Advancements in Storage and File Systems in Windows 8.1

Andy Herron
Microsoft Corporation

Overview

- Who am I?
  - Developer in Windows Storage and File Systems group
    - Worked on Spaces Tiering for 8.1
- What are we covering?
  - Enhancements to local storage stack
    - Windows Client 8.1
    - Windows Server 2012 R2

Specifics of what we will cover

- Solid State Hybrid Drives
- NVMe Devices
- Storage Spaces Advancements
  - Storage Tiering
- SM-API Management Advancements
- File System Advancements
  - NTFS
  - REFS

Solid State Hybrid Drives

Solid State Hybrid Drive

- Support added for Windows Client 8.1
  - Currently not supported in Windows Server 2012 R2
- Also known as
  - “SSHD”
  - “Hybrid Drive”
  - “Hybrid Disk”

What is a Solid State Hybrid Drive?

- Spinning magnetic disk + nonvolatile flash in **a single physical device**
- The flash typically acts as a nonvolatile **cache** for data also available on the disk
  - Total storage capacity of the device is max(`flash size`, `disk size`)

Image Source: www.Seagate.com

“SSHD” Value Proposition

- High level goal:
  - SSD-like performance + HHD-like price and capacity
  - Users want large storage capacity for music, videos, photos, etc. that would be prohibitively expensive for OEMs to meet with flash only storage.
- SSHDs can dramatically improve performance for low to mid-range systems with reasonable cost increase

New Host-Hinted SSHD devices

- Standardized interface through SATA-IO
  - "TP_042v14_SATA31_Hybrid Information"
- Host-hinted SSHDs rely on the OS to provide "hints" as to which LBAs are important / high value.
- A hint is a **suggestion**, not an order.
  - Device can move an LBA into flash immediately, lazily, or not at all.

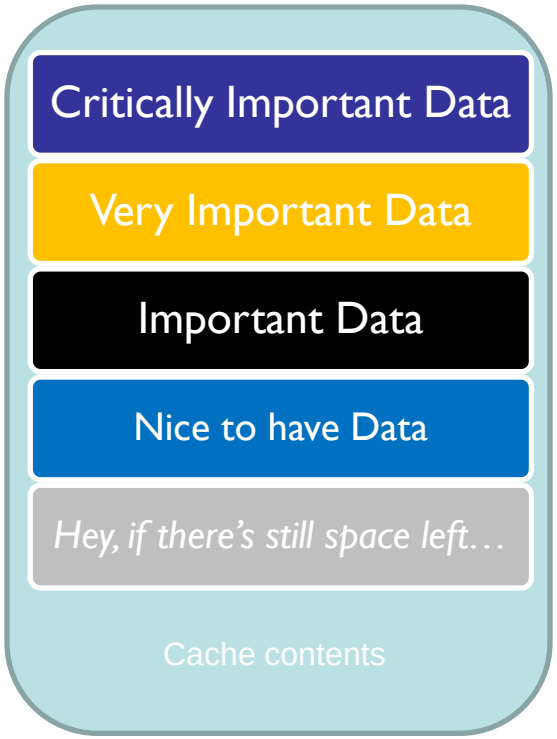
New Host-Hinted SSHD devices

- Cache remains a **black box** to the host. **There is no way to query its contents, or determine if a particular LBA is in the flash**.
  - We can get some high-level info like how much cache space is used.

- Drives can use their own proprietary self-pinning logic **in addition to** host hints.

Hinting Priorities

- Host hinted SSHDs take a “hint priority”
- Indicates how valuable an LBA is with respect to others in the cache.
- If there is not enough cache space, the device evicts LBAs from the lowest non-empty priority first to make room for higher priority LBAs.

