

NVM Express: Unlock Your Solid State Drives Potential

Janene Ellefson

SSD Product Market Manager – PCIe
Micron Technology

Agenda/Schedule

- **AM Session – 8:30am – 11:20am**
 - Brief introduction to NVMe - Janene Ellefson (Micron)
 - NVMe Ecosystem Development - Amber Huffman (Intel)
 - Microsoft's perspective on NVMe – Tobias Klima (Microsoft)
 - NVMe Applications for Datacenter, Enterprise, and Client – Swapna Yasarapu (STEC)
 - Break
 - NVMe Conformance & Interoperability - David Woolf (UofH IOL)
 - SATA Express & NVMe Vision in End User Computing – Munif Farhan (Dell)
 - Q&A
- **PM Session – 3:15pm – 4:25pm**
 - 1.1 Spec Overview and Future Directions – Peter Onufryk
 - Panel – “NVMe Deployment and What’s Next?”
 - Moderator: Sergis Mushell, Gartner
 - Panel Members: Steve Sardella (EMC), David Landsman (Sandisk), David Dale (NetApp), Sumit Puri (LSI)

What is NVMe?

- NVMe is a scalable host controller interface designed to address the needs of Enterprise, Datacenter, and Client
- Target for PCIe based SSDs
- Provides optimization
- 13 Promoter companies
 - Intel, Micron, LSI, Marvell, Cisco, EMC, Dell, Oracle, NetApp, sTec, Samsung, SanDisk, PMC Sierra
- Over 90 NVMe member companies
- Plugfest 1.0 Complete
- 1.1 Spec

Why NVMe?

- Deliver the full potential of NVM in Enterprise and Client platforms for PCIe based SSDs
- Architected for performance
 - Performance across multiple cores
 - Optimized Register interface and command set
 - Scalability
 - End to End data protection
 - Lower power consumption



nvmexpress.org



HOME



ABOUT



NEWS



PRODUCTS



RESOURCES



MEMBERS ONLY



BLOG

NVM EXPRESS

The Optimized PCI Express® SSD Interface

The NVM Express specification defines an optimized register interface, command set and feature set for PCI Express (PCIe®)-based Solid-State Drives (SSDs). The goal of NVM Express is to unlock the potential of PCIe SSDs now and in the future, and standardize the PCIe SSD interface.

Questions may be directed to | info@nvmexpress.org

NVM Express Overview & Ecosystem Update

Amber Huffman

Sr Principal Engineer, Intel

August 13, 2013

Agenda

- Importance of Architecting for NVM
- Overview of NVM Express
- Storage Drivers
- Form Factors and Connectors

PCIe* SSDs for the Datacenter

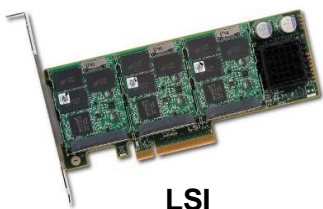
- PCI Express* is a great interface for SSDs
 - Stunning performance 1 GB/s per lane (PCIe* Gen3 x1)
 - With PCIe* scalability 8 GB/s per device (PCIe* Gen3 x8) or more
 - Lower latency Platform+Adapter: 10 μ sec down to 3 μ sec
 - Lower power No external SAS IOC saves 7-10 W
 - Lower cost No external SAS IOC saves ~ \$15
 - PCIe* lanes off the CPU 40 Gen3 (**80** in dual socket)



Virident



Fusion-io



LSI



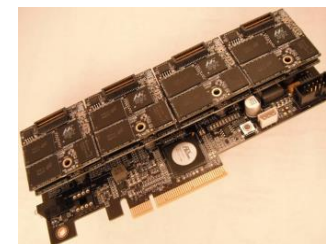
OCZ



Micron



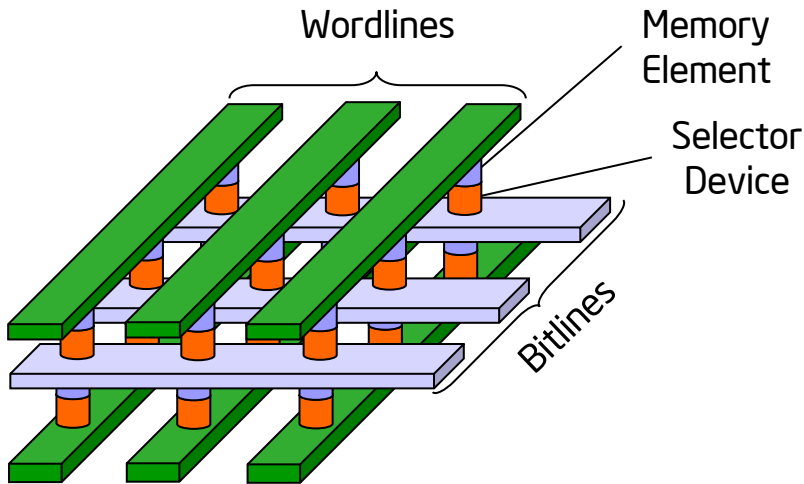
Intel



Marvell

Next Generation Scalable NVM

Scalable Resistive Memory Element



Cross Point Array in Backend Layers $\sim 4\lambda^2$ Cell

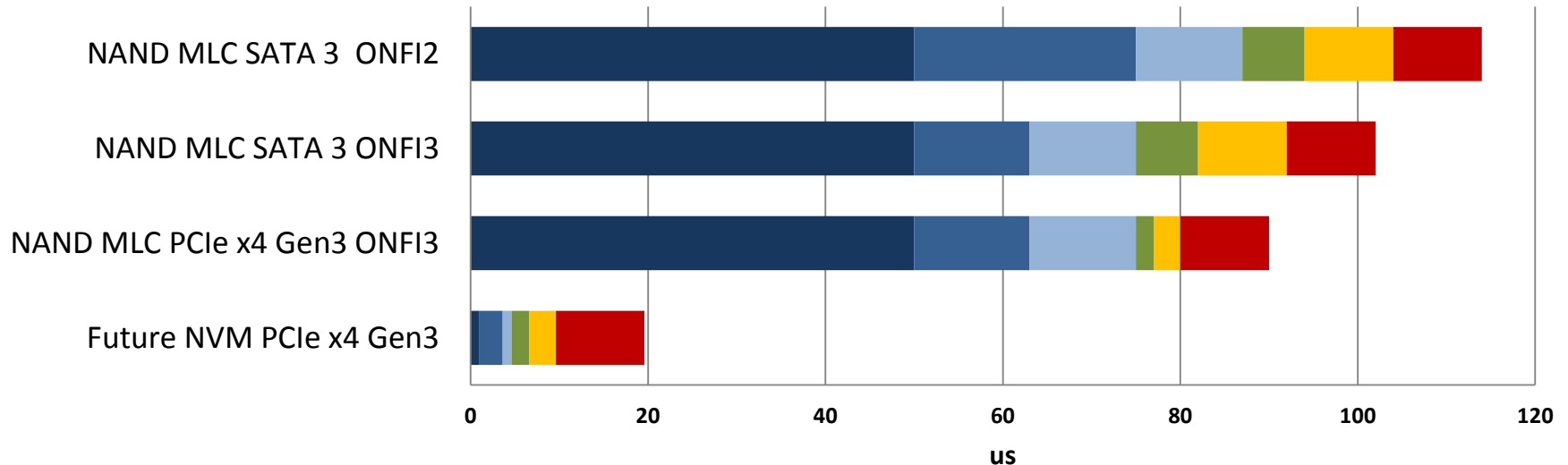
Resistive RAM NVM Options

| Family | Defining Switching Characteristics |
|--------------------------------|--|
| Phase Change Memory | Energy (heat) converts material between crystalline (conductive) and amorphous (resistive) <u>phases</u> |
| Magnetic Tunnel Junction (MTJ) | Switching of magnetic resistive layer by <u>spin-polarized electrons</u> |
| Electrochemical Cells (ECM) | Formation / dissolution of "nano-bridge" by <u>electrochemistry</u> |
| Binary Oxide Filament Cells | Reversible filament formation by <u>Oxidation-Reduction</u> |
| Interfacial Switching | <u>Oxygen vacancy drift</u> diffusion induced barrier modulation |

Many candidate next generation NVM technologies.
Offer $\sim 1000\times$ speed-up over NAND, closer to DRAM speeds.

Fully Exploiting Next Gen NVM *Requires Platform Improvements*

App to SSD IO Read Latency (QD=1, 4KB)

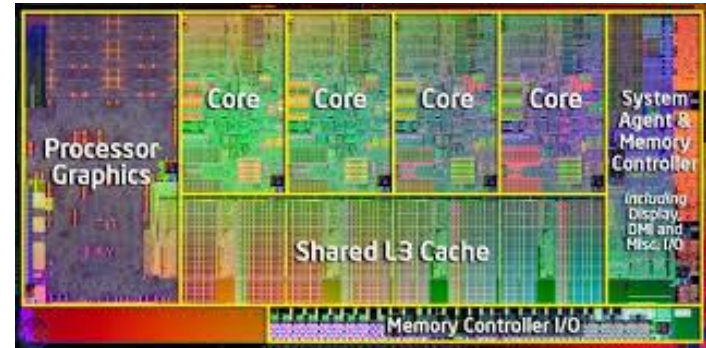


■ NVM Tread ■ NVM xfer ■ Misc SSD ■ Link Xfer ■ Platform + adapter ■ Software

- With Next Gen NVM, the NVM is no longer the bottleneck
 - Need optimized platform storage interconnect
 - Need optimized software storage access methods

Transformation Required

- Transformation was needed for full benefits of multi-core CPU
 - App and OS level changes required
- To date, SSDs have used the legacy interfaces of hard drives
 - Based on a single, slow rotating platter
- SSDs are inherently parallel and next gen NVM approaches DRAM-like latencies



For full SSD benefits, must architect for NVM from the ground up.
NVMe is architected for NAND today and next gen NVM of tomorrow.

Agenda

- Importance of Architecting for NVM
- Overview of NVM Express
- Storage Drivers
- Form Factors and Connectors

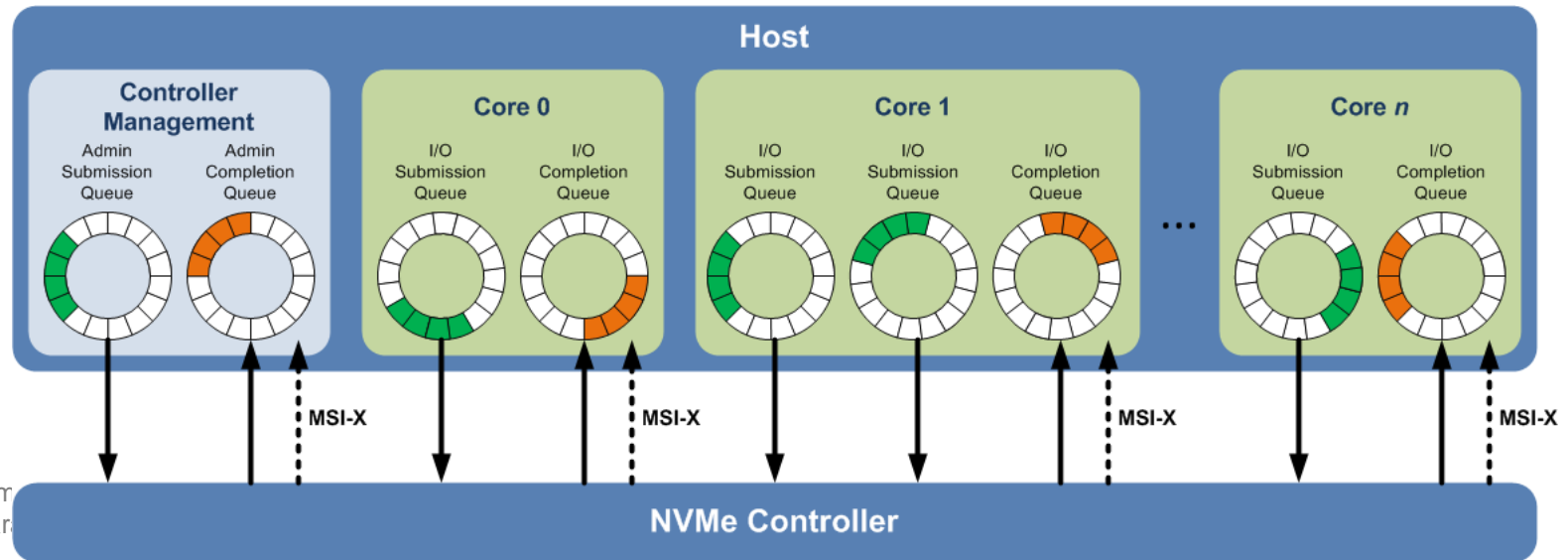
NVM Express

- NVM Express (NVMe) is the standardized high performance host controller interface for PCIe* SSDs
- NVMe was architected from the ground up for non-volatile memory, scaling from Enterprise to Client
 - The architecture focuses on latency, parallelism/performance, and low power
 - The interface is explicitly designed with next generation NVM in mind
- NVMe was developed by an open industry consortium of 90+ members and is directed by a 13 company Promoter Group



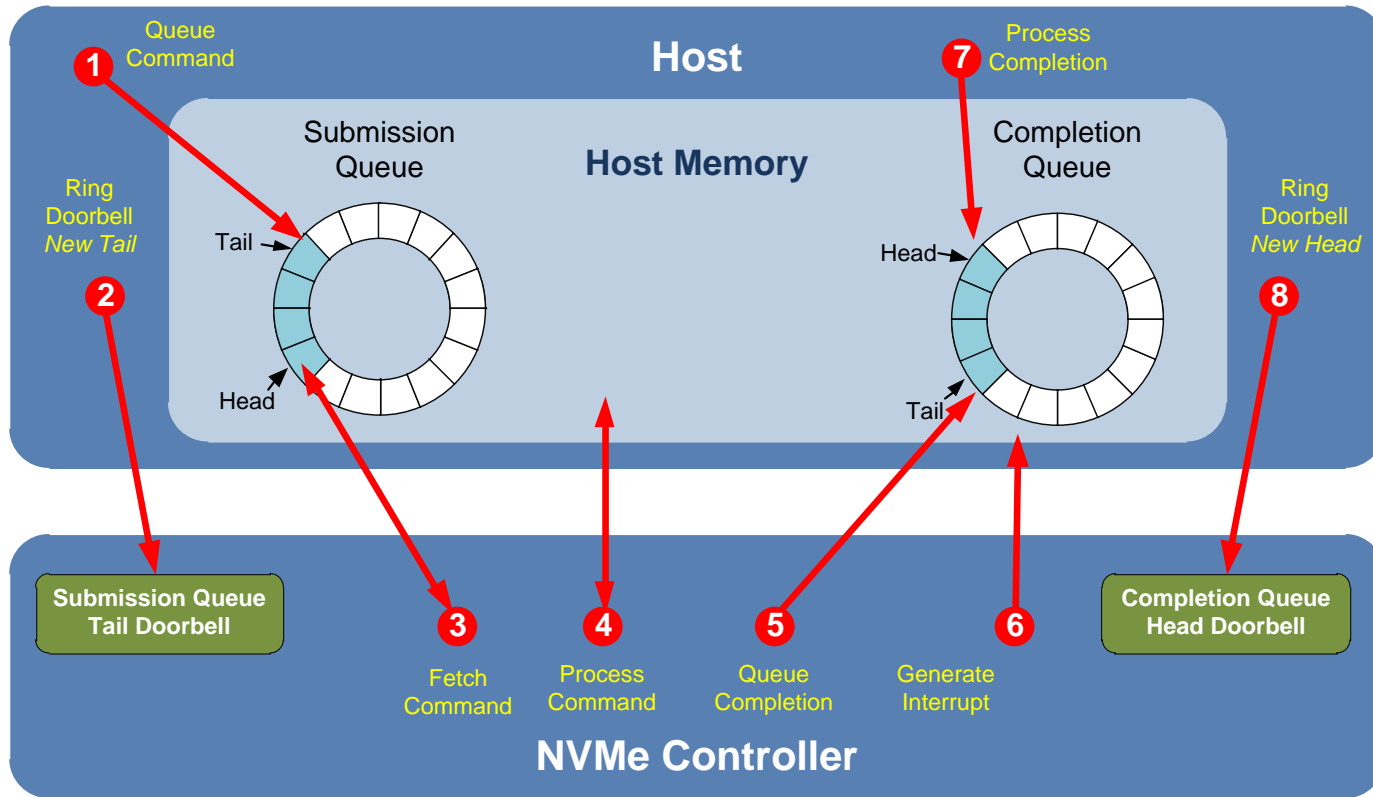
Technical Basics

- All parameters for 4KB command in single 64B command
- Supports deep queues (64K commands per queue, up to 64K queues)
- Supports MSI-X and interrupt steering
- Streamlined & simple command set (13 required commands)
- Optional features to address target segment (Client, Enterprise, etc)
 - Enterprise: End-to-end data protection, reservations, etc
 - Client: Autonomous power state transitions, etc
- Designed to scale for next generation NVM, agnostic to NVM type used



Queuing Interface

Command Submission & Processing



Command Submission

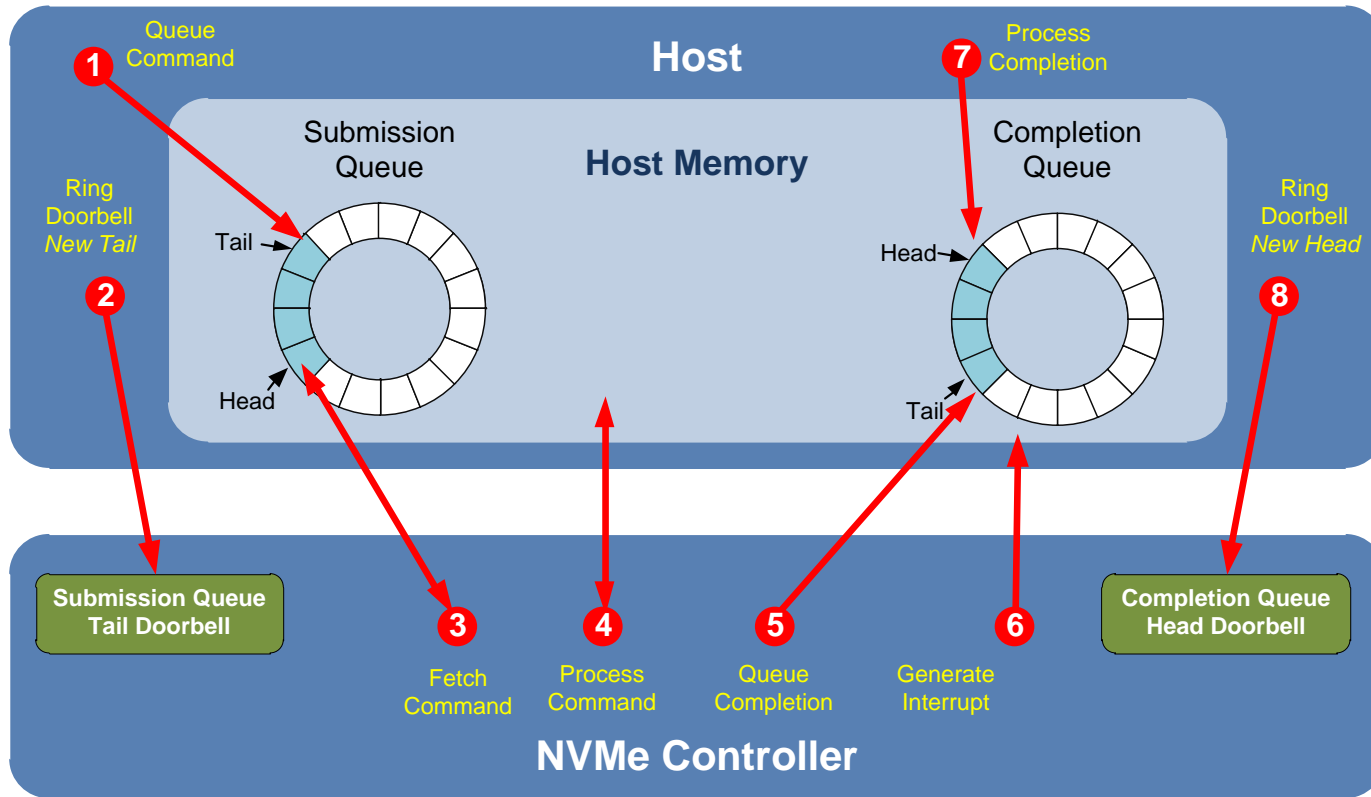
1. Host writes command to Submission Queue
2. Host writes updated Submission Queue tail pointer to doorbell

Command Processing

3. Controller fetches command
4. Controller processes command

Queuing Interface

Command Completion



Command Completion

- | | |
|---|--|
| 5. Controller writes completion to Completion Queue | 7. Host processes completion |
| 6. Controller generates MSI-X interrupt | 8. Host writes updated Completion Queue head pointer to doorbell |

Simple Optimized Command Set

Admin Commands

| |
|---|
| Create I/O Submission Queue |
| Delete I/O Submission Queue |
| Create I/O Completion Queue |
| Delete I/O Completion Queue |
| Get Log Page |
| Identify |
| Abort |
| Set Features |
| Get Features |
| Asynchronous Event Request |
| <i>Firmware Activate (optional)</i> |
| <i>Firmware Image Download (optional)</i> |
| <i>Format NVM (optional)</i> |
| <i>Security Send (optional)</i> |
| <i>Security Receive (optional)</i> |

NVM I/O Commands

| |
|--|
| Read |
| Write |
| Flush |
| <i>Write Uncorrectable (optional)</i> |
| <i>Compare (optional)</i> |
| <i>Dataset Management (optional)</i> |
| <i>Write Zeros (optional)</i> |
| <i>Reservation Register (optional)</i> |
| <i>Reservation Report (optional)</i> |
| <i>Reservation Acquire (optional)</i> |
| <i>Reservation Release (optional)</i> |

Only 10 Admin and 3 I/O commands required.

