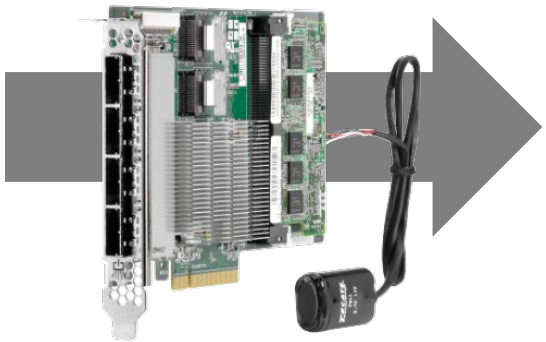
SNIA DSI '16 DATA STORAGE INNOVATION CONFERENCE
Innovation in Storage Products, Services, and Solutions
June 13-15, 2016 | Marriott San Mateo | San Mateo, CA

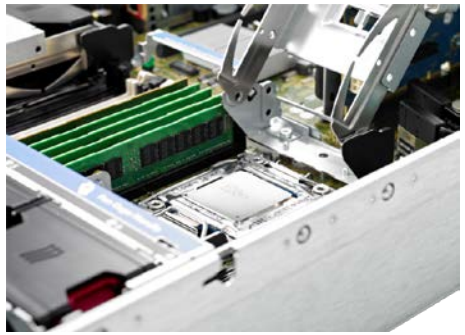
Preparing Applications for Persistent Memory

Doug Voigt
Hewlett Packard Enterprise

Persistent Memory (PM) Vision

Persistent Memory Brings Storage





Fast
Like Memory

Persistent
Like Storage

To Memory Slots

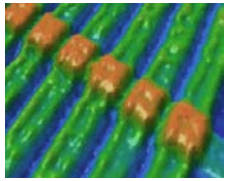
Make data durable without doing I/O

Technology Trends

- The flash revolution marches on
- NVDIMMs are delivering memory speed today
- New PM technologies bring similar speed with much larger capacity
- Fabric memory creates larger shared pools