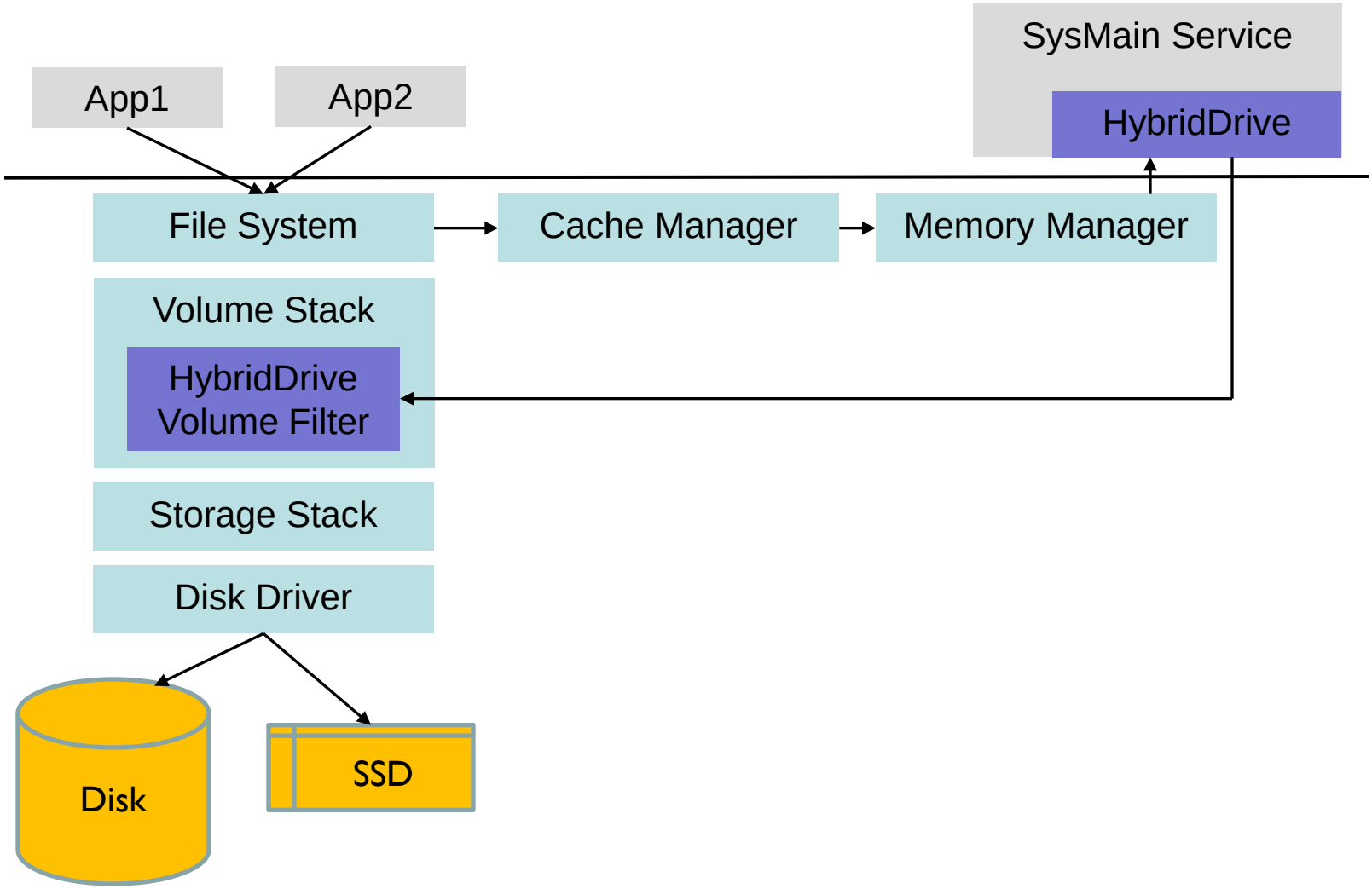


- Number of priority levels varies by device.
  - Windows 8.1 logo mandate at least 6 priority levels be exposed to the OS.

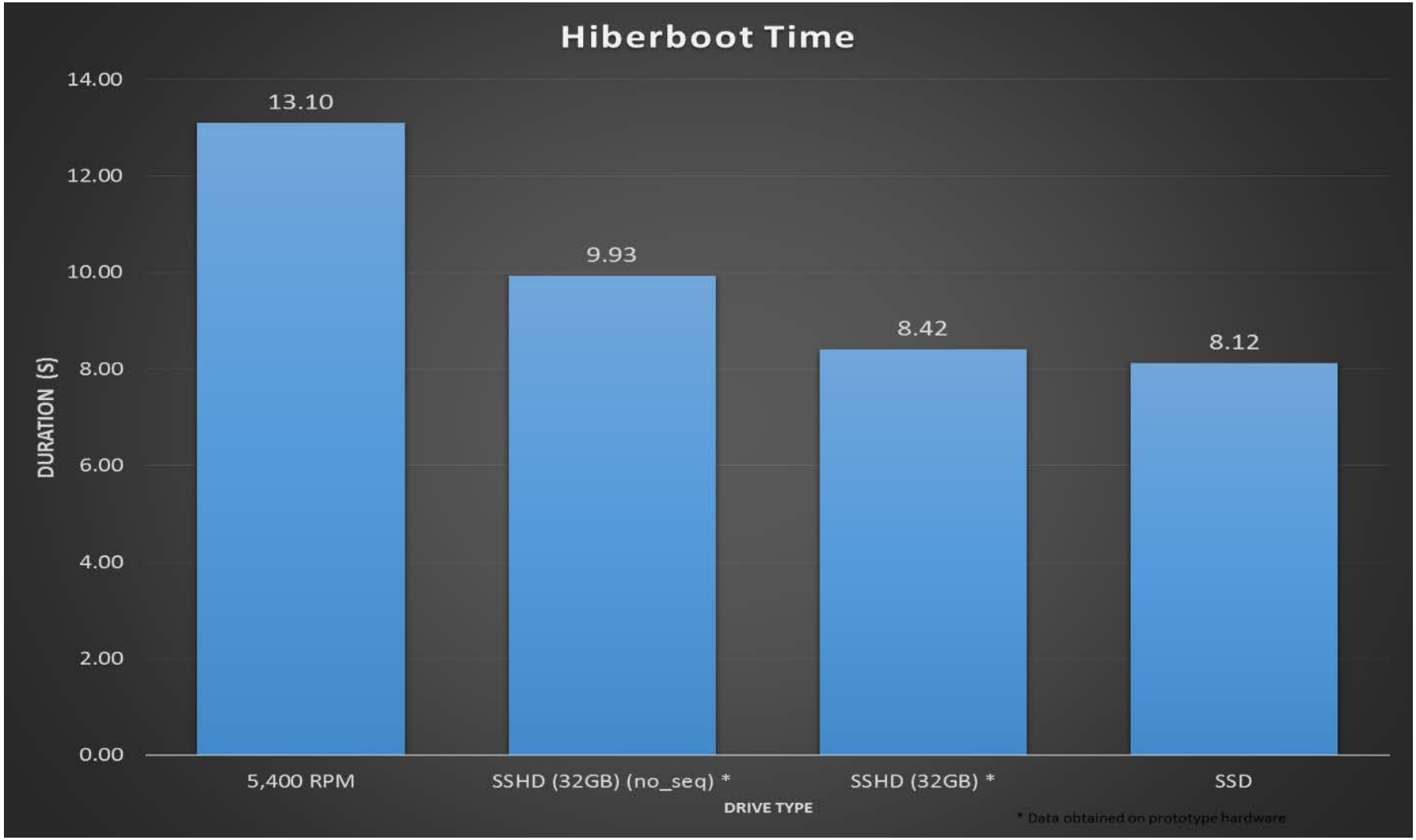
Windows Client 8.1 : Scenario-Focused Approach

