Managing the Next Generation Memory Subsystem

Scott Kirvan and Paul von Behren 9/22/2015

Agenda

- Memory technologies & management challenges
- Concepts and practices for managing next generation memory technologies
- Emerging management standards, open source code, documentation

Memory Technologies & Management USE Cases

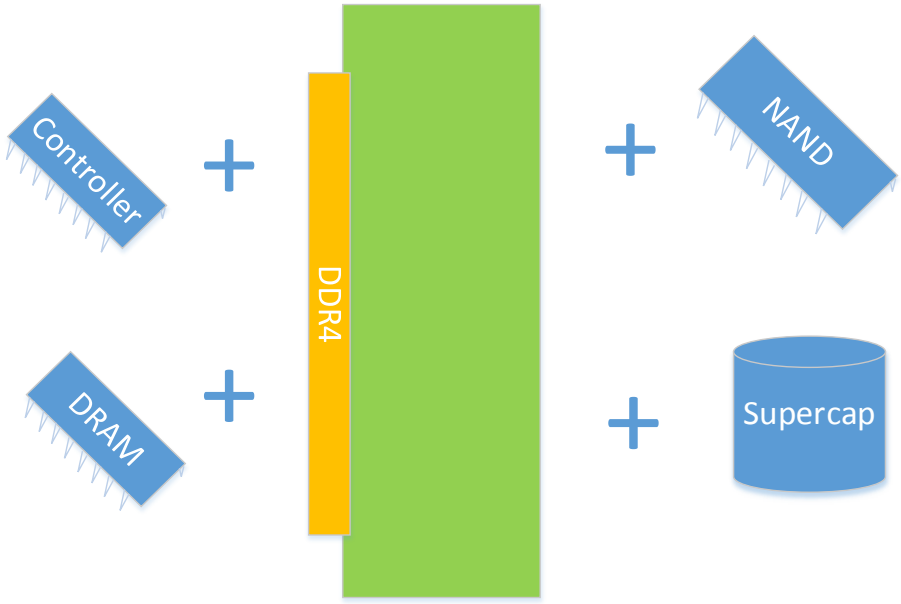
Today

Key use cases are around discovery and identification

Technology/Feature	Management Use Cases
Processor Cache	• How much do I have? • How much per core?
Memory controller, channels, slots DIMMs	• BIOS interleave configuration. • How much capacity do I have? • How many empty slots do I have?
Memory and channel clock speed	• What is the speed of my installed memory?
Redundancy: rank sparing, mirroring	• BIOS configuration • Mirror intact? Spare consumed?
SSDs cache for HDDs	• Set cache size, determine status • Select files/directories cached • Select pinned files

NVDIMM-N

- Non-volatile memory created by combining volatile & non-volatile media with a power source
  - DRAM speeds
  - Triggering mechanism like ADR to save volatile media contents to non-volatile media
  - Platform support, BIOS support
  - Interleaved separately from DRAM
  - BIOS uniquely identifies volatile/non-volatile memory regions.