SSD

NVDIMM

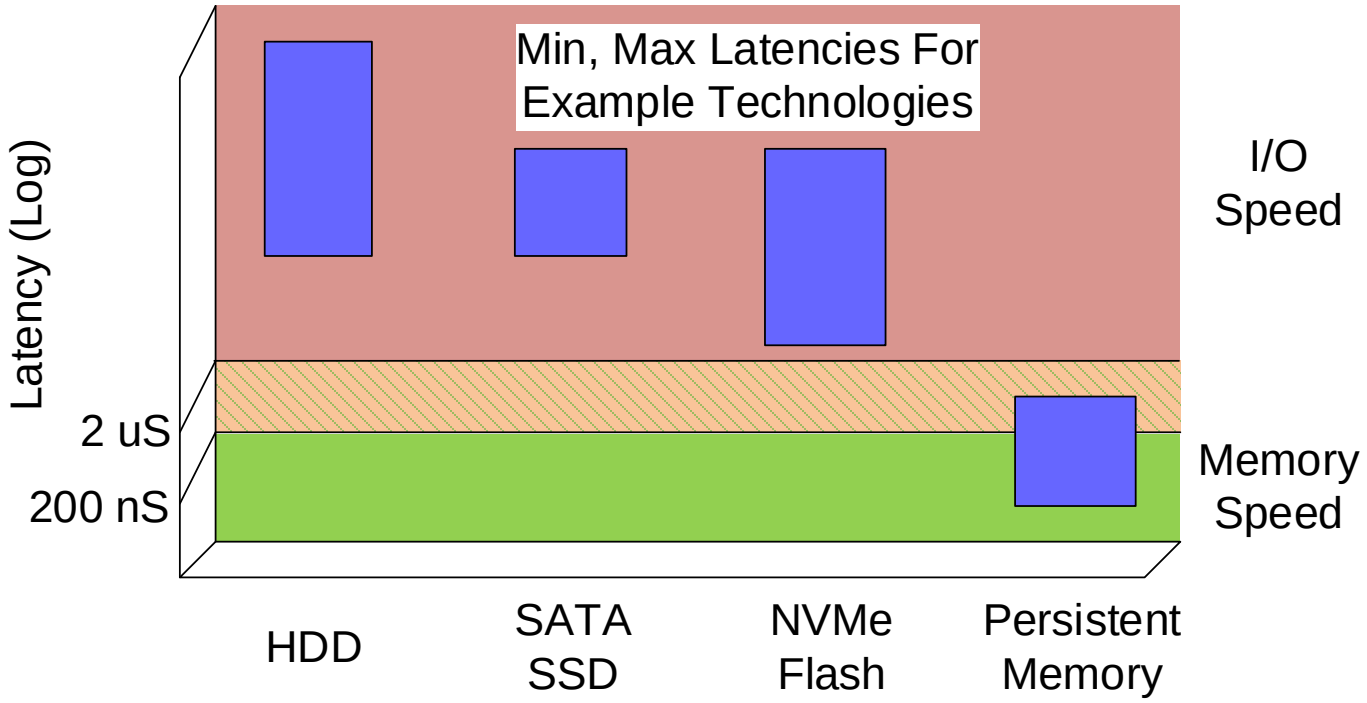


Persistent Memory

Fabric Memory

Advantages of memory speed

- Order of magnitude latency reduction
- Expands in-memory application trend

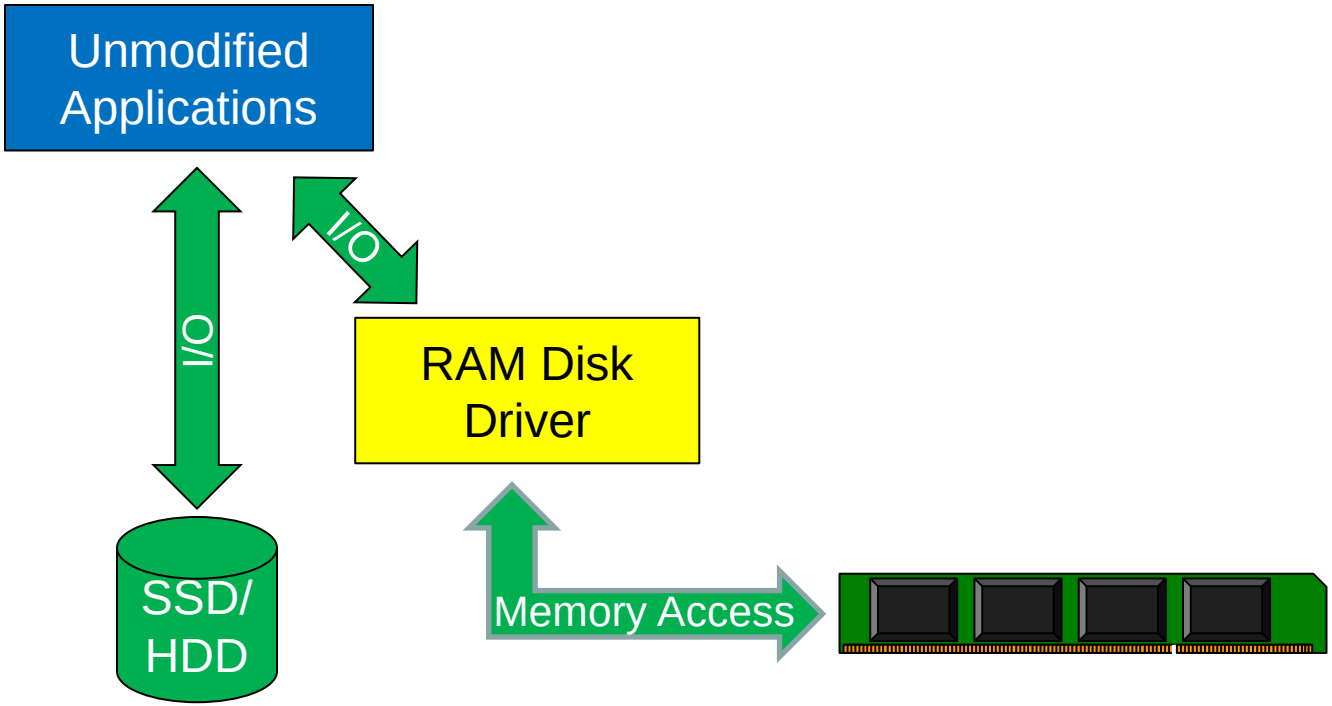


Mission of the NVM PM TWG

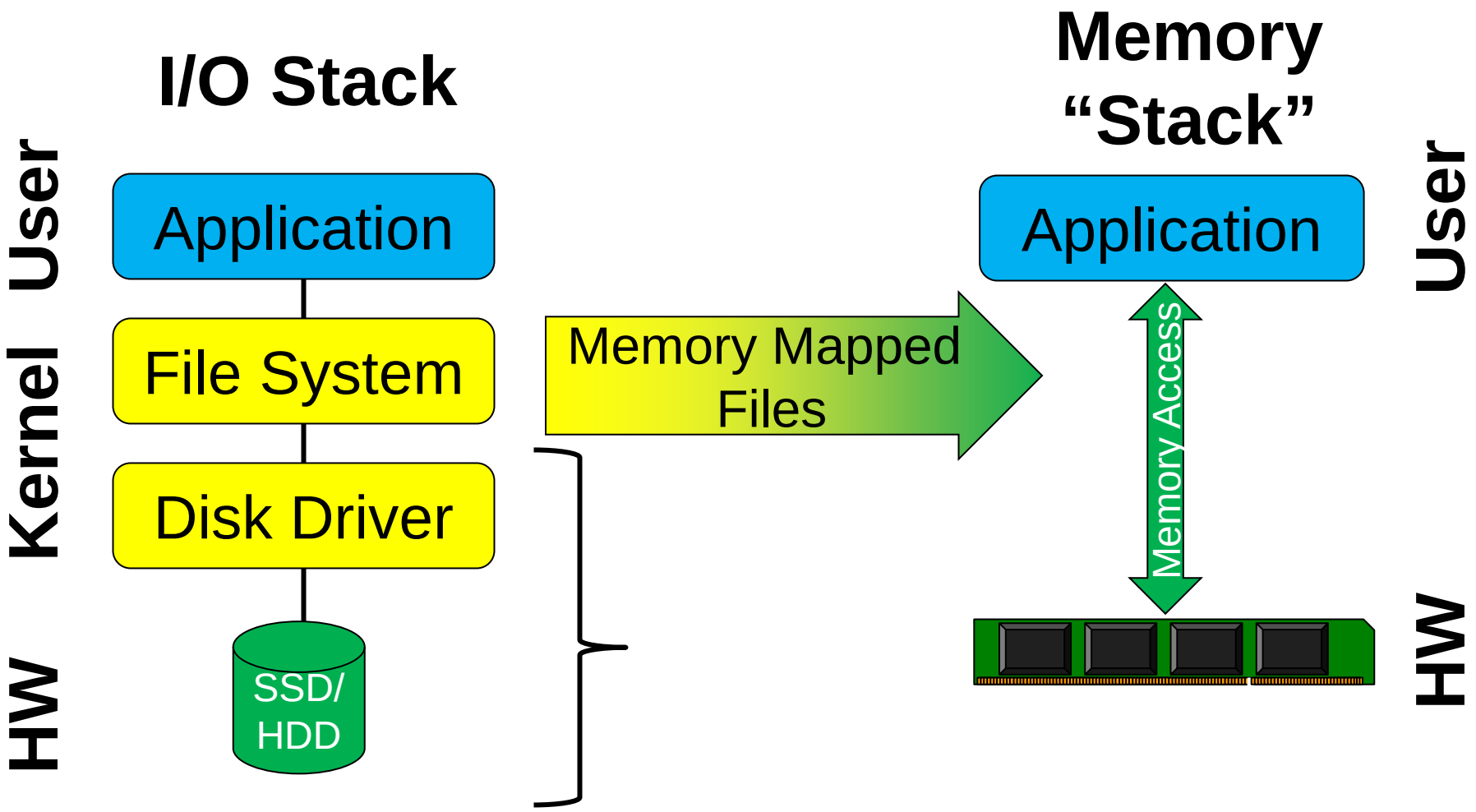
- Accelerate the availability of software that enables NVM (Non-Volatile Memory) hardware.
  - Hardware includes SSD's and PM
  - Software spans applications and OS's
- Create the NVM Programming Model
  - Describes application visible behaviors
  - Allows API's to align with OS's
  - Has exposed optimization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