- Other solutions look at recent access history
  - Cold data can be pushed out by hot data – but the cold data could still be important
- Window's SuperFetch has context on data accessed during important scenarios
  - Boot, Logon, Hibernate, Resume, and Standby
  - App Launch and Fast User Switch
- Scenarios are assigned to different priorities
  - Device will evict LBAs from less important scenarios before important ones.
  - Cold (but important) data can stay in the cache.

Hybrid Disk Support Architecture



SSHD Results in Windows 8.1



* Pre-Production Hybrid Hard Disk

NVMe Devices

NVMe in Windows

- The Protocol
  - Standardized PCIe Storage

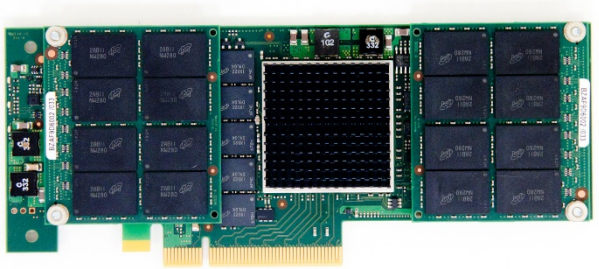
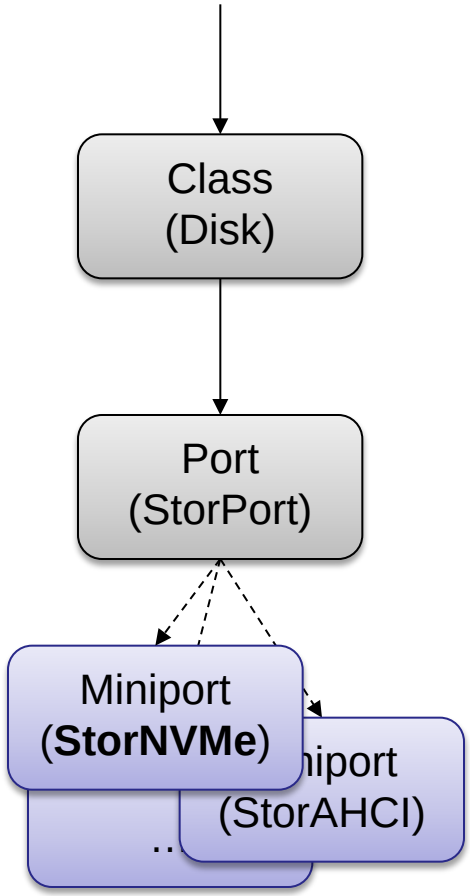


Image Source: www.Micron.com

- Windows OS Support
  - Windows Inbox Driver (StorNVMe.sys)
    - Windows Server 2012 R2 (high-density/performance)
    - Windows 8.1 (small form factors)
  - Stable Base Driver for all NVMe Compliant Devices