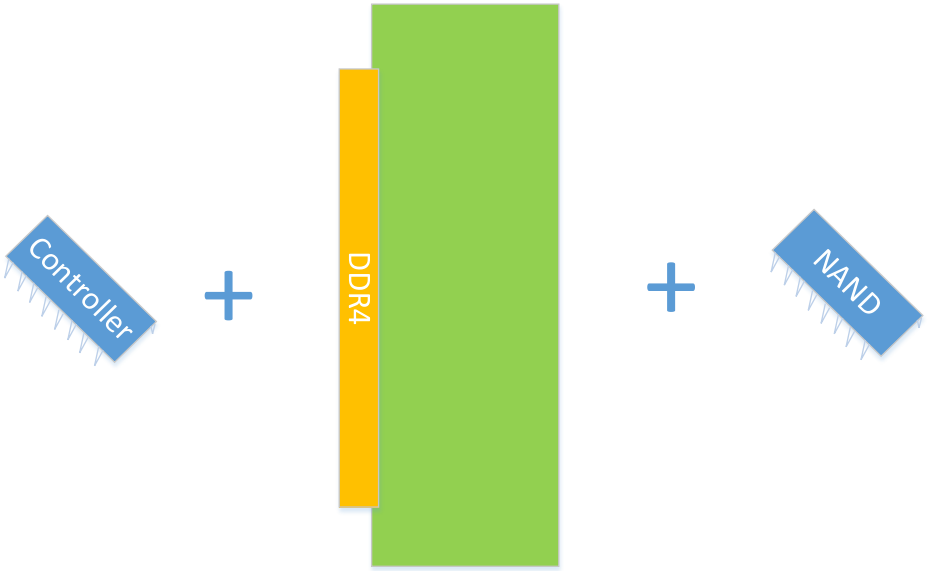


NVDIMM-N Management Use Cases

- Unique use cases are save/restore related
  - Trigger save, trigger restore
  - Monitor save/restore status
  - Monitor energy source, flash health, save readiness
- Standard NVDIMM use cases apply as well
  - Replace a failed (interleaved) DIMM
  - Update firmware on DIMM
  - Decommission, erase sensitive persistent content

NVDIMM-F

- Block device, DIMM form factor
  - Custom BIOS uniquely identifies block NVDIMM-F capacity in the system address map
  - Co-exists with DRAM
  - Some system DRAM may be used for caching.
  - Custom diver presents DIMMs as standard block device to the OS
  - Better than SSD performance.

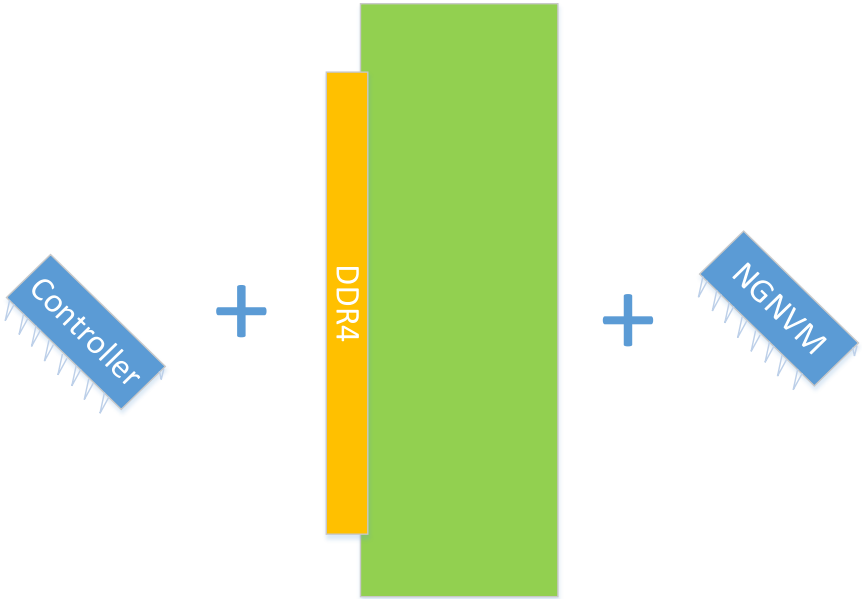


NVDIMM-F Management Use Cases

- Unique use cases are storage oriented
  - Monitor flash spare/wear & other drive like SMART metrics
  - Standard block device partitioning/formatting
  - Software RAID
- Standard NVDIMM use cases apply as well
  - Update firmware on DIMM
  - Decommission, erase sensitive persistent content
  - Backup

NVDIMM-P (proposed)

- Non-volatile memory fast enough for direct MC access or directly accessible DRAM & NAND
  - Near-DRAM speeds, directly accessible by MC
  - Very large capacities
  - May include multi-mode capable, byte and/or block addressable
  - BIOS uniquely identifies volatile/non-volatile memory regions.



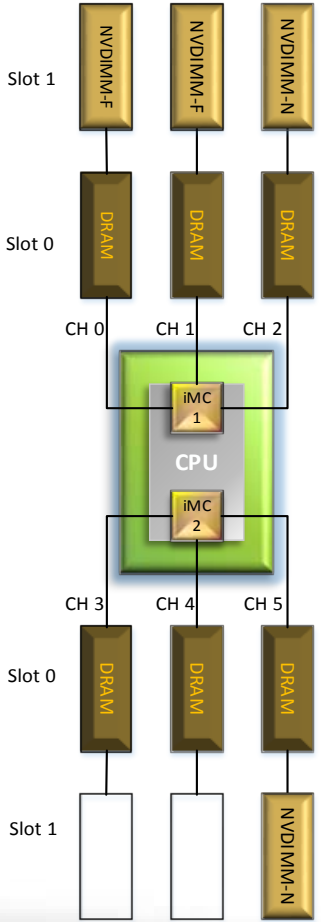
NVDIMM-P Management Use Cases

- Unique use cases are configuration oriented
  - Configure RAS and performance characteristics via BIOS
  - Configure block & direct access devices via driver
  - Optimize configuration for a given workload
- But many of the NVDIMM-N/F cases apply as well.
  - Replace a failed (interleaved) DIMM
  - Update firmware on DIMM
  - Decommission, erase sensitive persistent content

