NVMe SSDs with Persistent Memory Regions

Chander Chadha
Lead Technical Product Management, Enterprise & Datacenter SSD

Toshiba Memory America, Inc.

Agenda

- ❑ Why Persistent Memory is needed
- ❑ Key attributes of Persistent Memory
- ❑ Concept of NVM Express® (NVMe®) SSD with Persistent Memory
- ❑ PMR SSD mode of operation
- ❑ Key Benefits with PMR SSD
- ❑ Use Cases
- ❑ Next Steps

Why Persistent Memory is needed

- Log for software RAID & erasure coding systems
- Commit log device for NOSQL databases as well as Relational (MySQL, etc.) databases
- Journal for file systems
- Buffer for write-coalescing in caching systems
- Metadata
- Staging for de-dupe, compression, etc.
- NVMeoF™ RDMA transactions
- Utilized for In Memory Applications acceleration
  - Cassandra™, MongoDB®, STORM™, KAFKA™, SPARK™ …

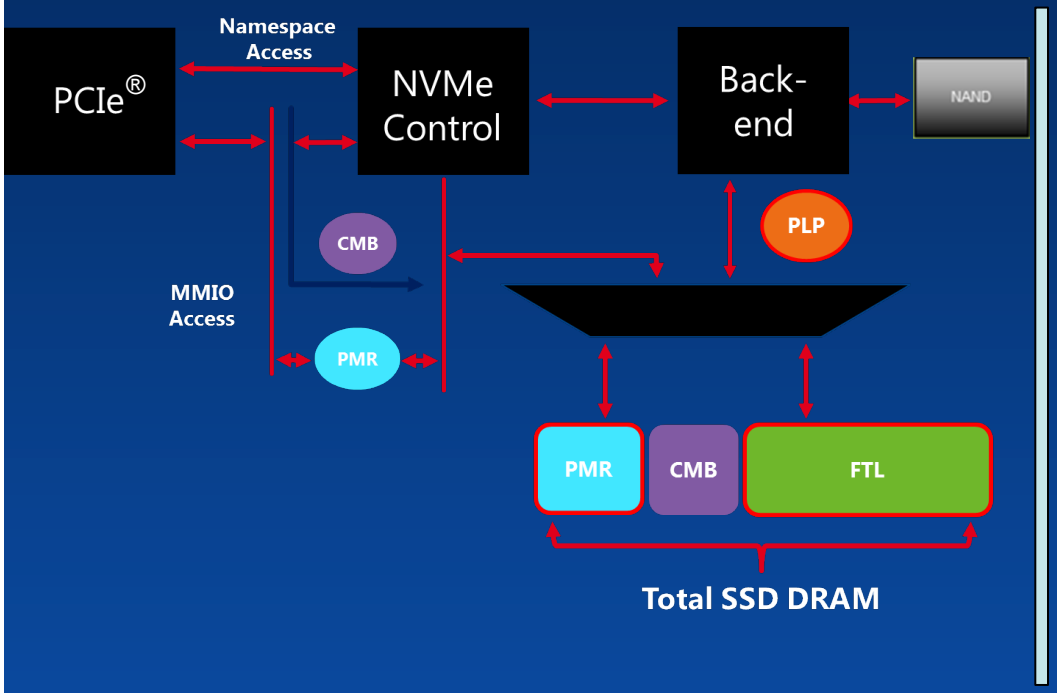
Key Attributes of Persistent Memory

Key Attributes

- ❑ Data Power Loss Protected
- ❑ Low Latency
- ❑ High Endurance
- ❑ Byte Addressable through CPU Load/Store Memory Instructions
- ❑ Block Addressable through software changes
- ❑ Today Served by
  - Battery backed DIMM's
  - NVDIMM's with Flash Storage
  - ST-MRAM & 3DXP

Wikipedia Definition…
In computer science persistent memory is any method or apparatus for efficiently storing data structures such that they can continue to be accessed using memory instructions or memory APIs even after the end of the process that created or last modified them.[1]

NVMe SSD with PMR : Concept



Key pieces for making PMR SSD:

- ❑ NVMe Enterprise SSD
- ❑ Additional DRAM for PMR function
- ❑ Persistent (PFAIL) Data path
- ❑ PMR configurability

Single Device offering for both block storage and PMR (byte) needs

PMR Mode of Operation

- Memory Mapped PMR after enumeration
  - Driver reads capability register and allocates Persistent Memory to Host (application)
  - DMA access from other PCIe EPs in the system (Peer-2-Peer)
- Accessibility through PCIe bus
  - MMIO Mode for Byte Access
- Writes and Reads Transactions:
  - Writes are “posted writes” based on PCIe “no ACK”
  - Reads are end to end from PMR to Host CPU
- In case of power loss, PMR Data gets saved to Flash
- PMR Data gets restored from Flash on next power up

Key benefits of SSD-based PMR

- Single Device with Persistent Byte Memory and Block storage
- Saves DIMM slots
- Dual port accessibility for higher reliability
- Aggregation of PMR's from multiple drives
- Provides persistent memory away from the CPU DDR bus
- Provides persistent memory in a CPU agnostic fashion without requiring ADR
- Robust and mature PCIe interface
  - Standard platform
  - Solid debug platform
  - Tools, analyzer fully available

Thoughts on Next Steps….

