SNIA CSI | CLOUD STORAGE SNIA SSSI | SOLID STATE STORAGE

Containers and Persistent Memory

Live Webcast
July 27, 2017

SNIA Legal Notice

SNIA CSI | CLOUD STORAGE
SNIA SSSI | SOLID STATE STORAGE

- The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Today's Presenters

Chad Thibodeau
Veritas

Arthur Sainio
SMART Modular

Mark Carlson
Toshiba

Alex McDonald
NetApp

SNIA-At-A-Glance

SNIA-at-a-Glance

160
unique member companies

2,500
active contributing members

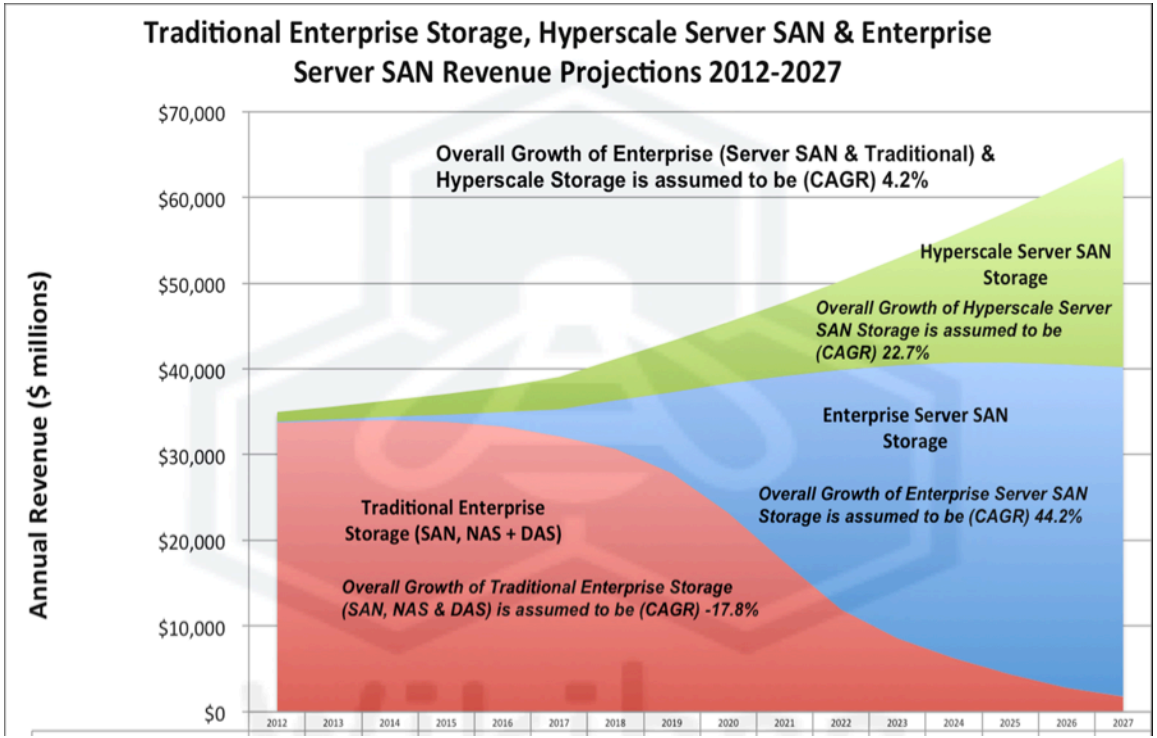
50,000
IT end users & storage pros worldwide

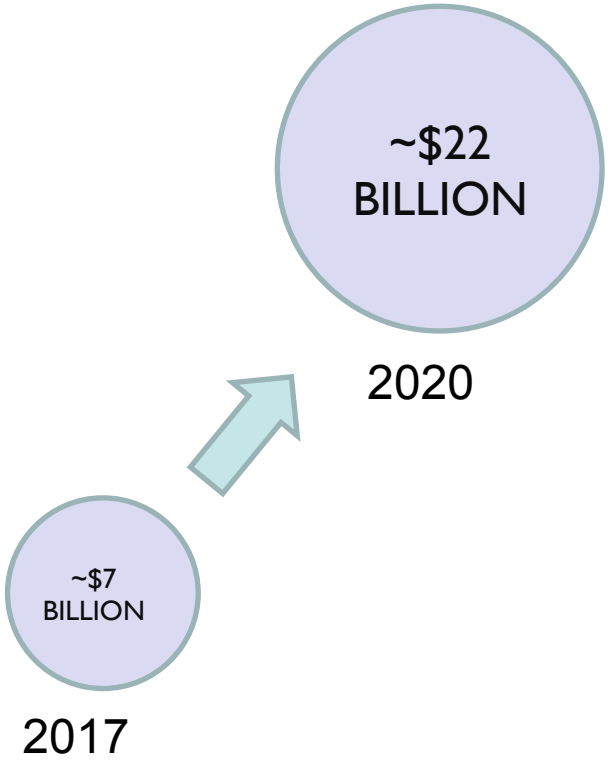
Agenda

- Persistent Storage for Containers
- Persistent Memory for Containers
- Infrastructure Software Changes for Persistent Memory-based Containers

- Persistent Storage for Containers
- Persistent Memory for Containers
- Infrastructure Software Changes for Persistent Memory-based Containers

