



Enabling Persistent Memory: The NVM Programming Model

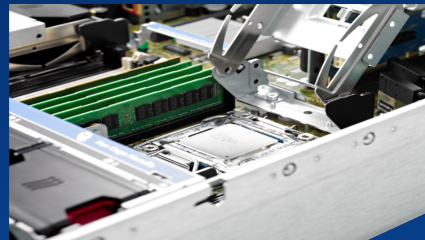
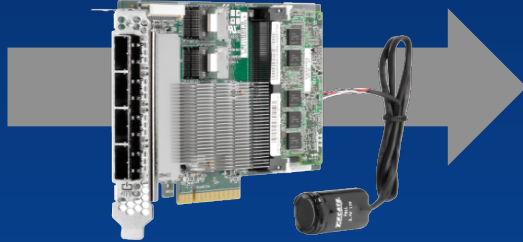
Doug Voigt,
Hewlett Packard Enterprise

Santa Clara, CA
August 2016



Persistent Memory (PM) Vision

Persistent Memory Brings Storage



To Memory Slots

Fast
Like Memory

Persistent
Like Storage

Make data durable without doing I/O



Mission of the NVM Programming Model TWG

- Accelerate the availability of software that enables NVM (Non-Volatile Memory) hardware.
 - Hardware includes SSD's and PM
 - Software spans applications and OS's
- Create the NVM Programming Model
 - Describes application visible behaviors
 - Allows API's to align with OS's
 - Exposes opportunities in networks and processors

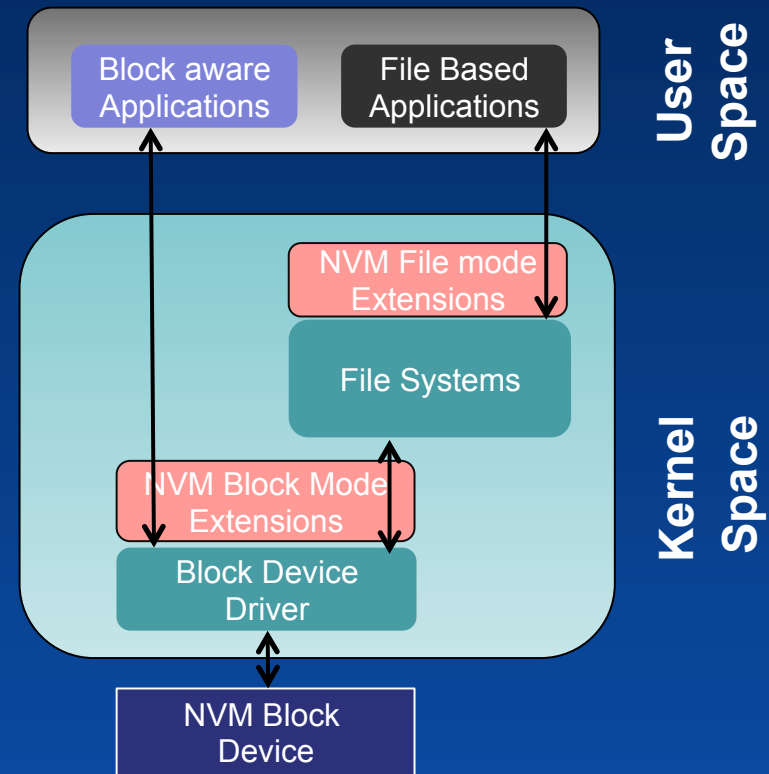


Programming Model Modes

- Block and File modes use IO
 - Data is read or written using RAM buffers
 - Software controls how to wait (context switch or poll)
 - Status is explicitly checked by software
- Volume and PM modes enable Ld/St
 - Data is loaded into or stored from processor registers
 - Processor makes software wait for data during instruction
 - No status checking – errors generate exceptions

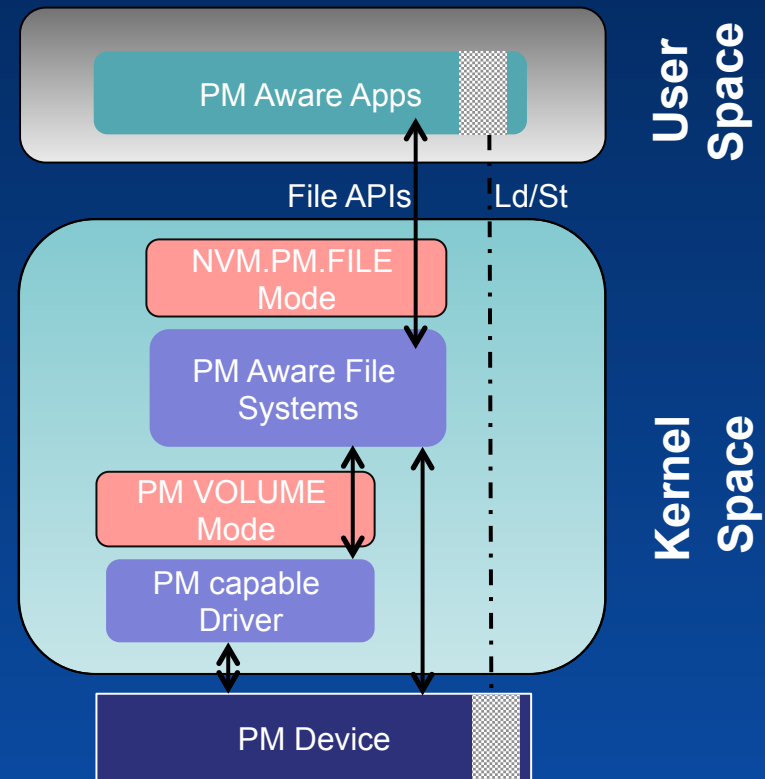
File and Block Mode Extensions

- NVM.BLOCK Mode
 - Targeted for file systems and block-aware applications
 - Atomic writes
 - Length and alignment granularities
 - Thin provisioning management
- NVM.FILE Mode
 - Targeted for file based apps.
 - Discovery and use of atomic write features
 - Discovery of granularities