NEXT GENERATION MEMORY MANAGEMENT CONCEPTS

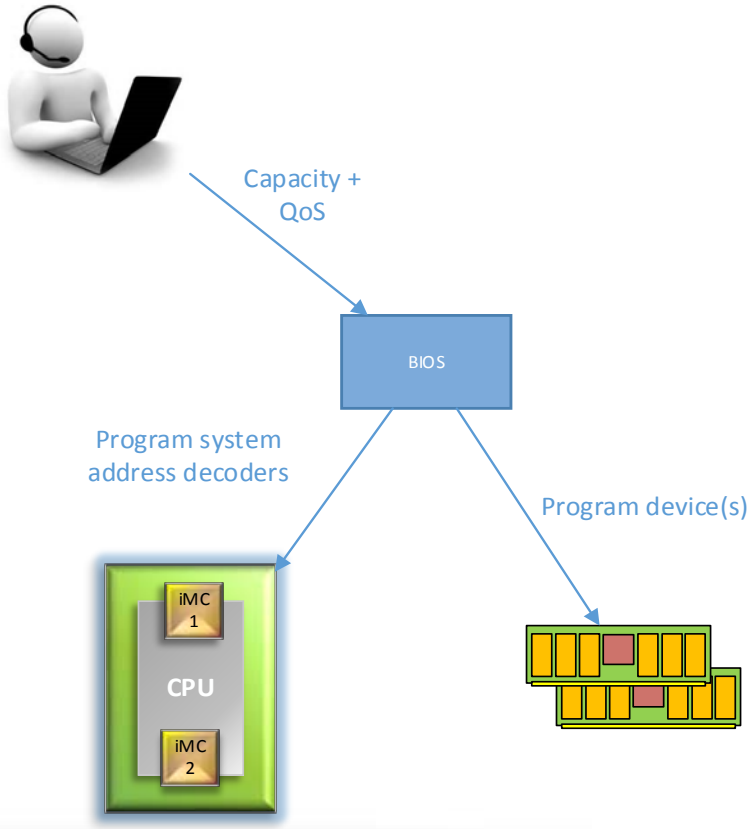
Memory is not a monolithic resource

- There can be multiple types of devices plugged into the memory bus
  - Devices may co-exist or require their own channel
  - They may work cooperatively or may be segregated
  - BIOS recognizes distinct device characteristics
    - MC programming, memory map reflect differences
    - Report uniquely in SMBIOS,ACPI (E820)
  - Management tools need to differentiate memory types and manage accordingly