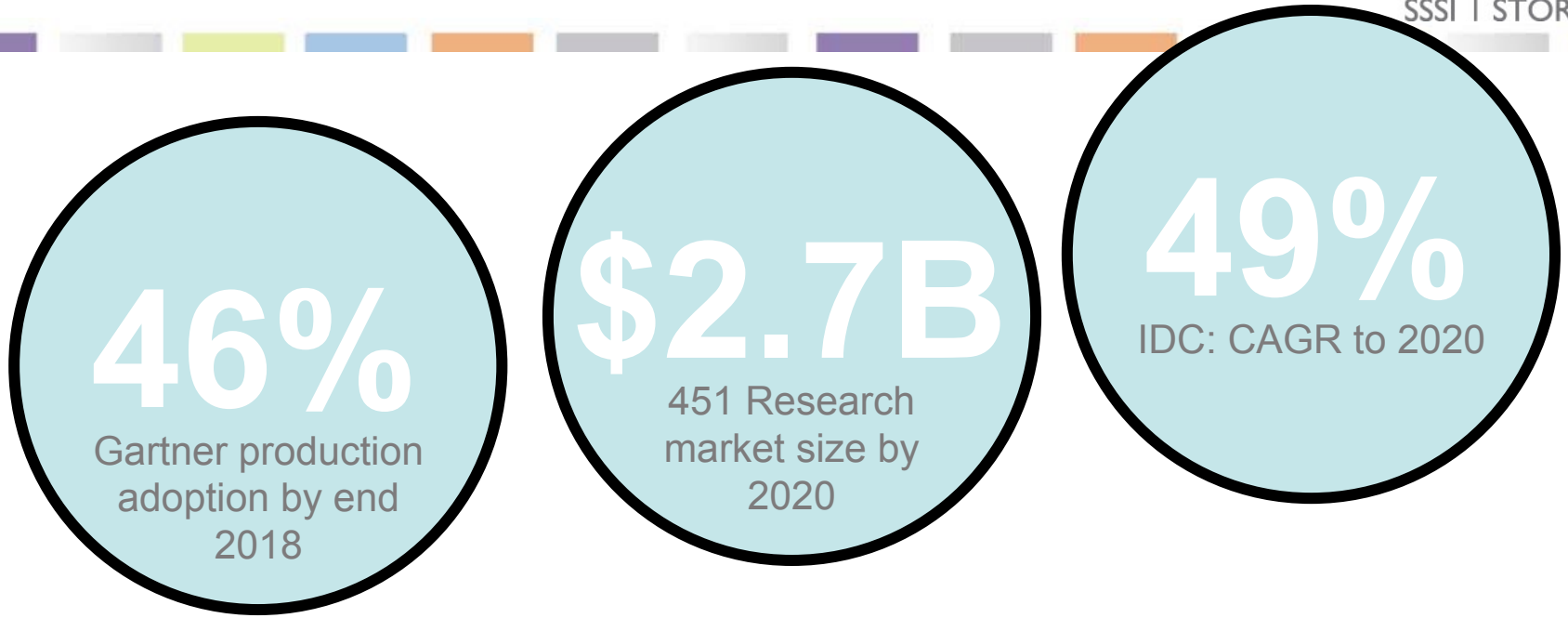
The SDS Market





Source: August 2016 Wikibon report "Server SAN Readies for Enterprise and Cloud Domination"

The Containers Landscape



"78% of DevOps early adopters have already invested in or plan to invest in containers and container orchestration within the next 12 months"

IDC March 2017

Storage Services for Containers

Persistent Storage Management	Scale-Out	Snapshots	I/O Acceleration
Quality of Service	Encryption	Disaster Recovery	Heterogeneous Support

Gaps in the Container Storage Ecosystem

SNIA CSI | CLOUD STORAGE
SNIA SSSI | SOLID STATE STORAGE

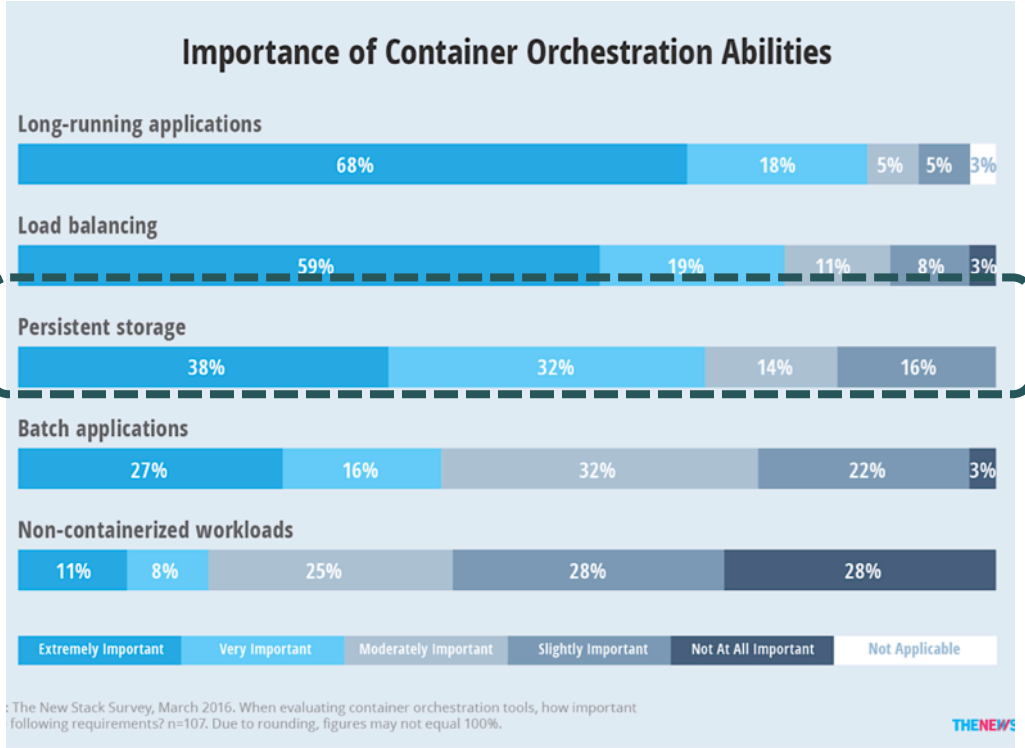
Persistence		Ensure state data is preserved and available across container lifetimes
Performance & Resiliency		Meet performance and resiliency needs specific to each application(s)
Complexity		Provide simplified storage operations across a variety of infrastructure
Legacy Applications		Enable legacy applications to take advantage of container volumes

Stateful vs Stateless

"**Stateful container** apps represent the next big **IT challenge**(1)"

"**Persistent storage** among top issues for container enterprise-readiness in **production**(2)"

