NVMe Software Drivers: What’s New and What’s Supported?

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

Scott Lee

Name

Insert Company Logo

Name

Insert Company Logo
Windows Inbox NVMe Driver

Scott Lee

Principle Software Engineer Lead

Microsoft
Agenda

• New Additions for Windows 10 version 1903, May 2019 Update (19H1)
• New Additions for Windows Next
• Futures
Windows 10 version 1903, May 2019 Update

- Endurance Group & NVM Set
- Improved diagnostics of NVMe hardware issues
- Runtime D3 for NVMe
- Device Self-Test
- Host Controlled Thermal Management Feature
- Controller Fatal Status
Windows Next

- Non-Operational Power State Config Feature
- NVMe LED
- ???
Futures*

- Native NVMe Storage Stack
- Zoned Namespace
- Device Firmware Hang
- ???

* Not plan of record
Questions?
vSphere NVMe Driver Support

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

Sudhanshu (Suds) Jain

Murali Rajagopal
NVMe Focus @VMWare

**Driver**
- Boot (UEFI)
- Firmware Update
- End-to-end protection
- Deallocate/TRIM/Unmap
- 4K
- SMART, Planned hot-remove

**Core Stack**
- Reduced serialization
- Locality improvements
- vNVMe Adaption layer
- Multiple completion worlds support in NVMe

**Virtual Devices**
- NVMe 1.0e spec
- Hot-plug support
- VM orchestration

**vSphere 6.5**
- Performance enhancements
- Extended CLI
- Name space management
- Async event error handling
- Enhance diagnostic logs

**vSphere 6.7**
- Optimal stack - Highly parallel execution for single path local NVMe devices
- Reach target of 90%+ performance of device spec

**Future Direction**
- PCIe Native Hot-plug
- LED Management
- NVMe Over Fabric
- Multiple fabric option
- Sanitize

- Next Generation Storage Stack with ultra-high IOPS
- End-to-end NVMe Stack
- NVMe Multi-pathing, ANA

- Rev the specification
- Parallel execution @backend
- 4K Support
- Scatter-gather support
- Interrupt coalescing
Hardware:
- Intel® Xeon® E5-2687W v3 @3.10GHz (10 cores + HT)
- 64 GB RAM
- NVM Express® 1M IOPS @ 4K Reads

Software:
- vSphere® 6.0U2 vs. Future prototype
- 1 VM, 8 VCPU, Windows® 2012, 4 VMDK eager-zeroed
- IOMeter:
  - 4K seq reads, 64 OIOs per worker, even distribution of workers to VMDK

The information in this presentation is intended to outline our general product direction and it should not be relied on in making a purchasing decision. It is for informational purposes only and may not be incorporated into any contract.
(Future) NVMe Driver Architecture

ESXi Storage Stack

ESXi Next Generation Storage Stack

NVMe Transport Device Driver Framework

PCIe Transport Driver

RDMA Transport Driver (RoCEv1, RoCEv2, iWarp)

Fibre Channel Transport Driver

Stack Interface 1

SCSI NVMe Translation

Stack Interface 2

NVMe Core Functionality

CLI

NVMe-oF Transport Abstraction

Driver Interface

vmknvme
VMware’s NVMe Driver Ecosystem

- Available as part of base ESXi image from vSphere 6.0 onwards
  - Faster innovation with async release of VMware NVMe driver
- VMware Opensource its NVMe Driver to encourage ecosystem to innovate
  - [https://github.com/vmware/nvme](https://github.com/vmware/nvme)
- Broad VMware NVMe Driver Ecosystem
  - Close to 300 third party NVMe devices certified on VMware NVMe driver
- Beyond NVMe PCI Driver (Future)
  - Actively working with broad I/O controller and storage array partners to bring NVMe-oF solutions
Questions?
Accelerating NVMe with SPDK

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

James Harris
Storage Performance Development Kit

**User Space Storage Software Stack**
- Extreme performance (10M+ IO/s on one thread)
- Block device abstraction and device drivers
- Network and virtualization protocols
- Resets, timeouts, I/O splitting, volume management