Proof Point: NVMe Latency

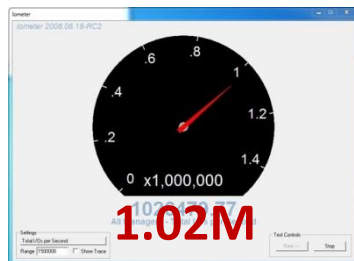
- NVMe reduces latency overhead by more than 50%
 - SCSI/SAS: 6.0 μ s 19,500 cycles
 - NVMe: 2.8 μ s 9,100 cycles**
- Increased focus on storage stack / OS needed to reduce latency even further

Chatham NVMe Prototype

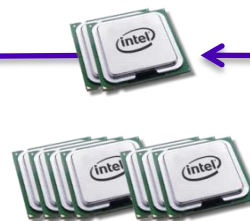


Measurement taken on Intel® Core™ i5-2500K 3.3GHz 6MB L3 Cache Quad-Core Desktop Processor using Linux RedHat® EL6.0 2.6.32-71 Kernel.

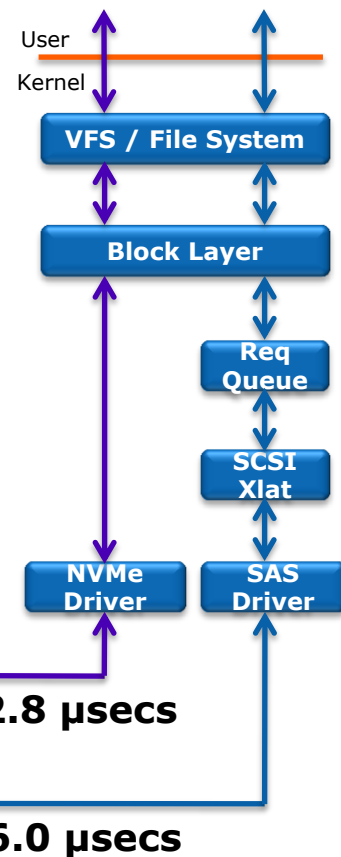
Prototype Measured IOPS



Cores Used for 1M IOPS



Linux* Storage Stack





NVMe Deployment Beginning

- NVM Express 1.0 specification published in March 2011
 - Additional Enterprise and Client capabilities included in NVMe 1.1 (Oct 2012)
- First plugfest held May 2013 with 11 companies participating
 - Interoperability program run by University of New Hampshire Interoperability Lab, a leader in PCIe*, SAS, and SATA compliance programs

"NVMe was designed from the ground up to enable non-volatile memory to better address the needs of enterprise and client systems. This Plugfest highlights the rapid development and maturity of the NVMe specification and the surrounding infrastructure as well as supporting PCIe SSD devices."

**JH Lee, Vice President,
Flash Memory Product Planning and Enabling,
Samsung Electronics**

FOR IMMEDIATE RELEASE

NVM Express Workgroup Holds First Plugfest

Milestone in Process to Deliver Standards-based Interoperability for PCI Express Solid-State Drives

WAKEFIELD, Mass., May 29, 2013 – The [NVM Express Workgroup](#), developer of the NVM Express specification for accessing solid-state drives (SSDs) on a PCI Express (PCIe) bus, held its first Plugfest at the University of New Hampshire InterOperability Lab in Durham, N.H., May 13-16, 2013. This event provided an opportunity for participants to measure their products' compliance with the NVM Express (NVMe) specification and to test interoperability with other NVMe products.

The NVMe specification defines an optimized register interface, command set and feature set for PCIe-based Solid-State Drives (SSDs). NVMe refers to non-volatile memory, as used in SSDs. The goal of NVMe is to unlock the potential of PCIe SSDs now and in the future, and to standardize the PCIe SSD interface. Participating in the Plugfest were Agilent Technologies, Dell Inc., Fastor Systems, Inc., HGST, a Western Digital company, Integrated Device Technology, Inc., Intel Corporation, Samsung Electronics Co., Ltd., SanDisk Corporation, sTec, Inc., Teledyne LeCroy, and Western Digital Corporation.

NVM Express products targeting Datacenter expected to ship 2H'13.

Agenda

- Importance of Architecting for NVM
- Overview of NVM Express
- Storage Drivers
- Form Factors and Connectors

Driver Developments on Major OSes

Windows*

- Windows* 8.1 includes inbox driver
- Open source driver in collaboration with OFA

Linux*

- Native OS driver since Linux* 3.3 (Jan 2012)

Unix

- FreeBSD driver upstream; ready for release

Solaris*

- Solaris driver will ship in S12

VMware

- vmklinux driver certified release in Dec 2013

UEFI

- Open source driver available on SourceForge

Native OS drivers already available in Windows and Linux!

Windows* Open Source Driver Update

Release 1

- Q2 2012 (released)
- 64-bit support on Windows* 7, Windows* Server 2008 R2
- Mandatory features

Release 1.1

- Q4 2012 (released)
- Added 64-bit support Windows* 8
- Public IOCTLs and Windows* 8 Storport updates

Release 1.2

- Q2 2013 (released)
- Added 64-bit support on Windows* Server 2012
- Signed executable drivers

Release 1.3

- Target: Q4 2013
- Added 32-bit support on all supported OS versions
- End-to-end Data Protection

Three major releases of the Windows* community driver since 2012.
Code contributions from Huawei, IDT, Intel, LSI, and SanDisk.

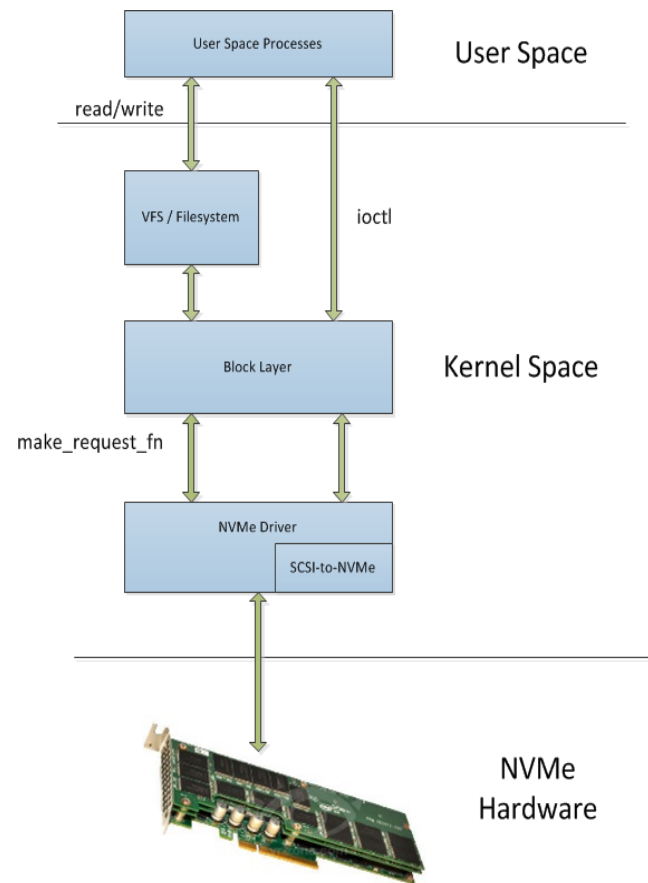
Recent Feature Additions

- Deallocate (i.e., Trim support)
- 4KB sector support (in addition to 512B)
- SCSI IOCTL support
- MSI support (in addition to MSI-X)
- Disk I/O statistics
- Many bug fixes

Community Effort

- Contributions from Fastor, IDT, Intel, Linaro, Oracle, SanDisk, and Trend Micro
- 59 changes since integrated into kernel

Current work includes power management, support for end-to-end data protection, sysfs manageability, and NUMA optimizations.



FreeBSD Driver Update

- NVMe support is upstream in the head and stable/9 branches
- FreeBSD 9.2 will be the first official release with NVMe support, slated for end of August



FreeBSD NVMe Modules

nvmecontrol

User space utility,
including firmware update

nvd

NVMe/block layer shim

nvme

Core NVMe driver

Solaris* Driver Update

- Current Status from Oracle team:
 - Stable and efficient working prototype conforming to 1.0c
 - Direct block interfaces bypassing complex SCSI code path
 - NUMA optimized queue/interrupt allocation
 - Support 8K memory page size on SPARC system
 - Plan to validate driver against Oracle SSD partners
 - Plan to integrate into S12 and a future S11 Update Release
- Future Development Plans:
 - Boot & install on SPARC and X86
 - Surprise removal support
 - Multipath, SR-IOV, SGL, etc

VMware Driver Update

- Initial “vmklinux” based driver in final stages of development
 - First release in mid-Oct, 2013
 - Certified release in Dec, 2013
- Native NVMe driver with pluggable Vendor Extensions planned for the future
- VMware’s IOVP program includes workflow for bugs/issues

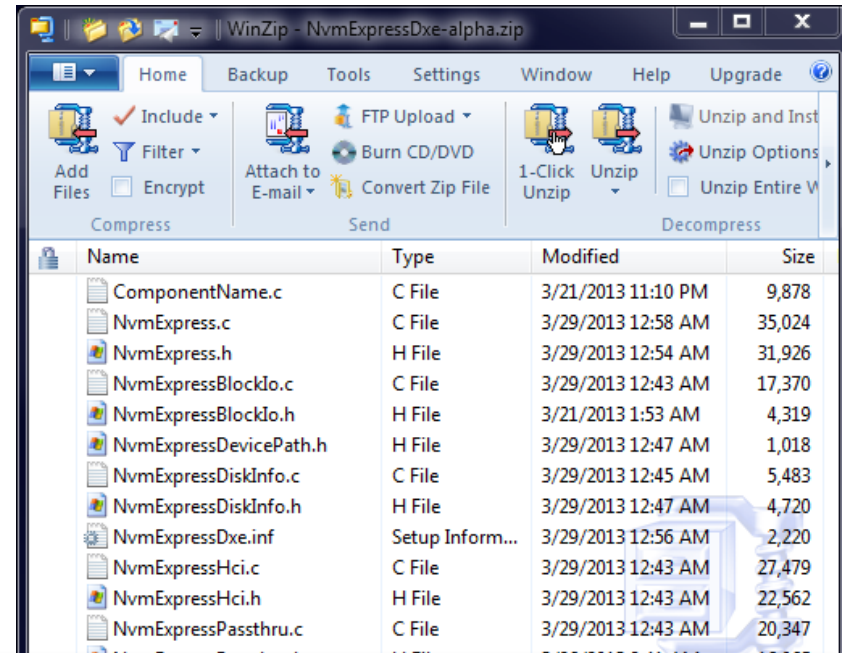


UEFI Driver

- The UEFI 2.4 specification available at www.UEFI.org contains updates for NVMe
- An open source version of an NVMe implementation for UEFI is at:
<https://sourceforge.net/projects/edk2/files/EDK%20II%20Releases/other/NvmExpressDxe-alpha.zip/download>
- The UEFI driver has been validated using the qemu virtual platform
 - The team is ready to test against real hardware as it rolls in

"AMI is working with vendors of NVMe devices and plans for full BIOS support of NVMe in 2014."

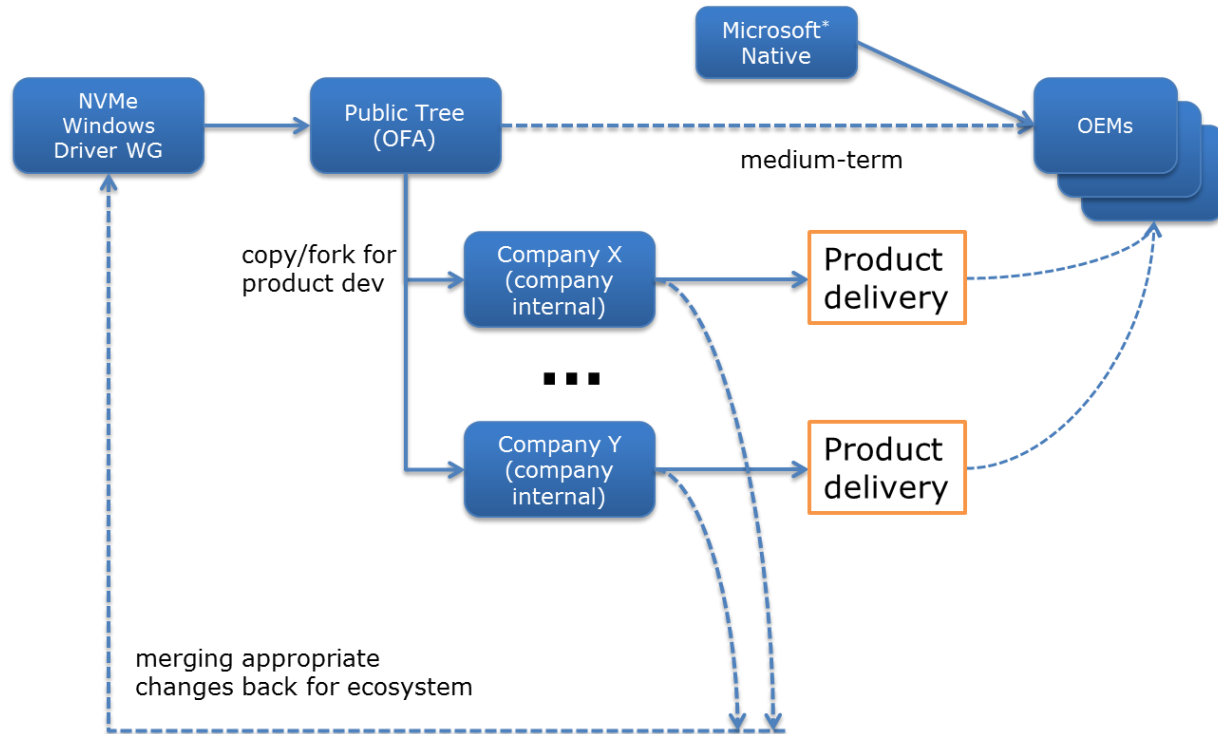
Sandip Datta Roy
VP BIOS R&D, AMI



NVMe boot support with UEFI will start percolating releases from Independent BIOS Vendors in 2014.

Fork and Merge for IHV Value-Add Drivers

- The NVMe Workgroup continues to recommend a “Fork and Merge” approach when IHVs provide their own value add driver
- Benefits of this strategy include: 1) maximum reference code re-use, 2) continuous improvement of reference code, 3) enables product team to focus on delivery (not basic driver)



Agenda

- Importance of Architecting for NVM
- Overview of NVM Express
- Storage Drivers
- Form Factors and Connectors

M.2 Emerging as Primary Client Form Factor

- In client, as SSDs move first to PCIe, OEMs are using the optimized M.2 form factor
- As native OS support of NVMe becomes pervasive, OEMs will move from AHCI to NVMe to take full advantage of PCIe

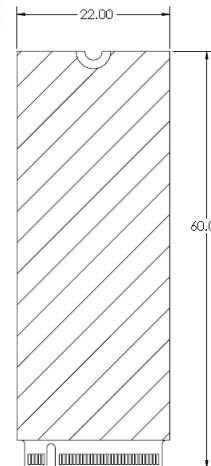
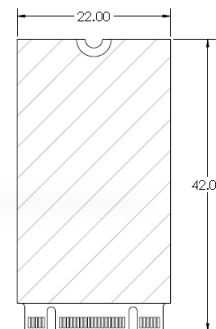
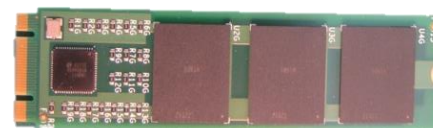


VAIO* Pro 13 Ultrabook™

The world's lightest 13.3" touch Ultrabook²¹.

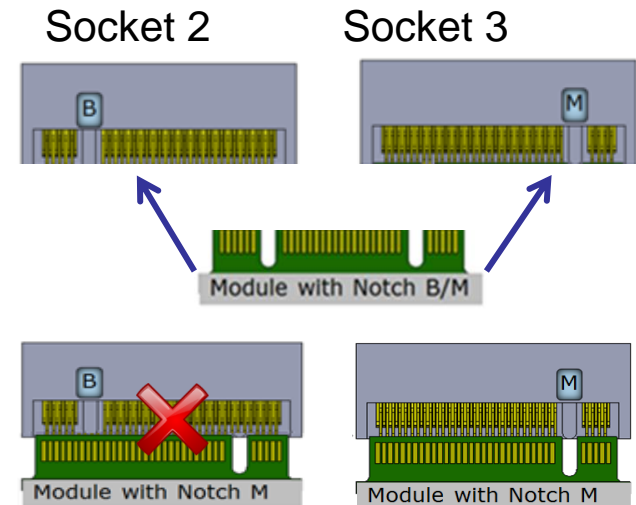
Features:

- 4th gen Intel® Core™ i7 processor available
- Windows 8 Pro available
- Full HD TRILUMINOS IPS touchscreen (1920 x 1080)
- Super fast 512GB PCIe SSD available
- Ultra-light at just 2.34 lbs.



M.2 Provides OEM Choice: Max Performance or Flexibility

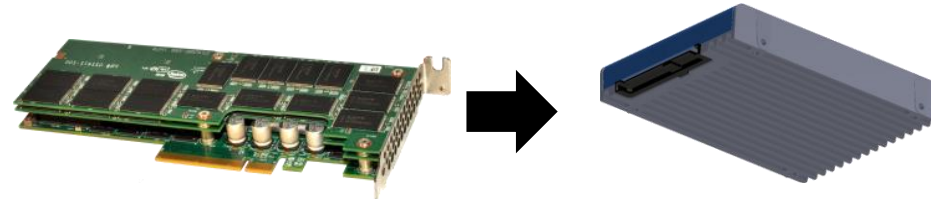
- Three families of modules:
 - Socket 1: Wi-Fi/Connectivity only
 - Socket 2: WWAN, Storage (SATA*, PCIe* x1, PCIe* x2), other
 - Socket 3: Storage only (SATA*, PCIe* x1, PCIe* x2, PCIe* x4)
- OEMs choose the socket to include
 - Socket 2: Most flexibility
 - Socket 3: Highest performance



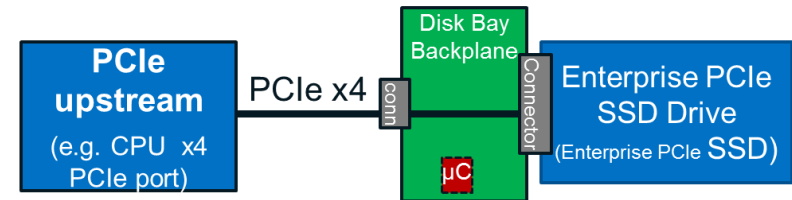
With M.2, client OEMs can choose maximum performance with 4 lanes, or they can choose flexibility with SATA and WWAN options.