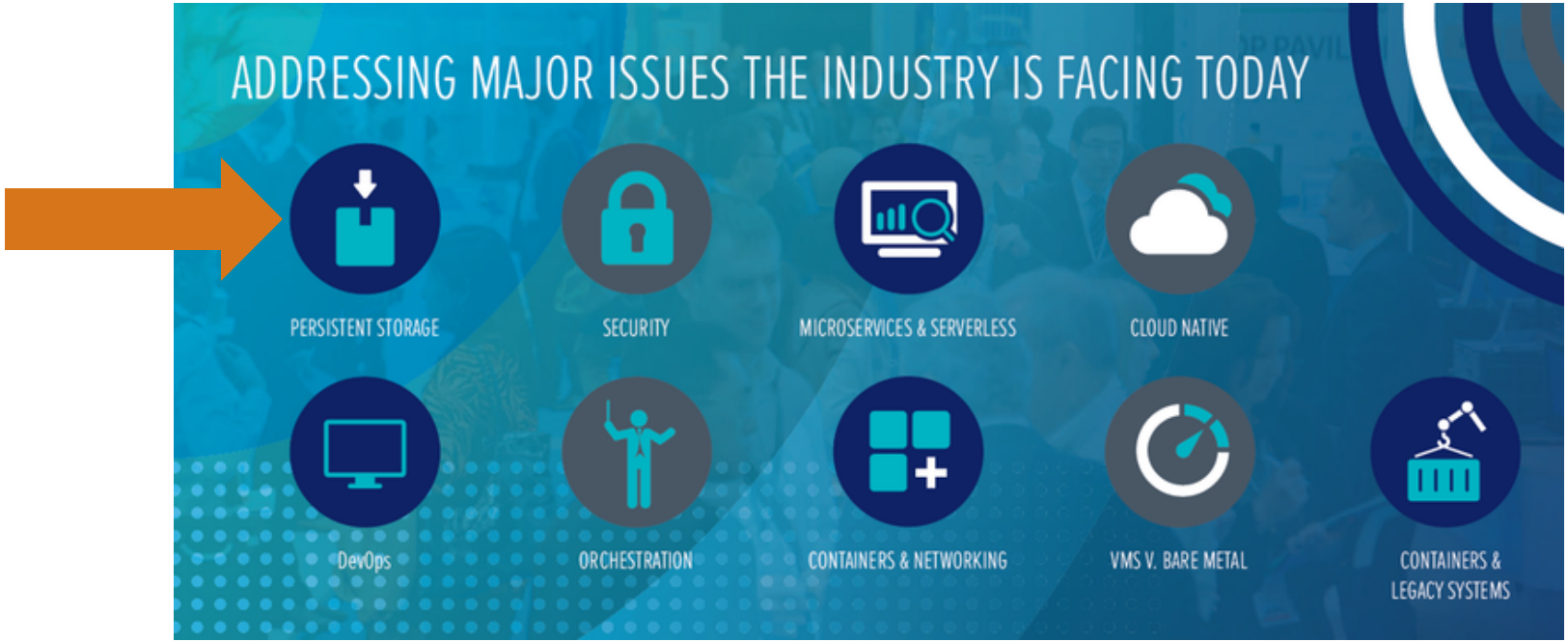
"**Stateful Database** applications such as Redis, MySQL, MongoDB among **most pulled** images on Docker Hub(2)"



(1) Container Journal
(2) Gartner

Many Container Challenges

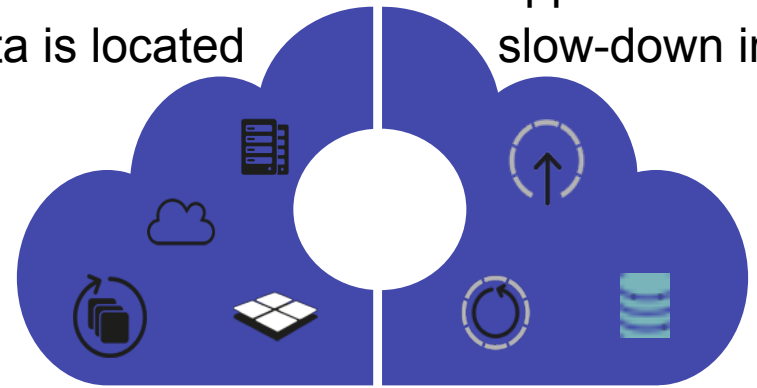
Container World 2017



Storage Persistence — Why

Data Accessibility
Run containers anywhere without worries about where data is located

Deployment Cycles
Traditional storage approaches slow-down innovation



Data Availability
Data needs to be always on no matter what happens

Storage Costs
Storage defined by Software using any commodity HW or Cloud

Docker Storage Examples

Registry — Cold storage of container images

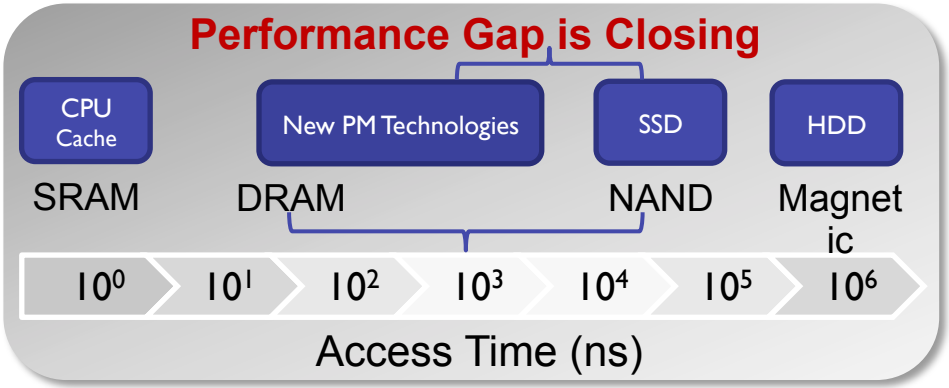
Graph — Active storage of running container images

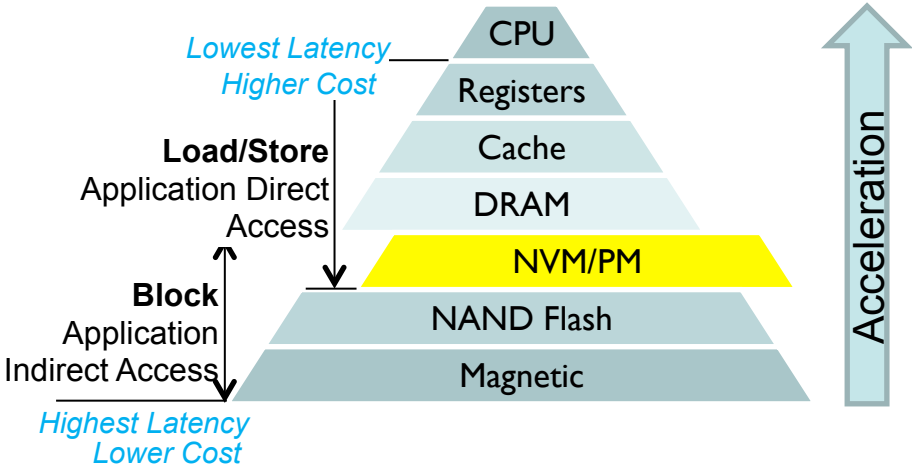
Volume — Persistent block storage for data

- Persistent Storage for Containers
- Persistent Memory for Containers
- Infrastructure Software Changes for Persistent Memory-based Containers