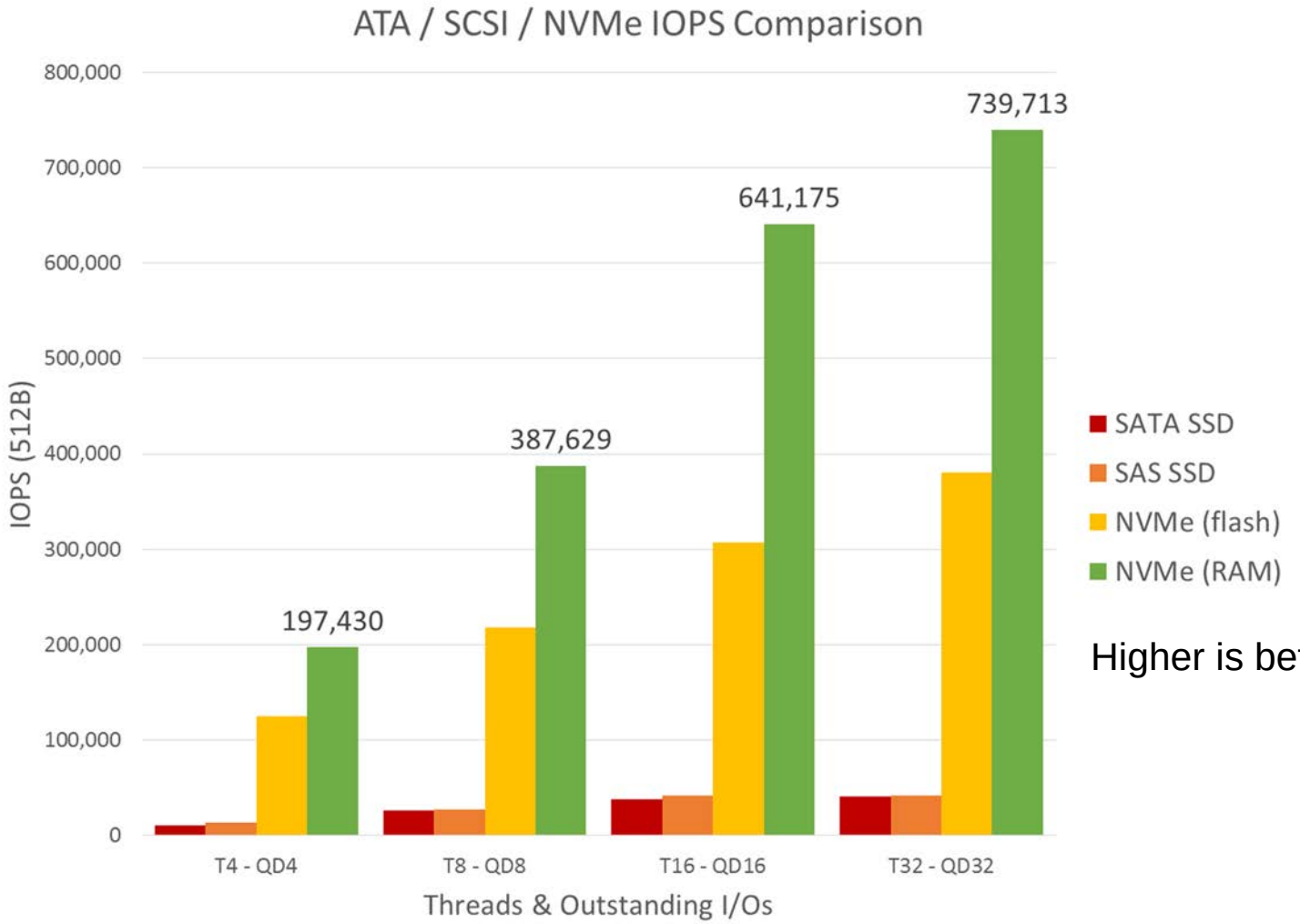
NVMe in Windows Storage Stack

- The Storport Model
  - Reduced development cost
    - Offloads Basics: PnP, Power, Setup, Crash, Boot*
  - Mature / Hardened Model
  - Storport optimized for performance
  - RAM-backed NVMe device
    - > 1 million IOPS | < 20µs latencies



