Linux NVMe Driver

Keith Busch
Software Engineer
Intel Corp
NVMe & Linux: Agenda

• Linux driver development process and history
• Implementation details
• NVMe inspired kernel optimizations
• How to get involved
NVMe: Linux community development process

Maintainer Tree (infradead.org) -> Linux Mainline (kernel.org) -> Distros

Company X (company internal) -> Product delivery
Company Y (company internal) -> Product delivery

- copy/fork for product dev
- merging appropriate changes back for ecosystem
- medium-term
Active development community:
• 59 Change sets since initial commit
• 7 companies contributed patches
• Multiple Linux distributors ported driver to earlier kernel releases
NVMe: Linux PCI-e Driver

- **NVMe Device Driver**
  - Register PCI driver
  - Hot Add/Remove
  - Power Management
  - PCI Errors
  - PCI enable, select BARs, IO Remap
  - DMA Settings, mem pools
  - IRQ, MSI, MSI-x

- **Linux PCI Driver + Arch Specific**
  - PCI config read/write

- **NVM-Express Controller**
  - PCI Class Code: 010802h
NVMe: Queue allocation details

- Submission/Completion queue pairs
  - Round Robin Arbitration
  - One pair per CPU and assigned to that CPU
  - MSI-x interrupt affinity pinned to a CPU core per pair
    - Resort to MSI, then INTx, and finally polling if all else fails
  - Scalable: minimize lock contention, maximize cache hits
NVMe: Anatomy of Linux block software stack
NVMe: Detailed IO Process

Linux Block Layer

nvme_make_request

- Select NVMe IO Queue for currently running CPU
- Allocate command id
- Allocate resources to build command

bio_end_io

- Free IO Resources
- Read NVMe completion
- Set BIO success or -EIO
- Increment CQ Head

 ISR / Poll

- Get CQ head entry

New entry?

Yes

- Write new CQ Head Doorbell
- Return

No

- Split request SGE that breaks PRP constraints
- Requeue for submission
- Return

PRP ?

Yes

- Build command
- Write new doorbell tail
- Return

- Map BIO Vector to PRP

Read/Write

IO Flags

Flush/Discard

- Map BIO Vector to PRP

PRP ?

No
NVMe: Linux SCSI SG_IO IOCTL

For “Legacy” SCSI Management (not fast-path)

- Read/Write 6, 10, 12, 16
- Inquiry (Std, VPD 0, 80, 83, 86, B1)
- Mode Sense 10/16
- Mode Select 10/16
- Log Sense
- Read Capacity 10/16
- Report LUNS
- Request Sense
- Security Protocol In/Out
- Start Stop Unit
- Test Unit Ready
- Write Buffer
- Unmap
NVMe: Device management

- NVMe Device Driver
  - Create Misc Device
  - NVMe Admin Passthrough Command

- Character Device /dev/nvme#
  - open/close
  - IOCTL

- User Space
  - open/close
  - IOCTL

- Linux Misc Device
  - Create Misc Device
  - open/close
  - IOCTL

- NVM-Express Controller
  - PCI Class Code: 010802h
Asynchronous IO Latency sources:

- For low latency devices, context switch and interrupt dominate user observed latency.
Linux block layer performance optimizations: beyond NAND
Linux performance optimizations: T10 DIF Protection Information

- Linux calculates CRC-16 Guard via table lookups and is **expensive**!
- x86-64 improvement: PCLMULQDQ; merged in linux crypto-dev tree
Linux performance optimizations: T10 DIF Protection Information

Throughput T10 DIF Comparison

Block Size; IO Depth = 1

- None
- PCLMULQDQ
- Table
Linux NVMe: Get involved!

- Subscribe and contribute to mailing list: http://lists.infradead.org/mailman/listinfo/linux-nvme
- Clone, compile, and enhance driver: http://git.infradead.org/users/willy/linux-nvme.git
- Some TODO items:
  - Enhanced manageability via sysfs
  - Asynchronous events
  - Meta-data, T10 DIF/DIX
  - Power management
  - Performance enhancements/experiments
  - CPU hotplug
  - Advanced error handling
  - Enhanced PCI error handling
  - NVMe 1.1 spec updates
  - Device specific handling
Linux NVMe: Get involved!