Rapid Deployment of PM using RAM Disks

- Avoids application disruption
- Limits potential performance advantage



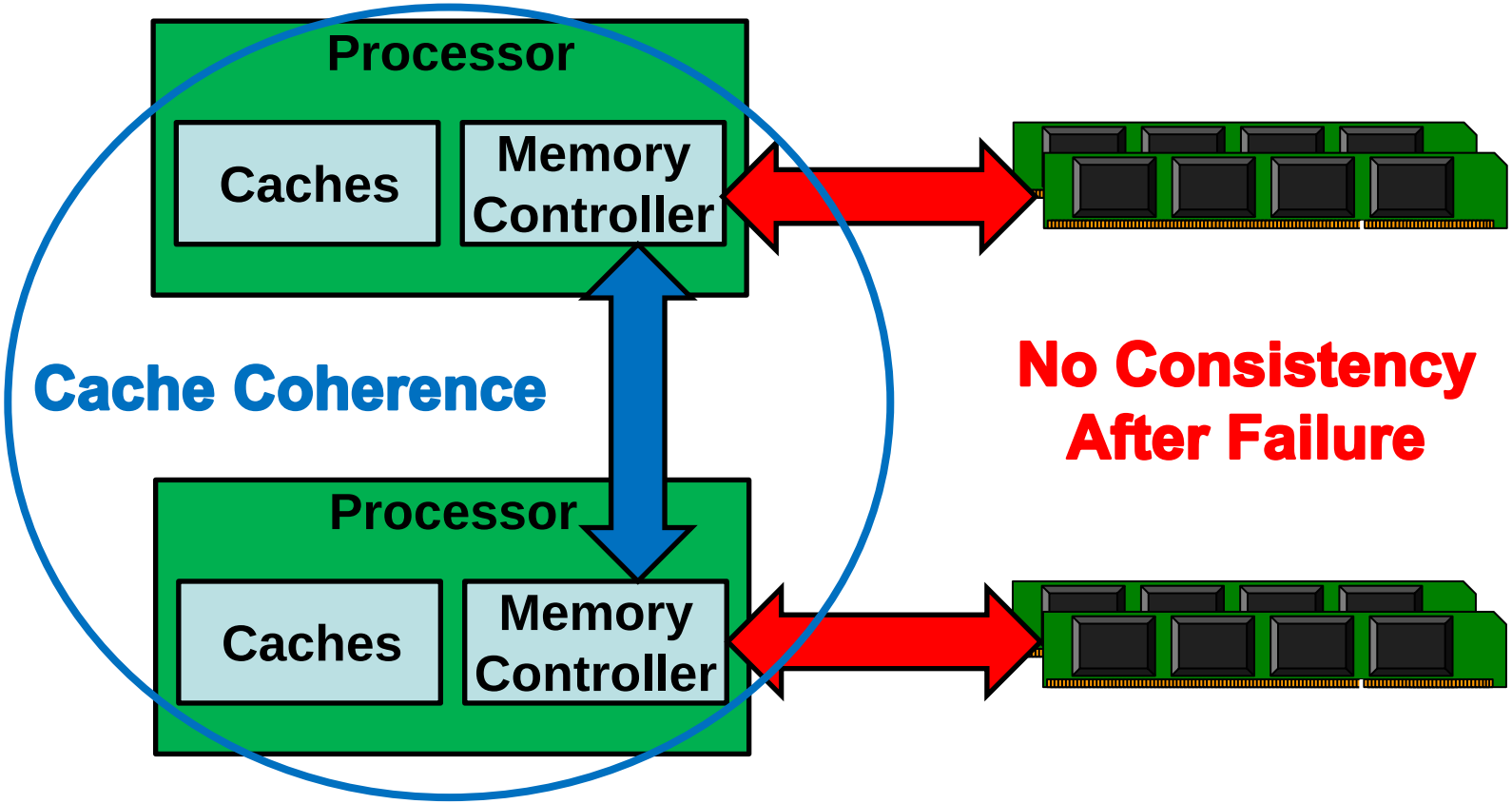
Memory mapped files



Challenges of memory speed

- Applications must change for optimal performance
  - Use PM data structures
  - Different style of error handling
- Processors have volatile write caches
  - Normal write flow does not account for PM
  - Requires the use of flush instructions
- Elimination of Disk IO has ripples
  - Failure atomicity required for data recoverability
  - Leads to more transactional memory access

What Processors Automatically Provide

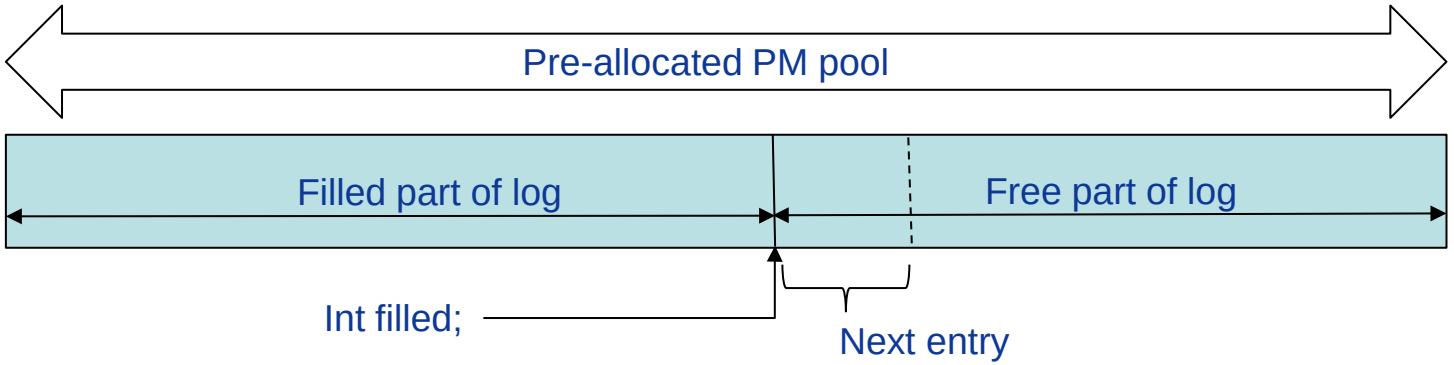


Data recoverability requires cache flushing

Persistent Memory Data Structures

- Persistent memory data structures must assure recoverability after abrupt power loss
- Single data structures can be made intrinsically recoverable
  - Limit in-place updates
  - Groups of changes are slaved to a single memory access
- Groups of data structures must use transactions
  - Classical approaches apply
  - Implemented using PM data structures

Trivial Example: Append Only Log



Append pseudocode:

```
<Create new log entry in free space>
Flush(new entry);
filled = filled + size(new entry);  # Atomic update to fundamental data type
Flush(filled);
```

