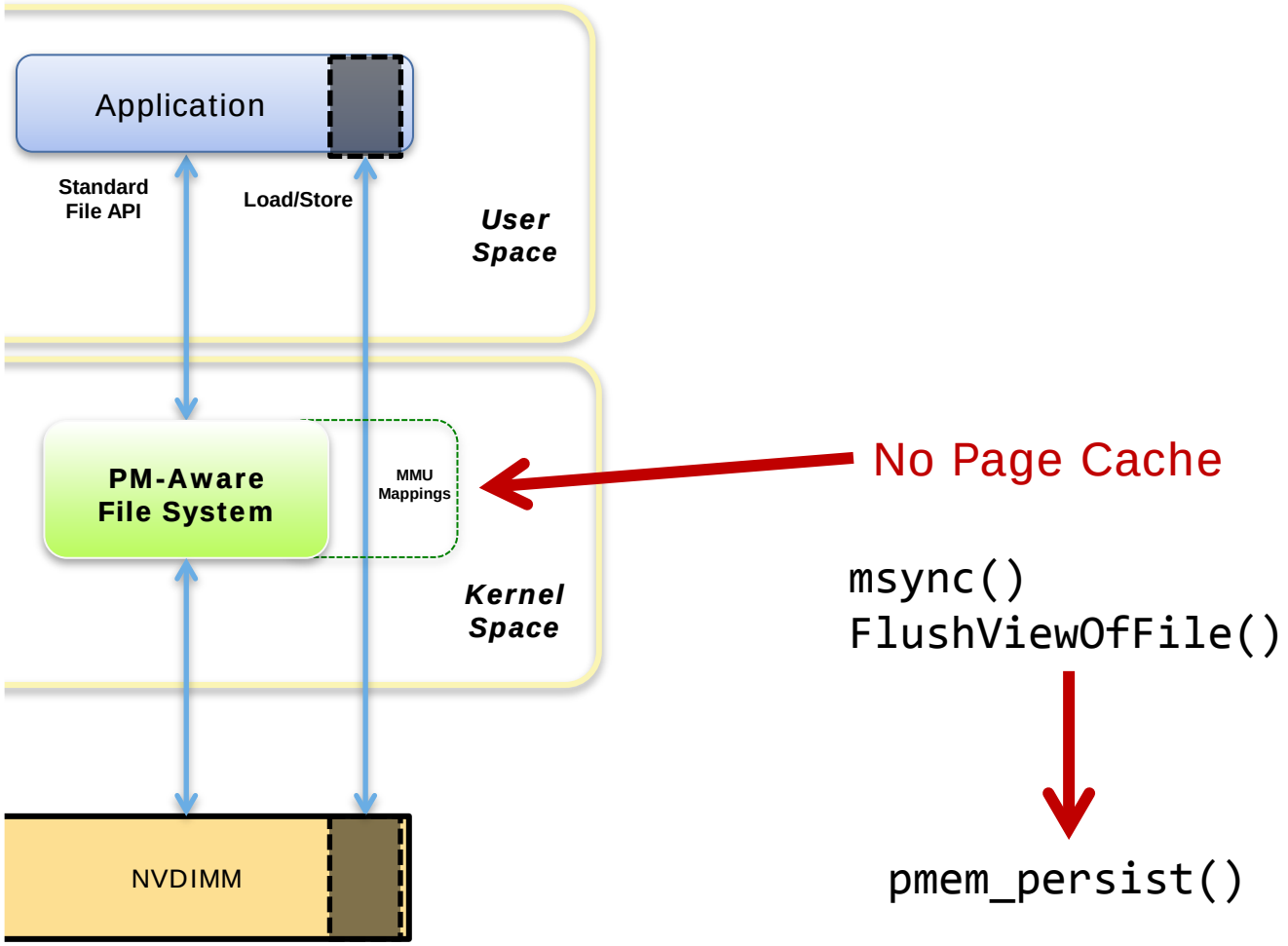
NVM Summit

January 20th 2015

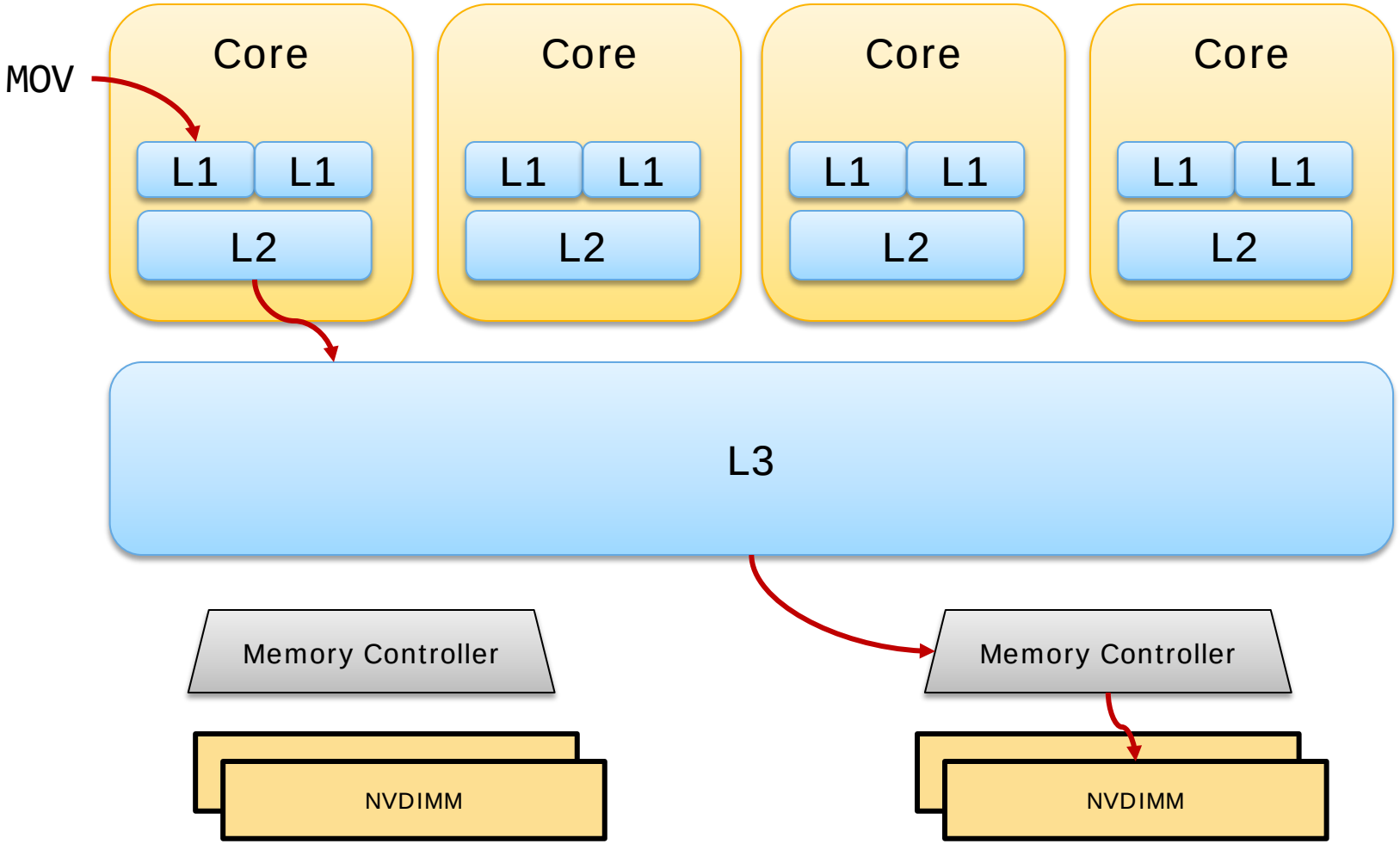
Challenges Introduced by NVM Programming


Mgmt.
Block
File
Memory
Management UI
Management Library
Application
Application
Application
Standard
Raw Device
Access
Standard
File API
Standard
File API
Load/Store
User
Space
File System
PM-Aware
File System
MMU
Mappings
NVDIMM Driver
Kernel
Space
NVDIMM