- No hardware? No problem.
  - Machine emulator and virtualizer with NVMe support freely available from qemu.org
    - Good for testing features and basic functionality
    - Bad for analyzing performance and power characteristics
Questions:
keith.busch@intel.com
References

- NVM-Express
  [http://nvmexpress.org/](http://nvmexpress.org/)
- Linux NVMe Repository:
  [http://git.infradead.org/users/willy/linux-nvme.git](http://git.infradead.org/users/willy/linux-nvme.git)
- Linux NVMe Mailing list:
- When Polling is Better than Interrupt:
- Block polling in Linux:
  [http://lwn.net/SubscriberLink/556244/309ec42e8b9a4fcf/](http://lwn.net/SubscriberLink/556244/309ec42e8b9a4fcf/)
- CRC-16 T10 DIF PCLMULQDQ:
  [https://lkml.org/lkml/2013/5/1/449](https://lkml.org/lkml/2013/5/1/449)
NVMe OFA Open Source Windows Driver

Kwok Kong
Director of Software Engineering
PMC-Sierra
Agenda

- Status Update
- Driver Architecture
- Driver Features
- Future Features
## Status 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* OFA community driver since 2012. Code contributions from Huawei, IDT, Intel, LSI, and SanDisk.

*Other names and brands may be claimed as the property of others.*
Windows Storage Architecture

- Applications
  - Win32 I/O APIs
  - I/O Manager
  - Cache Manager
    - Partition Manager
    - Mount Manager
    - Volume Manager
  - File System
    - File System
    - Storage Class Driver
      - Storage Port / Storage Miniport
        - NVMe Miniport Driver
        - Driver
      - SCSI Port / SCSI Miniport
        - SCSI Miniport Driver
        - Driver
      - ATA Port / ATA Miniport
        - ATA Miniport Driver
        - Driver
      - IDE Port / IDE Miniport
        - IDE Miniport Driver
        - Driver

- Storage Port / Storage Miniport
  - NVMe Miniport Driver
  - Driver

- SCSI Port / SCSI Miniport
  - SCSI Miniport Driver
  - Driver

- ATA Port / ATA Miniport
  - ATA Miniport Driver
  - Driver

- IDE Port / IDE Miniport
  - IDE Miniport Driver
  - Driver

- Applications
  - Partition Manager
  - Mount Manager
  - Volume Manager

- File System
  - File System
  - Storage Class Driver
    - Storage Port / Storage Miniport
      - NVMe Miniport Driver
      - Driver
    - SCSI Port / SCSI Miniport
      - SCSI Miniport Driver
      - Driver
    - ATA Port / ATA Miniport
      - ATA Miniport Driver
      - Driver
    - IDE Port / IDE Miniport
      - IDE Miniport Driver
      - Driver

- Storage Port / Storage Miniport
  - NVMe Miniport Driver
  - Driver

- SCSI Port / SCSI Miniport
  - SCSI Miniport Driver
  - Driver

- ATA Port / ATA Miniport
  - ATA Miniport Driver
  - Driver

- IDE Port / IDE Miniport
  - IDE Miniport Driver
  - Driver

- Applications
  - Partition Manager
  - Mount Manager
  - Volume Manager

- File System
  - File System
  - Storage Class Driver
    - Storage Port / Storage Miniport
      - NVMe Miniport Driver
      - Driver
    - SCSI Port / SCSI Miniport
      - SCSI Miniport Driver
      - Driver
    - ATA Port / ATA Miniport
      - ATA Miniport Driver
      - Driver
    - IDE Port / IDE Miniport
      - IDE Miniport Driver
      - Driver

- Storage Port / Storage Miniport
  - NVMe Miniport Driver
  - Driver

- SCSI Port / SCSI Miniport
  - SCSI Miniport Driver
  - Driver

- ATA Port / ATA Miniport
  - ATA Miniport Driver
  - Driver

- IDE Port / IDE Miniport
  - IDE Miniport Driver
  - Driver
Driver Architecture

Storport Driver

NVMe Storport Miniport Driver

Initialization
- DriverEntry
- HWFindAdapter
- HWInitialize

I/O Services
- HWBuildIo
- HWStartIo
- HWInterrupt

PnP/PM, Error Recovery
- HWAdapterControl
- HWResetBus

Init State Machine
- Interrupt Config.
- I/O Process

SCSI/NVMe Tx I/F
- Q Management
- Register Access
IO Queue Allocation Diagram

Start

Allocate 1st IO Queue pair

Failed?

Yes

Driver Initialization fails

End

No

Allocate one IO Queue pair

Failed?