* For machines that have compatible UEFI boot environment

Windows Stack Performance

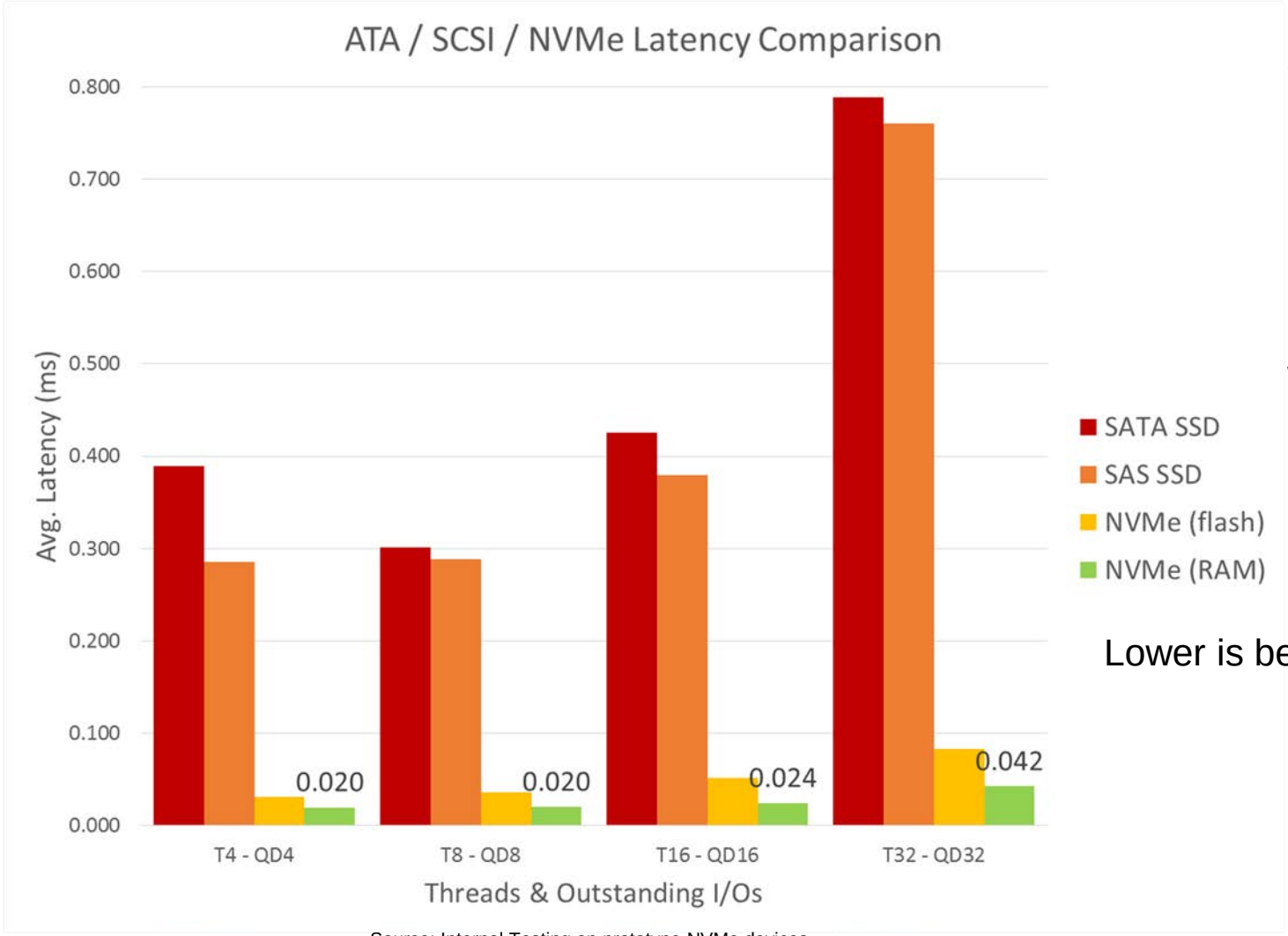


System:
Romley-based Server
2 Sockets
16 physical processors
32 hyper-threaded
Random Read Workload

Higher is better

Source: Internal Testing on prototype NVMe devices

Windows Stack Latency



System:
Romley-based Server
2 Sockets
16 physical processors
32 hyper-threaded
Random Read Workload