SFF-8639 Brings Full Storage Capabilities to Enterprise

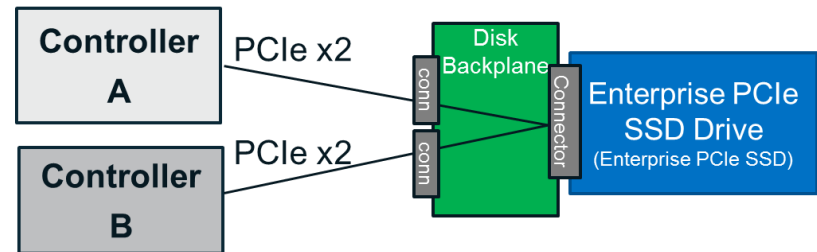
- SFF-8639 brings a 2.5" pluggable form factor to the Enterprise
- For Enterprise PCIe SSDs, this includes support for a typical server and storage configuration
- Server: Single x4 PCIe SSD
- Storage: High availability dual ported solution



Typical Server configuration

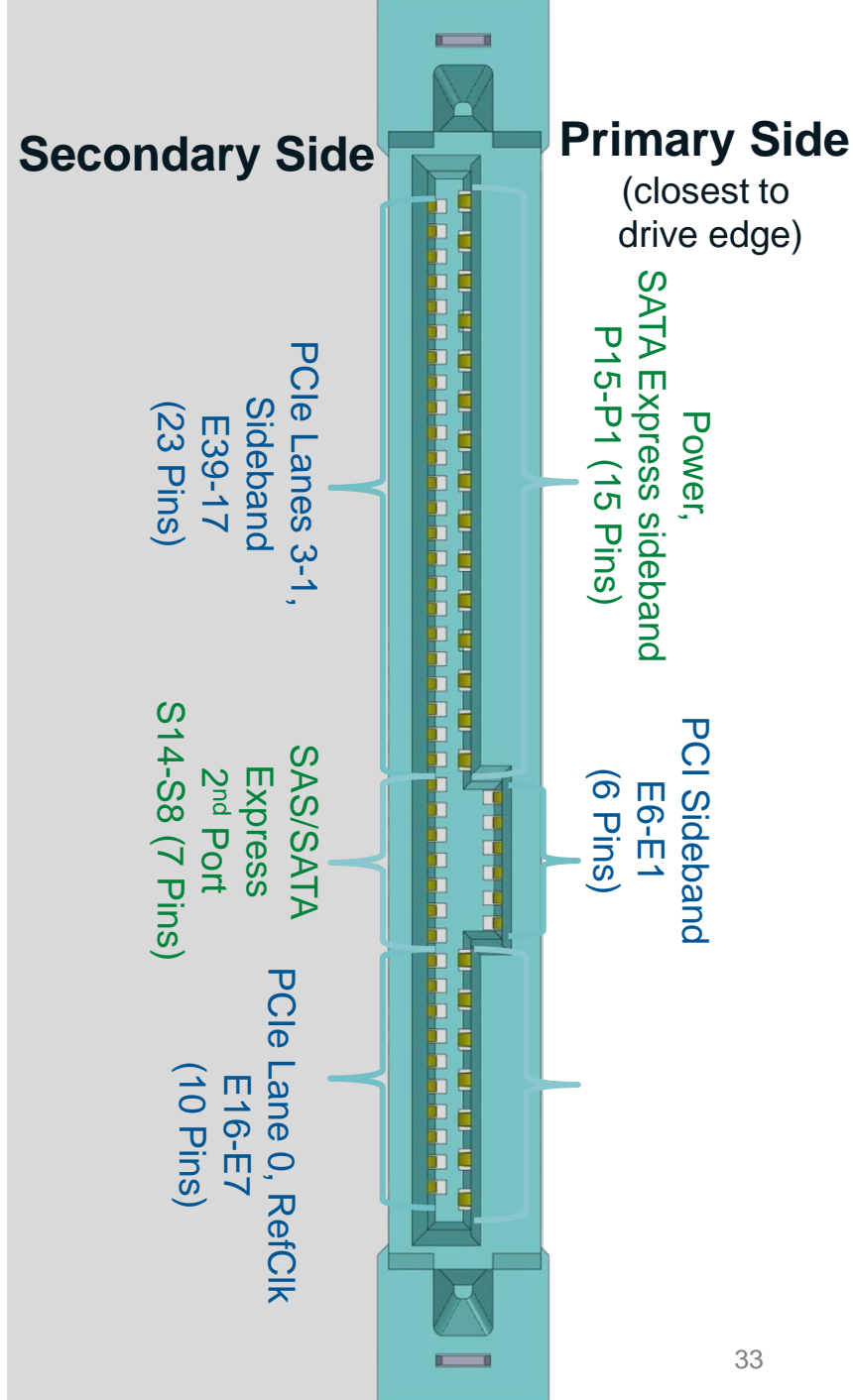


Typical High Availability Storage configuration



SFF-8639 Flexibility

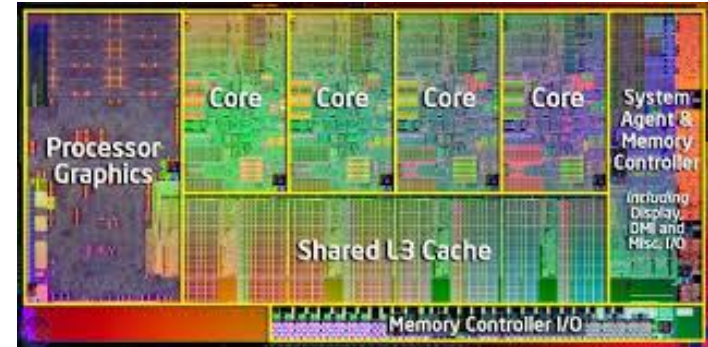
- SFF-8639 supports:
 - Enterprise PCIe x4 SSDs
 - Existing SAS drive (dual port)
 - Existing SATA drives
- As ecosystem develops:
 - Client 2.5" PCIe (often referred to as SATA Express)
 - x4 SAS
- Supports flexible backplanes
 - Enterprise x4 PCIe SSDs
 - SAS/SATA HDDs



Summary: Transformation Required

Recall:

- Transformation was needed for full benefits of multi-core CPU
 - Application and OS level changes required
- To date, SSDs have used the legacy interfaces of hard drives
 - Based on a single, slow rotating platter..
- SSDs are inherently parallel and next gen NVM approaches DRAM-like latencies



For full SSD benefits, must architect for NVM from the ground up.
Future proof your PCIe storage investment by adopting NVMe.

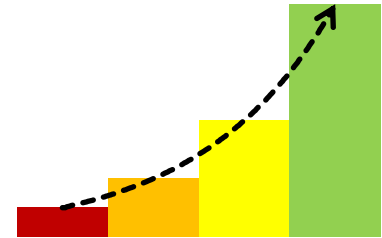


Windows' Perspective on NVMe

Tobias Klima – PM

Windows Core – Storage and File
Systems

- The Protocol
 - Standardized PCIe Storage
 - Natural Progression
- The OS
 - Windows Inbox Driver (StorNVMe.sys)
 - Windows Server 2012 R2 (high-density/performance)
 - Windows 8.1 (small form factors)
 - Stable Base Driver





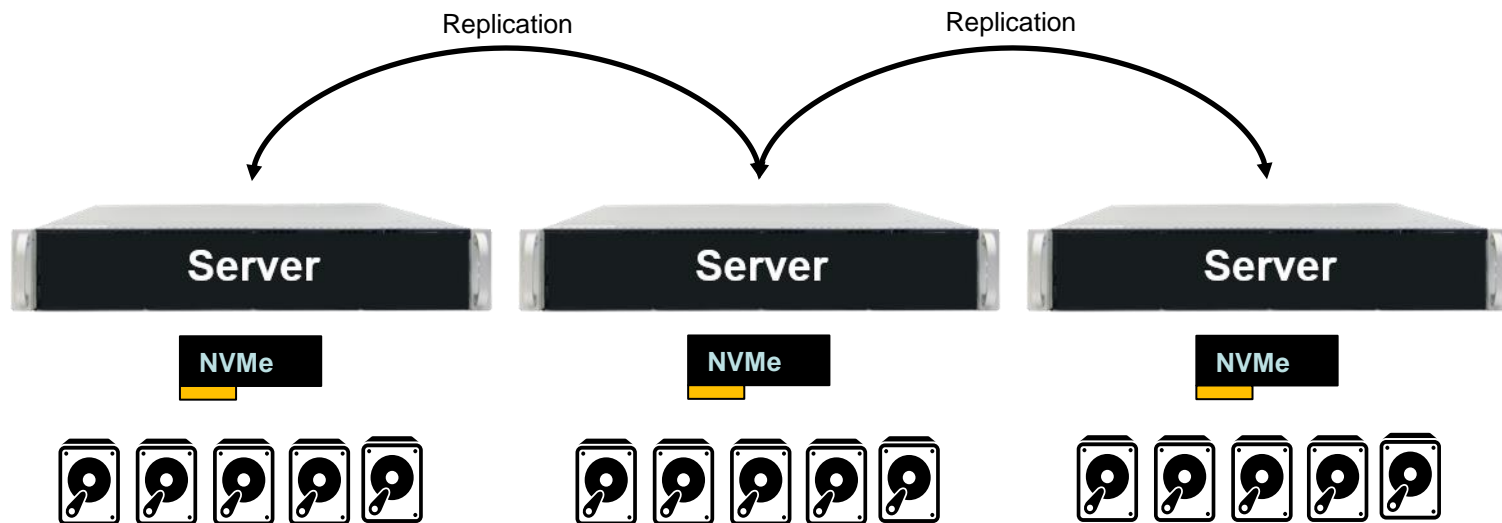
Server/ Client Considerations

- Server
 - First devices are enterprise-class
 - High-Density / Performance
 - Closing the latency gap with RAM

- Client
 - Boot
 - UEFI\Platform support required first
 - Granular Power Management needed
 - AHCI PCIe SSDs causing confusion

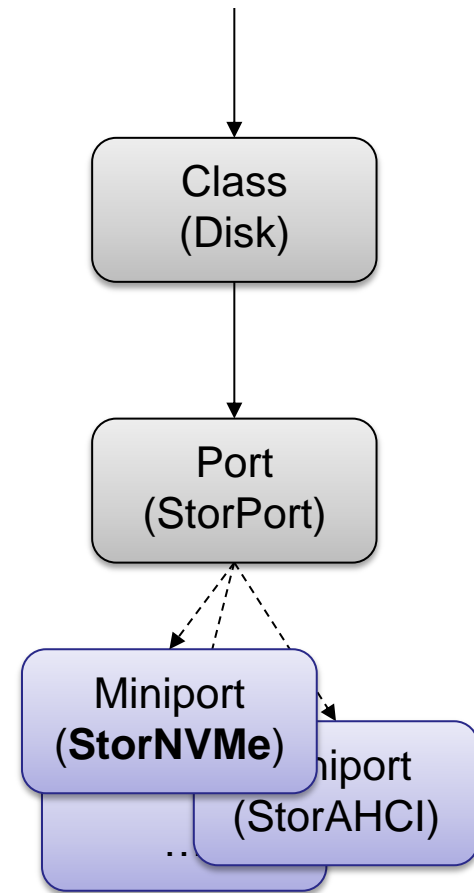
NVMe Use Cases

- Replicated Systems / Custom Deployment
- Non-Clustered Storage Spaces

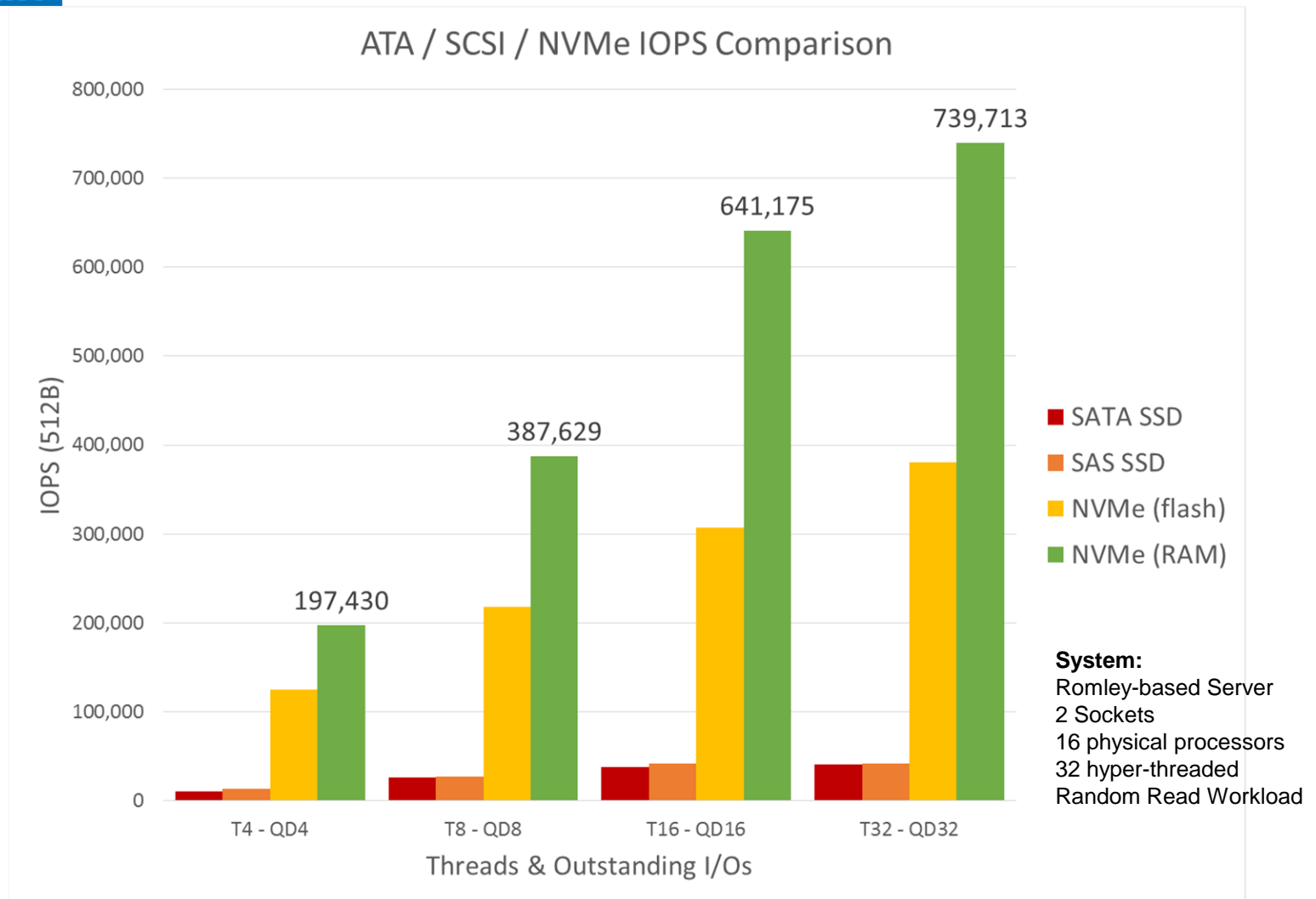


The Windows Storage Driver Model

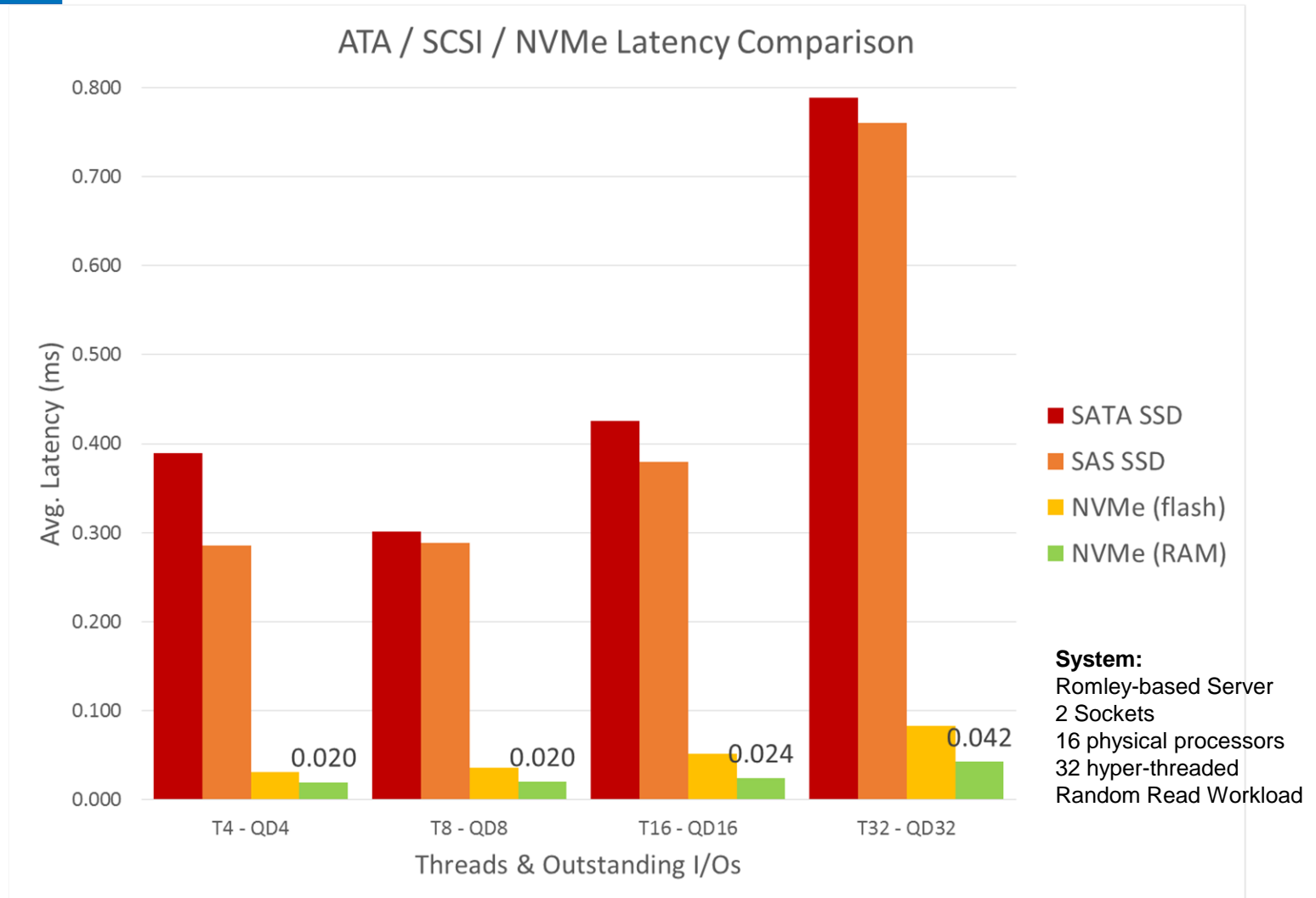
- The Storport Model
 - Reduced development cost
 - Offloads Basics: PnP, Power, Setup, Crash, Boot*
 - Mature / Hardened Model
 - Storport optimized for performance
 - RAM-backed NVMe device
 - > 1 million IOPS | < 20μs latencies



Windows Stack Performance



Windows Stack Latency



Future Challenges

- Shareable Devices
 - High Availability (Clustering)
 - Fault Tolerance (Storage Spaces)
- Form Factor
 - Small Devices, High Density, Power
- Transition
 - SATA → NVMe