**Widely Adopted**
- Powering major storage systems in production today

**C Libraries and Applications**
- Open Source (GitHub, BSD License)
- Active Community (~50 contributors each quarter)
# SPDK Architecture

## Block Storage Protocols
- **Networking**
  - NVMe-oF (RDMA, TCP), iSCSI
- **Virtualization**
  - vhost-scsi, vhost-blk, vhost-nvme

## Block Storage Services
- **Partitioning**
  - Logical Volumes, GPT
- **Caching**
  - OCF
- **Host FTL**
  - Open Channel
- **Pooling**
  - RAID-0
- **Transforms**
  - Crypto, Compression

## File Storage Services
- **Filesystems**
  - BlobFS

## Integration
- **Database**
  - RocksDB, Ceph
- **Orchestration**
  - Cinder

## Tools
- **Management**
  - nvme-cli, spdk-cli
- **Benchmarking**
  - fio

## Block Storage Providers
- NVMe, Ceph RBD, Linux AIO, virtio, iSCSI, pmemblk, malloc, null

## Drivers
- NVMe (PCIe, RDMA, TCP), virtio (scsi, blk), Intel Quickdata
PCIe NVMe Performance

Single Thread Random Read
(in millions of I/O per second)

- Intel® Xeon® Platinum 8280L CPU
  - Turbo 4.0GHz
- 21 SSDs Attached
  - Intel® P4610
  - Intel® P4800X

11.0M @ 57us avg latency!
SPDK and Kernel

SPDK has better performance and efficiency compared to interrupt-driven kernel mode approaches

BUT...

SPDK is not a general-purpose solution

- covers some use cases very well – others not at all (or at least not well)

Polled mode design and userspace implementation drove much of the SPDK design
NVMe Performance: Avoid MMIO

- Past: Simple completion queue doorbell batching
  - Ring doorbell after processing first 3 completions

- Recent: Leverage polling
  - Delay ringing submission queue doorbell until end of poll call

- Future: Advanced completion queue batching
  - Track number of free cq slots
  - Only ring doorbell when slots are needed
NVMe Transport Abstraction

Enables different implementations for different transports

- construct/destruct controller
- set/get register value
- create/delete I/O queue pair
- submit request
- process completions
NVMe-oF Target

Spec-compliant, fully functional NVMe-oF target

- No modifications on client/compute node

Supports broad range of storage services – including:

- Sharing SSD across multiple clients (Logical Volumes)
- At-rest data encryption with crypto offload
- SSD pooling/striping
NVMe/TCP

NVMe TP ratified November 2018

SPDK added TCP transport for

- NVMe driver
- NVMe-oF target

Supports alternative TCP stack implementations
Host Block FTL

Host FTL enabling smart data placement

- Based on OC2.0 specification

Block FTL support added to bdev nvme module

Long term goal: Zoned Namespace API
- With ZNS/OC adapters
Supported Features

Explicit Queue Pair Allocation
Metadata and Data Protection
Controller Memory Buffer
Timeout Handling
SGL
Asynchronous Attach
AER
Error Injection
Queue Pair Creation

```c
struct spdk_nvme_qpair *
spdk_nvme_ctrlr_alloc_io_qpair(struct spdk_nvme_ctrlr *ctrlr,
    const struct spdk_nvme_io_qpair_opts *opts,
    size_t opts_size);
```

Queues are *not* preallocated

- admin commands issued when qpair allocated

```c
struct spdk_nvme_io_qpair_opts
```

- Priority (for WRR)
- I/O queue size, # I/O requests
Metadata Support

Contiguous metadata
- Uses “standard” I/O functions
  - i.e. spdk_nvme_ns_cmd_read

Separate metadata buffer
- spdk_nvme_ns_cmd_read_with_md()
  - and variants

End-to-end Data Protection
- All I/O commands take io_flags parameter
Questions?