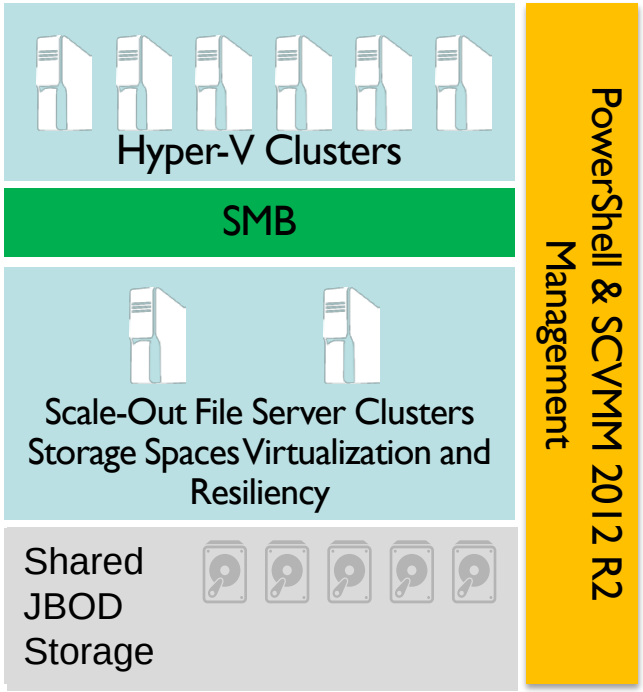
Lower is better

Source: Internal Testing on prototype NVMe devices

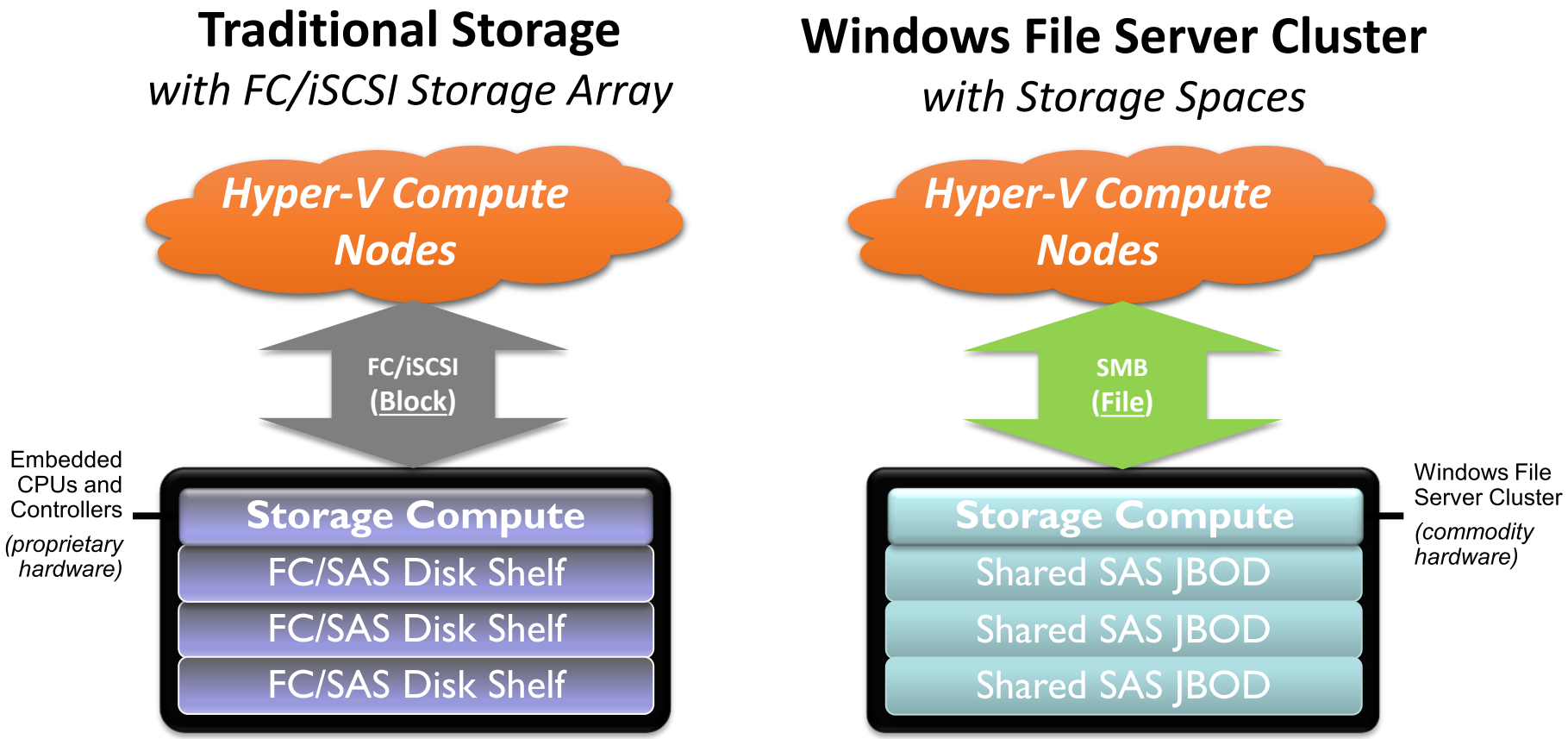
Storage Spaces Advancements

Infrastructure-as-a-Service Storage Vision

- IAAS : Cloud hosted VM for generic workloads
- Dramatically lowering the costs and effort of delivering IaaS storage services
- Disaggregated compute and storage
  - Independent manage and scale at each layer
- Industry standard servers, networking and storage
  - Inexpensive networks
  - Inexpensive shared JBOD storage



Storage Spaces Deployment Model



Storage Spaces Advancements

- New Resiliency Schemes
  - Windows Server 2012 had simple, mirrored, and single parity
  - Windows Server 2012 R2 now adds dual parity with erasure coding
    - Requires at least seven disks in pool for dual parity
    - Dual parity guarantees resiliency in the case of two concurrent device failures
    - Uses a write journal for crash consistency
      - Battery backup of storage not required

Storage Spaces Advancements

- Parity Spaces are now supported in clustered scenarios.
- Space allocation for Parity Spaces are now Enclosure Aware.
  - Distributed across enclosures such that it is resilient to failure of one full enclosure and another disk in a different enclosure.

Storage Spaces Advancements

- Fast (Parallel) Rebuild
  - When a physical disk fails, Storage Spaces regenerates the data on the replaced physical disk in parallel.
  - In the past, priority was placed on not interfering with IOs rather than completing the rebuild quickly.
  - Configurable on a per disk basis and defaults to fast rebuild enabled.

Storage Spaces Advancements

- Persistent Write Back Cache
  - Available for simple and mirrored spaces
    - Parity gets this through journal on SSD
  - Uses SSDs to log small writes and then destage over time. Bypassed for large writes.
  - Configurable per space.
  - Requires same degree of resiliency as space it is caching data for.

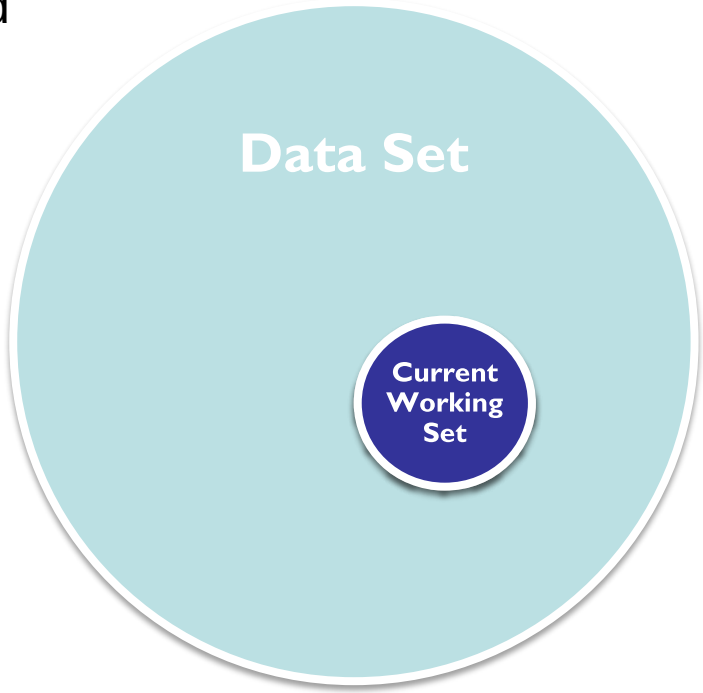
Tiered Storage on Spaces

Tiered Storage – High Level

- Enhancement to Windows Storage Spaces
- Capacity of large pool of disks with benefits of flash performance
- Flash drives (SSDs) and hard drives (HDDs) on same volume
- Write back cache (WBC) on flash is used to buffer writes to volume
  - WBC only caches blocks going to disk regions, not flash regions

Tiered Storage

- Match Workload Characteristics to Drives
- Common Workload Characteristics
  - Large data set, and majority of data is cold
  - Minority of data is in active use, and is hot
    - *The hot data is the "working set"*
  - Working set changes over time
- Common Drive Characteristics
  - Hard Disk Drives
    - *Capacity* Optimized
  - Solid State Drives
    - *Performance* Optimized



Tiered Storage

- Traditionally done as remap layer of LBA below file system
  - Pros
    - Transparent to file system
    - Faster media can be shared across volumes
  - Cons
    - Yet another table lookup at IO time
    - Mapping must be hardened against loss at flush or FUA write time
    - Underlying remap layer not aware when a file is moved to new LBA. Has no knowledge about what LBAs make up a particular file.