Yes

More core in current NUMA node

No

More NUMA node

Yes

All cores share 1st allocated IO Q

No

No IO Queue Shared

Yes

More NUMA node

No

No core in current NUMA node

Yes

More NUMA node

No

No core in current NUMA node

Yes

More NUMA node

No

No core in current NUMA node
Driver Initialization State Machine

- NVMeRunningStartAttempt
- WaitOnRDY
- WaitOnIdentifyCtrl
- WaitOnIdentifyNS
- WaitOnSetFeatures
- WaitOnSetupQ
- WaitOnLearnMapping
- WaitOnReSetupQ

State Arbiter

Enter new state

Issue command(s)

Command timeout?
- Yes
  - DriveLoading Failed and Return FALSE
  - DriveLoading Succeeded and Return TRUE

Command(s) completed?
- Yes
  - StartComplete
- No
  - StartFailed

Start Complete?
- No
  - Any errors?
- Yes
  - No
IO Process Diagram (Read)

NVMeBuildIo called

- Pre-processing checks succeeds?
  - Yes: Succeeds?
    - Yes: Return TRUE
    - No: Return FALSE
  - No: Populate error status
    - SCSI/NVMe cmd Tx:
      1. CDB
      2. SG to PRP
      3. Prepare Sub entry in SRB Ext
    - StorPortNotification (RequestComplete)
      1. Get current process#
      2. Get associated sub/cpl queue pair
      3. Get Cmd ID/Entry
      4. Copy sub entry from SRB Ext to the associated sub queue
    - Succeeds?
      - Yes: Return TRUE
      - No: Schedule DPC

NVMeStartIo called

- Acquire StartIO Lock
- Ring doorbell to issue cmd
- Succeeds in issuing cmd?
  - Yes: Return TRUE
  - No: Release StartIO Lock

ISR called

- Acquire StartIO Lock
- Schedule DPC
- Return TRUE

DPC called

- Acquire DPC Lock
- Determine which Cpl queue(s) to process
- More queue to process?
  - Yes: Update Cpl Hd Ptr
    - If any entries processed
      - Yes: Pending Cpl entry?
        - Yes: Return TRUE
        - No: Return FALSE
      - No: Release DPC Lock
      - Recycle Cmd ID/entry
      - Call completion routine if exists.
      - Call StorPortNotification (RequestComplete) if necessary
  - No: Release DPC Lock
- Release DPC Lock
- Recycle Cmd ID/entry
- Mark SrbStatus as SRB_STATUS_BUSY
- Call StorPortNotification (RequestComplete)
- Pending Cpl entry?
Queue-Core-Vector Mappings
## System Features Overview

<table>
<thead>
<tr>
<th>Features</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Versions (64-bit only)</td>
<td>7, 8, Server 2008R2, Server 2012</td>
</tr>
</tbody>
</table>
| NUMA Optimized Queues and Memory| Dedicated IO queues per CPU core  
Single Admin Queue for all CPU cores  
Queue memory allocated local to NUMA Node |
| Interrupt                       | MSI-X (Vectors mapped to NUMA optimized IO Queue Pairs)  
MSI INTX                                      |
| Queue Arbitration / Priority    | Round Robin only  
1:1 mapping between Submission and Completion queues |
<p>| Pass Through                    | Yes (with DeviceIoControl())                                              |
| Multiple Namespaces             | Up to 16, Mapped to Bus 0, Tgt 0, LUN 0-15                               |
| Registry Parameters             | Name Space, Max Transfer Size, Admin Queue Size, IO Queue size, Interrupt Coalescing Time/Entries |
| End to End Protection           | No                                                                        |</p>
<table>
<thead>
<tr>
<th>Commands</th>
<th>Native Support</th>
<th>Pass Through</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete I/O Submission Queue</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Create I/O Submission Queue</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Get Log Page</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Delete I/O Completion Queue</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Create I/O Completion Queue</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Identify</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Abort</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Set Features</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Get Features</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Asynchronous Event Request</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Firmware Activate</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Firmware Image Download</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Format NVM</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Security Send</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Security Receive</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Vendor specific</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## NVM command Sets

<table>
<thead>
<tr>
<th>Commands</th>
<th>Native Support</th>
<th>Pass Through</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flush</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Write</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Read</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Write Uncorrectable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Compare</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dataset Management</td>
<td>Yes (Deallocate)</td>
<td>Yes</td>
</tr>
<tr>
<td>Vendor specific</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Future Features

- **1.3 Release (end of 2013)**
  - Windows 32-bit
  - End to End Protection
  - Hibernation Support on Boot Drive
  - NVM Format Enhancement

- **2014 and Beyond**
  - NVMe 1.1 Features

Need you contribution to write the future
OFA NVMe Driver Working Group

- Founding Companies
  - PMC Sierra (IDT) – Chairperson
  - Intel – Code maintainer
  - LSI (SandForce)

- To Contribute
  - Join the mailing list
  - Email your patch to the WG mailing list
  - Code is checked in if approved by two out of three founding companies
Useful Information

- **Join the Driver Mailing List to Contribute**
  - [http://lists.openfabrics.org/cgi-bin/mailman/listinfo/nvmewin](http://lists.openfabrics.org/cgi-bin/mailman/listinfo/nvmewin)

- **Development Tools Description**

- **Driver Source Code**
  - [http://www.openfabrics.org/svnrepo/nvmewin/](http://www.openfabrics.org/svnrepo/nvmewin/)