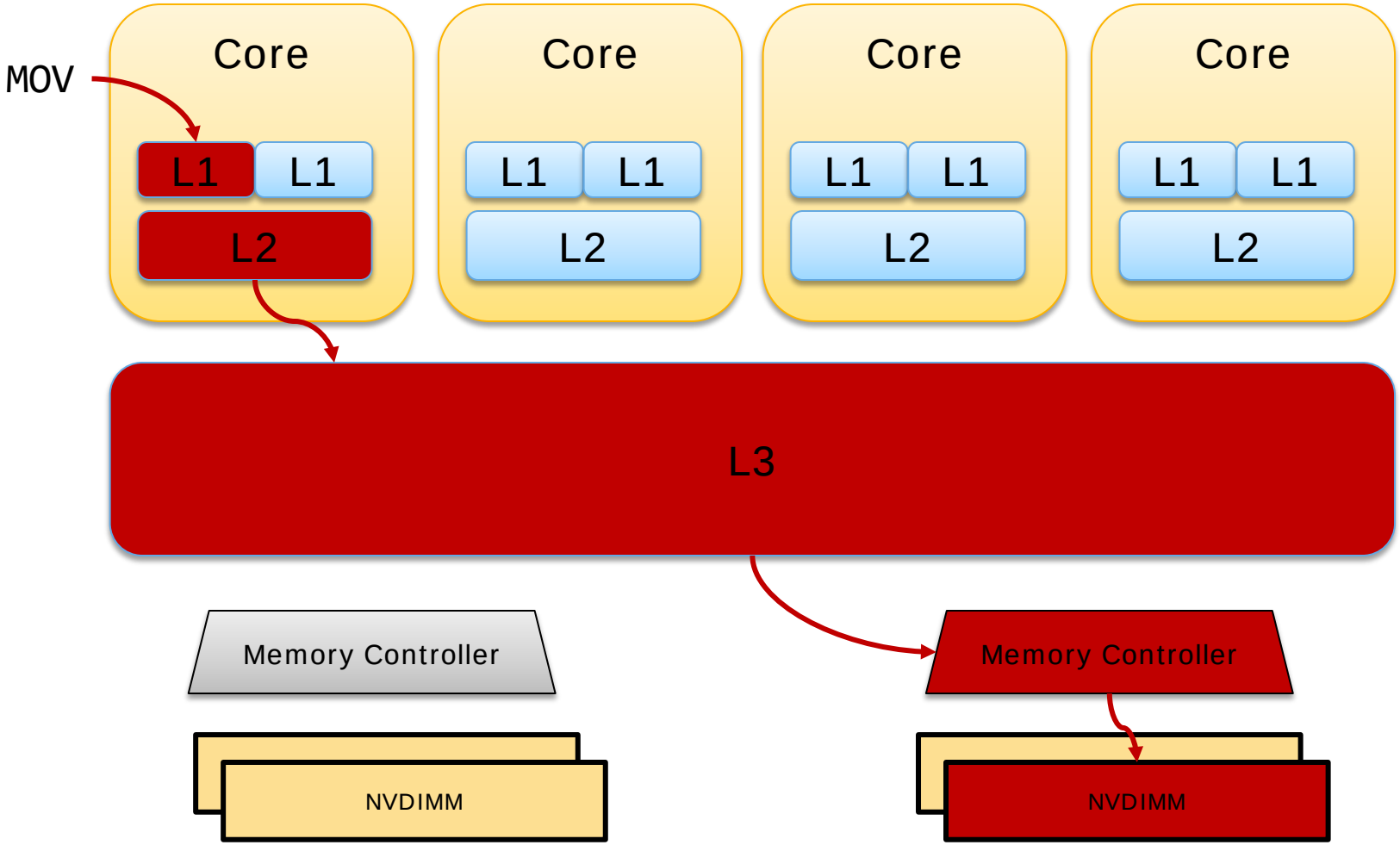
Where Writes are Cached



The Road to Persistence

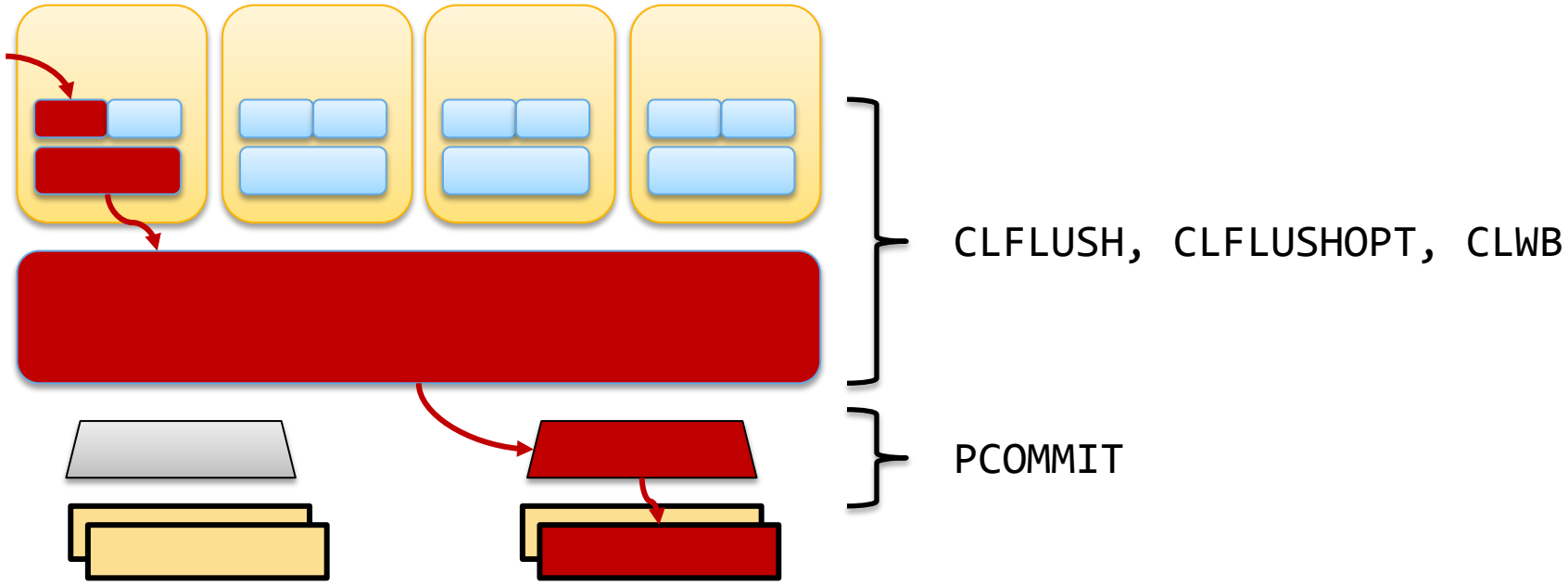


Hiding Places



Instruction-Level Support

Two Levels of Flushing Writes



Flushing Writes from Caches

Instruction	Meaning
`CLFLUSH addr`	Cache Line Flush: Available for a long time
`CLFLUSHOPT addr`	Optimized Cache Line Flush: New to allow concurrency
`CLWB addr`	Cache Line Write Back: Leave value in cache for performance of next access
`WBINVD`	Heavy Hammer: used by OS to flush entire cache Privileged

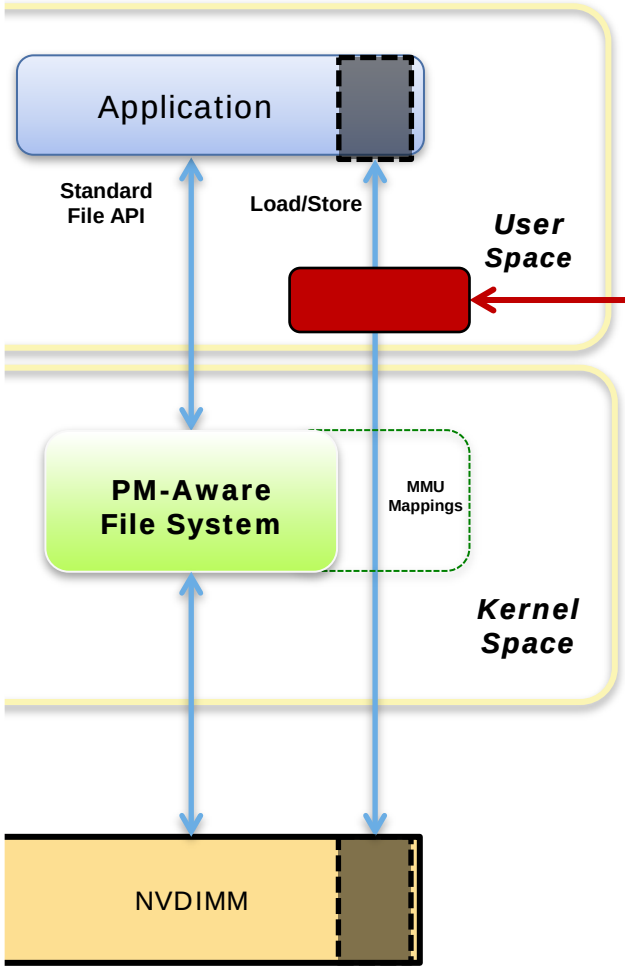
Flushing Writes from Memory Controller

Instruction	Meaning