Tiered Storage – A new approach

- Allocate large blocks of different media types into explicit regions of logical volume.
- Make configuration of tiers and explicit mappings for a volume known to the file system.
- Use file extent list in file system the authoritative mapping of what portions of what file go into which regions.
- Measure read/write heat at logical file offset level so that when file moves, heat moves too.

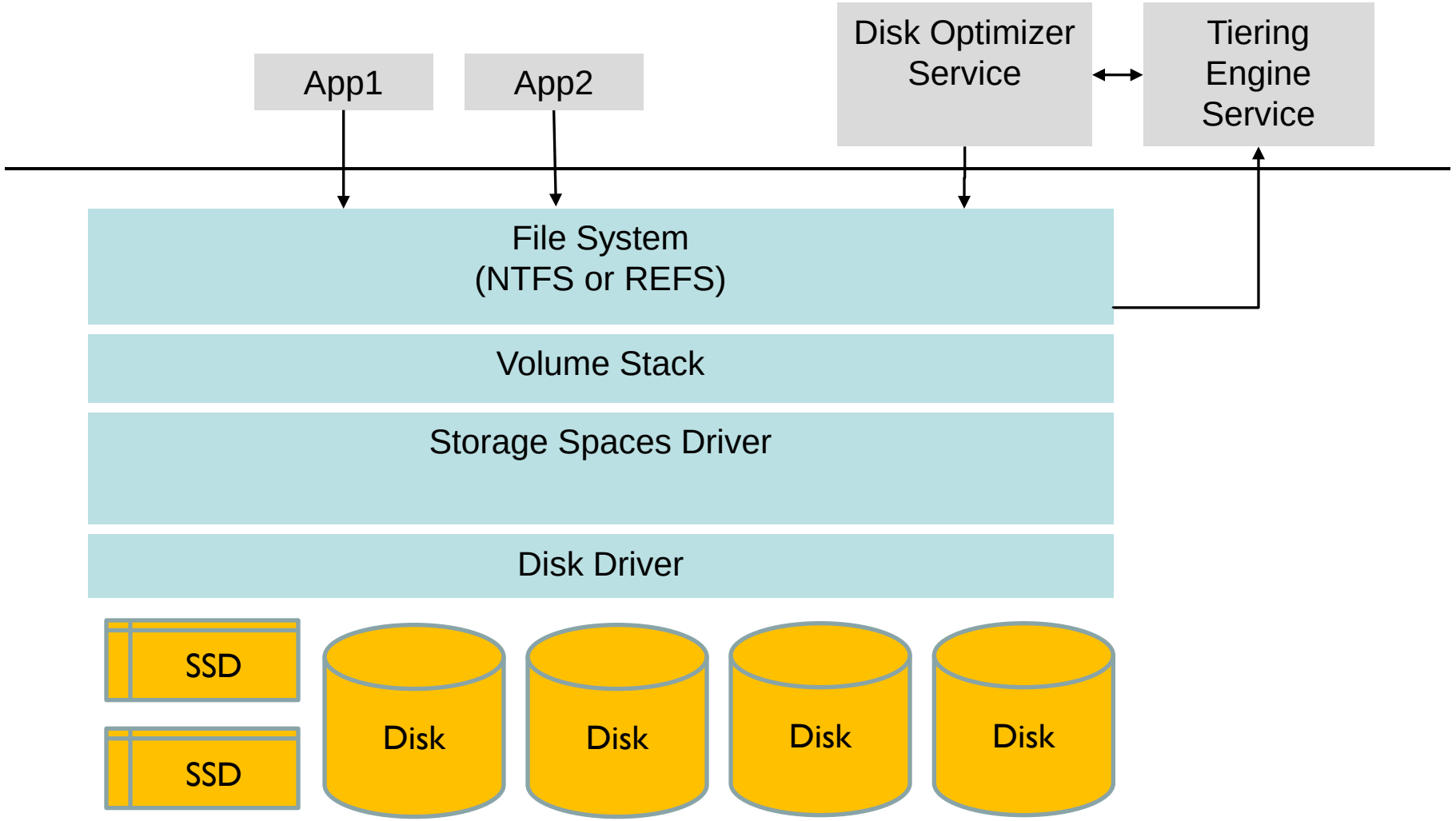
Tiered Storage

- Benefits to new approach vs Traditional Remap
  - No secondary lookup at IO time through remap layer.
  - No extent mapping to be hardened at write or flush.
  - Can provide an API that allows administrators to specify tier at the file level.
  - File systems can get initial placement correct.
  - When a file is moved to new LBA, current tier can be taken into account and maintained.

Tiered Storage

- File system is aware of different regions and can have different allocation policies for different tiers.
- Read/Write IO is measured at logical file offset layer so that when file is moved, heat follows the file.
  - Important for REFS integrity streams
- REFS during Allocate-On-Write can preserve tier during new allocations for same file or file range.

Tiered Volume Architecture



Tiered Storage

- Tiering Engine Service
  - Tracks IO workload on volume and moves data once a day by default.
  - Cold data existing on flash regions is moved to disk regions.
  - Hot data existing on disk regions is moved to flash regions.
  - Configurable how much data is moved and how often.
  - Defrag interface to file system is used to move data between tier regions.