NVM Library

- http://pmem.io/nvml
- PM assist functions
  - Map, Flush, Allocation
- PM Data Structures
  - Log, Block and more
- PM Object
  - Root, Transactions, Type Safety and more

Language extensions for persistent memory

- Features similar to the NVM library can be integrated into standard programming languages
  - More convenient
  - More sophisticated
  - Safer

Atlas: Leveraging Locks for Non-volatile Memory Consistency

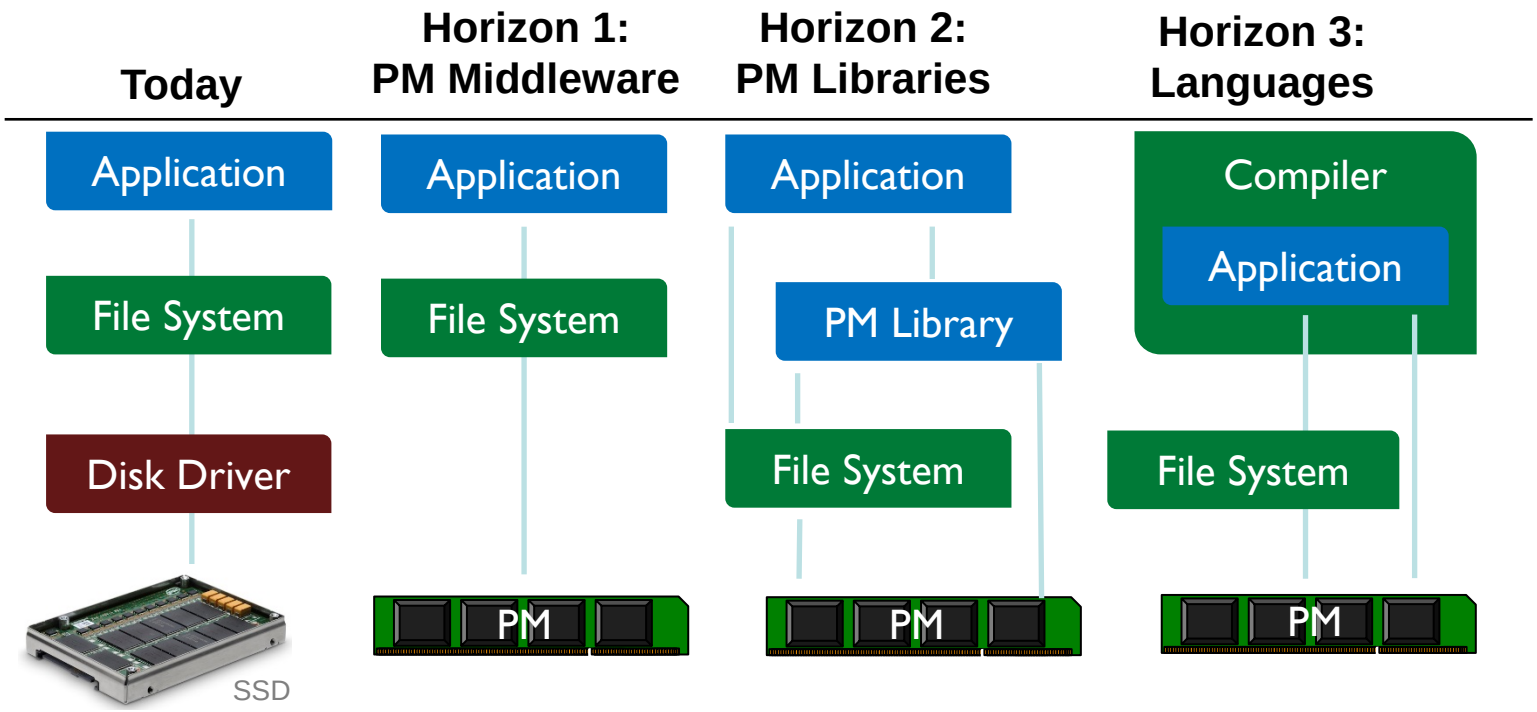
Failure atomic code sections based on existing critical sections

NVM support for C Applications

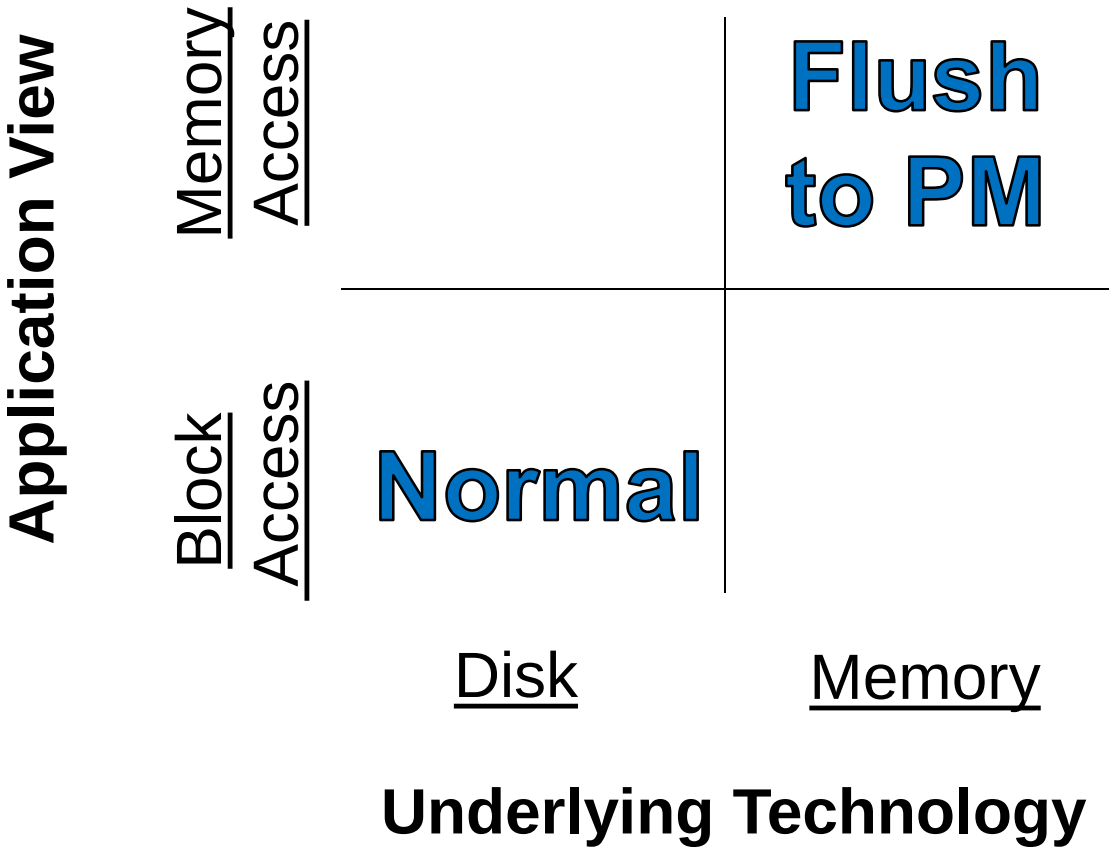
PM region file management, transactions with locks, heaps

