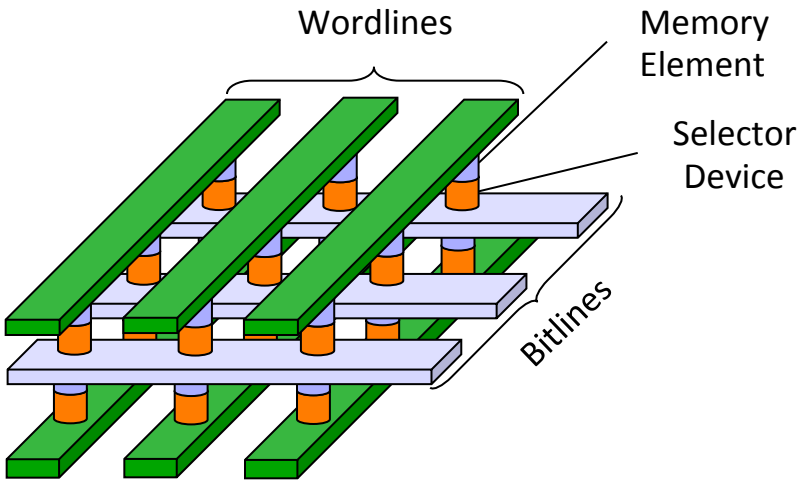
Exposing Persistent Memory

Next Generation Scalable NVM

Scalable Resistive Memory Element



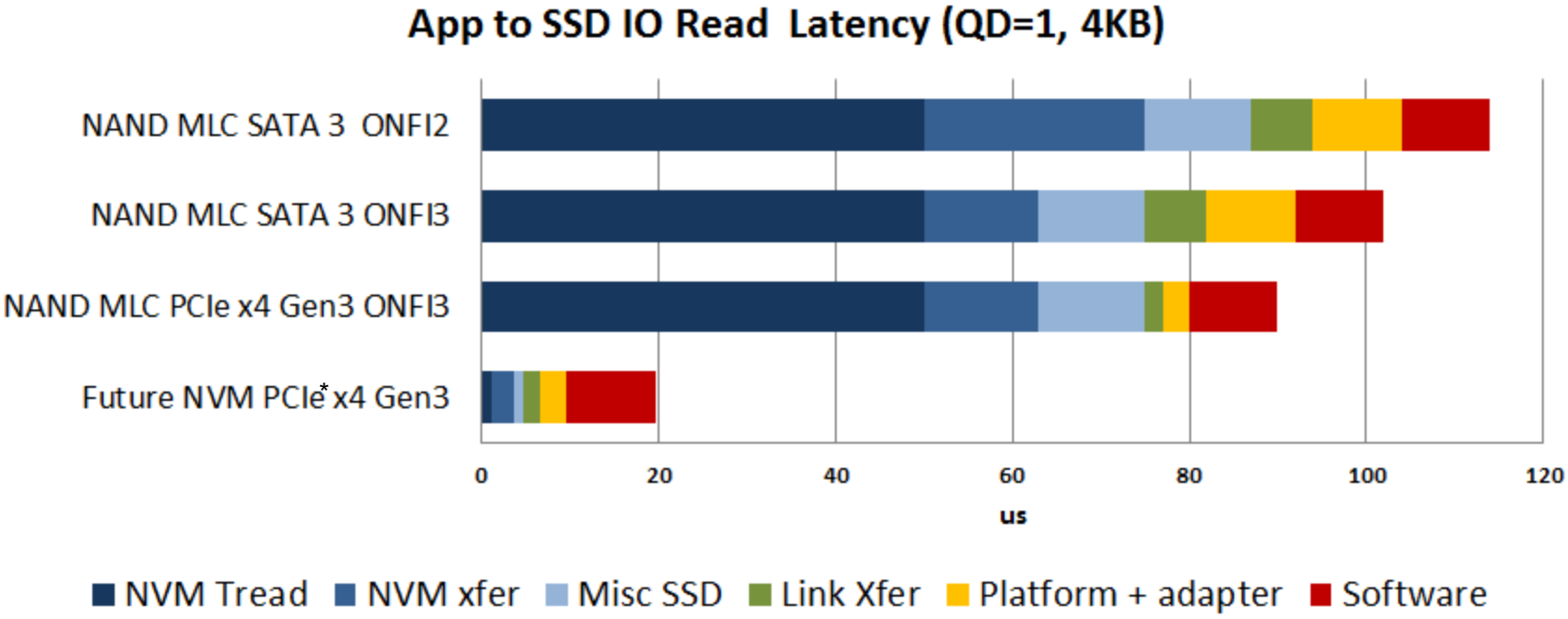
Cross Point Array in Backend Layers ~$4l^2$ Cell