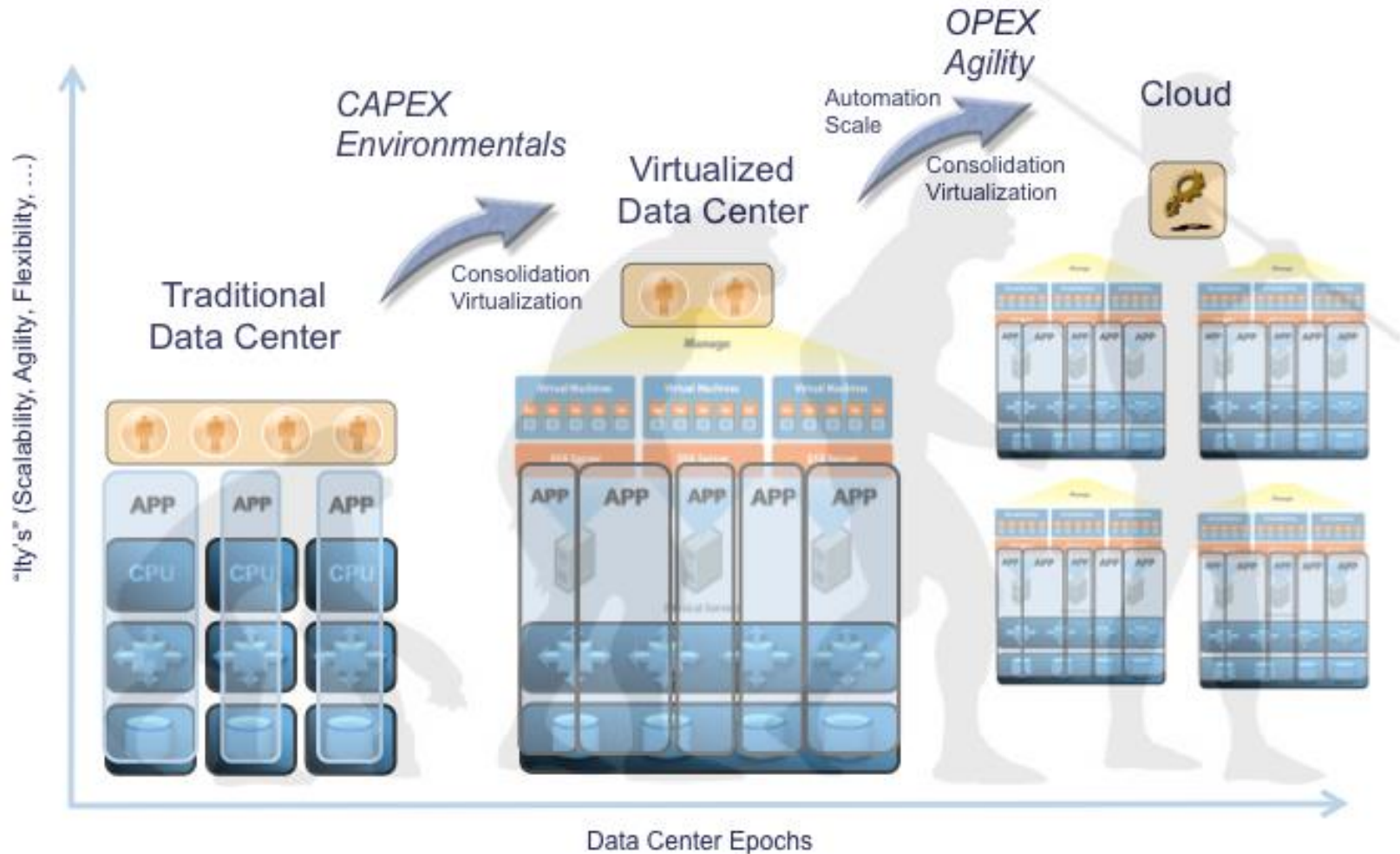
Summary

Test it, Send us Feedback!

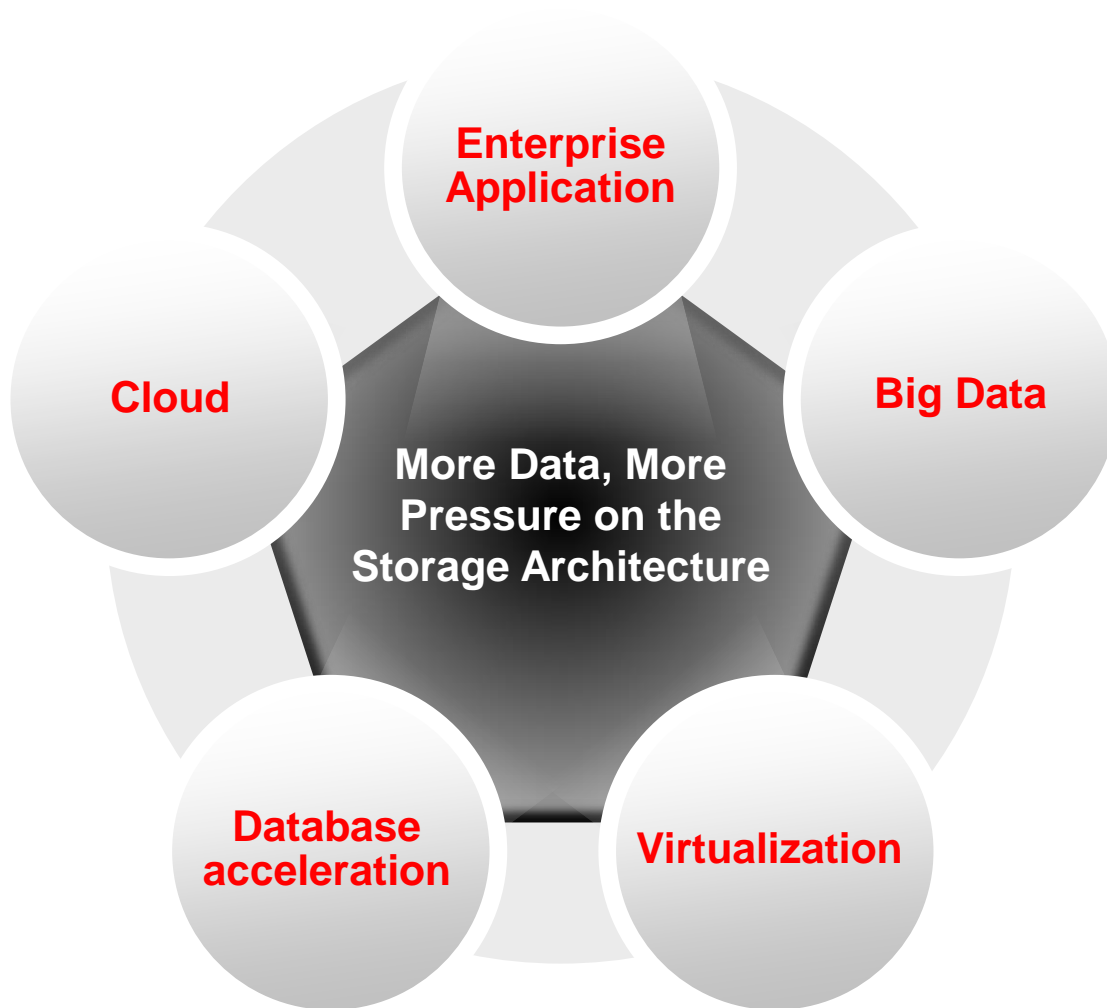
NVMe Applications for Enterprise Datacenter

Swapna Yasarapu
Director of SSD Product Marketing -
sTec

Data Center Evolution

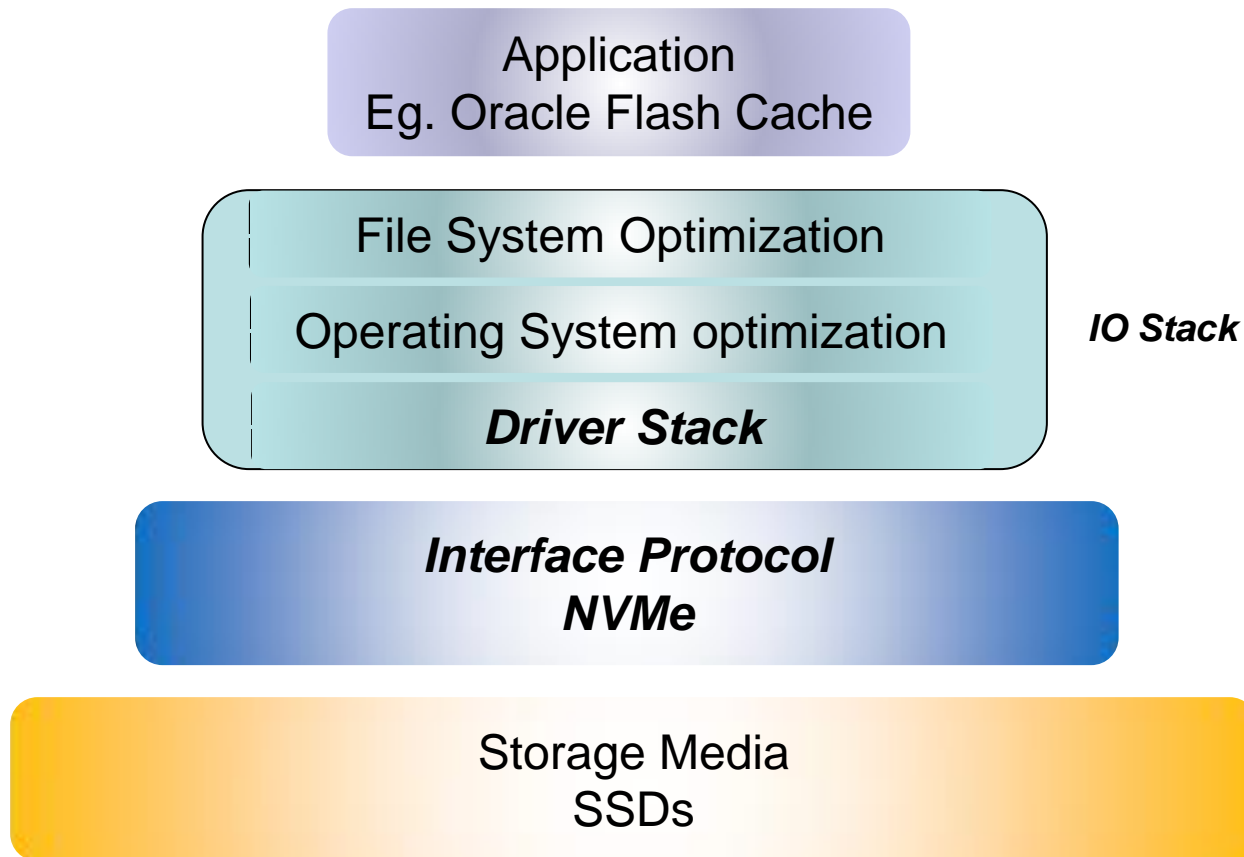


Pressure on Storage architectures



**Traditional
Architectures
Can't Keep Pace**

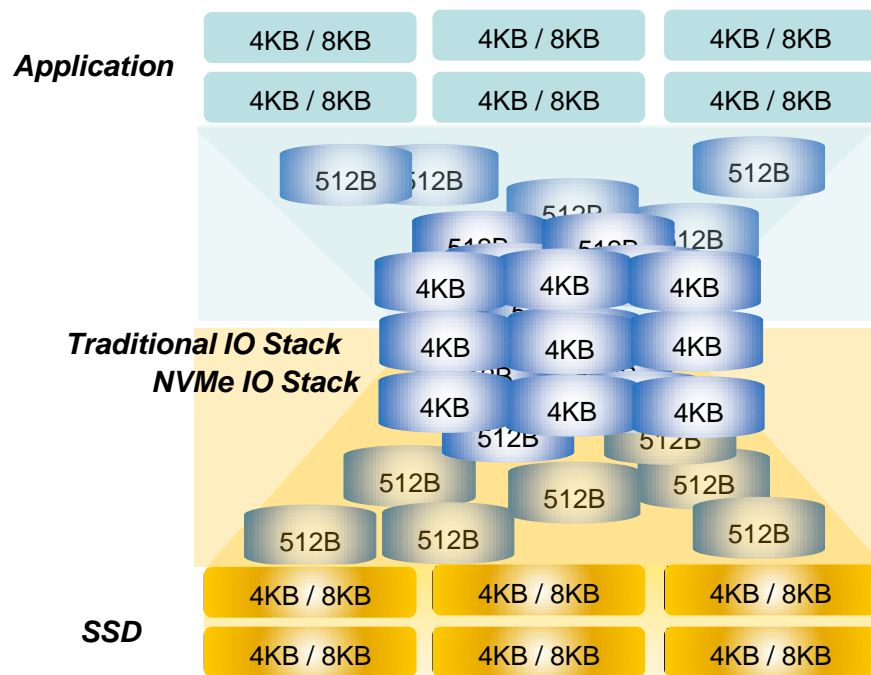
Improvements required through IO stack





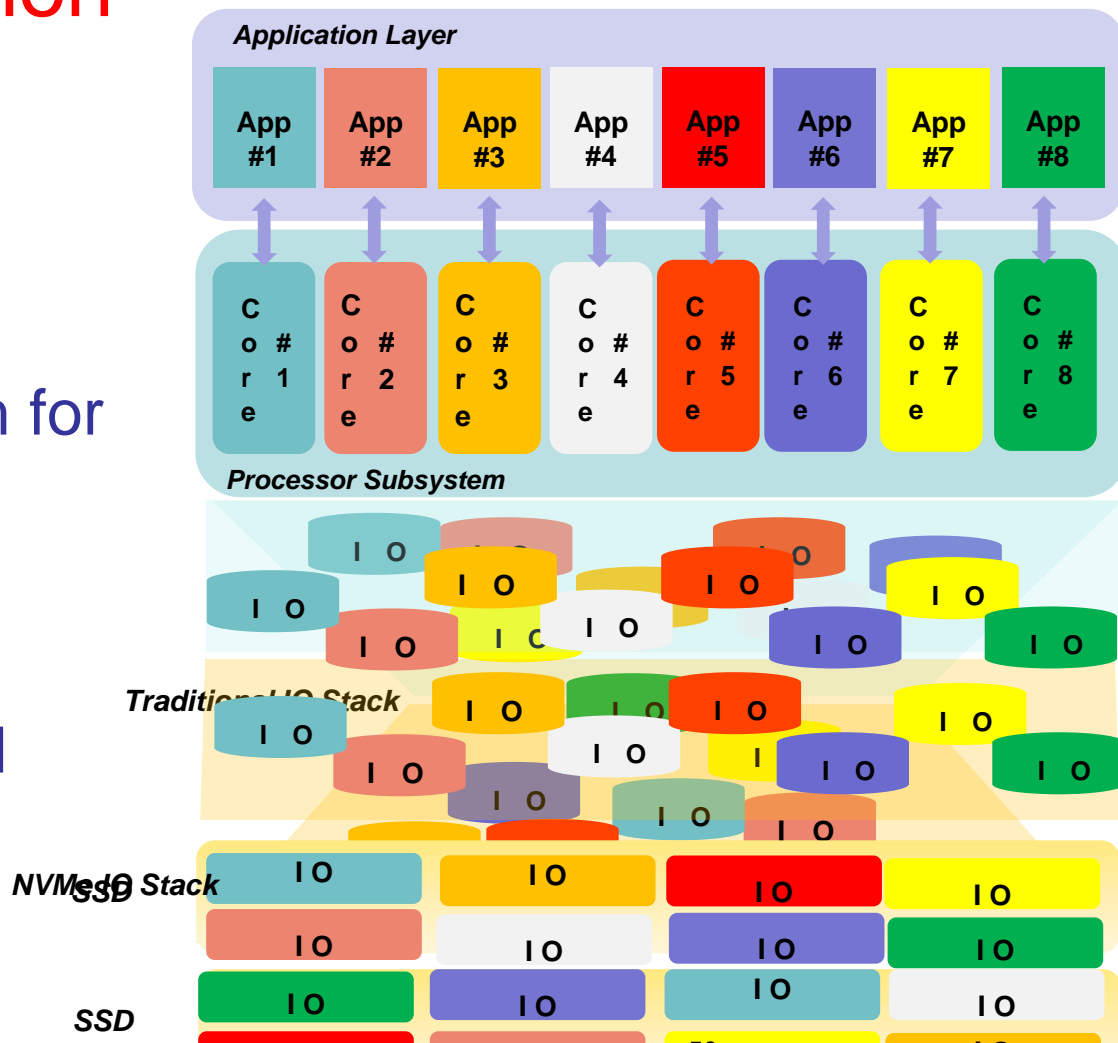
NVMe built to accelerate enterprise applications

- Traditional IO stack optimized for rotational media
 - Eg. 512B block size transfers
- Database applications tuned for larger IO granularities
 - Oracle 4KB block granularity
 - MySQL 8KB block granularity
- NVMe provides native atomic IO size affinity for databases



Enable faster and more efficient database performance
NVMe matches the natural application IO granularities to solid state media

- NVMe has affinity to multi core architecture
- End to End parallelism for improved VM performance
- Balanced QoS per VM application



Extend parallelism inherent in virtualization IO stack all the way to storage media

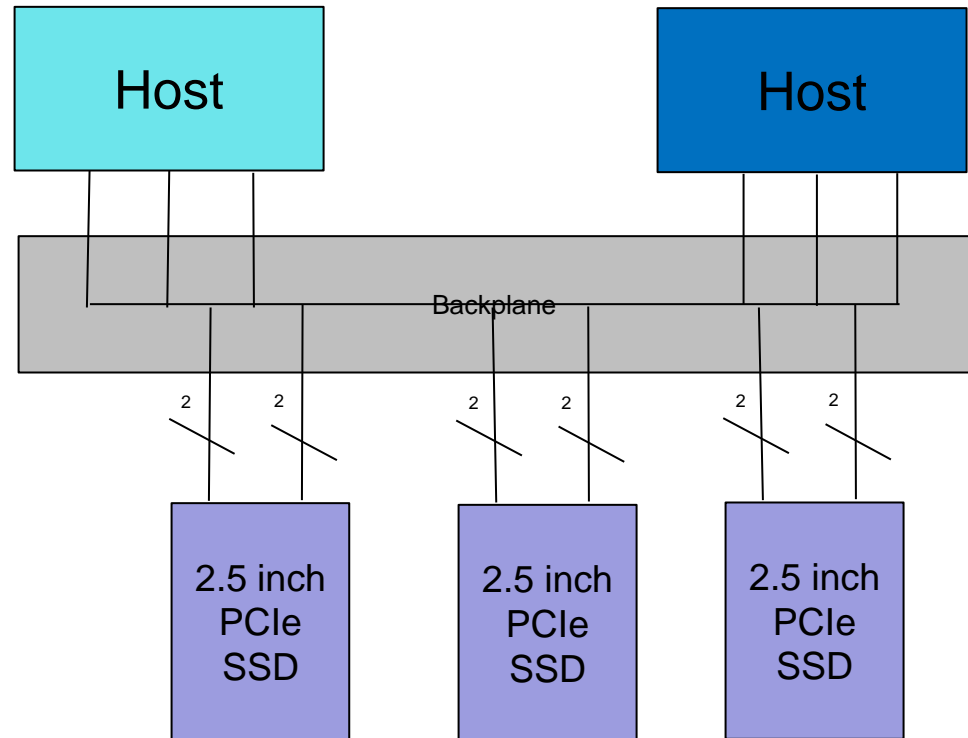
Cost-effective storage system acceleration

- Single device – multi- workload optimization
- Modern flash aware file systems require write log and read cache devices
- Single NVMe device enables workload optimized per domain performance

***Eliminate wasted SSD capacity;
Lower operating costs***

PCIe in Enterprise Applications

- Active-active shared storage combined with high speed PCIe and low latency
- SFF 8639 2 x2 PCIe interface connectivity plus NVMe enabled multipath IO



Path to highly available PCIe deployments

Deployment Efficiency

- Cross platform standardized drivers
 - Windows, linux, Unix, Solaris, Vmware
- Industry sponsored interop
 - Plugfests, standardized test suites, compliance
- Faster test and easy deployment



How does it help the IT manager?