Application Data Access Evolution

Reaping the advantages of persistent memory with step-wise innovation

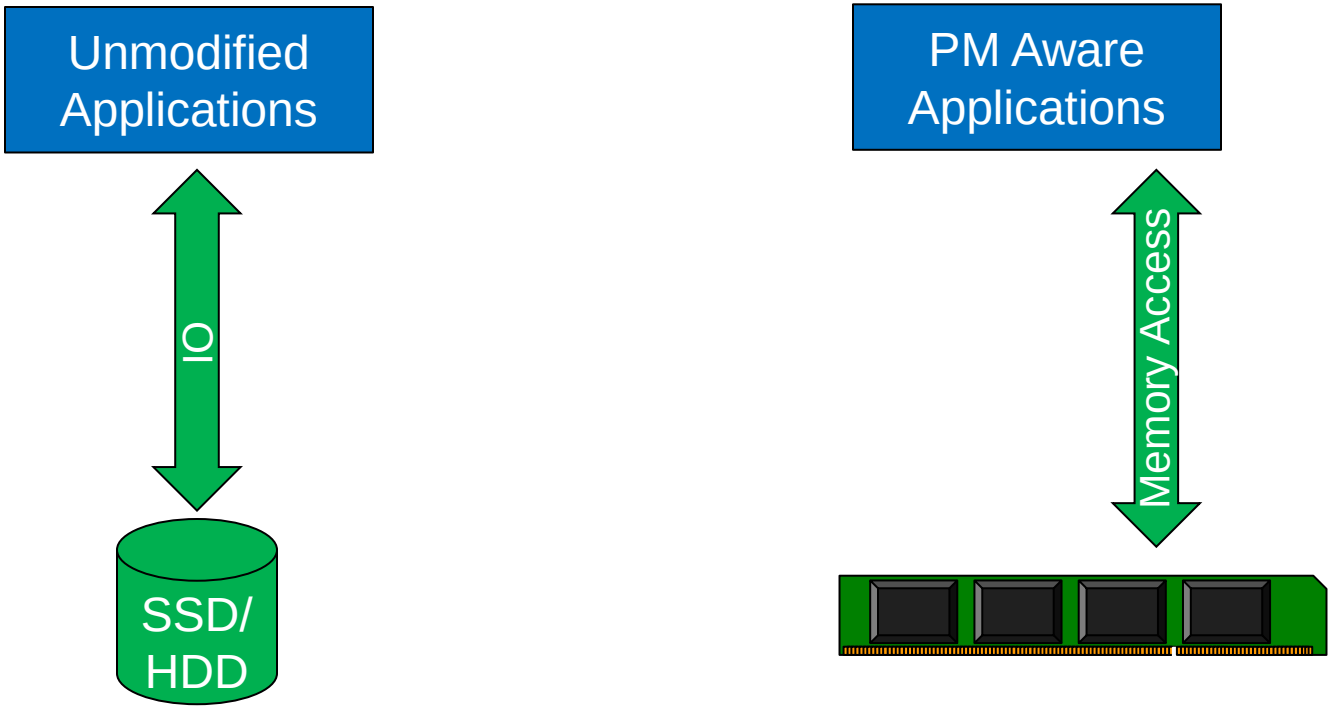


Technology Access Matrix

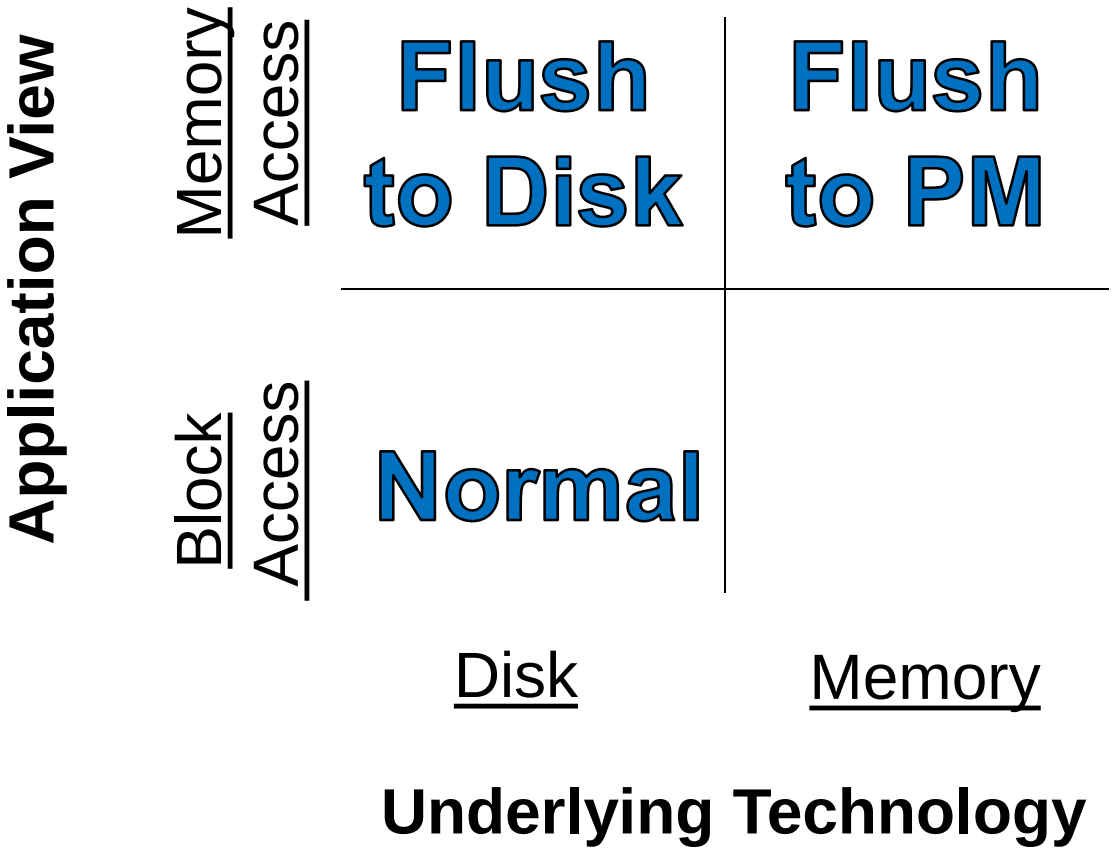


Separate Stacks

Application requires a system with the correct technology