Memory – Storage Hierarchy

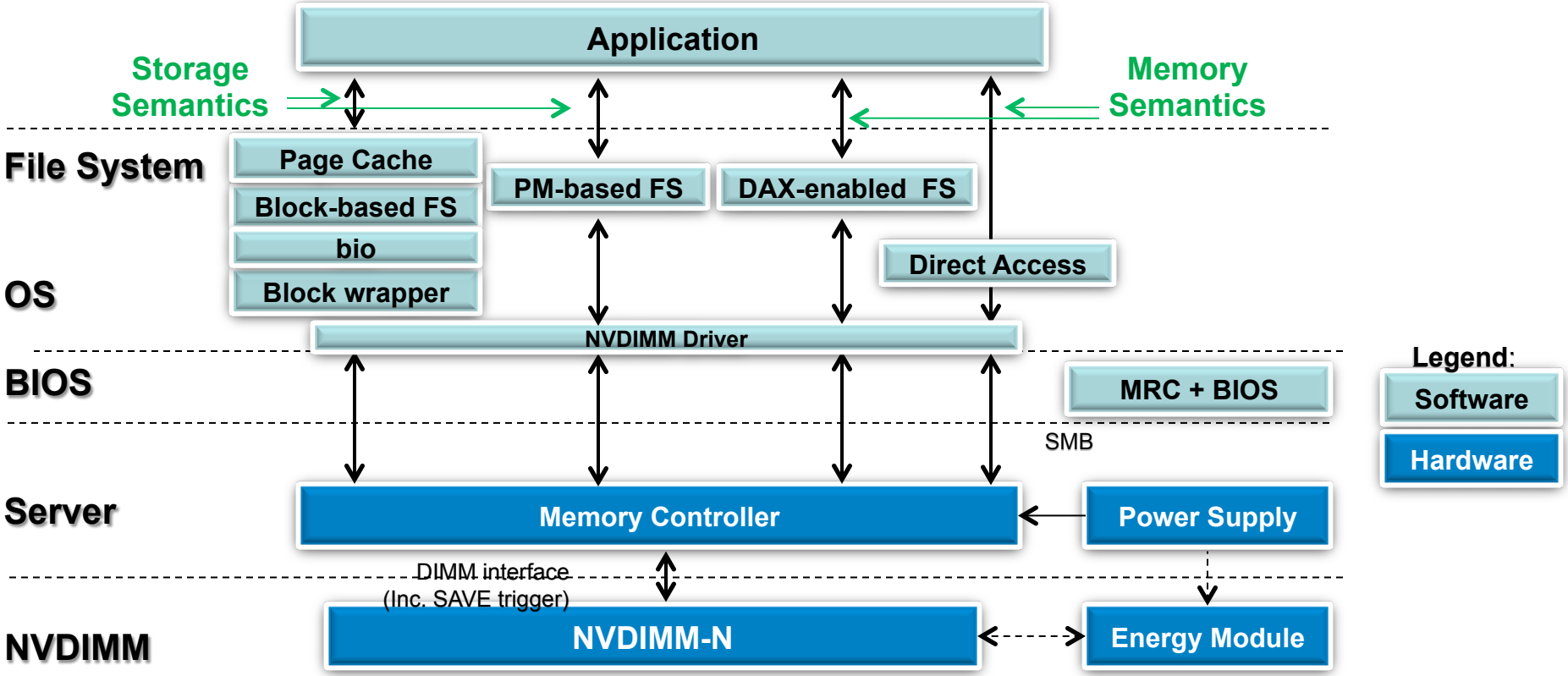
- Data-intensive applications need fast access to storage
- Persistent memory is the ultimate high-performance storage tier
- NVDIMMs have emerged as a practical next-step for boosting performance





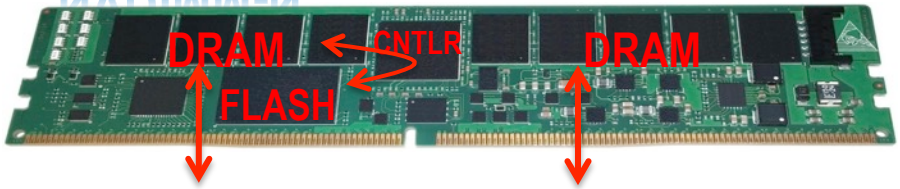
Source; HPE/SNIA 2015

NVDIMM-Enabled Ecosystem



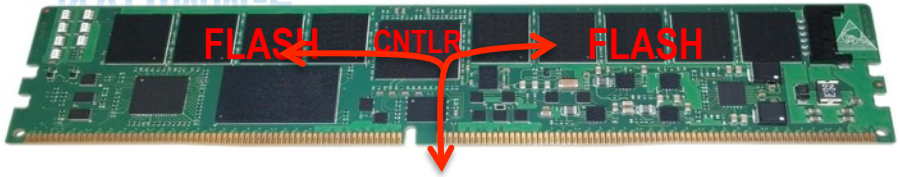
NVDIMM Types

NVDIMM-N



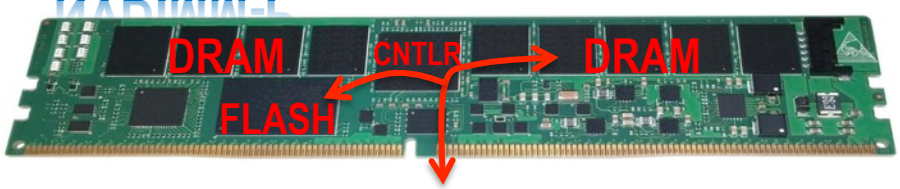
- Host has direct access to DRAM
- CNTLR moves DRAM data to Flash on power fail
- Requires backup power (typically 10's of seconds)
- CNTLR restores DRAM data from Flash on next boot
- Communication through SMBus (JEDEC std.)

NVDIMM-F



- Host accesses Flash through controller
- Block-access to Flash, similar to an SSD
- Enables NAND capacity in the memory channel (even volatile operation)
- Communication through SMBus (JEDEC std. TBD)

NVDIMM-P



- Functionality of -N and -F
- Host accesses memory through a media controller on the NVDIMM-P
- Developed to support adding Persistent Memory on the DDR5 host memory interface (e.g. NAND, MRAM, PCM, ReRAM, 3DXPoint)

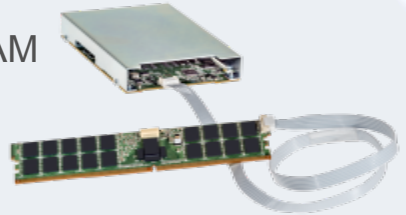
Application Access to NVDIMMs

- Disk-like NVDIMMs (Type F or P)
  - Appear as disk drives to applications
  - Accessed using disk stack
- Memory-like NVDIMMs (Type N or P)
  - Appear as memory to applications
  - Applications store variables directly in RAM
  - No IO or even DMA is required
- Memory-like NVDIMMs are a type of persistent memory
- NVDIMMs are available today!