NVMe delivering on Enterprise Requirements

- ✓ Higher Performance
- ✓ Scalable Architecture
- ✓ High availability
- ✓ Deployment Efficiency
- ✓ Lower Operating Costs
- ✓ Serviceability

NVMe Conformance & Interoperability

David Woolf
Senior Engineer - UofH IOL

Agenda

- **NVMe Test Program Overview**
- May 2013 NVMe Plugfest
- NVMe Integrators List
- Improving the Test Program – Continue to Climb



NVMe Test Program Overview



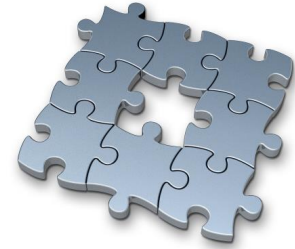
- NVMe Test Program is focused on a public Integrators List hosted by UNH-IOL
- Promoters group decides the qualifications for a product to get on the Integrators List
- UNH-IOL performs the testing to determine if a products meets those qualifications



NVMe Test Program Overview



- New Testing Requirements approved by NVMe Promoters group and debuted at plugfests
- Testing for Integrators List performed throughout the year at UNH-IOL.
- UNH-IOL makes test tools available to members to enable conformance checking



Agenda

- NVMe Test Program Overview
- **May 2013 NVMe Plugfest**
- NVMe Integrators List
- Improving the Test Program – Continue to Climb



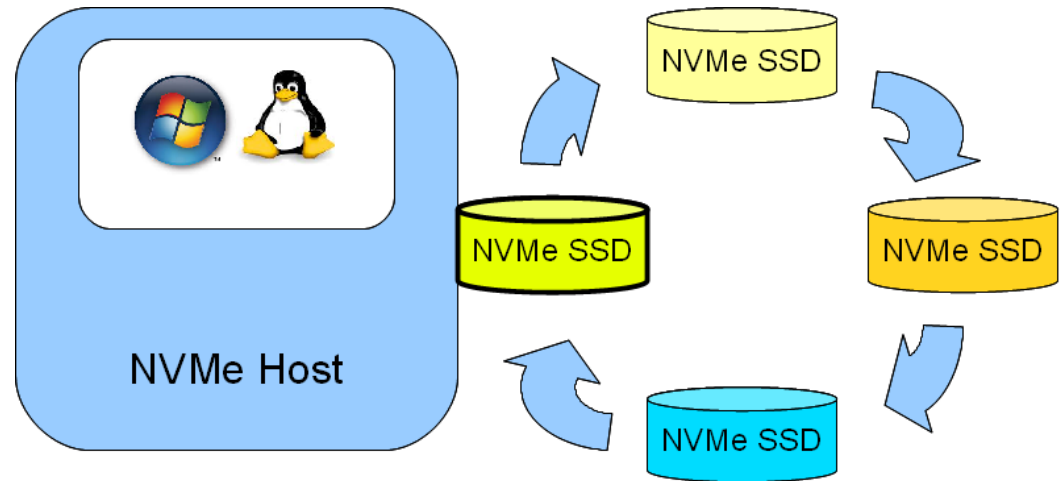


- 1st ever NVMe plugfest held at UNH-IOL May 13-16, 2013
- 11 Companies participating
- 6 products added to NVMe Integrators List





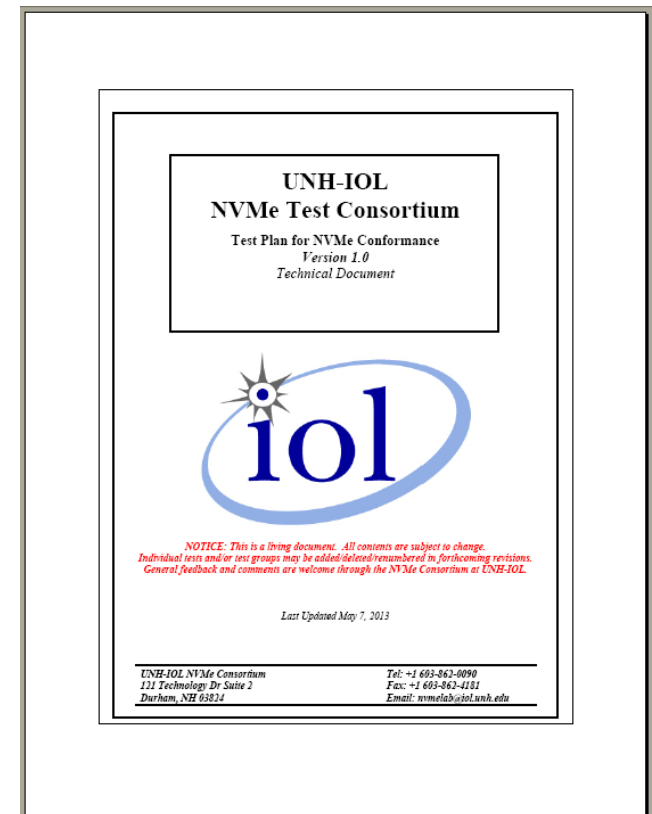
■ Interop Testing



- Tests defined in NVMe Interop Test Suite document, publically available at :
<https://www.iol.unh.edu/services/testing/NVMe/testsuites/>
- Cycle SSDs against each NVMe host
- Used vdbench on Windows and Linux OS to read/write/verify data

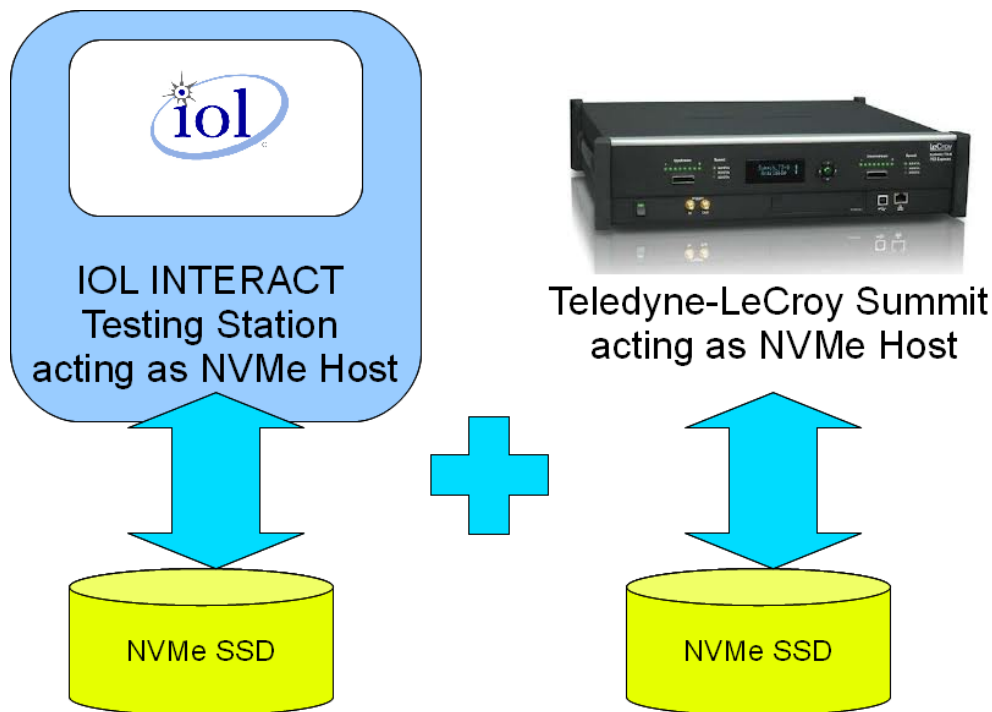


- Conformance Testing
 - Tests defined in NVMe Interop Test Suite document (publically available at www.iol.unh.edu)
 - Test tools used to check proper construction and response to different NVMe Stimuli
 - Admin Commands, NVM Commands, Controller Registers





- Conformance Testing
 - 2 types of test stations used



Agenda

- NVMe Test Program Overview
- May 2013 NVMe Plugfest
- **NVMe Integrators List**
- Improving the Test Program – Continue to Climb






- Hosted by UNH-IOL at www.iol.unh.edu
- Opt-in list of qualifying NVMe products
- No PCIe component to qualification today
 - UNH-IOL does offer PCIe testing to complement NVMe testing, but it is not a component of the NVMe IL Qualification





<https://www.iol.unh.edu/services/testing/NVMe/integratorslist.php>



University of New Hampshire
InterOperability Laboratory
Improving networks worldwide.

Log In Home

Services Education For Members Press Room

Location: Home » Services » Testing » NVMe

NVMe Integrators List

This Integrators List (IL) contains information about NVMe Products that UNH-IOL has performed interoperability and conformance testing on. Successful completion of such conformance tests when combined with satisfactory operation in UNH-IOL's interoperability tests provides a reasonable level of confidence that the Product Under Test will function properly in many NVMe environments.

Products listed here have met the requirements of the NVMe Integrators List Policy, documented here: [NVMe Integrators List Policy Document](#)

[NVMe Devices](#)
[NVMe Host Platforms](#)

NVMe Devices

| Product | Firmware Version | Test Suite Versions | Date Listed | Test ID | Further Info |
|--|------------------|--|-------------|---------|--------------|
| IDT Princeton NVMe Controller | 1611 | <ul style="list-style-type: none"> Interop TS: v1.0 Conformance TS: v1.0 | 5/31/13 | | |
| Samsung XS1715 | IPM04B20 | <ul style="list-style-type: none"> Interop TS: v1.0 Conformance TS: v1.0 | 5/31/13 | | |
| Western Digital Technologies, Inc. PCIe NVMe SSD | | <ul style="list-style-type: none"> Interop TS: v1.0 Conformance TS: v1.0 | 5/31/13 | | |

Member

- » Conso
- » Curren
- » Feedb
- » Reque

NVMe Co

- » Conso
- » Membr
- » Join
- » Test S
- » Test T
- » Knowle
- » Equipr
- » Contac

Related

- » SAS
- » SATA
- » PCIe

Testing P

- » All Tec

NVMe Integrators List as of August 7, 2013



- NVMe Host Qualification
 - Perform Interop Test against 4 different SSDs
 - Pass with 3 SSDs
 - Pass = data transfers without errors





- NVMe Device Qualification
 - Perform Interop Test against 4 different Hosts, pass with 3 Hosts
 - One of the hosts must be either the Windows or Linux Reference Driver
 - Pass all conformance tests



Agenda

- NVMe Test Program Overview
- May 2013 NVMe Plugfest
- NVMe Integrators List
- **Improving the Test Program – Continue to Climb**





- First NVMe Plugfest and Current IL Qualifications established baseline.
- We are making the testing program more rigorous on both the interop and conformance fronts.



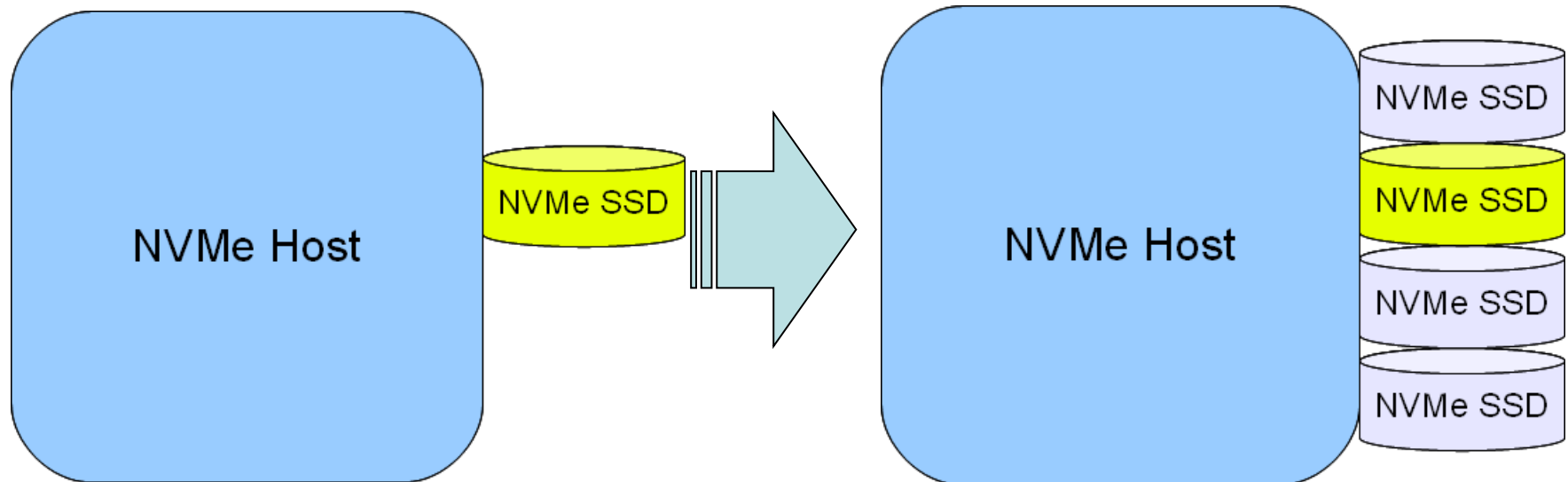


- Continue to Climb—How?
 - Make the Interop Tests more Rigorous
 - Raise the IL Qualification requirements
 - Improve the Tools



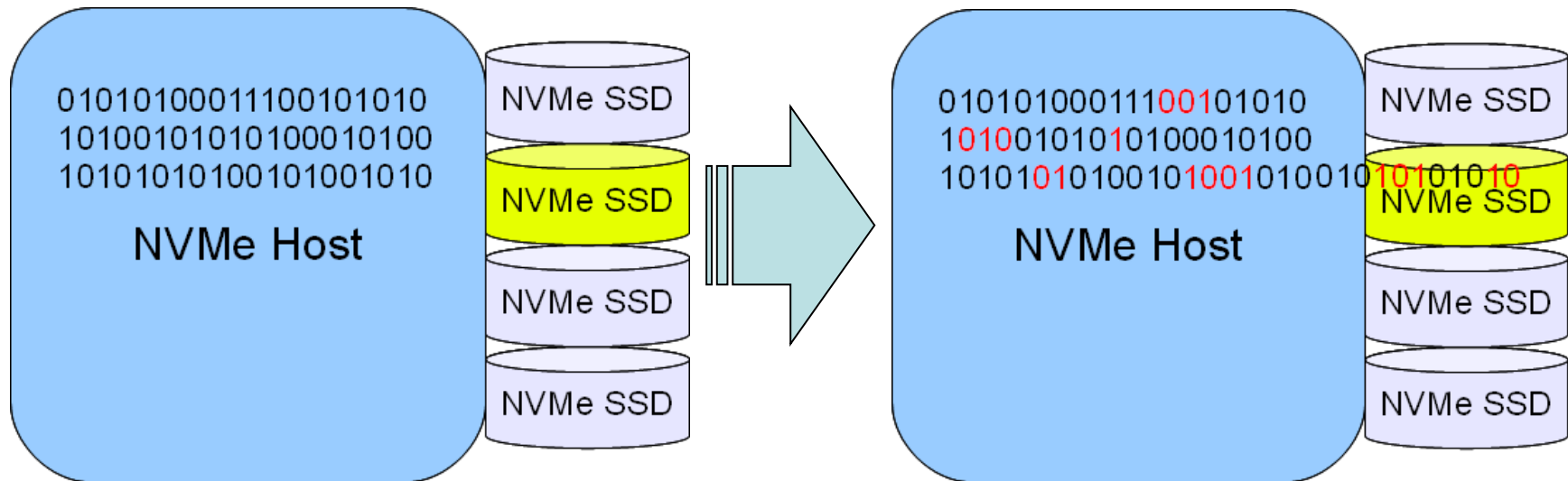


- Make the Interop Tests more Rigorous:
 - multiple SSDs in the system



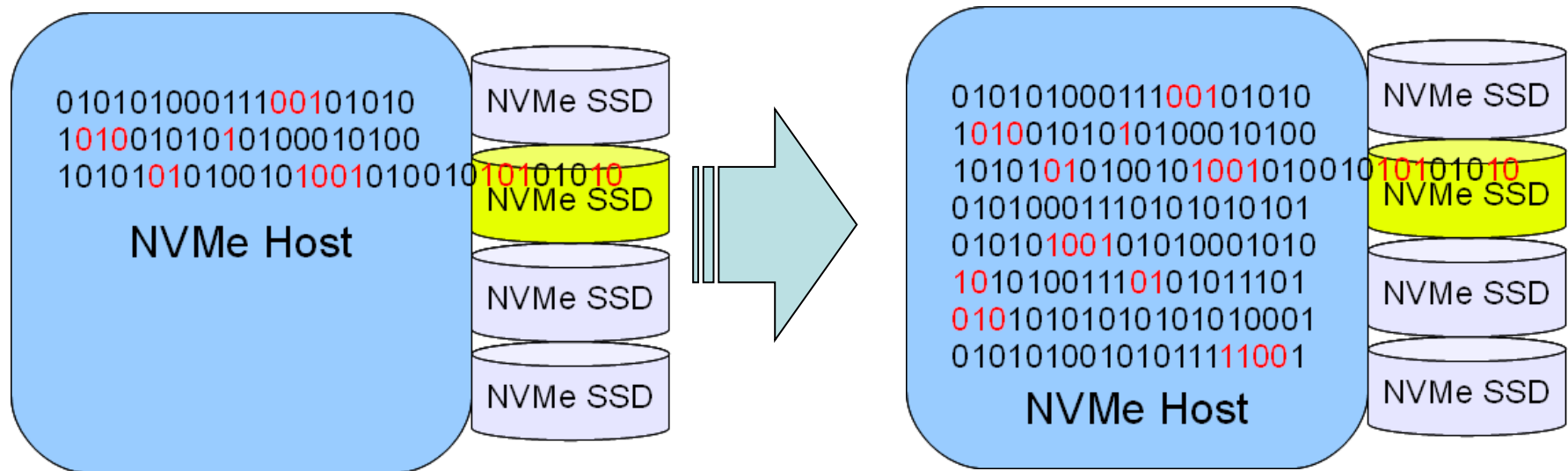


- Make the Interop Tests more Rigorous
 - stressing patterns



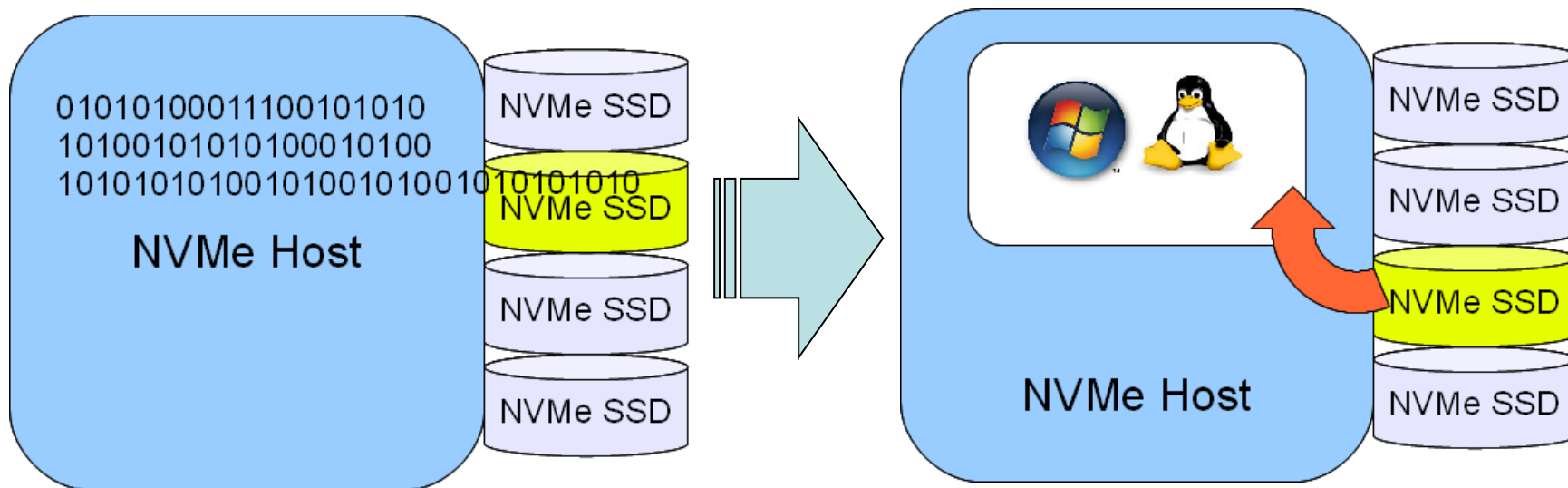


- Make the Interop Tests more Rigorous:
 - larger/varying transfer sizes





- Make the Interop Tests more Rigorous:
 - boot tests, rather than just data transfers





- Raise the IL Qualification Requirements:

| Today | Tomorrow |
|---|--|
| Test with 4 products, pass with 3 | Test with 5 products pass with 4 |
| Pass with Windows or Linux reference driver | Pass with Windows and Linux reference driver |



- Improve the Tools:
 - Add MSI-X support to Summit scripts.
 - Already undertaken by UNH-IOL and distributed to some members.