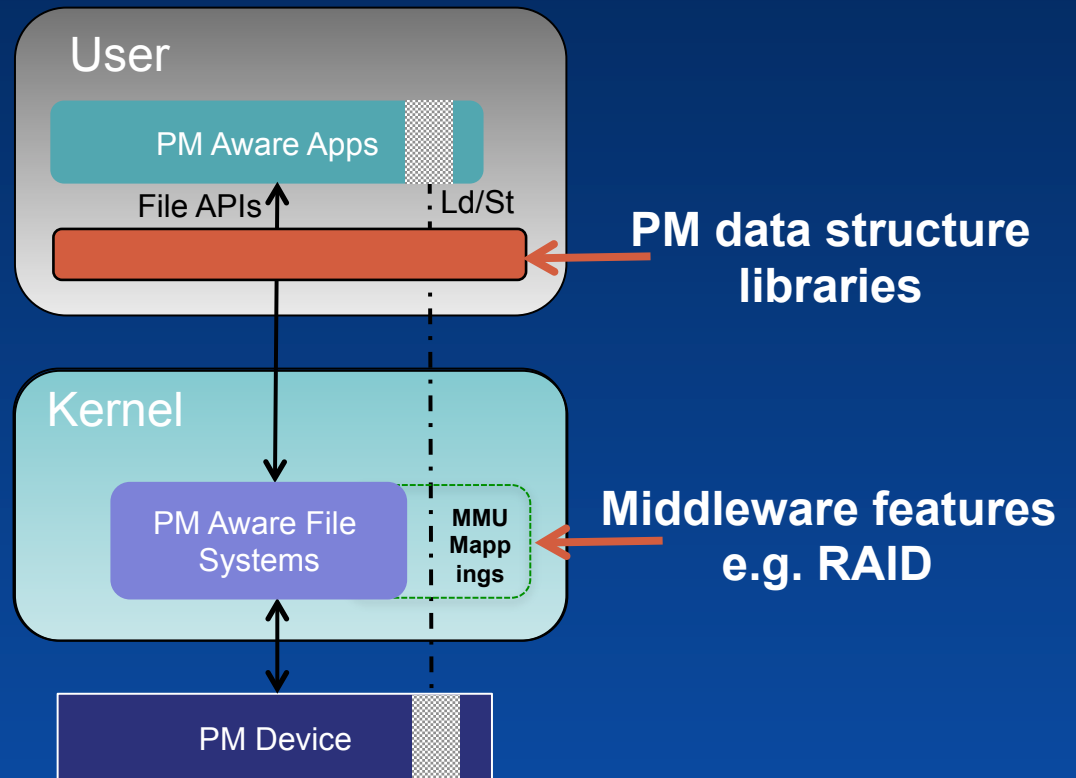
Persistent Memory (PM) Modes

- NVM.PM.VOLUME Mode
 - Software abstraction for persistent memory hardware
 - Address ranges
 - Thin provisioning management
- NVM.PM.FILE Mode
 - Application behavior for accessing PM
 - Mapping PM files to application address space
 - Syncing PM files



Recent NVMP TWG Work in Progress

- Atomicity White Paper in Final Review
Transactional PM Libraries
- Remote Access for HA White Paper Published
High Availability PM -
Remote Optimized Flush





Atomicity

- Inter-Process Atomicity vs. Failure Atomicity
- References NVM Library
- PM Data Structures with Intrinsic atomicity
Log, Block and more
- Transactions for more complex atomicity
Multiple Data Structures



Remote Access

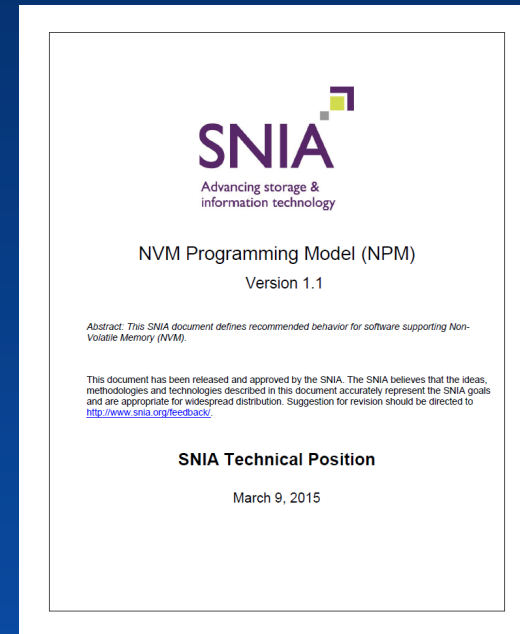
- SNIA PM Remote Access for High Availability
 - Remote access taxonomy
 - Data recoverability requirements
 - Model and requirements for remote flush
- Multiple industry parties are responding
 - Open Fabrics Alliance
 - InfiniBand Trade Association
 - Several vendors



Role of the NVM Programming Model

Rally the industry around a view of NVM that is:

- Application centric
- Vendor neutral
- Achievable today
- Beyond storage
 - Applications
 - Memory
 - Networking
 - Processors





Enabling Persistent Memory: The NVM Programming Model

Doug Voigt,
Hewlett Packard Enterprise