SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2015

Storage Class Memory Support in the Windows Operating System

Neal Christiansen

Principal Development Lead
Microsoft

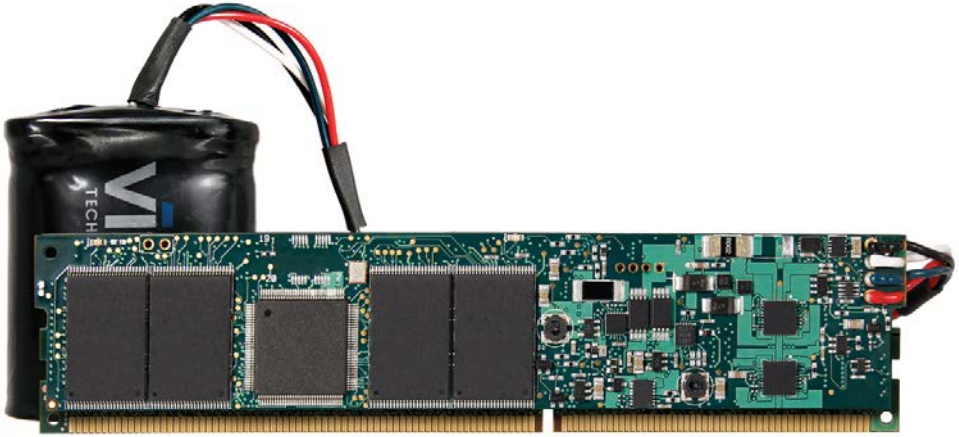
nealch@microsoft.com

What is “Storage Class Memory”?

- **Paradigm Shift:** A non-volatile storage medium with RAM-like performance characteristics - Low latency/high bandwidth.
- Resides on the memory bus
  - Underlying technology does not matter
- Several different terms in use:
  - Storage Class Memory (SCM)
  - Direct Access Storage (DAS)
  - Byte Addressable Storage (BAS)
  - Persistent Memory (PM)
  - Non-Volatile Memory (NVM)

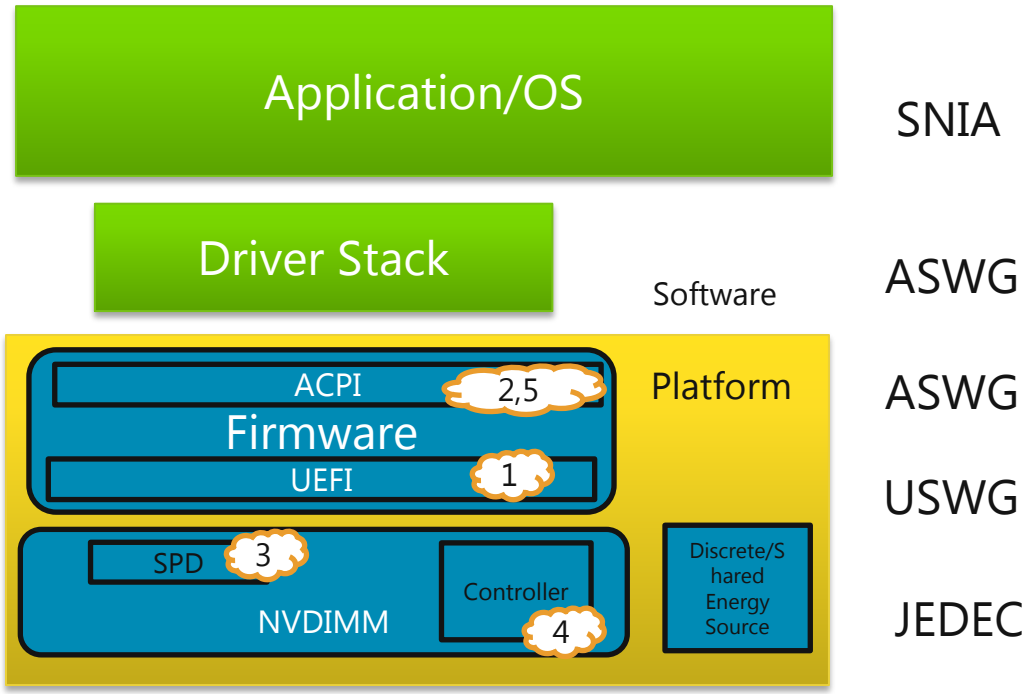
NVDIMM-N

- NVDIMM-N is an example of this new type of storage
  - Has DRAM and Flash on DIMM module
  - DRAM contents saved to Flash on power fail
  - Requires per module or central backup power source
  - Requires Specific platform support
- Available today



Source: Viking Technology

Standardization



	Purpose	Body
1	UEFI Persistent Memory Type (UEFI 2.5)	UEFI Standards Working Group
2	ACPI Persistent Memory Type, NFIT, ACPI Namespace Object for NVDIMM devices (ACPI 6.0)	ACPI Standards Working Group
3	Identification for NVDIMM devices (DDR4 SPD Document Release 3)	JEDEC SPD TG
4	Standard interface to NVDIMM-N (Byte-Addressable Energy-Backed Interface Specification)	JEDEC TG456_2
5	NVDIMM-N Health and management data (Microsoft _DSM for JEDEC Byte-Addressable Energy-Backed Interface NVDIMMs)	Microsoft