- Improve the Tools:
 - Improve OS support (update to Ubuntu 13.04) for IOL INTERACT.
 - Bug Fixes:
 - Set NSID to FFFFFFFFh
 - Correct AB value





- Make the Interop Tests More Rigorous
 1. Multiple Devices in the System
 2. Stressing Patterns
 3. Larger/Varying Transfer Sizes
 4. Boot from NVMe
- Raise the IL Qualification Requirements:
 - Require these more rigorous tests to be performed against more interop partners.
- Improve the tools
 - MSI-X, better OS Support, bug fixes



- These improvements to be implemented by UNH-IOL prior to the next NVMe Plugfest:
Planning Q4 2013
@ UNH-IOL Durham, NH
- UNH-IOL will continue to work with member companies to add products to the Integrators List outside of plugfest events.

NVMe in End User Computing (EUC) Dell's Vision

Munif Farhan

Senior Principle Engineer – (EUC)
CTO office Dell Inc.

- Dell Storage vision in End User Computing (EUC)
- NVMe application examples
- EUC NVMe adaption
- Keys for success
- Challenges & opens

Storage in End User Compute

Background

- Digital content growth across multiple users & devices.
- Smart phone mobility behaviors penetrating traditional End User Compute
- Remote data service as one of primary storage solutions

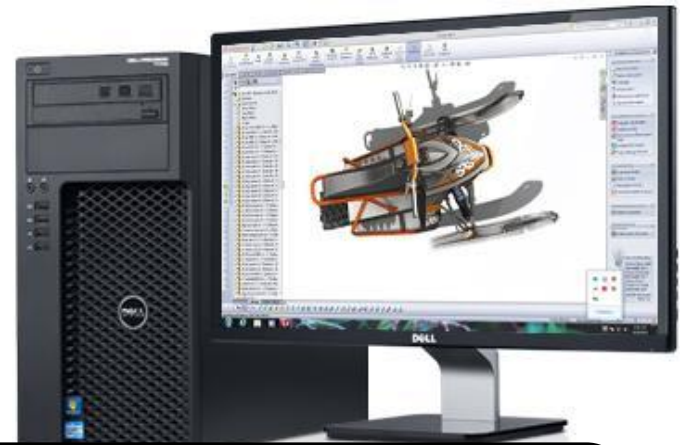
Vision

- Data storage baseline experiences:
 - Lowest \$/GB
 - A blend of performance and capacity
 - Best in class performance , form factor & power
- Define Dell differentiated experiences at the “Usage/Application” level through data management

NVMe Application in Dell EUC Examples

Professional Applications

- ✓ Low Latency media creation/editing
- ✓ Large data set file manipulation –CAD, Simulation,...



Gaming

- ✓ Fast loading times
- ✓ “Next level” transitions

Compelling overall value prop: the next step in storage performance!

- ✓ Best in class performance
- ✓ Feature equivalency with SATA devices
- ✓ Platform for system level differentiation

Dell EUC NVMe Adoption

Adaption plans: Seamless transition

- ✓ Minimum impact to baseline platforms design & cost
- ✓ Interchangeable with ALL existing SATA devices

| | |
|--------------|--|
| Tablets | BGA (Primary Storage) x2 M.2 (Primary Storage) |
| Ultra Mobile | 2.5" PCIe (Primary Storage) x2 M.2 (Primary Storage, Cache) |
| Notebooks | 2.5" PCIe (Primary Storage) x2 M.2 (Cache) |
| Desktops | 2.5" PCIe (Primary Storage) x2 M.2 (Cache) |

Keys For Success

Eco System enablement

- Chipset support & validation
- Interface & lane flexibility & availability
- One system image to support all SATA and PCIe Storage devices

Industry standards

- Connector/cable definition
- Clocks
- Low power
- Ultra small form factors?

Challenges & Opens

- Cost premium on host connector for broad adoption
- One cable and host connector for SATA and PCIe storage devices
- Support SRIS (Separate Refclk Independent SSC)
 - Cables and notebook, no REFCLK.
 - Cable definition and impact on shielding
- 8639 connector adaption is challenging in Dell Client system
 - Must maintain plugin compatibility with SATA devices

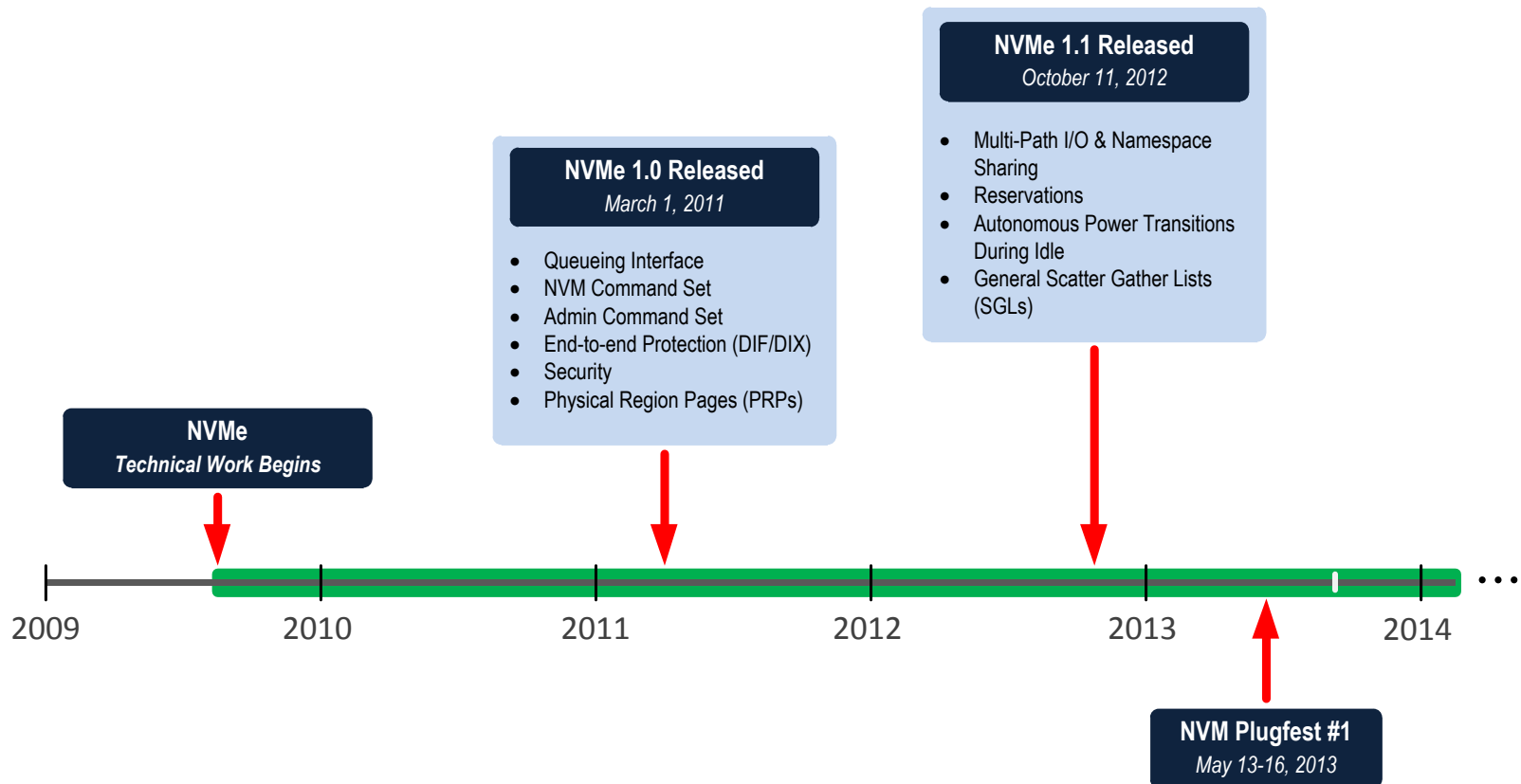


What's New in NVMe 1.1 and Future Directions

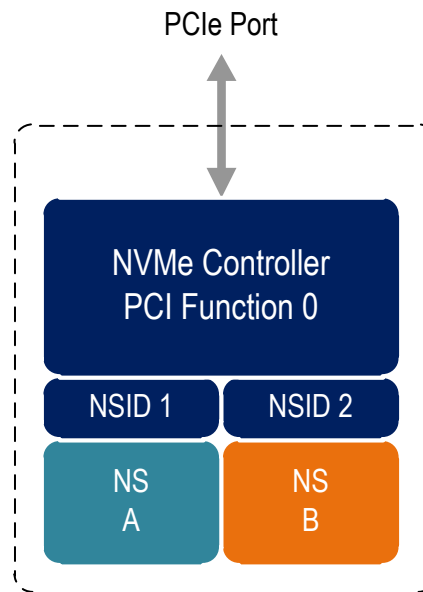
Peter Onufryk
Sr. Director, Product Development
PMC-Sierra



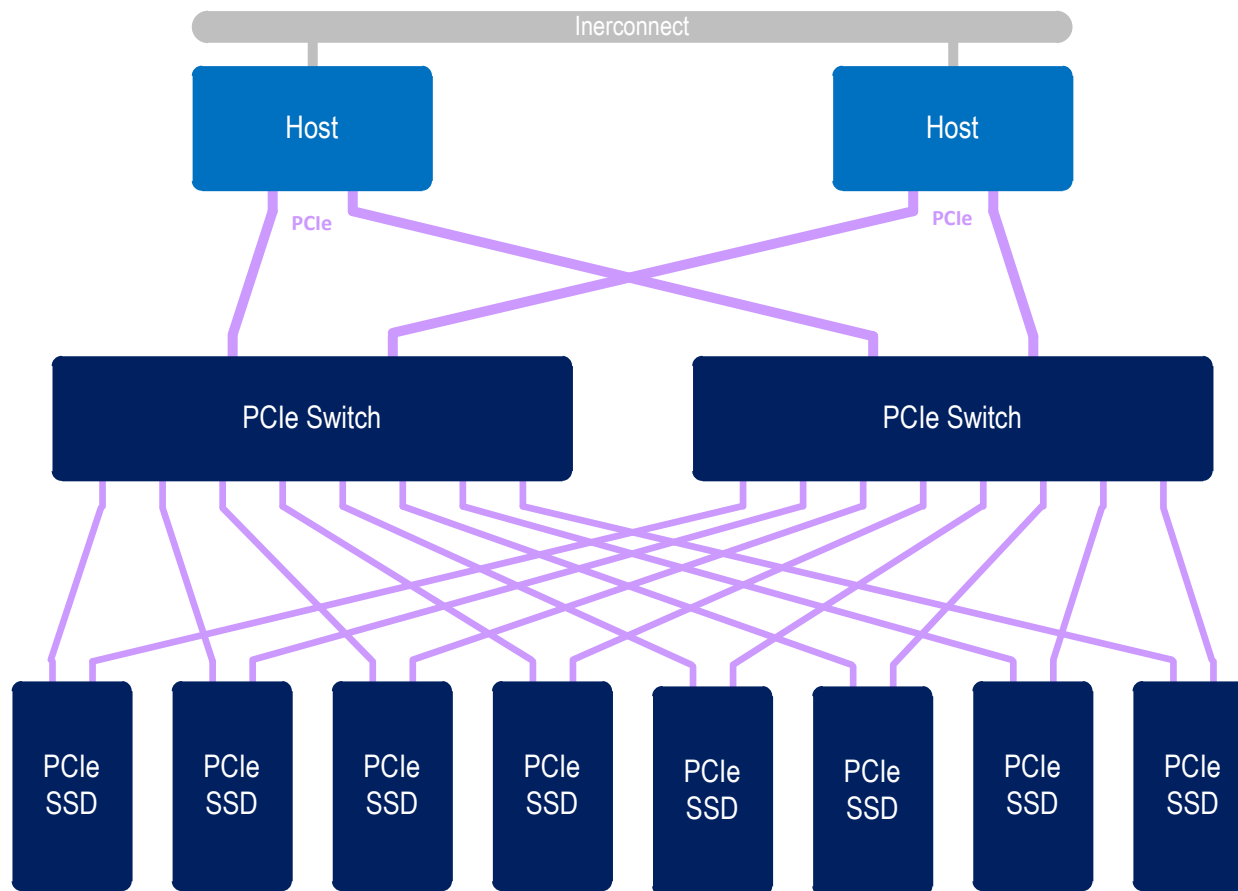
NVMe Development Timeline



Architectural Model of NVMe Controller

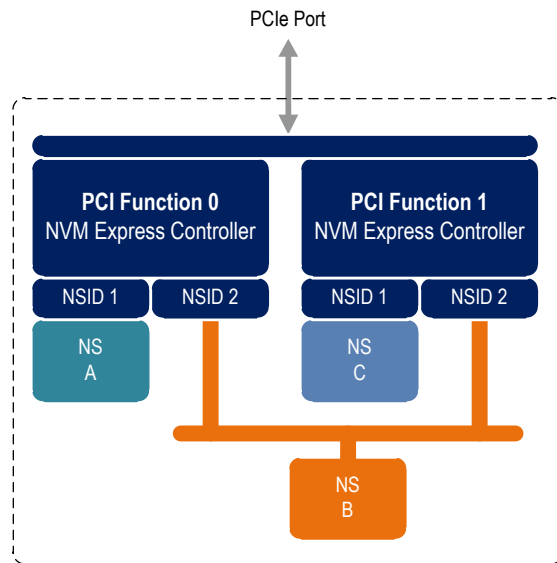


PCIe Multi-Path Usage Model

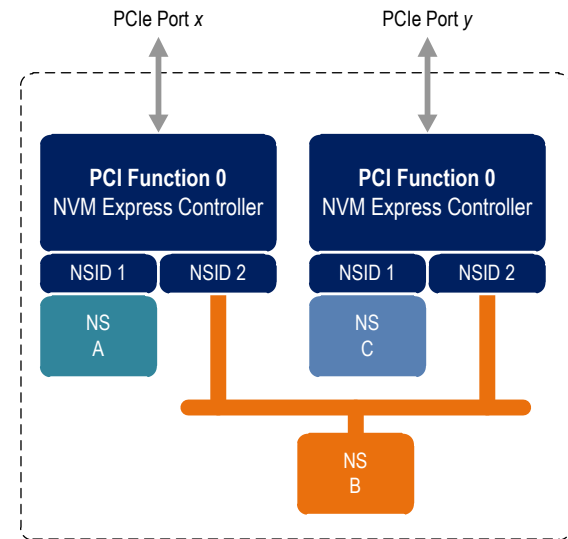


Multi-Path I/O and Namespace Sharing

- **NVM Subsystem** - one or more controllers, one or more namespaces, one or more PCI Express ports, a non-volatile memory storage medium, and an interface between the controller(s) and non-volatile memory storage medium

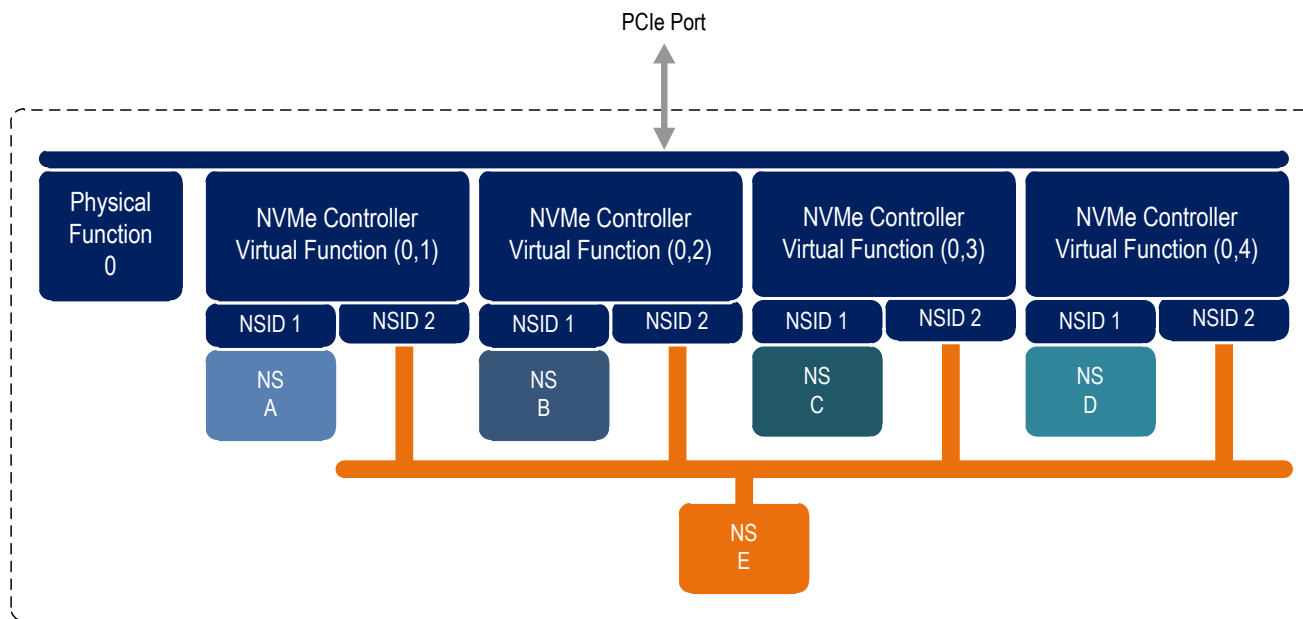


NVM Subsystem with Two Controllers and One Port



NVM Subsystem with Two Controllers and Two Ports

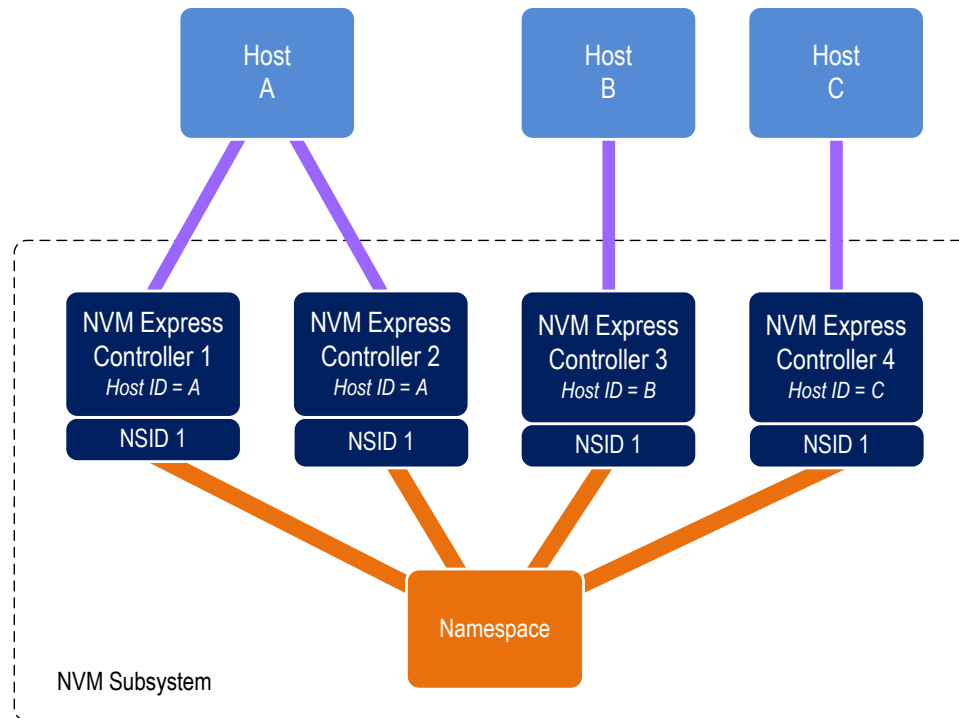
PCI Express SR-IOV



Reservation Overview

- Reservations allow two or more hosts to provide coordinate access to a shared namespace
- Reservations are on a namespace
- Reservations are used to restrict access to a namespace
- Capabilities are provided to allow recovery from a reservation held by a failing or uncooperative host