NVDIMM-N Summary

NVDIMM-N
Standardized

- Memory mapped DRAM. Flash is not system mapped
- Access Methods -> byte- or block-oriented access to DRAM
- Capacity = DRAM DIMM (1's -10's GB)
- Latency = DRAM (10's of nanoseconds)
- Energy source needed for backup
- DIMM interface (HW & SW) defined by JEDEC



Industry Standardization

JEDEC
SNIA

Processors

intel
AMD
Others…

OS Support

Linux
Microsoft

Systems

intel
SUPERMICRO
DELL
Hewlett Packard Enterprise
Others…

- NVDIMM firmware interface table (NFIT) added in ACPI 6.0

DDR4 NVDIMM-N JEDEC Design Standard (Revision 1.0 Published Sep '16)

- Defines electrical and mechanical requirements for 288-pin, 1.2V, DDR4 NVDIMM-N
- NVDIMM-N modules adhere to the Byte Addressable Energy Backed Interface (BAEBI) Standard, JESD245, that provides detailed logical behavior, interface, and register definitions
  - SAVE_n: pin 230 sets an efficient interface to signal a backup
  - 12V: pin 1, 145 provides power for backup energy source
  - EVENT_n: pin 78 asynchronous event notification pin
  - Byte Addressable SMBus interface (JESD245)
  - JEDEC defined SPD/Registers to comply with DDR4 RDIMM

NVDIMM-N Applications

- In-Memory Database: Journaling, reduced recovery time, Ex-large tables
- Traditional Database: Log acceleration by write combining and caching
- Enterprise Storage: Tiering, caching, write buffering and meta data storage
- Virtualization: Higher VM consolidation with greater memory density
- High-Performance Computing: Check point acceleration and/or elimination
- Rendering software in computer graphics imaging

How NVDIMM-N's Improve Performance

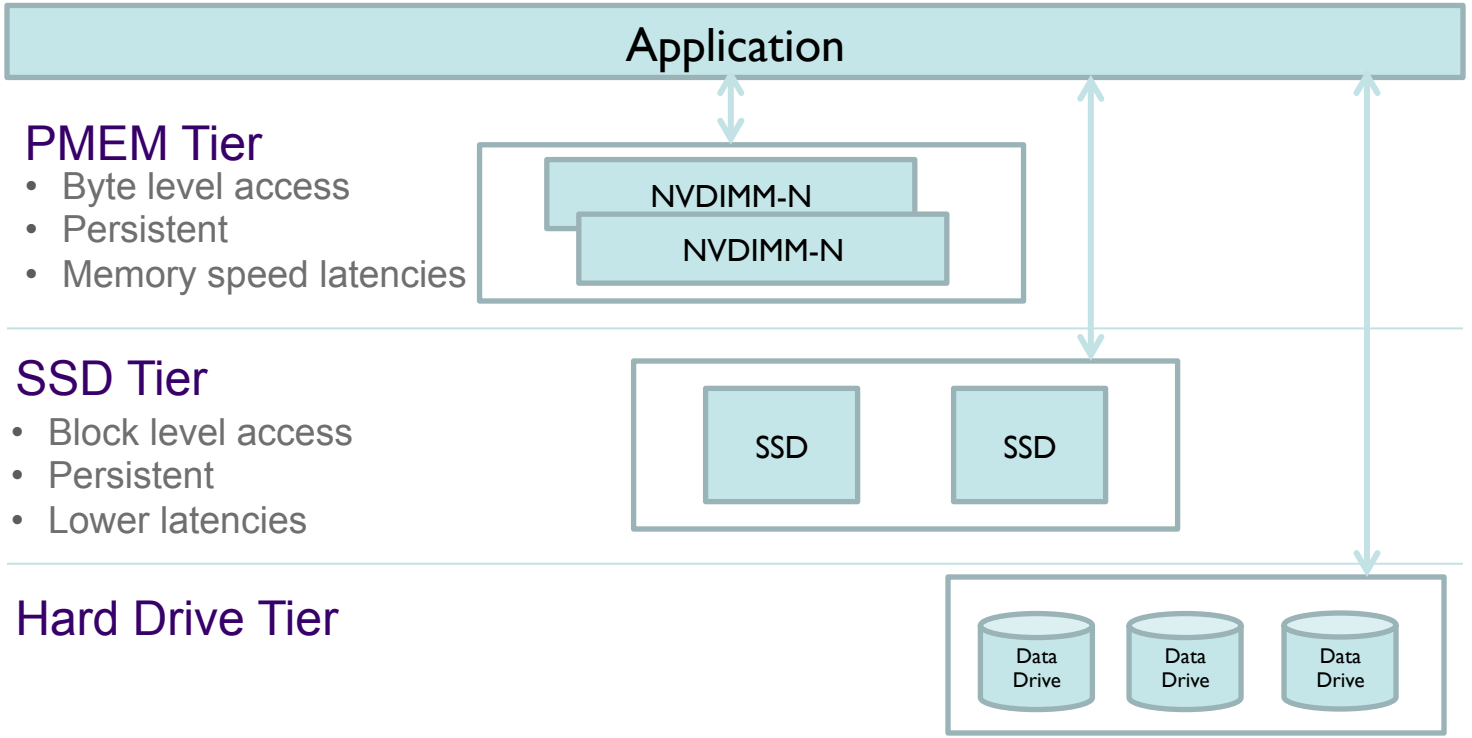
- NVDIMM-Ns are byte addressable and can store any type of transient data
- Direct access to records removes disk IO and all the software overhead
- A memcached structure is dramatically faster than even the best solid-state solution
- Since NVDIMM-Ns appear as DRAM to the system, using RDMA to create redundancy and cluster sharing is a given

NVDIMMs: Overcoming Challenges for Adoption

- Prior to JEDEC standardization of the Byte Addressable Energy Backed Interface (BAEBI) specification, NVDIMM-N vendors had proprietary non-volatile controller register interfaces

- Qualifying NVDIMM-N's is more of a platform validation process.
  - Separate combinations of processor, motherboard, memory reference code (MRC), power supply and platform memory configurations need to be tested