Resistive RAM NVM Options

Family	Defining Switching Characteristics
Phase Change Memory	Energy (heat) converts material between crystalline (conductive) and amorphous (resistive) phases
Magnetic Tunnel Junction (MTJ)	Switching of magnetic resistive layer by spin-polarized electrons
Electrochemical Cells (ECM)	Formation / dissolution of "nano-bridge" by electrochemistry
Binary Oxide Filament Cells	Reversible filament formation by Oxidation-Reduction
Interfacial Switching	Oxygen vacancy drift diffusion induced barrier modulation

Many candidate next generation NVM technologies. Offer ~ 1000x speed-up over NAND, closer to DRAM

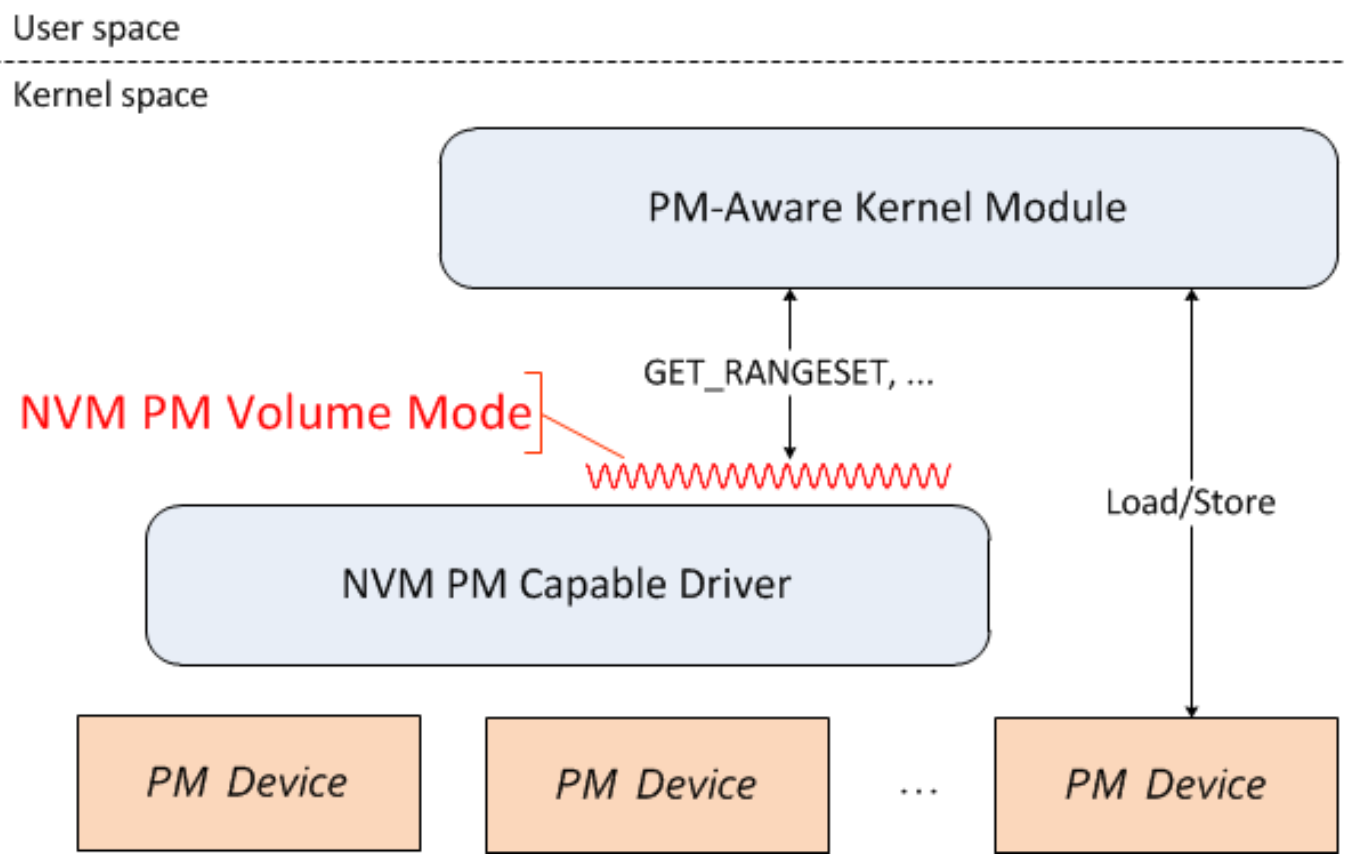
Exploiting Next Generation NVM



- With Next Generation NVM, the NVM is no longer the bottleneck
  - Need optimized platform storage interconnect
  - Need optimized software storage access methods