Example Multi-Host System



Host Identifier (Host ID) preserves reservation properties across all controllers associated with same host

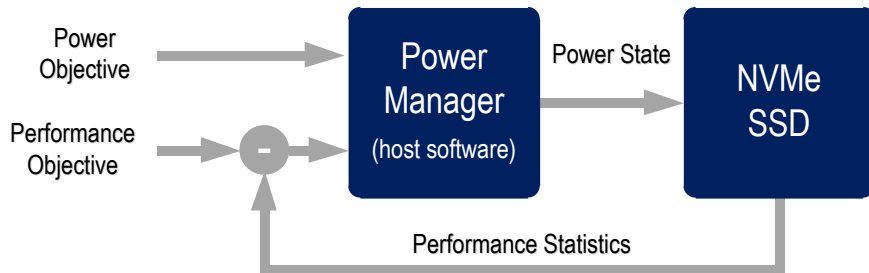
New NVM Reservation Commands

| NVM I/O Command | Operation |
|----------------------|---|
| Reservation Register | <ul style="list-style-type: none"> • Register a reservation key • Unregister a reservation key • Replace a reservation key |
| Reservation Acquire | <ul style="list-style-type: none"> • Acquire a reservation on a namespace • Preempt reservation held on a namespace • Preempt and abort a reservation held on a namespace |
| Reservation Release | <ul style="list-style-type: none"> • Release a reservation held on a namespace • Clear a reservation held on a namespace |
| Reservation Report | <ul style="list-style-type: none"> • Retrieve reservation status data structure <ul style="list-style-type: none"> Type of reservation held on the namespace (if any) Persist through power loss state Reservation status, Host ID, reservation key for each host that has access to the namespace |

Command Behavior In Presence of a Reservation

| Reservation Type | Reservation Holder | | Registrant | | Non-Registrant | | Reservation Holder Definition |
|-------------------------------------|--------------------|-------|------------|-------|----------------|-------|---|
| | Read | Write | Read | Write | Read | Write | |
| Write Exclusive | Y | Y | Y | N | Y | N | One Reservation Holder |
| Exclusive Access | Y | Y | N | N | N | N | One Reservation Holder |
| Write Exclusive - Registrants Only | Y | Y | Y | Y | Y | N | One Reservation Holder |
| Exclusive Access - Registrants Only | Y | Y | Y | Y | N | N | One Reservation Holder |
| Write Exclusive - All Registrants | Y | Y | Y | Y | Y | N | All Registrants are Reservation Holders |
| Exclusive Access - All Registrants | Y | Y | Y | Y | N | N | All Registrants are Reservation Holders |

NVMe Power Management



Power State Descriptor Table

| Power State | Maximum Power | Operational State | Entry Latency | Exit Latency | Relative Read Throughput | Relative Read Latency | Relative Write Throughput | Relative Write Latency |
|-------------|---------------|-------------------|---------------|--------------|--------------------------|-----------------------|---------------------------|------------------------|
| 0 | 25 W | Yes | 5 μ s | 5 μ s | 0 | 0 | 0 | 0 |
| 1 | 18 W | Yes | 5 μ s | 7 μ s | 0 | 0 | 1 | 0 |
| 2 | 18 W | Yes | 5 μ s | 8 μ s | 1 | 0 | 0 | 0 |
| 3 | 15 W | Yes | 20 μ s | 15 μ s | 2 | 1 | 2 | 1 |
| 4 | 7 W | Yes | 20 μ s | 30 μ s | 1 | 2 | 3 | 1 |
| 5 | 1 W | No | 100 mS | 50 mS | - | - | - | - |
| 6 | .25 W | No | 100 mS | 500 mS | - | - | - | - |

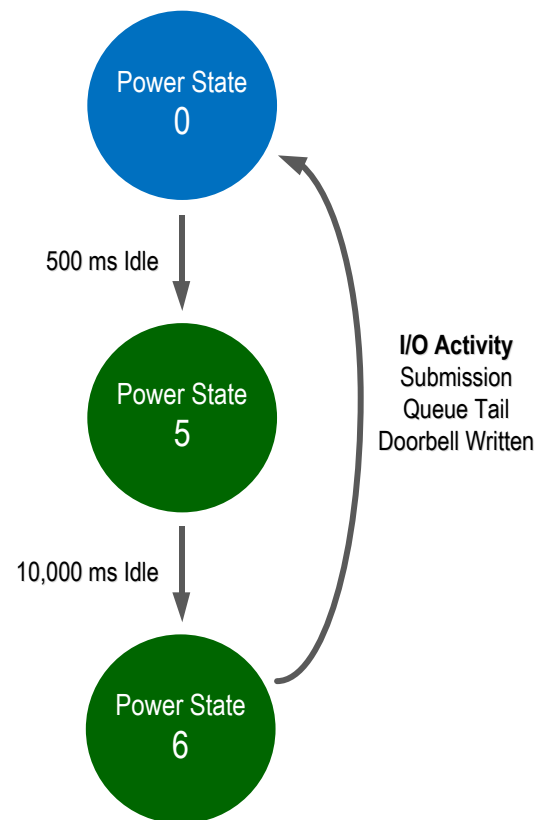
Autonomous Power State Transitions

Power State Descriptor Table

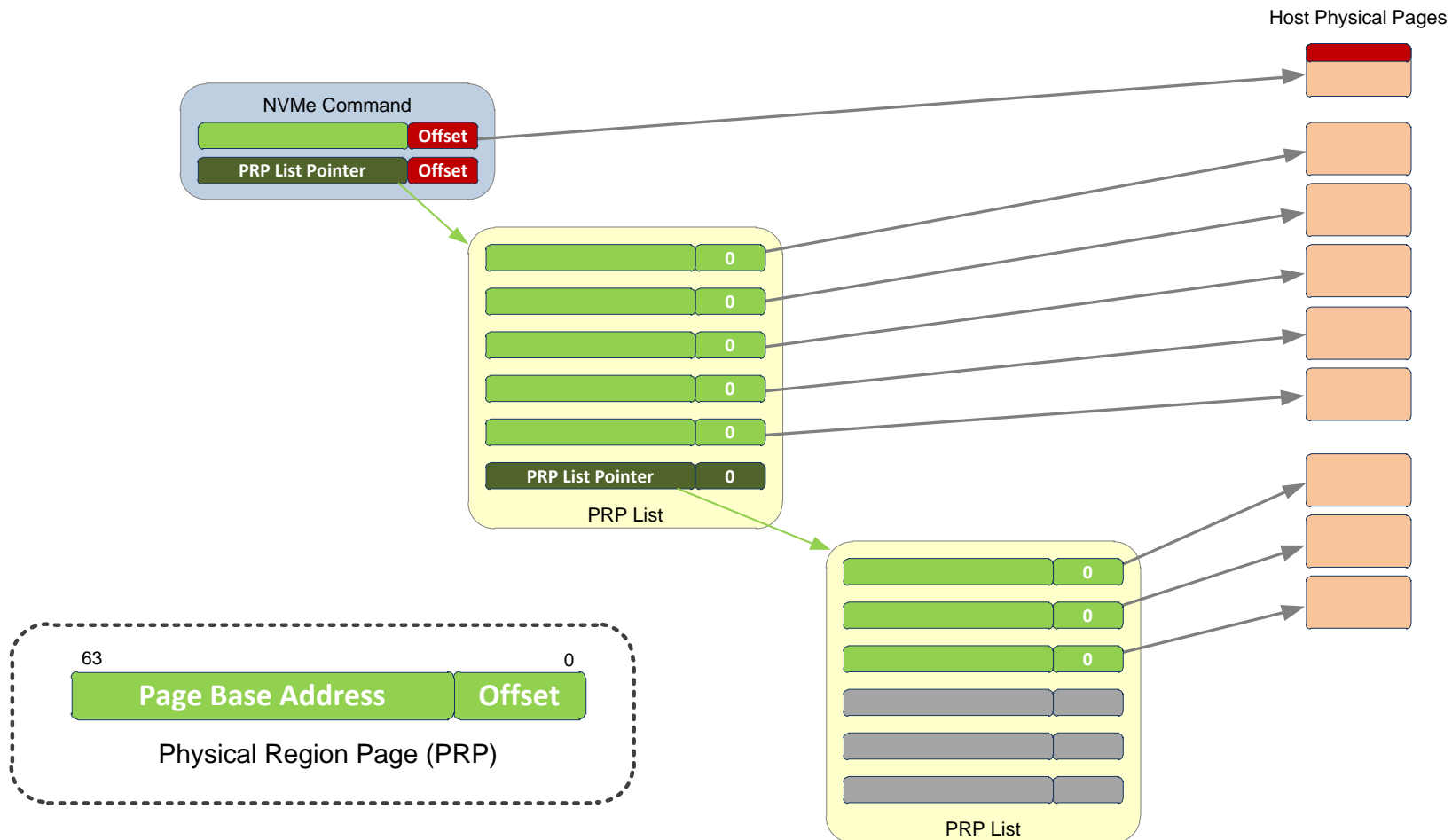
| Power State | Maximum Power | Operational State | Entry Latency | Exit Latency |
|-------------|---------------|-------------------|---------------|--------------|
| 0 | 25 W | Yes | 5 μ s | 5 μ s |
| 1 | 18 W | Yes | 5 μ s | 7 μ s |
| 2 | 18 W | Yes | 5 μ s | 8 μ s |
| 3 | 15 W | Yes | 20 μ s | 15 μ s |
| 4 | 7 W | Yes | 20 μ s | 30 μ s |
| 5 | 1 W | No | 100 mS | 50 mS |
| 6 | .25 W | No | 100 mS | 500 mS |

Autonomous Power State Transition Table

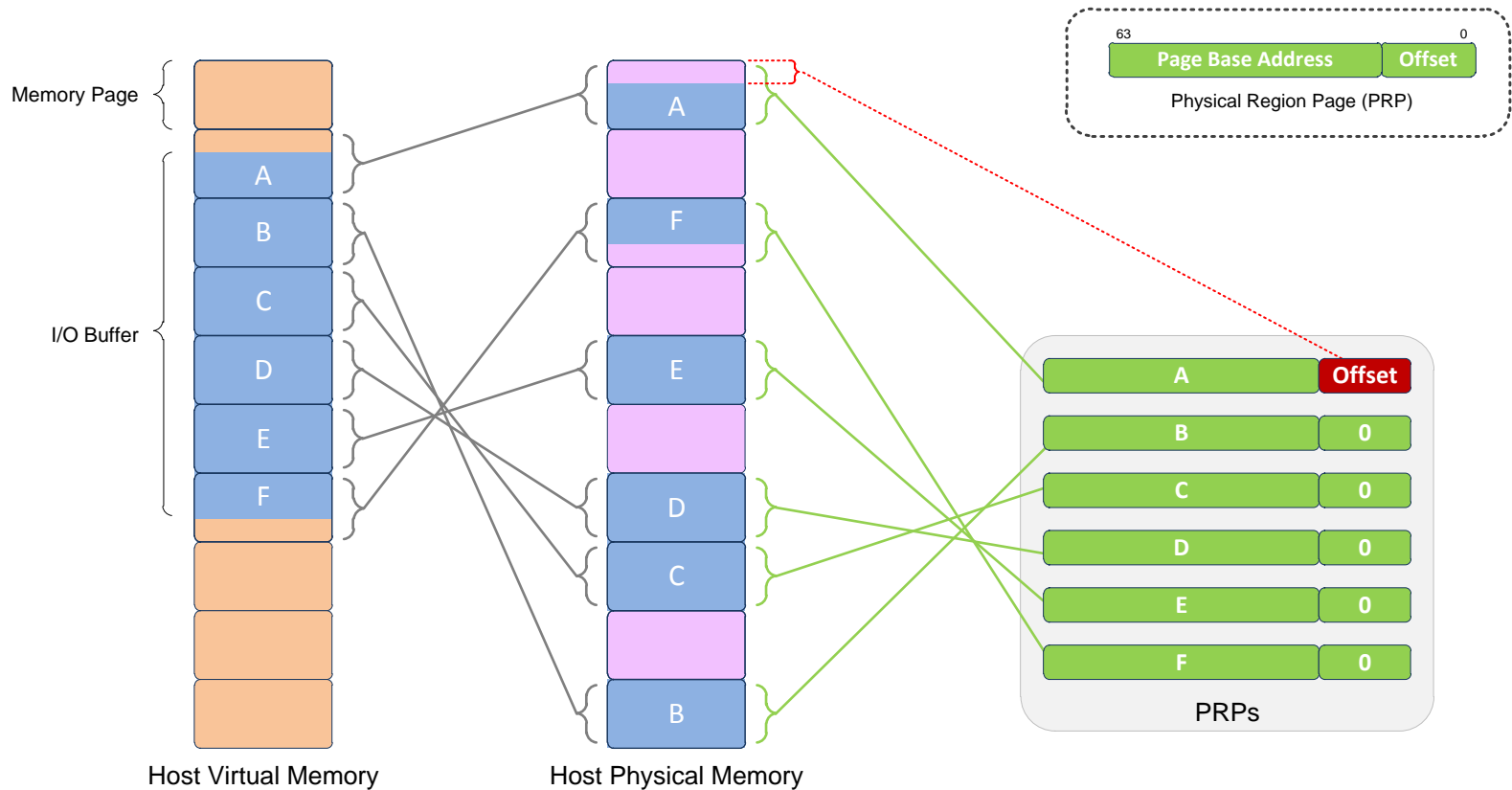
| Idle Time Prior to Transition | Idle Transition Power State |
|-------------------------------|-----------------------------|
| 500 ms | 5 |
| 500 ms | 5 |
| 500 ms | 5 |
| 500 ms | 5 |
| 500 ms | 5 |
| 10,000 ms | 6 |
| - | - |



Physical Region Pages (PRPs)

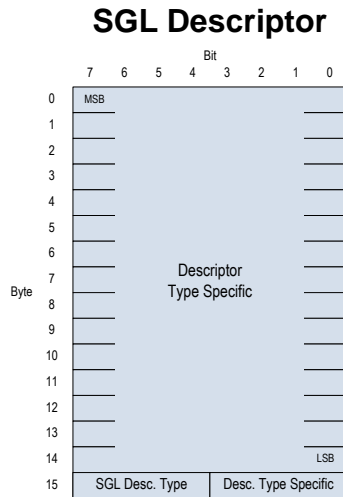


Why PRPs?

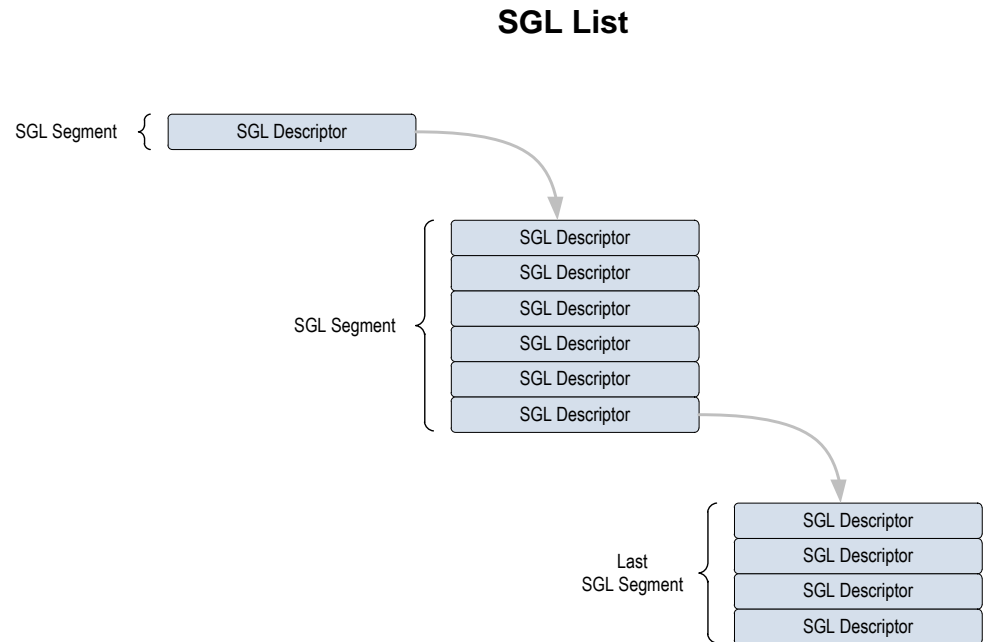


Fixed Size PRPs Accelerate Out of Order Data Delivery

Scatter Gather List (SGLs)

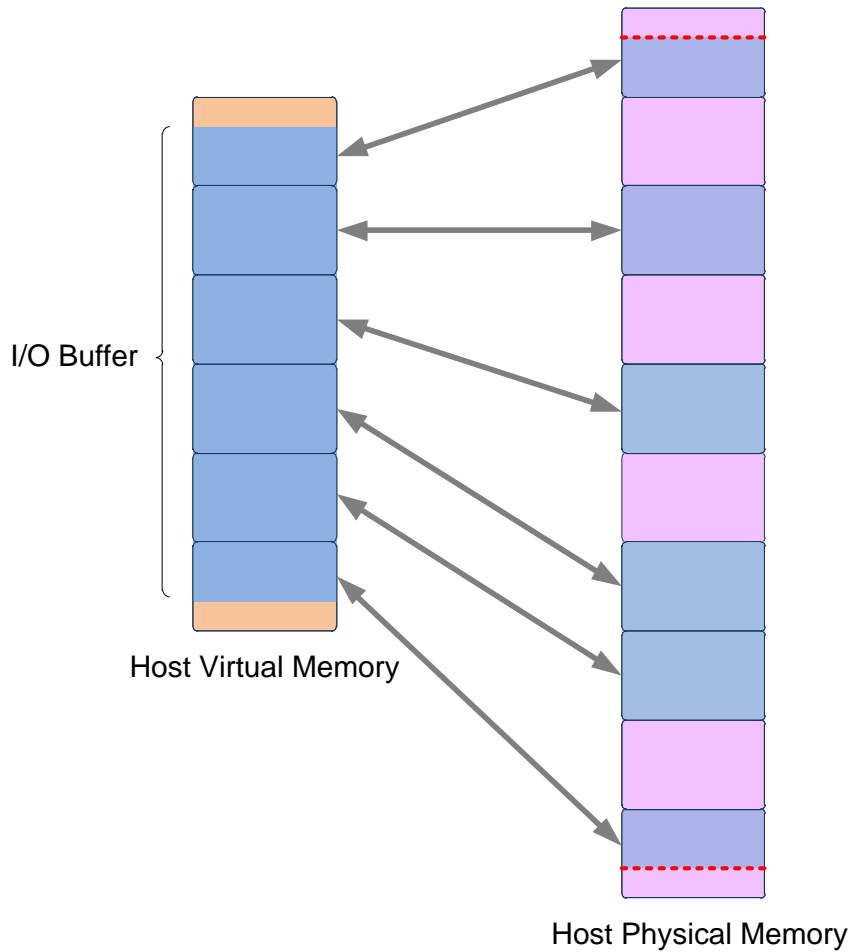


| Code | SGL Descriptor Type |
|---------|---------------------|
| 0h | SGL Data Block |
| 1h | SGL Bit Bucket |
| 2h | SGL Segment |
| 3h | SGL Last Segment |
| 4h - Eh | Reserved |
| Fh | Vendor Specific |