PCOMMIT	Persistent Commit: Flush stores accepted by memory subsystem
Asynchronous DRAM Refresh	Flush outstanding writes on power failure Platform-Specific Feature


How to Use This

NVM Library: pmem.io

64-bit Linux Initially

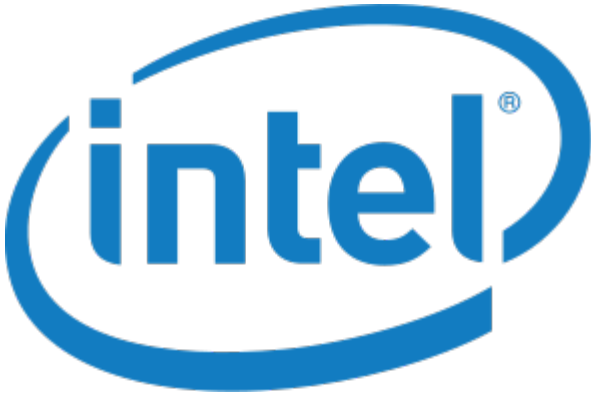


- Open Source
  - http://pmem.io
- Libpmem
- Libpmemobj
- Libpmemblk
- Libpmemlog
- Libvmem

Transactional

Summary

- New Instructions Emerging
  - Support the SNIA NVM Programming Model
- Stay Tuned… more to come…
  - But you probably want to use a library: http://pmem.io
- Intel Architecture Instruction Set Extensions Programming Reference
  - https://software.intel.com/en-us/intel-isa-extensions


intel