NVDIMM Use Case
Application Persistent Data Tier

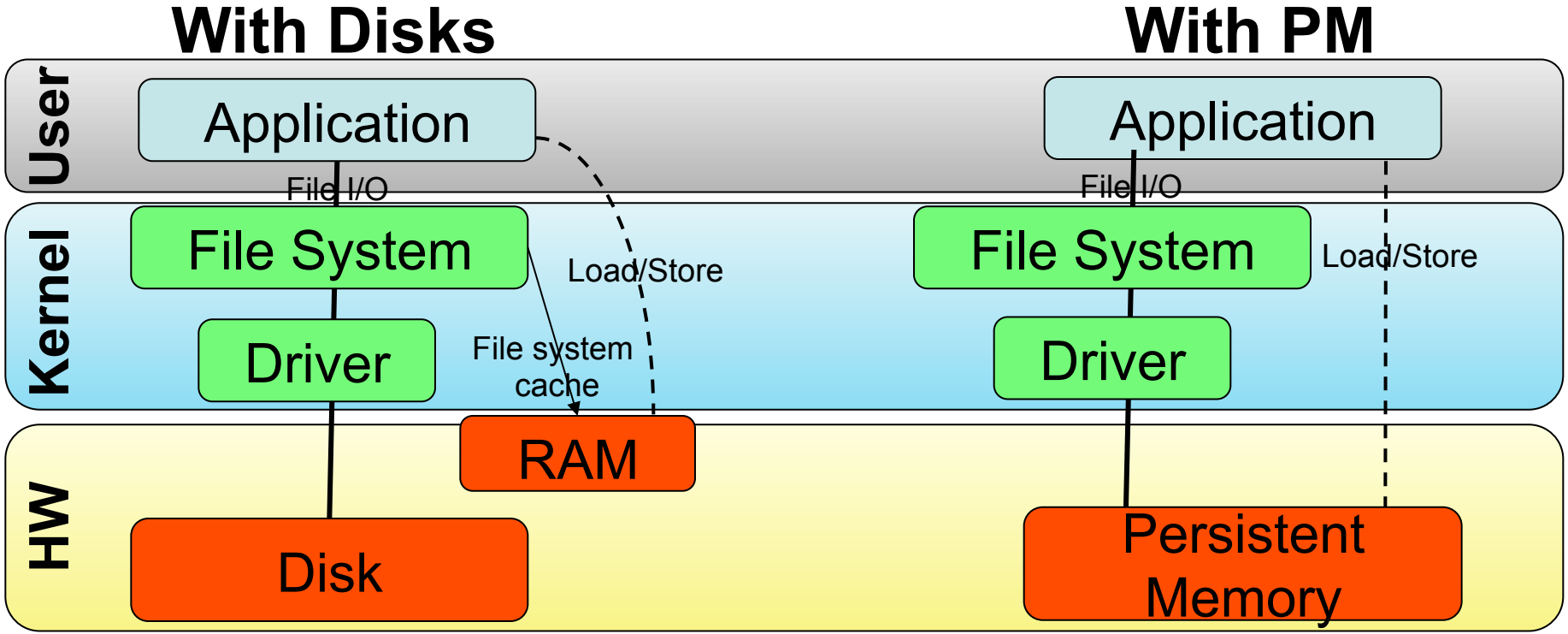


- Persistent Storage for Containers
- Persistent Memory for Containers
- Infrastructure Software Changes for Persistent Memory-based Containers

How This New Paradigm Changes the Infrastructure

- Applications will use persistent memory in several ways
- To minimize the impact on existing applications, their use of I/O interfaces are “wrapped” by new filesystem and block drivers
  - However while this speeds things up, this legacy interaction is not optimal
- Applications can be re-written, new applications can be created for optimal use of persistent memory
- The SNIA has modeled the new interactions and programming constructs necessary for this
  - The NVM Programming Model will influence infrastructure architecture and design as a result

Memory Mapped File Programming Model



NVM Programming Model Specification Organization

- Disk-like non-volatile memory
  - Appears as disk drives to applications
  - Accessed as traditional array of blocks
- Memory-like non-volatile memory
  - Appears as memory to applications
  - Applications store data directly in byte-addressable memory
  - No IO or even DMA is required

"Persistent memory" refers to Memory-like non-volatile memory

SNIA *NVM Programming Model*

- Version 1.2 approved by SNIA in June 2017
  - https://www.snia.org/sites/default/files/technical_work/final/NVMProgrammingModel_v1.2.pdf
  - Major new installment on error handling
  - Optimized Flush Allowed
  - Deep Flush
- Use of memory mapped files for persistent memory
  - Existing abstraction that can act as a bridge
  - Limits the scope of application re-invention
  - Open source implementations available
- Programming Model, not API
  - Described in terms of attributes, actions and use cases
  - Implementations map actions and attributes to API's

The Four Modes

Block Mode Innovation

- Atomics
- Access hints
- NVM-oriented operations

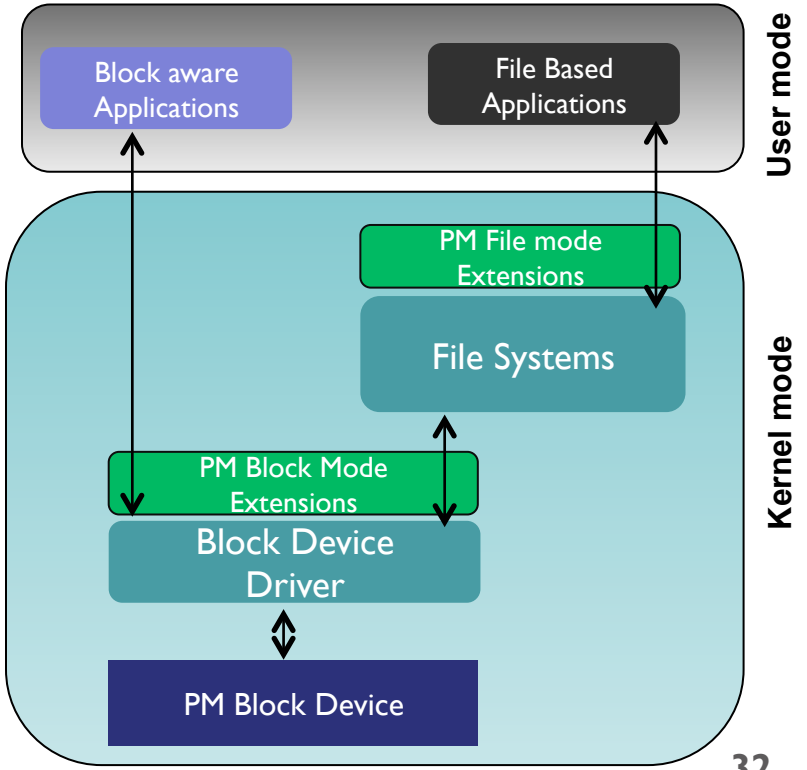
Emerging NVM Technologies

- Performance
- Performance
- Performance, cost

	Traditional	Persistent Memory
User View	NVM.FILE	NVM.PM.FILE
Kernel Protected	NVM.BLOCK	NVM.PM.VOLUME
Media Type	Disk Drive	Persistent Memory
NVDIMM	Disk-Like	Memory-Like