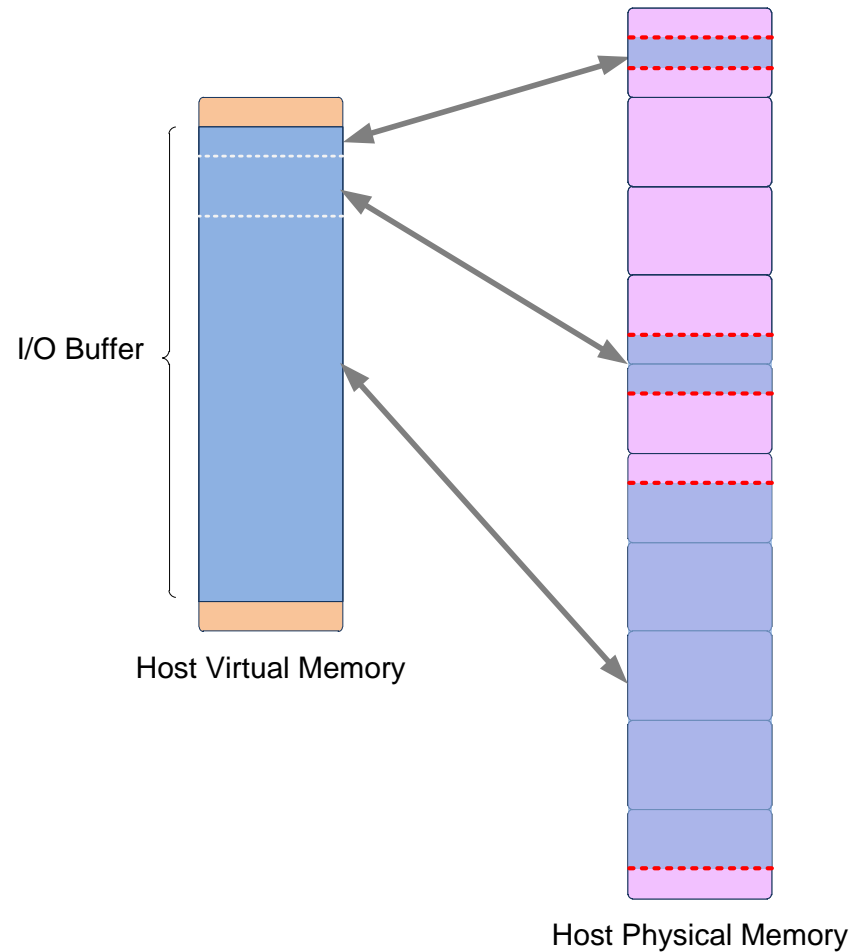


SGLs Enable Arbitrary Data Transfer Size and Byte Alignment

Comparing SGLs with PRPs



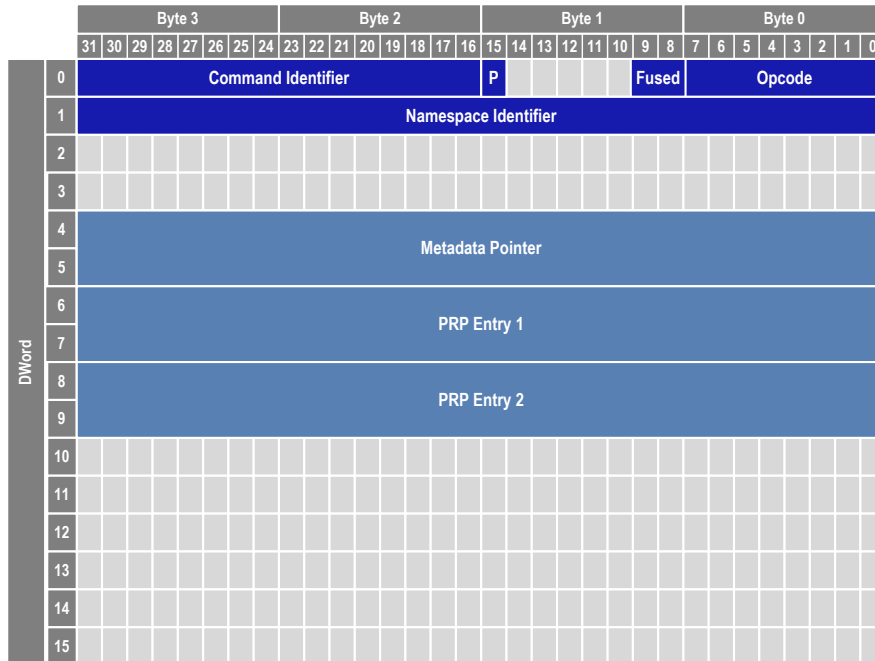
PRP Data Transfer



SGL Data Transfer

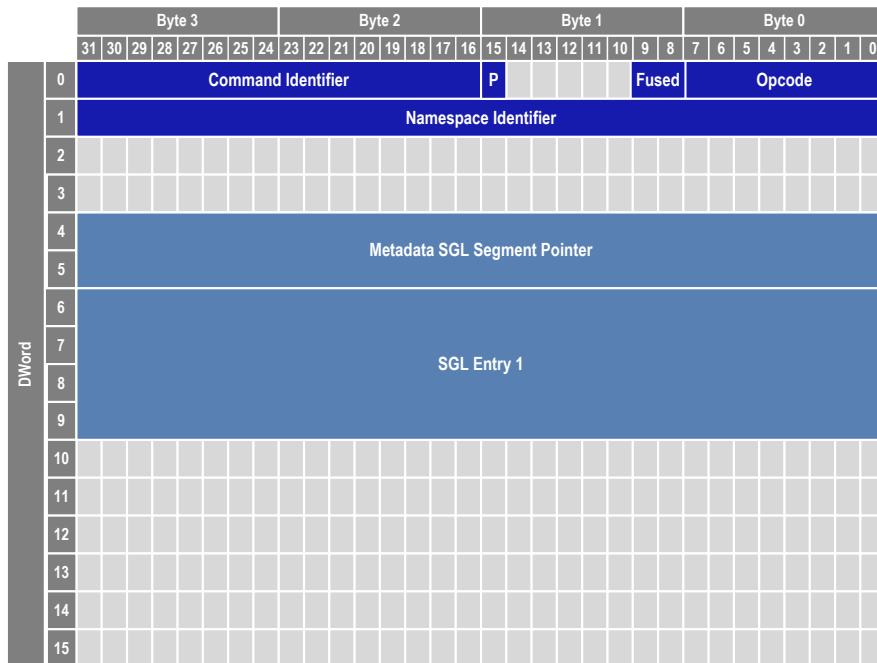


Command Format (PRP)



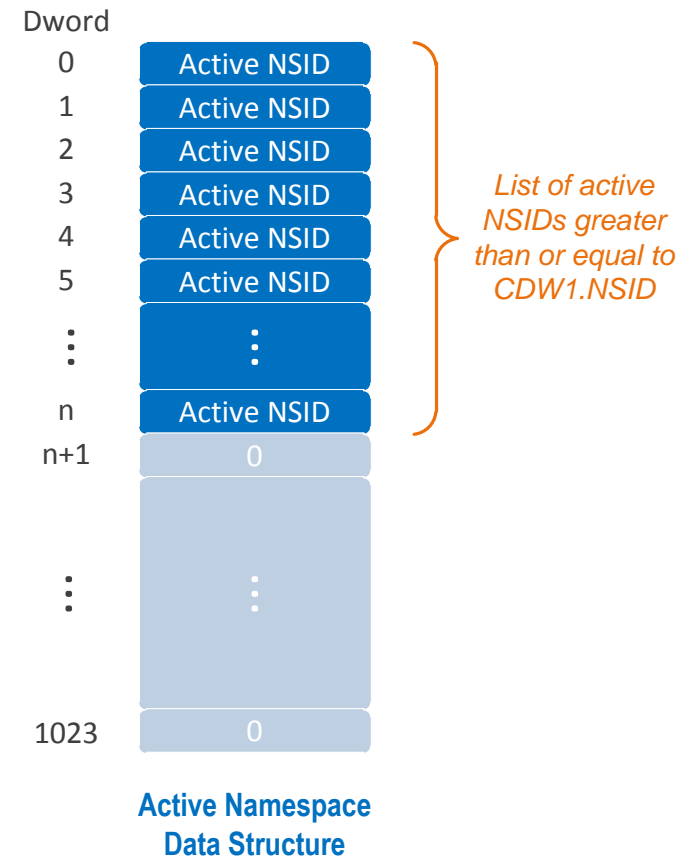
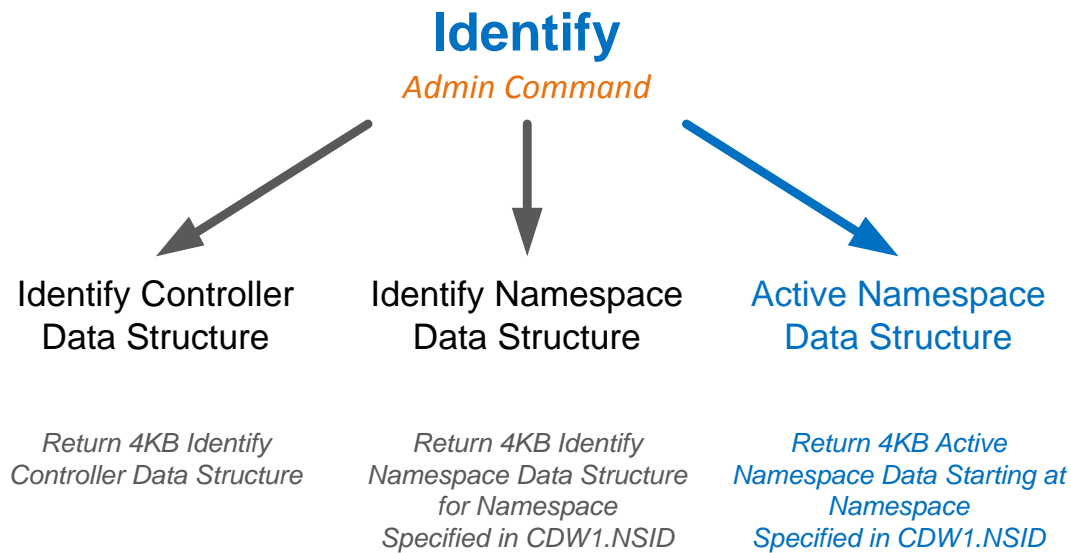
- **Opcode** - Command to execute
- **Fused** – Indicates two commands should be executed as atomic unit
- **P** - Use PRPs or SGLs for data transfer
- **Command Identifier** - Unique ID associated with command
- **Namespace Identifier** - Namespace on which command operates
- **Metadata Pointer** – Pointer to contiguous buffer containing metadata in “DIX” mode
- **PRP Entry 1 & 2** – PRP or PRP list

Command Format (SGL)



- **Opcode** - Command to execute
- **Fused** – Indicates two simpler commands should be executed as atomic unit
- **P** - Use PRPs or SGLs for data transfer
- **Command Identifier** - Unique ID associated with command
- **Namespace Identifier** - Namespace on which command operates
- **Metadata SGL Segment Pointer** – Pointer to metadata SGL segment in “DIX” mode
- **SGL Entry 1** – First SGL segment associated with data transfer

Active Namespace Reporting



Other New Features

- Write Zeros Command
- Subsystem Reset
- Persistent Features Across Power States
- Atomic Compare and Write Unit

NVMe 1.1 New Commands

Admin Commands

| |
|---|
| Create I/O Submission Queue |
| Delete I/O Submission Queue |
| Create I/O Completion Queue |
| Delete I/O Completion Queue |
| Get Log Page |
| Identify |
| Abort |
| Set Features |
| Get Features |
| Asynchronous Event Request |
| <i>Firmware Activate (optional)</i> |
| <i>Firmware Image Download (optional)</i> |

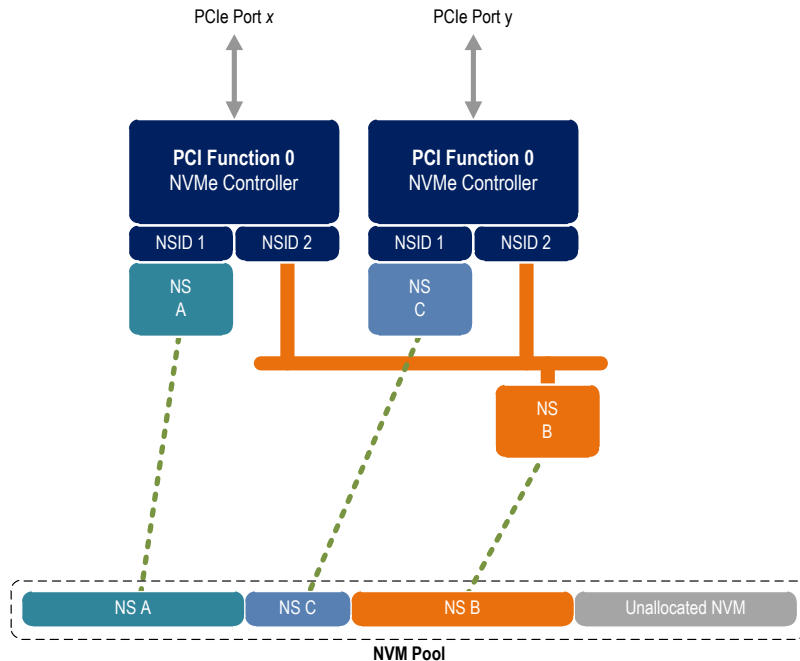
NVM Admin Commands

| |
|------------------------------------|
| <i>Format NVM (optional)</i> |
| <i>Security Send (optional)</i> |
| <i>Security Receive (optional)</i> |

NVM I/O Commands

| |
|---|
| Read |
| Write |
| Flush |
| <i>Write Uncorrectable (optional)</i> |
| <i>Compare (optional)</i> |
| <i>Dataset Management (optional)</i> |
| <i>Write Zeros (optional)</i> |
| <i>Reservation Register (optional)</i> |
| <i>Reservation Report (optional)</i> |
| <i>Reservation Acquire (optional)</i> |
| <i>Reservation Release (optional)</i> |

Future Direction Namespace Management



- Ability to create, resize (larger or small), and delete a namespace
- Ability to attach or detach a namespace to/from a specific controller in the NVM subsystem
- Namespace and NVM pool status reporting

Figure 84: Identify – Identify Namespace Data Structure, NVM Command Set Specific

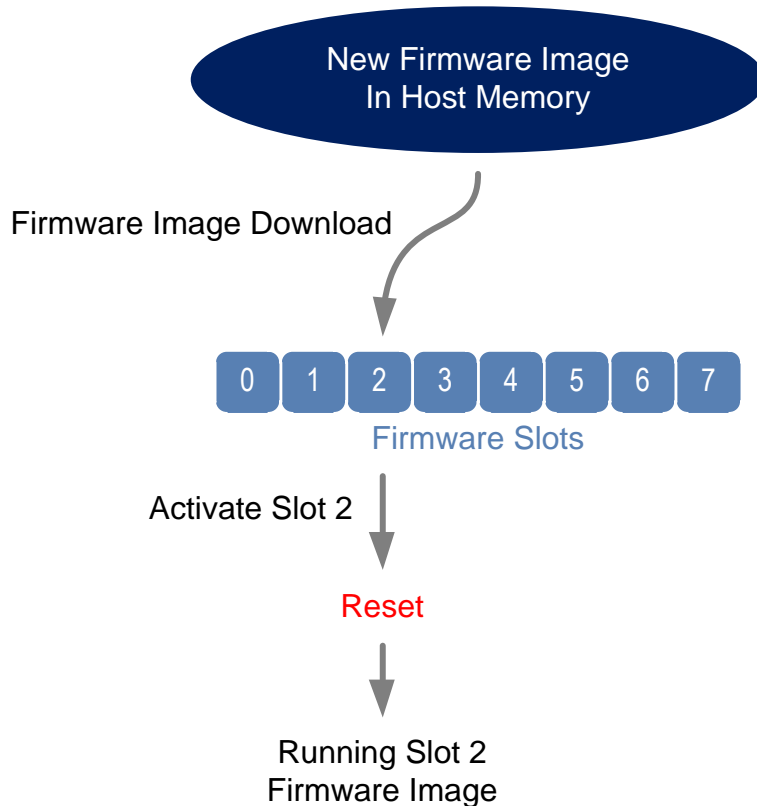
| Bytes | O/M | Description |
|-------|-----|--|
| 7:0 | M | <p>Namespace Size (NSZE): This field indicates the total size of the namespace in logical blocks. A namespace of size n consists of LBA 0 through $(n - 1)$. The number of logical blocks is based on the formatted LBA size. This field is undefined prior to the namespace being formatted.</p> <p>Note: The creation of the namespace(s) and initial format operation are outside the scope of this specification.</p> |
| 15:8 | M | <p>Namespace Capacity (NCAP): This field indicates the maximum number of logical blocks that may be allocated in the namespace at any point in time. The number of logical blocks is based on the formatted LBA size. This field is undefined prior to the namespace being formatted. This field is used in the case of thin provisioning and reports a value that is smaller than or equal to the Namespace Size. Spare LBAs are not reported as part of this field.</p> <p>A value of 0h for the Namespace Capacity indicates that the namespace ID is an inactive namespace ID.</p> <p>A logical block is allocated when it is written with a Write or Write Uncorrectable command. A logical block may be deallocated using the Dataset Management command.</p> |
| 23:16 | M | <p>Namespace Utilization (NUSE): This field indicates the current number of logical blocks allocated in the namespace. This field is smaller than or equal to the Namespace Capacity. The number of logical blocks is based on the formatted LBA size.</p> <p>When using the NVM command set: A logical block is allocated when it is written with a Write or Write Uncorrectable command. A logical block may be deallocated using the Dataset Management command.</p> |
| 24 | M | <p>Namespace Features (NSFEAT): This field defines features of the namespace.</p> <p>Bits 7:1 are reserved.</p> <p>Bit 0 if set to '1' indicates that the namespace supports thin provisioning. Specifically, the Namespace Capacity reported may be less than the Namespace Size. When this feature is supported and the Dataset Management command is supported then deallocating LBAs shall be reflected in the Namespace Utilization field. Bit 0 if cleared to '0' indicates that thin provisioning is not supported and the Namespace Size and Namespace Capacity fields report the same value.</p> |
| 25 | M | <p>Number of LBA Formats (NLBAF): This field defines the number of supported LBA data size and metadata size combinations supported by the namespace. LBA formats shall be allocated in order (starting with 0) and packed sequentially. This is a 0's based value. The maximum number of LBA formats that may be indicated as supported is 16. The supported LBA formats are indicated in bytes 128 – 191 in this data structure.</p> <p>The metadata may be either transferred as part of the LBA (creating an extended LBA which is a larger LBA size that is exposed to the application) or it may be transferred as a separate contiguous buffer of data. The metadata shall not be split between the LBA and a separate metadata buffer.</p> |

...

- Optionally generate asynchronous event when certain fields in the Identify Namespace data structure change
 - Log page indicates which namespaces are affected
- May be used by host to determine when a namespace change has occurred

Future Direction

Firmware Activation Without Reset



- Current firmware update process
 - Download firmware image to firmware slot
 - Activate firmware image
 - Perform reset to cause new firmware image to run
 - Controller reset
 - PCIe conventional reset
 - Subsystem reset
- Enhance firmware update process to allow new firmware image to run without a reset



Future Direction Other Enhancements

- More power management enhancements
- SGL enhancements
- Enhanced error reporting

Summary

- NVMe 1.1 adds enhancements for client and enterprise applications
- NVMe 1.1 enhancements maintain backward compatibility
- Future enhancements are planned for NVMe
- NVMe continues to maintain core philosophy of simplicity and efficiency