NVM Express
The Interface Standard for PCI Express SSDs

Kevin Marks
Principal Engineer / Technologist
Dell Inc.

Peter Onufryk
Director of Engineering
IDT
Benefits of PCIe as an SSD Interface

• PCIe is High Performance
  ▪ Full duplex, multiple outstanding requests, and out of order processing
  ▪ Scalable port width (x1 to x16)
  ▪ Scalable link speed (2.5 GTps, 5 GTps, 8 GTps)
  ▪ Low latency (no HBA overhead or protocol translation)

• PCIe is Low Cost
  ▪ High volume commodity interconnect
  ▪ Direct attach to CPU subsystem eliminates HBA cost

• PCIe Provides Effective Power Management
  ▪ Direct attach to CPU subsystem eliminates HBA power
  ▪ Link power management
  ▪ Optimized Buffer Flush/Fill (OBFF)
  ▪ Power Budgeting and Dynamic Power Allocation
  ▪ Slot Power Limit
NVM Express (NVMe)

- NVMe defines an optimized queuing interface, command set, and feature set for PCIe SSDs
  - Architected to scale from client to enterprise
- Standardization accelerates industry adoption
  - Standard drivers
    - Eliminates need for OEMs to qualify a driver for each SSD
    - Enables broad adoption across a wide range of industry standard and proprietary OSes
  - Consistent feature set
    - All SSDs implement required features
    - Optional features are implemented in a consistent manner
  - Industry ecosystem
    - Development tools
    - Compliance and interoperability testing
Example usage models for NVMe devices

<table>
<thead>
<tr>
<th>Server Caching</th>
<th>Server Storage</th>
<th>Client Storage</th>
<th>External Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
<td><img src="image3.png" alt="Diagram" /></td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- **Server Caching**
  - Used for temporary data
  - Non-redundant
  - Used to reduce memory footprint

- **Server Storage**
  - Typically for persistent data
  - Redundant (i.e., RAID’ed)
  - Commonly used as Tier-0 storage

- **Client Storage**
  - Used for Boot/OS drive and/or HDD cache
  - Non-redundant
  - Power optimized

- **External Storage**
  - Used for Metadata or data
  - Multi-ported device
  - Redundancy based on usage
“The integration of solid state storage technology will have a profound impact on computer architectures over the next few years. The NVMe Work Group, driving a standard interface for the industry’s most popular hardware building blocks, is ensuring that Solid State becomes a ubiquitous component of server and enterprise storage, enabling companies like NetApp to innovate at the system level and deliver unprecedented value to customers.”

David Dale
Director of Industry Standards
NetApp
The Value of NVMe

“EMC continues to be at the forefront of helping customers exploit the cost and performance benefits of SSDs, and is a strong supporter of standardization. Cloud and big data are prime examples of the drivers for insatiable performance demands customers are facing. Non-Volatile memory has the raw speed to meet these challenges, but requires an optimized interface across the industry to fully realize its potential. NVM Express can be that interface and EMC is pleased to help drive this important standard.”

Bill DePatie
Vice President of Hardware Engineering
EMC
NVM Express Specification

• NVM Express 1.0 completed on March 1, 2011
  ▪ Specification available at nvmexpress.org
• Specification cooperatively developed by more than 80 companies within the NVMe Work Group
  ▪ NVMe Work Group is directed by a multi-member Promoter Group of Companies consisting of Cisco, Dell, EMC, IDT, Intel, Micron, NetApp, Oracle, SandForce and STEC
NVMe Ecosystem

- **Products**
  - In development pipelines for 2012 releases

- **Drivers**
  - Linux driver available at [nvmexpress.org](http://nvmexpress.org)
  - Windows driver in development (IDT, Intel, and SandForce effort)

- **Development Tools**
  - LeCroy PCIe analyzers available with NVMe decode

- **Compliance and Interoperability Testing**

NVMe Promoter Group has begun working with the University Of New Hampshire IOL to create an interoperability test suite and implementers list.
NVMe Structure

NVM Express Specification

Queuing Interface

Command Set

Admin Command Set

I/O Command Sets

NVM Cmd Set

Rsvd #1

Rsvd #2

Rsvd #3
Queuing Interface

1. Queue Command
2. Ring Doorbell New Tail
3. PCIe TLP
4. Process Command
5. Queue Completion
6. Generate Interrupt
7. Process Completion
8. Ring Doorbell New Head