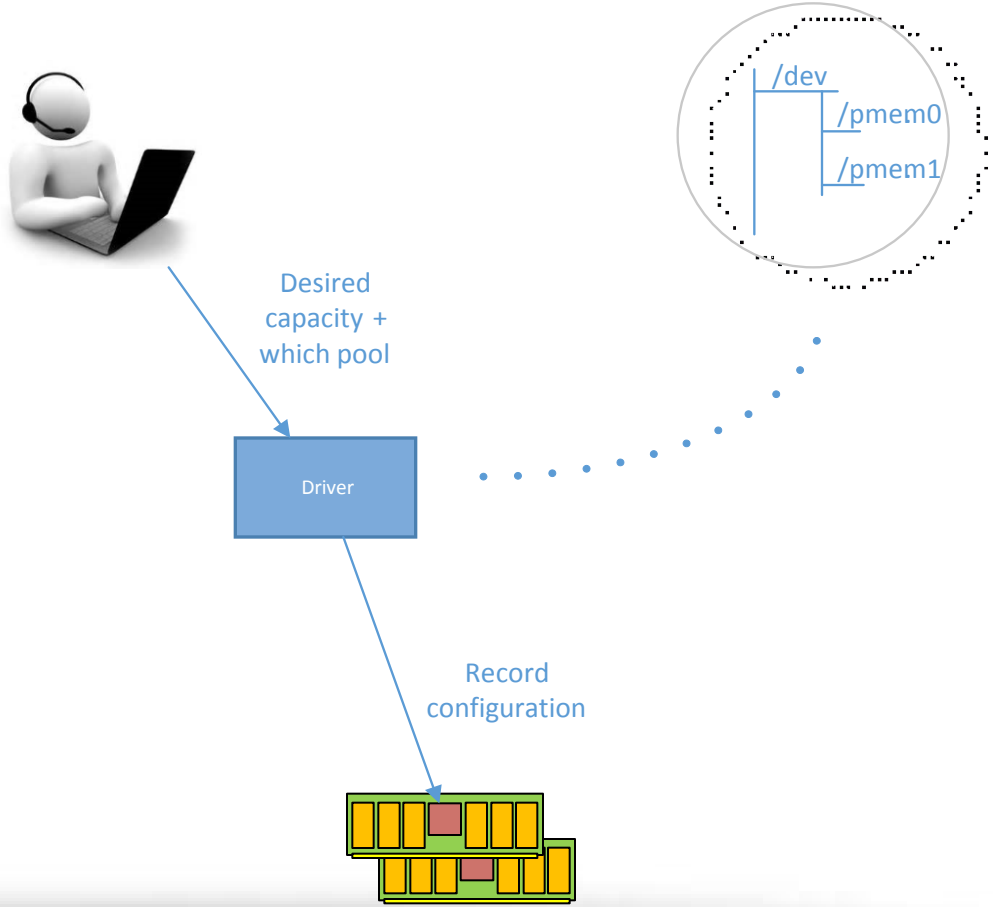
Configuration Required

- BIOS needs help optimizing memory configuration
  - Choices
    - Volatile vs. persistent
    - Interleaved, mirrored
    - Block access, byte access
    - Cooperative relationships
  - Constraints
    - Topology restrictions
    - OS support
    - Workload requirements
  - Management tools translate user requests



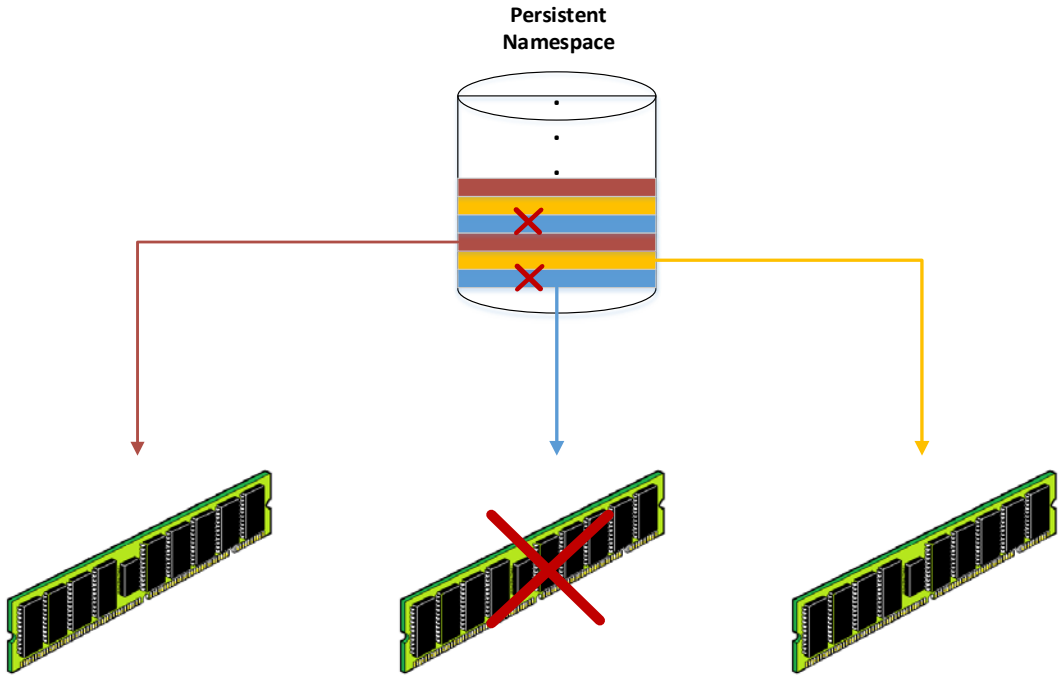
Persistent Handles

- Applications must be able to access same persistent memory regions across restarts
  - File systems + drivers support exactly this type of behavior for other resources
  - Must be able to allocate & label a region of persistent memory
  - Potentially allocate from a pool with particular QoS
  - Deallocate when done, modify if needed