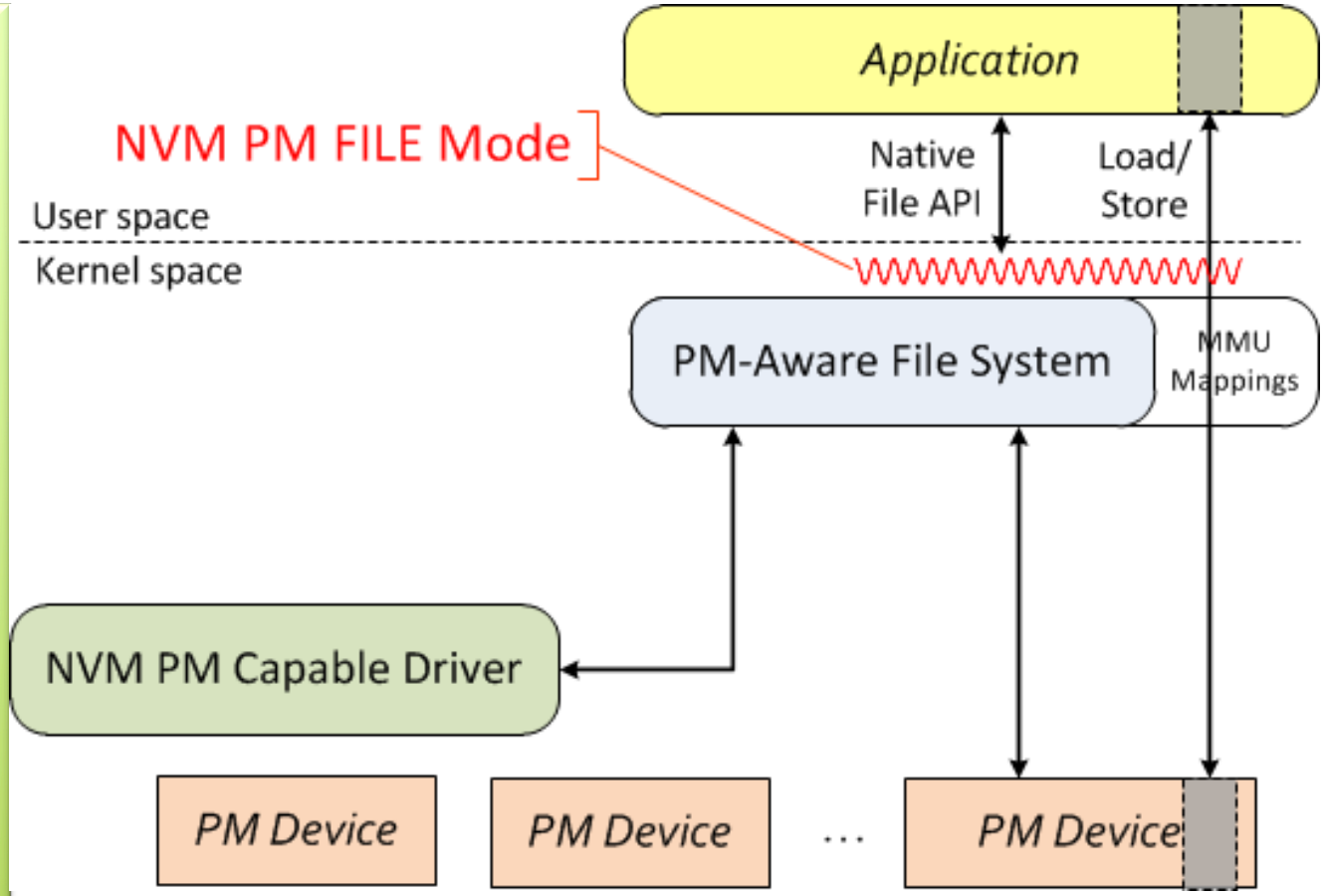
Exposing PMem for Other Kernel Modules

- NVM Volumes
  - PM Capable
  - A list of physical ranges of NVM
- Operations
  - GET_RANGESET
  - …



Exposing PMem for Applications

- NVM Files
  - PM Capable
  - Native file APIs and management
- Operations
  - Native open/close read/ write
  - NVM.PM.FILE.MAP
  - …