Submission Queue
- Head
- Tail

Completion Queue
- Head
- Tail

Host Memory

NVMe Controller
- Submission Queue Tail Doorbell
- Completion Queue Head Doorbell
- Fetch Command
- Process Command
- Generate Interrupt

PCIe TLP
PRP Scatter/Gather Lists

Fixed Size PRP Lists Accelerate Out of Order Data Delivery
Power Management

Example Power State Descriptor Table
(Indentify Controller Data Structure)

<table>
<thead>
<tr>
<th>Power State</th>
<th>Maximum Power</th>
<th>Entry Latency</th>
<th>Exit Latency</th>
<th>Relative Read Throughput</th>
<th>Relative Read Latency</th>
<th>Relative Write Throughput</th>
<th>Relative Write Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25 W</td>
<td>5 mS</td>
<td>5 mS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>18 W</td>
<td>5 mS</td>
<td>7 mS</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>18 W</td>
<td>5 mS</td>
<td>8 mS</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>15 W</td>
<td>20 mS</td>
<td>15 mS</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>10 W</td>
<td>20 mS</td>
<td>30 mS</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>8 W</td>
<td>50 mS</td>
<td>50 mS</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>5 W</td>
<td>20 mS</td>
<td>5000 mS</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
# Admin Command Set

<table>
<thead>
<tr>
<th>Command</th>
<th>Required or Optional</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create I/O Submission Queue</td>
<td>Required</td>
<td>Queue Management</td>
</tr>
<tr>
<td>Delete I/O Submission Queue</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Create I/O Completion Queue</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Delete I/O Completion Queue</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Identify</td>
<td>Required</td>
<td>Configuration</td>
</tr>
<tr>
<td>Get Features</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Set Features</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Get Log Page</td>
<td>Required</td>
<td>Status Reporting</td>
</tr>
<tr>
<td>Asynchronous Event Reporting</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Abort</td>
<td>Required</td>
<td>Abort Command</td>
</tr>
<tr>
<td>Firmware Image Download</td>
<td>Optional</td>
<td>Firmware Update / Management</td>
</tr>
<tr>
<td>Firmware Activate</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>I/O Command Set Specific Commands</td>
<td>Optional</td>
<td>I/O Command Set Specific</td>
</tr>
<tr>
<td>Vendor Specific Commands</td>
<td>Optional</td>
<td>Vendor Specific</td>
</tr>
</tbody>
</table>
# NVM Command Set

<table>
<thead>
<tr>
<th>Command</th>
<th>Required or Optional</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Required</td>
<td>Required Data Commands</td>
</tr>
<tr>
<td>Write</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Flush</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Write Uncorrectable</td>
<td>Optional</td>
<td>Optional Data Commands</td>
</tr>
<tr>
<td>Compare</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Dataset Management</td>
<td>Optional</td>
<td>Data Hints</td>
</tr>
<tr>
<td>Vendor Specific Commands</td>
<td>Optional</td>
<td>Vendor Specific</td>
</tr>
</tbody>
</table>
NVMe - Architected for Performance

- **No practical limit on number of outstanding requests**
  - Up to 64K I/O queues each with up to 64K entries
  - 32-bit controller unique command identifier (16-bit Queue ID + 16 Command ID) allows up to $2^{32}$ outstanding commands

- **Supports many-core processors without locking**
  - Each processor may be configured with its own submission/completion queues and MSI-X interrupt

- **At most one doorbell write to issue a command**
  - Multiple commands may be issued with a single doorbell write

- **Streamlined NVM command set avoids burdening controller with legacy command support requirements**

- **Fixed size 64B commands and 16B completions enable fast and efficient command decode and execution**
  - Commands contain 2 PRPs allowing 4KB or 8KB reads and writes be processed without fetching any additional information (e.g., scatter/gather list)

- **PRP based scatter/gather list allow efficient out-of-order data delivery**
NVMe Standardized Features

- Logical block data and metadata
- End-to-end data protection (T10 DIF and DIX compatible)
- Security (Trusted Computing Group collaboration)
- Submission queue arbitration and QoS
- Firmware update and activation
- Dynamic power management
- Robust error reporting
- Interrupt coalescing configuration/control
- Capability discovery and configuration
Summary

• NVMe is a high performance queuing interface and command set optimized for PCIe SSDs

• NVMe is scalable from client to enterprise applications

• NVMe 1.0 specification is complete and available at www.nvmexpress.org

• Industry ecosystem is forming around NVMe
  ▪ Standard drivers and development tools