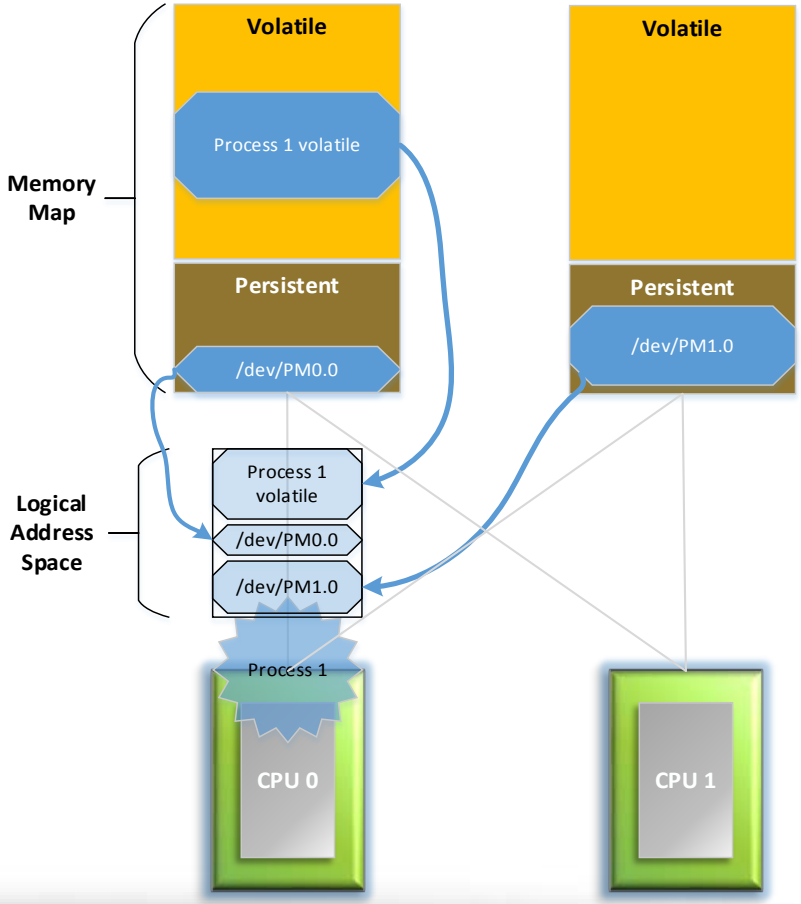
Data Management Needed

- Persistence creates reliability, serviceability & security concerns
  - Interleaving NVDIMMs complicates failure domains.
  - Replace failed DIMMs, rebuild logical storage entities and restore data from backup
  - Failed server –need to migrate NVDIMMs to a new server and locate logical storage entities
  - Repurposing NVDIMMs –must be able to securely erase data



Optimization is Hard

- NUMA just one example of configuration driven performance degradation
  - OS attempts to co-locate process and memory in multi-socket systems
  - May not be possible if persistent allocations are not well thought out.
  - Cross-socket access increases latency
  - Management tools need to expose socket relationships