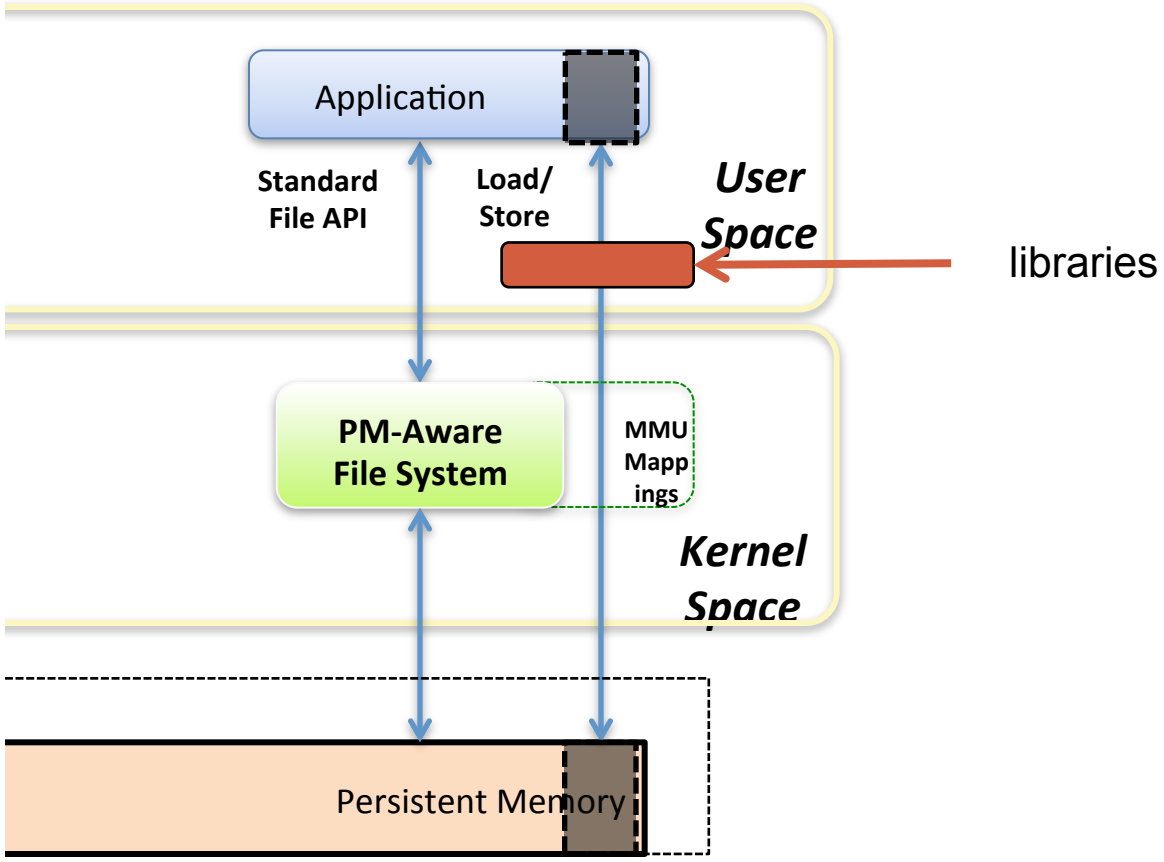
Backup Slides

The Value of Persistent Memory

- Data sets addressable with no DRAM footprint
  - At least, up to application if data copied to DRAM
- Typically DMA (and RDMA) to PM works as expected
  - RDMA directly to persistence – no buffer copy required!
- The “Warm Cache” effect
  - No time spend loading up memory
- Byte addressable