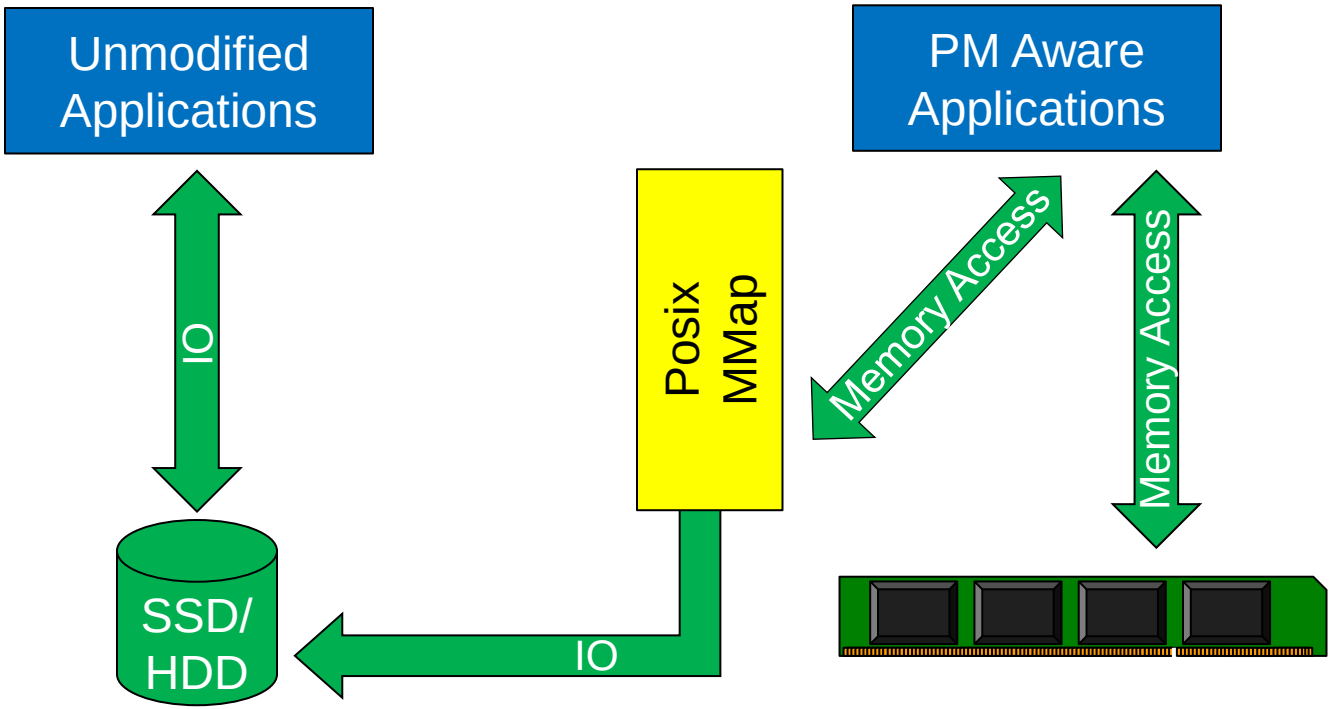


Technology Access Matrix

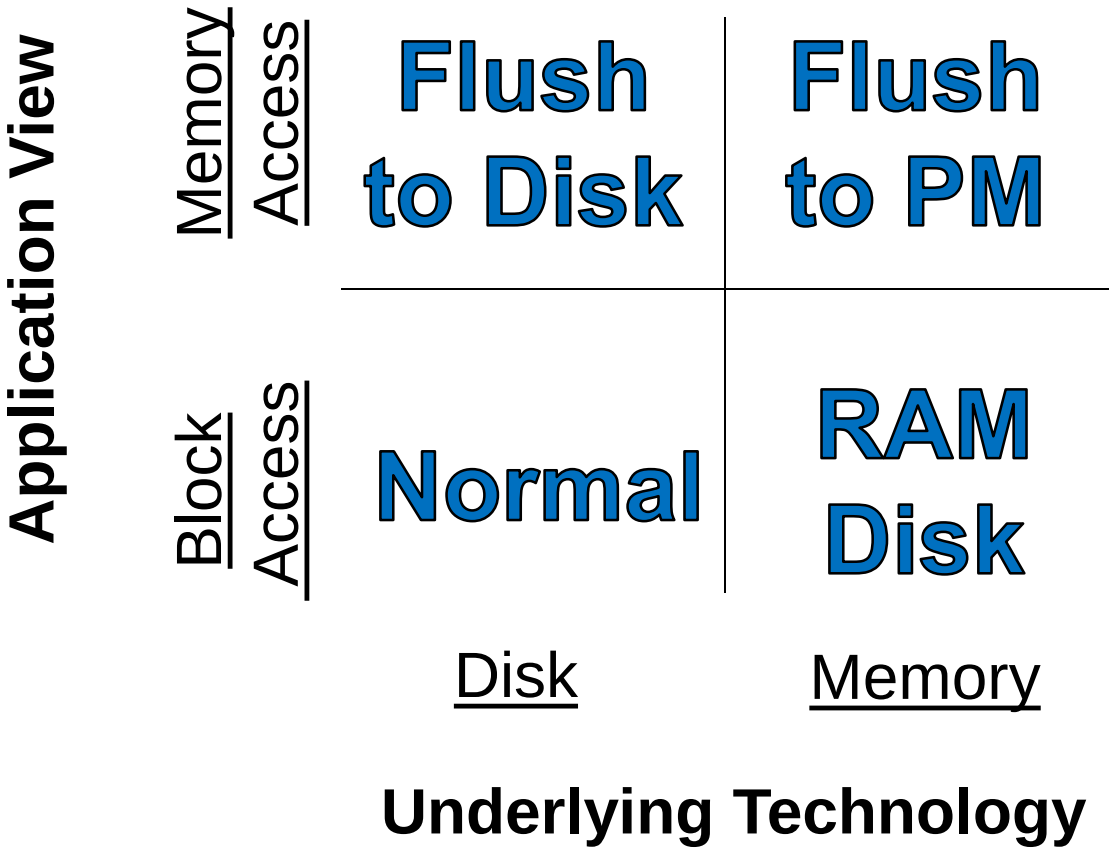


Dual Stack Scenario with Posix MMap

Emulate memory access with flush to disk
(Existing Posix file system feature called Mmap)