SCM Storage Drivers

- New driver model
  - SCM Bus Driver
    - Enumerates the physical and logical SCM devices on the system
  - SCM Disk Drivers
    - Driver for logical SCM devices
    - Storage abstraction layer to rest of the OS
    - Hardware-specific
      - Can support both standards or vendor-specific driver
- New interfaces to expose byte addressable storage functionality and to support SCM management

Windows Goals for Storage Class Memory

1. Support zero-copy access to persistent memory
2. Most existing user-mode applications will run without modification
3. Provide an option to support 100% backward compatibility
   - Introduces new types of failure modes
4. Make available sector granular failure modes for application compatibility

BTT – Block Translation Table

- Algorithm created by Intel
- Provides sector level atomicity of writes
  - No sub-sector torn writes
  - On power loss either see contents of old sector or new sector
  - Maintains compatibility with existing applications that have built-in assumptions around storage failure patterns
- SCM Storage Drivers will support BTT
  - May have an option to disable

File Systems and Storage Class Memory

- **SCM is a disruptive technology**

- Customers want the fastest performance
  - System software is in the way!
- Customers want application compatibility
- Can be conflicting goals

A Storage Class Memory Aware File Systems for Windows

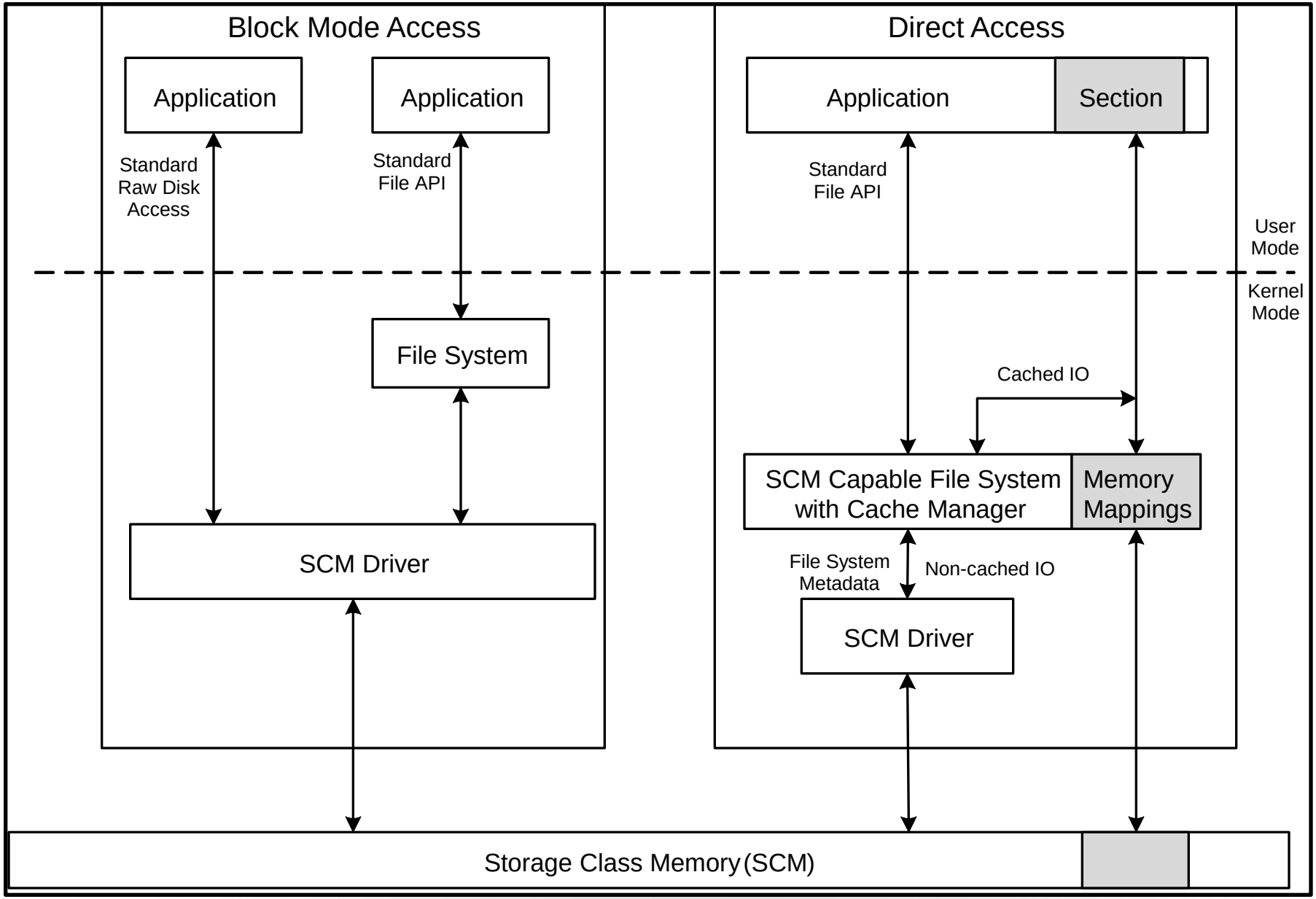
- A volume can be in one of the following modes:
  - Block mode
  - DAS mode
- The mode is chosen at format time