- Direct user-mode access
  - No kernel code in data path
- Intel collaborating with the industry to ensure open standards and choice

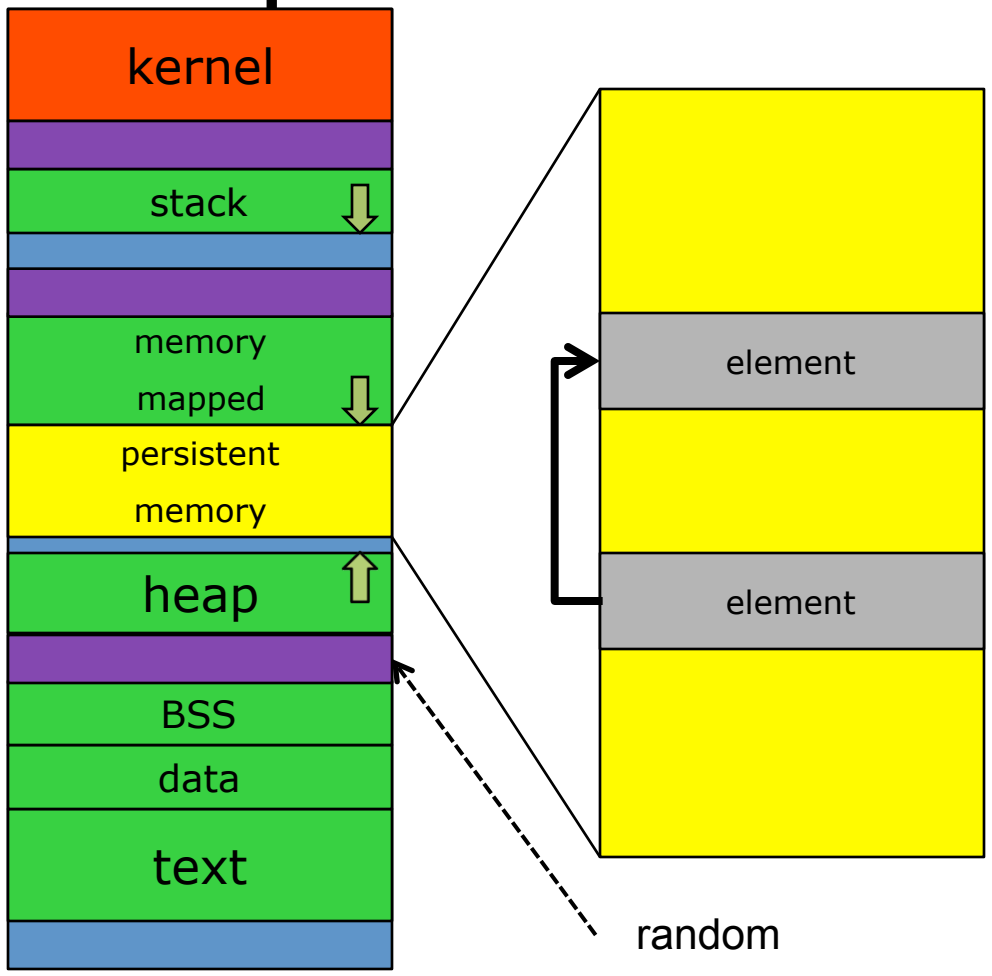
Building on the Basic PM Model

- NVM.PM.FILE programming model "surfaces" PM to application
- Still somewhat raw at that point
- Build on it with additional libraries
- **Eventually turn to language extensions**