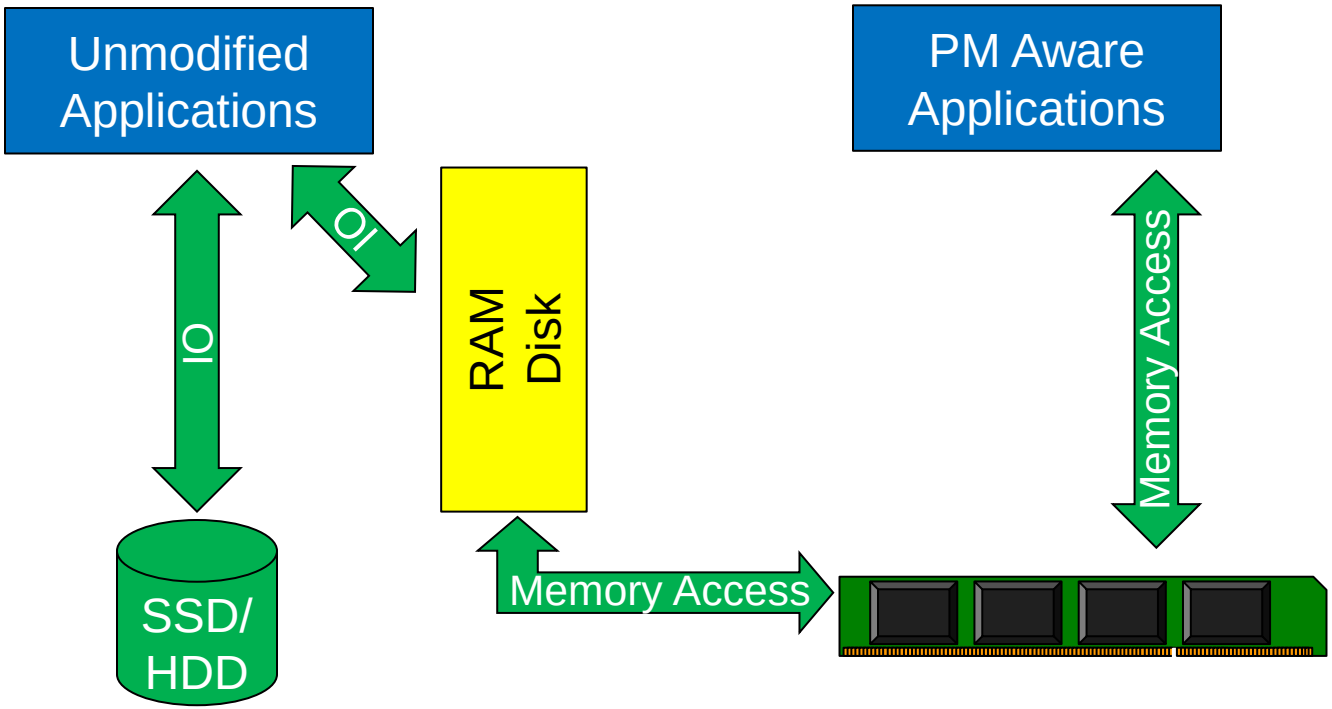


Technology Access Matrix

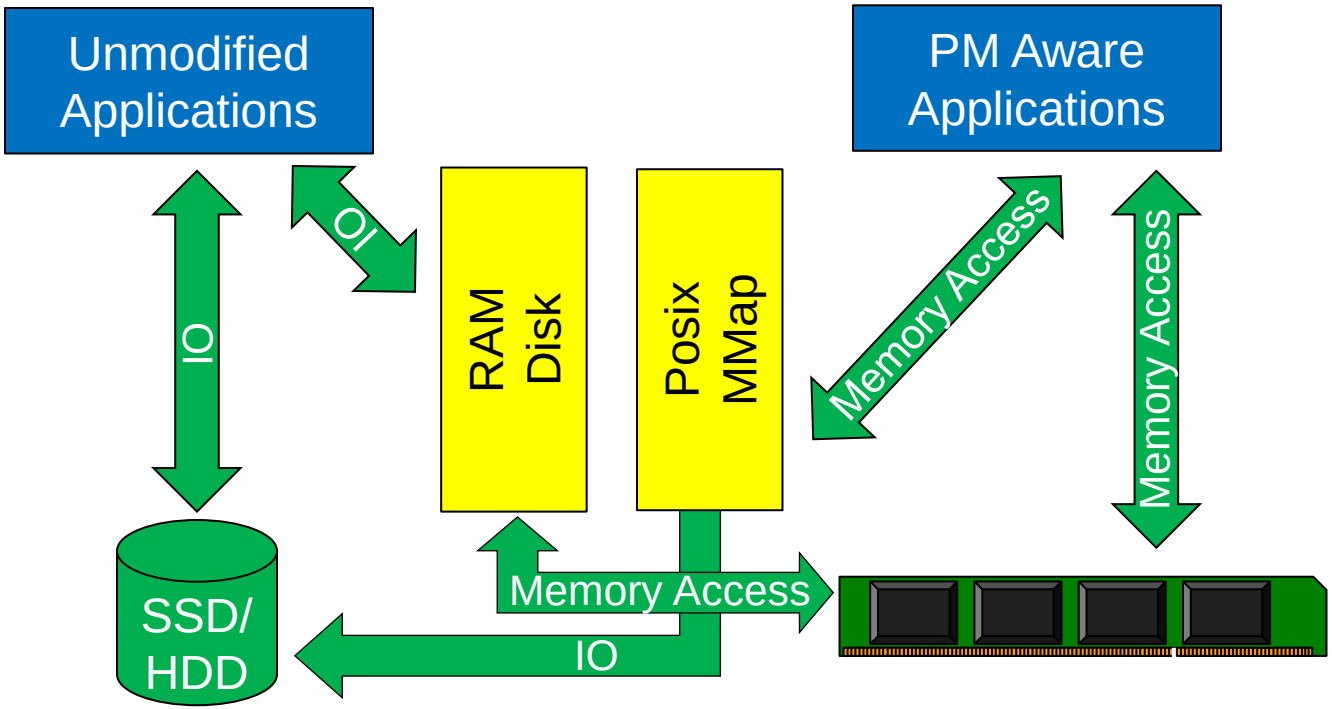


Dual stack Scenario with RAM Disk

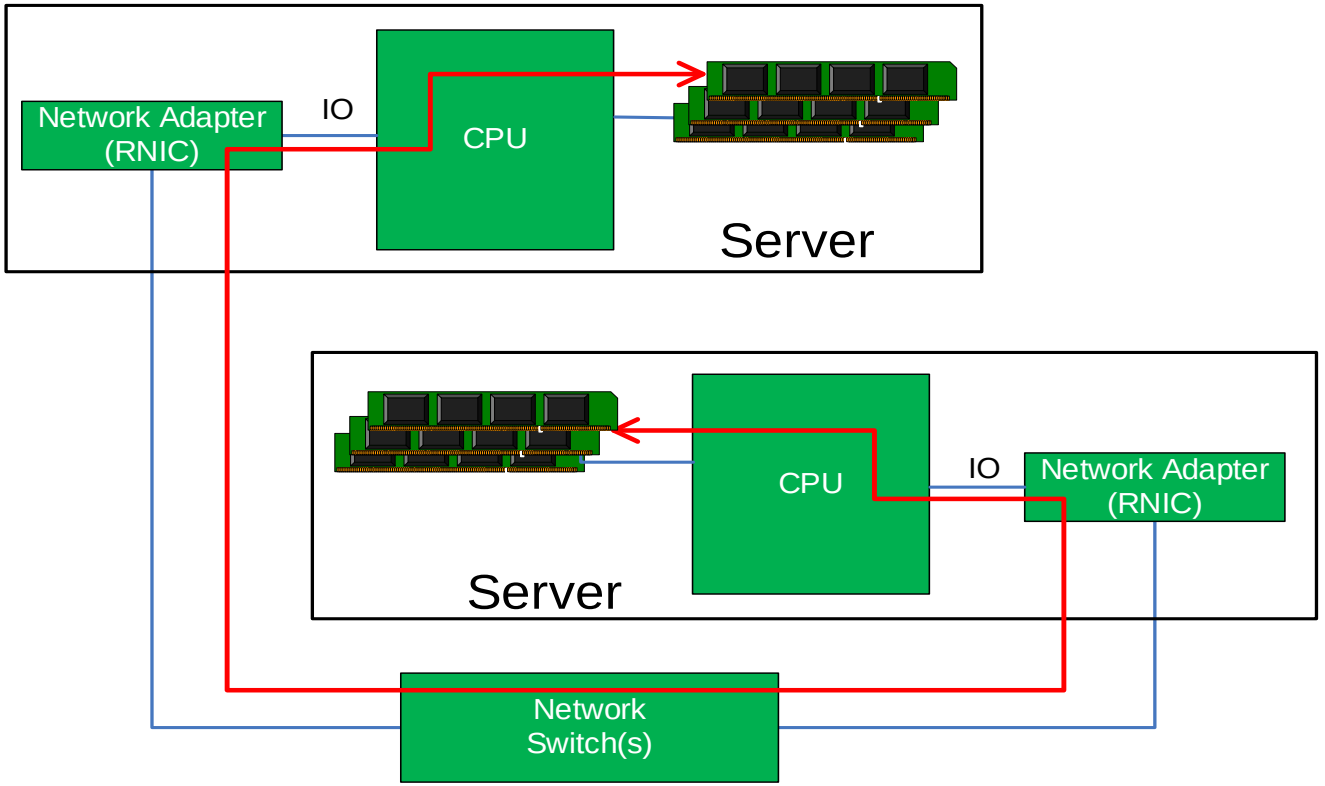
Emulate disk with RAM disk driver backed by PM



Dual stack Scenario



Do You Want High Availability With That PM?



- There is no hardware flush action over PCI
- AND there is no RDMA completion that indicates persistence
- SO software is involved at the remote node

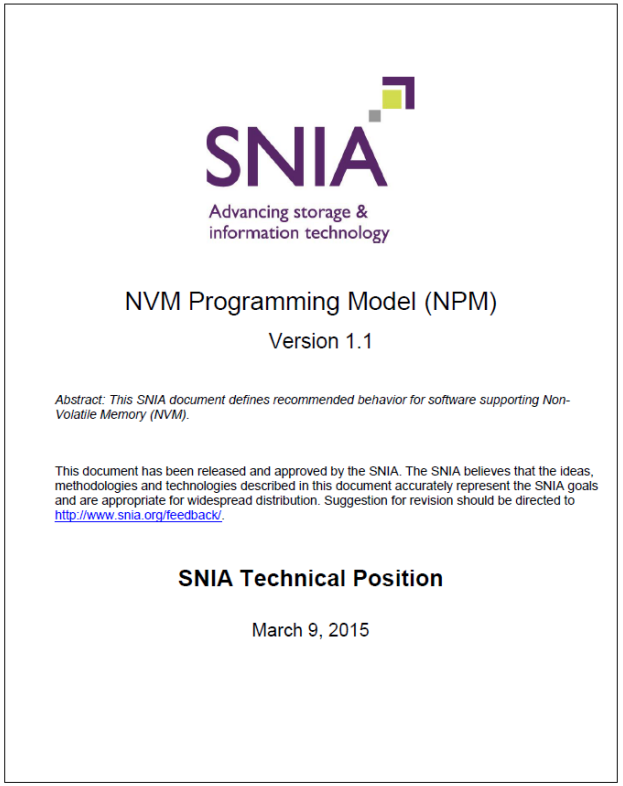
Remote Access Challenge

- How can remote access PM latency be reduced?
- 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
- Reaches beyond storage
  - Applications
  - Memory
  - Networking
  - Processors


SNIA
Advancing storage & information technology

NVM Programming Model (NPM)

Version 1.1

Abstract: This SNIA document defines recommended behavior for software supporting Non-Volatile Memory (NVM).

This document has been released and approved by the SNIA. The SNIA believes that the ideas, methodologies and technologies described in this document accurately represent the SNIA goals and are appropriate for widespread distribution. Suggestion for revision should be directed to http://www.snia.org/feedback/.

SNIA Technical Position

March 9, 2015

SNIA DSI '16 DATA STORAGE INNOVATION CONFERENCE
Innovation in Storage Products, Services, and Solutions
June 13-15, 2016 | Marriott San Mateo | San Mateo, CA

Preparing Applications for Persistent Memory

Doug Voigt
Hewlett Packard Enterprise