The State of NVMe® Interoperability
David Woolf – UNH-IOL

Sponsored by NVM Express organization, the owner of NVMe®, NVMe-oF™ and NVMe-MI™ standards
Speakers

David Woolf

University of New Hampshire
InterOperability Laboratory
Agenda

• Architecting for Interoperability
• Prioritizing Interoperability
• Enabling Interoperability
• Observations at the Lab
• Ensuring Future Interop
Architected for Interoperability

NVMe® technology is the language of storage

- Agnostic to underlying media
- Different Flash Memory types
Architected for Interoperability

NVMe® technology is the language of storage

- Agnostic to underlying transport
- Leverage iterations in transport
  - PCIe 3.0, 4.0, 5.0
  - Ethernet 100G, 200G, 400G
  - FC 32GFC, 64GFC, 128GFC
Architected for Interoperability

Flexible and Interoperable

• SSDs can be designed for very different use cases

• SKUs with varying cost, capacity, media, form factor, are interoperable
Prioritizing Interoperability

Spec development process scrutinizes and reviews new features for interoperability.

1. Specification Developed and Released
2. Proposal for New Feature Introduced
3. Is it backwards compatible?
4. Tests Developed for new features
5. Tests feedback into specification
Prioritizing Interoperability

Testing Program

Program has Interop and Compliance Components

Test program is updated twice a year to align with additions and modifications to the specification.

Interop Testing

- checking functionality with different OS and HW platforms

Compliance Testing

- 250+ tests checking adherence to specification requirements
- Tests run according to products claimed specification version support
Prioritizing Interoperability

Integrators List Testing

100’s of products listed on NVMe Integrators list after passing a set of interop and compliance tests
Enabling Interoperability

Common Toolsets
- nvmecli
- INTERACT Compliance Test Tool used by UNH-IOL Can be included in in-house development process and in regression tests.

Tight coupling of driver development and specification development
- Key contributors to open source NVMe® SSD driver participate in specification development
- NVMe Organization coordinated initial efforts to create Fabrics drivers

Enabling Interoperability

Review of most recent updates to Compliance Program for November 2020

- Align to NVMe-MI™ 1.1 specification
- Align to NVMe-oF™ 1.1 specification
- Align to NVMe® 1.4 specification ECNs and TPs
  - 42 PCIe DUT Tests FYI>Mandatory
  - 26 Fabric DUT Tests FYI>Mandatory
  - Sanitize
  - UUID List and CNS 17h
  - Endurance Groups
  - Telemetry enhancements
Enabling Interoperability

Align to NVMe-MI™ 1.1 specification

- Management Endpoint Buffer Read/Write and SES Send/Receive Commands
- Command Initiated Auto Pause requirements
- Topology Multirecord requirements
- PCIe Port Numbering
- Test Cases added to address VPD Read/Write requirements.
- Test Case added address Identify, Get Log, and Get Features Commands over NVMe-MI architecture for NVMe® Storage Device
- Test Cases added to address handling of NVMe Admin commands prohibited for NVMe Enclosures
- Test Cases added to address handling of NVMe-MI commands during a Sanitize operation
Enabling Interoperability

**Align to NVMe-oF™ 1.1 specification**

- Test cases added to address NVMe-oF v1.0 ECN 001, 002, 005 requirements
- When controller is enabled, it shall accept all supported Admin commands in addition to Fabrics commands
- host and controller association is preserved for at least 2 minutes after CC.EN transitions to ‘0’.
- requirements around accepting commands on channels that are not yet authenticated/secure.
- if a Fabrics command that is not supported on an I/O Queue is sent on an I/O Queue, that command shall be aborted with a status code of Invalid Field in Command.
- requirements around Controller ID of FFFFh.
- requirements around shutdown and the CC.SHN field.
- requirements for if the controller detects an NVMe Transport connection loss
- requirements around the Disconnect command
- checking for AER and Keep Alive support when explicit persistent connections are supported.
- Check the error condition when persistent connections are requested and the DUT does not support persistent connections.
- Check for SQ Flow Control Negotiation, and proper behavior when SQ Flow Control is disabled.
Enabling Interoperability

Align to NVMe 1.4® specification ECN 001 and 002 and new features
- Test around the THINP bit and use of the Namespace Utilization field
- Tests around Telemetry Host-Initialed Data Generation Number increment
- Tests to check that Write Uncorrectable commands and Write Zeroes commands shall not impact the Data Units Written value.
- Updates around terminating DST operations with Format NVM
- Checking Proper use of the Sanitize Config command
- Checking when to mark a block as allocated when a Write, Write Uncorrectable, Write Zeroes commands.
- Tests around Endurance Group Enhancements
- Tests Read Recovery Levels
Observations at the Lab

PCIe 4.0 Architecture
• Works well with systems we have in lab
• Most PCIe issues are misconfigurations of purchased IP

NVMe Boot
• worked very well since UEFI support came

Hot Plug
• Not widely implemented prior to NVMe® architecture
• Great improvements over last 3 years, especially as enterprise and cloud use expands
• Expect this to carry over with EDSFF use case (Mandatory at November plugfest)

Open Source Driver
• Very few issues found
• Interop issues typically addressed in days/weeks
Ensuring an Interoperable Future

NVMe® technology continues to evolve and adapt, but interoperability will remain a key focus.

Areas that interop efforts will focus on in the coming year:

• Key Value Command Set
• Zoned Namespaces Command Set
• Multi-Domain Subsystems over Fabrics
• Computational Storage
• Specification Refactor
Conclusions

NVMe® technology is the language of storage

• NVMe technology is architected for interop
• NVMe community prioritizes interop
• NVMe resources enable interoperability
• The Future of NVMe technology is Interoperable!
Questions?