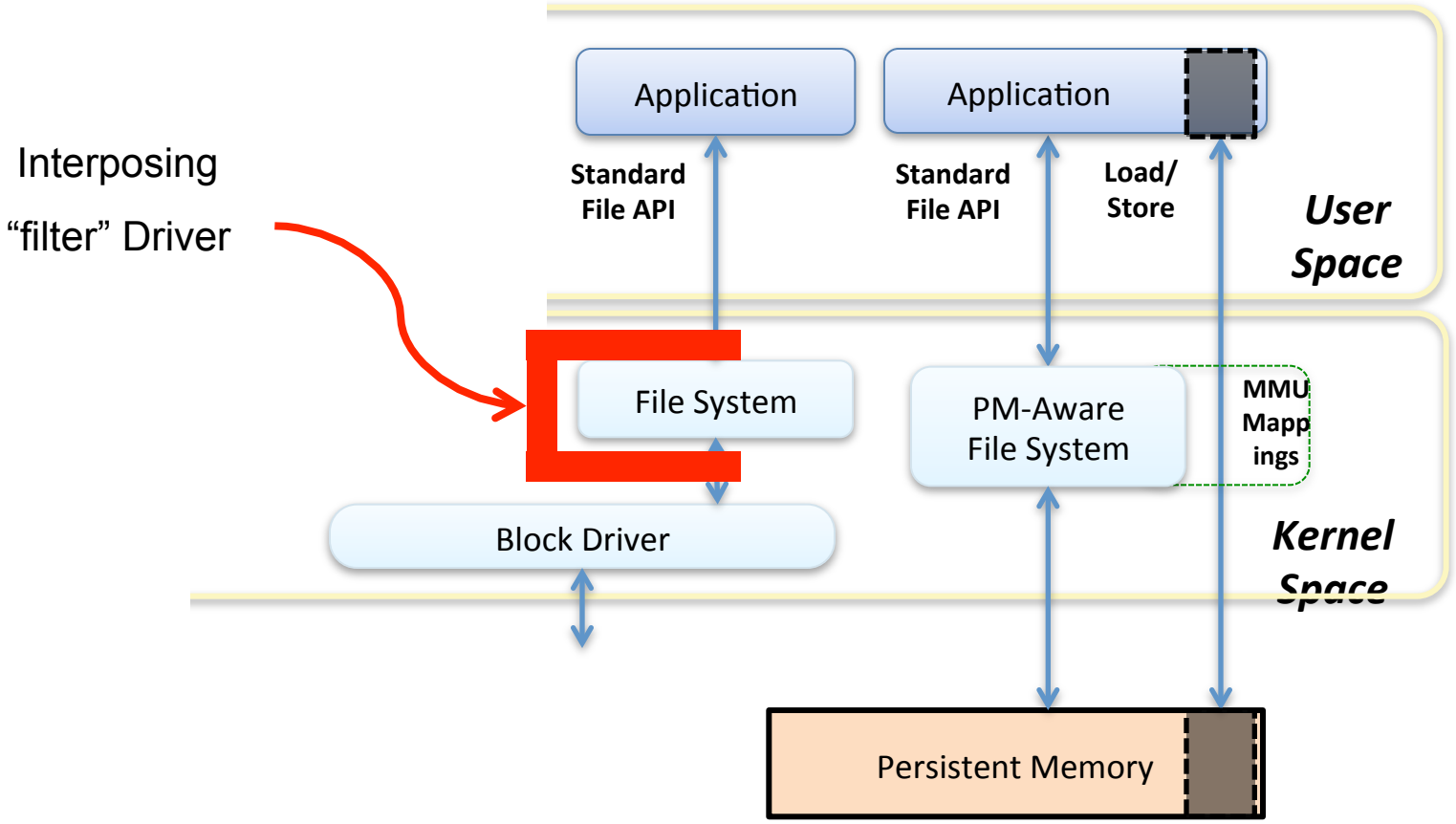


Position Dependence

- Can pointers work across sessions?
- Will a PM file be mapped at the same address every time?



The “C-Clamp”


Interposing
“filter” Driver
Application
Application
Standard
File API
Standard
File API
Load/
Store
User
Space
File System
PM-Aware
File System
MMU
Mapp
ings
Block Driver
Kernel
Space
Persistent Memory

PM Interposition Points