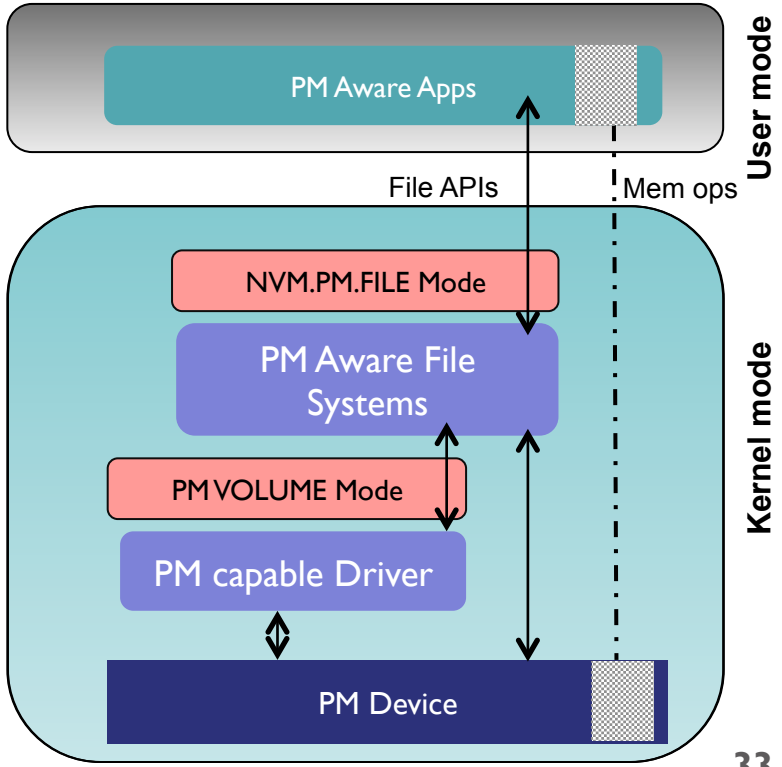
File and Block Mode Extensions

- 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



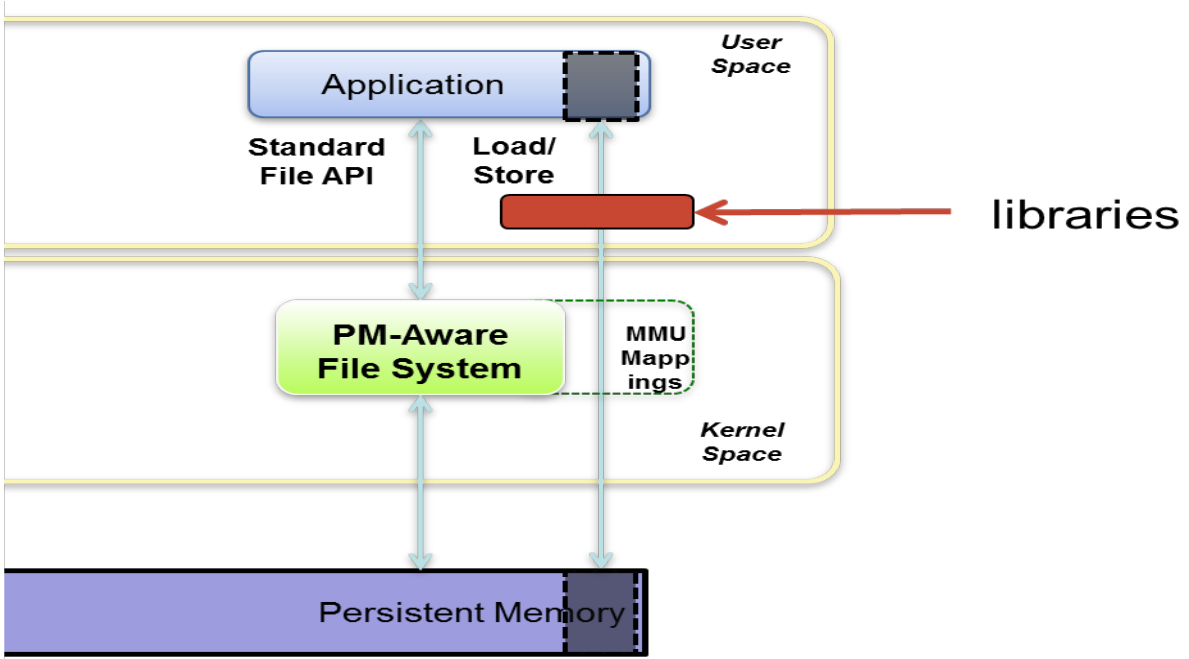
Persistent Memory (PM) Modes

- 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



Building On the Basic PM Model

- NVM.PM.FILE programming model "surfaces" PM to application
- Considering refinements to model with additional libraries
- Some may evolve into language extensions

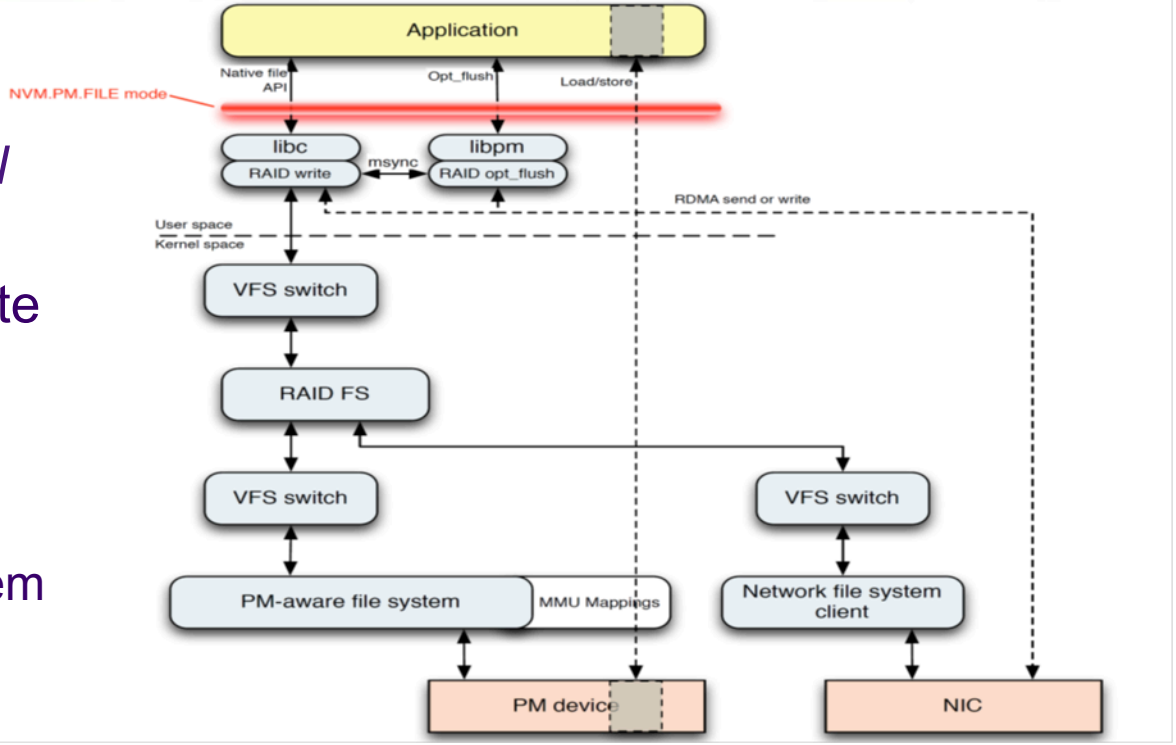


Infrastructure Changes

- Operating System
  - Filesystem changes for memory mapped files
  - Memory Management software
- Hypervisors
  - Allocation of Persistent Memory to Guests
  - Coordinating with Guest's use of PM
- Containers
  - User space libraries supporting PM
  - Support for legacy interfaces with PM aware implementations
  - Securing application data in a multi-tenant environment