Tiered Storage

- Tier creation and configuration is integrated into core Storage Management UI
- Individual files can have their storage tier explicitly set via PowerShell command
- Administrative operations are done from PowerShell or UI.
  - No administrative actions required to get good results

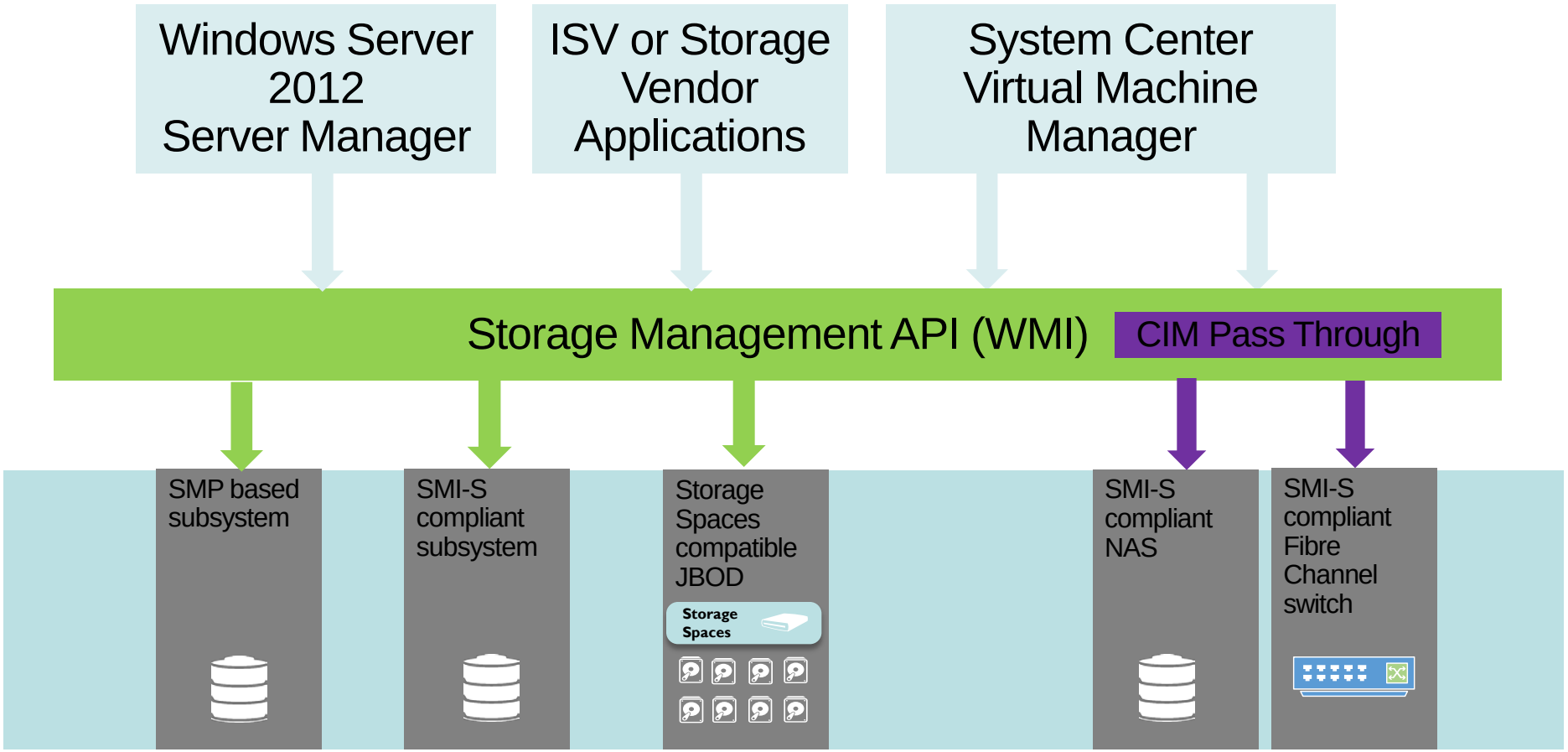
Tiering and Hybrid Disk Support

- Currently two distinct separate technologies
- Tiered Storage scenarios
  - Windows Server 2012 R2
  - Optimized for generic enterprise workload performance where the working set changes slowly over time.
- Hybrid Drive scenarios
  - Windows Client 8.1
  - Optimized for fast boot and application launch
  - Dramatically speeds up key scenarios in client

SM-API Management Advancements

SM-API Overview

Single standardized management interface to manage storage



SM-API Management

- Remote Spaces Management
  - Connect to a provider in cluster namespace or a standalone machine's local subsystem
  - Disabled connecting to local subsystem if node part of a cluster
  - Pool, Space, and Physical Disks can be remotely managed.
    - Virtual Disk, Volume, and Partition not yet supported remotely

SM-API Management

- Automatic Cluster Aggregation
  - Consistent view from any cluster node.
  - Management operations performed on subsystem basis
  - Automatic redirection of operations to owner node

SM-API Management

- New CreateVolume API automatically does following:
  - Creates a Storage Space
  - Initializes the disk
  - Creates a partition
  - Formats the volume
- Many performance improvements

NTFS & REFS Advancements

File System Advancements

- NTFS & REFS
  - Worry Free Thin Provisioning
    - A thin-provisioned volume no longer disappears if the underlying physical storage runs out of space to back an allocation.
    - An application now gets STATUS_DISK_FULL in this condition
  - API available to read copies of data exposed by Spaces
    - Allow for applications that implement their own checksum schemes
  - Lots of performance improvements

File System Enhancements

- REFS
  - Alternative Data Streams now supported
  - Self correcting on both mirror and parity Spaces volumes
    - File system metadata
    - User data with integrity enabled
  - More self healing
    - Automatically recovers from directory metadata corruption

- Questions?