Next steps …

- Effort to standardize PMR
  - Registers definitions for PMR settings - Done
  - Get/Set Features for PMR configuration
  - PMR as Namespace unit for security (Lock/Unlock)
  - Data units boundaries for moving data between PMR and Flash
- Programming Model API for accessing PMR

PMR SSD POC Test Results

Flash Memory Summit

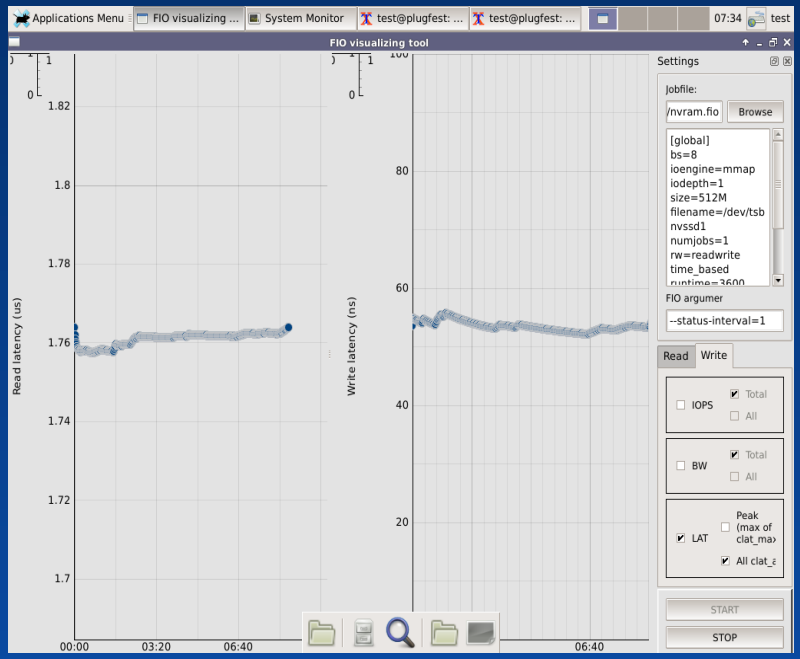
Test Setup

- Host: Ubuntu 14.04 LTS (Kernel v 3.14.14)
- System: Supermicro X9DRX (Intel Xeon 2.6 GHz, 8 cores)
- Benchmarking Tool: FIO v2.1.3
- GUI tools: ksysguard for Bandwidth/IOPS, FIO visualizer for latency
- Custom Driver: TSBNVSSD

Toshiba PMR SSD POC Drive
User Capacity :2TB, PMR :1GB
Latency

Operation	Block-Size	Jobs	Total QDepth	Latency
seq-write	8 byte	1	1	60 ns
seq-read	8 byte	1	1	1.75 us

Latency Chart



Disclaimers & Notes

Definition of capacity: Toshiba Memory Corporation defines a gigabyte (GB) as 1,000,000,000 bytes. A computer operating system, however, reports storage capacity using powers of 2 for the definition of 1GB = 2^30 bytes = 1,073,741,824 bytes and therefore shows less storage capacity. Available storage capacity (including examples of various media files) will vary based on file size, formatting, settings, software and operating system, such as Microsoft Operating System and/or pre-installed software applications, or media content. Actual formatted capacity may vary.

NVM Express, NVMe, NVMe-oF are trademarks of NVM Express, Inc.
Cassandra, Storm, Kafka, and Spark are trademarks of Apache Software Foundation.
PCIe is a registered trademark of PCI-SIG.
All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Information in this presentation, including product pricing and specifications, content of services, and contact information is current and believed to be accurate on the date of the publication, but is subject to change without prior notice. Technical and application information contained here is subject to the most recent applicable Toshiba product specifications.

As with any test, the results and outcomes herein should not be interpreted as a guarantee or warranty of similar results. Results may vary, depending on the circumstances and conditions.

Thank You

Contact Info: Chander.Chadha@taec.toshiba.com

Backup

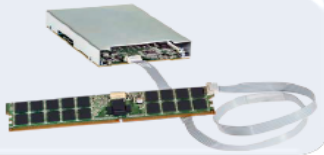
Evolution of Persistent Memory

Emergence of Persistent Memory Options (*source SNIA*)

NVDIMMS - JEDEC TAXONOMY

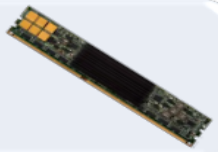
NVDIMM-N
Standardized

- 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 for backup
- DIMM interface (HW & SW) defined by JEDEC



NVDIMM-F
Vendor Specific

- Memory mapped Flash. DRAM is not system mapped.
- Access Method -> block-oriented access to NAND through a shared command buffer (i.e. a mounted drive)
- Capacity = NAND (100's GB-1's TB)
- Latency = NAND (10's of microseconds)



NVDIMM-P
Proposals in progress

- Memory-mapped Flash and memory-mapped DRAM
- Two access mechanisms: persistent DRAM (–N) and block-oriented drive access (–F)
- Capacity = NVM (100's GB-1's TB)
- Latency = NVM (100's of nanoseconds)