Software Context Example

- Standard file API
- *NVM Programming Model* optimized flush
- Uses replications to remote system to achieve HA
  - Local file system
  - Remote file system
    - via network file system client and NIC



Application Changes

- Applications can use modified implementations of legacy interfaces to start
  - Need support in Docker, other containerizers
- Start to move applications to become aware of NVM Programming Model semantics
  - Library support
  - Compiler support
  - High availability support

Linux Kernel 4.4+ NVDIMM-N OS Support

SNIA CSI | CLOUD STORAGE

SNIA SSSI | SOLID STATE STORAGE

- Linux 4.2 + subsystems added support of NVDIMMs. Mostly stable from 4.4
- NVDIMM modules presented as device links: /dev/pmem0, /dev/pmem1
- QEMO support (experimental)
- XFS-DAX and EXT4-DAX available

DAX	File system extensions to bypass the page cache and block layer to memory map persistent memory, from a PMEM block device, directly into a process address space.
BTT (Block,Atomic)	Block Translation Table: Persistent memory is byte addressable. Existing software may have an expectation that the power-fail-atomicity of writes is at least one sector, 512 bytes. The BTT is an indirection table with atomic update semantics to front a PMEM/BLK block device driver and present arbitrary atomic sector sizes.
PMEM	A system-physical-address range where writes are persistent. A block device composed of PMEM is capable of DAX. A PMEM address range may span an interleave of several DIMMs.
BLK	A set of one or more programmable memory mapped apertures provided by a DIMM to access its media. This indirection precludes the performance benefit of interleaving, but enables DIMM-bounded failure modes.

Open Source Contributions

- Linux DAX Extensions - PM-aware file system (NVM.PM.FILE)
  - Support ext4 on NV-DIMMs
  - http://lwn.net/Articles/588218/
  - DAX changes accepted in Linux kernel 4.0
  - Support for NVDIMM detection from BIOS in kernel 4.2
- PM transactional libraries
  - NVML: http://pmem.io/nvml/
  - NVM-Direct: https://github.com/oracle/NVM-Direct

Windows NVDIMM-N OS Support

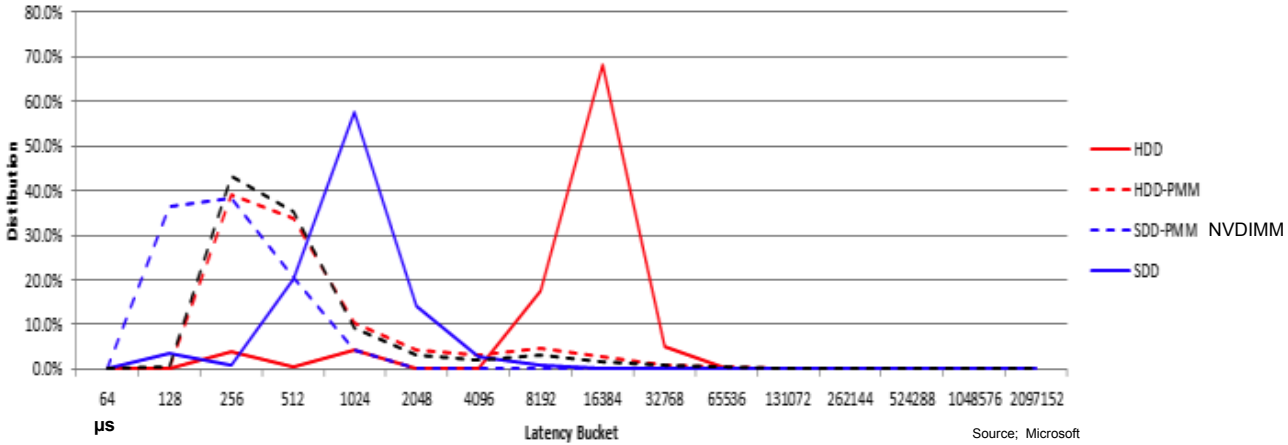
- Windows Server 2016 supports DDR4 NVDIMM-N
- Block Mode
  - No code change, fast I/O device (4K sectors)
  - Still have software overhead of I/O path
- Direct Access
  - Achieve full performance potential of NVDIMM using memory-mapped files on Direct Access volumes (NTFS-DAX)
  - No I/O, no queueing, no async reads/writes

4K Random Write	Thread Count	IOPS	Latency (us)
NVDIMM-N (block)	1	187,302	5.01
NVDIMM-N (DAX)	1	1,667,788	0.52

Source; Microsoft

Application Benefits – Windows Example

- Tail of Log in SQL 2016
  - Writes updates to SQL log through persistent memory first
  - Uses memory instructions to issue log updates to persistent memory directly
  - Utilizes memory-mapped files on NTFS Direct Access (DAX) volume



Flash Memory Summit

August 7-10, 2017 Santa Clara Convention Center

SNIA CSI | CLOUD STORAGE

SNIA SSSI | SOLID STATE STORAGE

- Learn more about Persistent Memory and NVDIMMs at these sessions:
  - ***Preconference Seminar on Persistent Memory –*** Mon. Aug. 7
  - ***Forum R-21 & R-22 – Persistent Memory – Convergence of Storage & Memory***, Wed. Aug. 9
  - ***Forum R-31 – NVDIMMs: Powerful Persistent Memory Arrives in a Familiar Form*** Thurs. Aug. 10

After This Webcast

- Please rate this webcast. We value your feedback
- This webcast and a copy of the slides will be on the SNIA Cloud Storage website and available on-demand
  - http://www.snia.org/forum/csi/knowledge/webcasts
- A Q&A from this webcast, including answers to questions we couldn't get to today, will be on the SNIACloud blog
  - http://www.sniacloud.com/
- Follow us on Twitter @SNIACloud & @SNIASolidState

SNIA CSI | CLOUD STORAGE SNIA SSSI | SOLID STATE STORAGE

Thank You!