Block Mode Volumes

- Maintains existing storage semantics
  - All IO operations traverse the storage stack to the SCM storage driver
    - Shorter path length through storage stack
- Fully compatible with existing applications
- Supported by all Windows file systems
- Works with existing file system and storage filters

DAS Mode Volumes

- Introduces new storage concepts
  - Memory mapped files provide applications with zero copy access to SCM
    - Maximizes performance
- Some existing functionality is lost
- DAS mode will be supported by both the NTFS and ReFS file systems



Memory Mapped IO in DAS mode

- On DAS formatted volumes creation of a memory mapped section will map directly to SCM hardware
  - No change to existing memory mapping APIs
- When an application creates a memory mapped section:
  - The memory manager asks the File System if the section should be created in DAS mode
  - The FS returns YES when:
    - The volume resides on SCM hardware
    - The volume has been formatted for DAS mode

Memory Mapped IO in DAS mode

- When a DAS mode section is requested
  - MM asks the file system for the physical memory ranges for a given offset and length of the file
  - The file system translates the given file offset and length into one or more volume relative extents (sector offset and length)
  - The file system then asks the storage stack to translate these extents into physical memory ranges
  - MM then updates is paging tables for the section which maps directly to the persistent storage

Memory Mapped IO in DAS mode

- This is true **zero-copy** access to storage
  - An application has direct access to persistent memory
  - BTT is not used
    - An application may see new failure patterns on power loss or system crash
- Important → No paging reads or paging writes will be generated

Cached IO in DAS mode

- When cached IO is requested for a file on a DAS enabled volume the Windows cache manager will create a DAS enabled cache map
- The cache manager will then copy directly between the user's buffer and SCM
  - Cached IO has **one-copy** access to storage

Cached IO in DAS mode

- Cached IO is coherent with memory mapped IO
- BTT is not used
  - An application may see new failure patterns on power loss or system crash
- As in the Memory Mapped IO case, no paging reads or paging writes will be generated

Non-cached IO in DAS Mode

- Will send IO operations down the storage stack to the SCM storage driver
  - Will use BTT
  - Maintains existing storage semantics for application compatibility

File System Metadata in DAS Mode

- File system metadata files will operate in block mode
  - Meaning paging reads/writes will be generated for all FS metadata operations
  - Needed to maintain existing ordered write guarantees for write-ahead logging
- One or more metadata files may switch to DAS mode access in the future

Impacts to File System Functionality in DAS Mode

- Direct access to storage by applications eliminates the traditional hook points that file systems use to implement various features
- Following is functionality that can not be supported on DAS enabled volumes:
  - No NTFS encryption support
  - No NTFS compression support
  - No NTFS TxF support
  - No ReFS integrity stream support
  - No ReFS cluster band support
  - No ReFS block cloning support
  - No volume encryption support via bitlocker
    - It is expected that SCM vendors will provide hardware encryption in the future
  - No volume snapshot support via volsnap
  - No mirrored or parity storage support via spaces or dynamic volumes

Impacts to File System Functionality in DAS Mode

- Functionality that is not currently supported but can be supported in the future:
  - Sparse files
- For writeable memory mapped files the file system no longer knows when the file has been modified
  - The following file system features are now updated at the time the file is memory mapped
    - Updating the file's modification time
    - Marking the file as modified in the USN Journal
    - Signaling directory change notification

File System Filters

- File system filters are drivers that layer above the file system and can interact with all operations as they come into and out of the file system
  - Filters have the ability to augment file system functionality
  - Example classes of filters: Anti-virus, replication, HSM, encryption, compression, quota, activity monitor, etc.

File System Filters in DAS Mode

To minimize compatibility issues:

- No existing filter will receive notification when a DAS volume is mounted
- At filter registration time filters will indicate via a new registration flag if they understand DAS mode semantics

Compatibility Issues with Filters in DAS Mode

- Data Transformation Filters
  - There is no opportunity for these filters (ex: encryption and compression) to do their work
- Anti-virus filters
  - Minimally impacted because scanning is performed on file open and close
  - Detecting when a file is modified will need to be updated
    - Watch for creation of writeable mapped sections

Intel NVML Library

- Open source library implemented by Intel
- Defines a set of application API's for directly manipulating files on SCM hardware
- Available for Linux today via GitHub
- Microsoft is working with Intel on a Windows port

Questions?