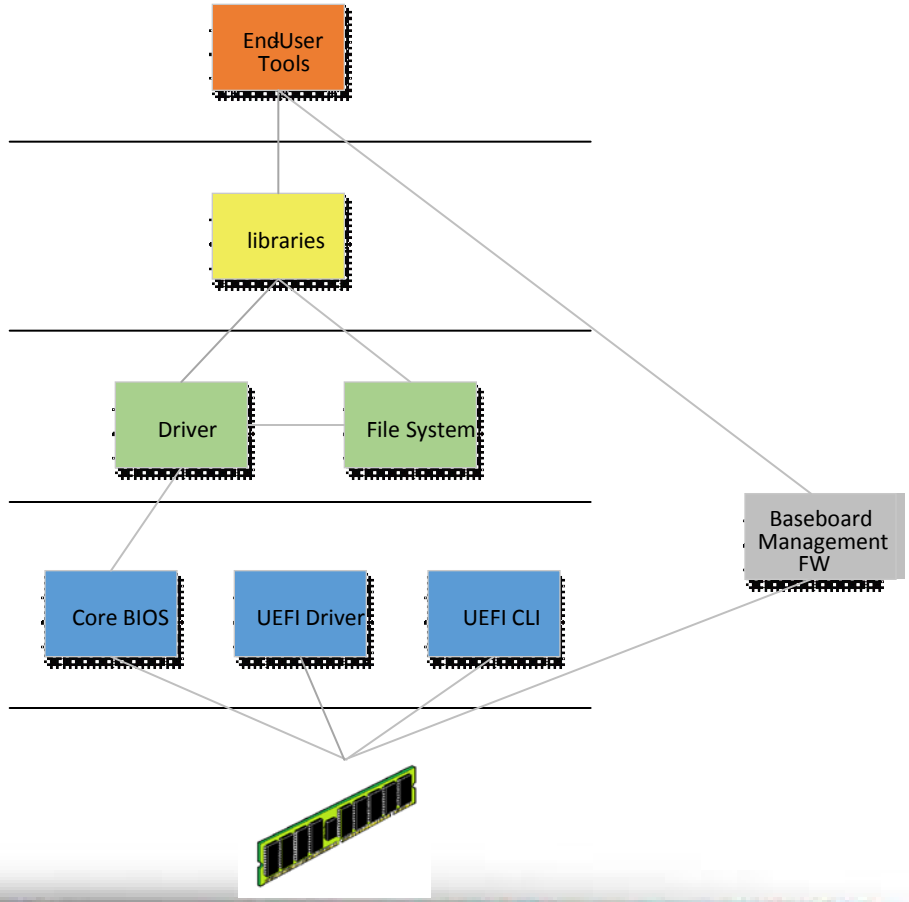


PUBLIC Resources

Standards, Documentation, Code

Software/Interface Ecosystem

- **End-user tools** –CLI, XML, etc.
- **Integration** –management libraries
- **Kernel** –sysfs, ioctl, etc.
- **Out-of-band** –IPMI, Redfish
- **BIOS** –ACPI, _DSM, UEFI driver protocols
- **Hardware** –registers, FW interfaces



BIOS

NVDIMM Firmware Interface Table (NFIT)

- BIOS tables that describe NVDIMM resources to OS
- Maps system physical address ranges to NVDIMMs including any interleaving schemes in use
- Describes QoS characteristics of the range (e.g. cacheable, write-protected, etc.)
- NVDIMM control surfaces (e.g. CSRs, block access mechanisms)
- http://www.uefi.org/sites/default/files/resources/ACPI_6.0.pdf

Device Specific Methods (_DSM)

- BIOS runtime interface to access NVDIMM functionality
- http://pmem.io/documents/NVDIMM_DSM_Interface_Example.pdf

Kernel

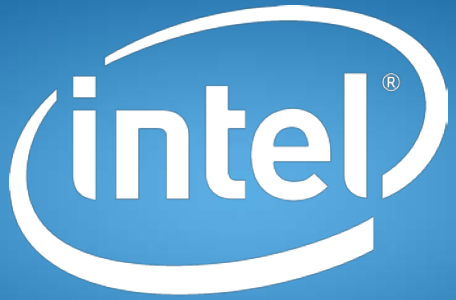
- Linux PMEM Driver, ndctl
  - Persistent RAM Driver --/sysfs management interface, persistent memory namespace management
    - Bulk of NVMDIMM extensions included in 4.2 kernel (kernel.org)
    - https://github.com/01org/prd
      - Intel repo with upstream kernel and emerging NVDIMM-related patches
    - http://pmem.io/documents/NVDIMM_Namespace_Spec.pdf
    - http://pmem.io/documents/NVDIMM_Driver_Writers_Guide.pdf
  - NDCTL –low level Linux only library for accessing NVDIMM management features.
    - https://github.com/pmem/ndctl

Emerging general-purpose NVDIMM management model

- DMTF DSP1071 – CIM static model for memory resource discovery.
  - http://www.dmtf.org/sites/default/files/standards/documents/DSP1071_1.0.0a.pdf
- SNIA Memory Configuration, Persistent Configuration – CIM models for creating the system address map and for allocating and labeling persistent memory regions.
  - In 2016, look for SMI-S 1.7.0 here:
    - http://www.snia.org/tech_activities/standards/curr_standards/smi
  - SNIA SMI TSG members, look for 1.7.0 Rev 3 or 4, Host Book here:
    - https://members.snia.org/members/smis/
  - Older version, but publicly available
    - http://www.snia.org/sites/default/files/SmisMemoryProfiles_v1.7r2.pdf

Stay Tuned!

- Additional documentation, reference/open source code, standards related to NVDIMMs are in the works.


intel
experience
what's inside™