LightNVM Brings SSDs to the Linux Kernel

Matias Bjørling, LightNVM Principal Architect
CNEX Labs, Inc.
Introduction

- Matias Bjørling, LightNVM Principal Architect, CNEX Labs, Inc.

CNEX Labs

- CNEX is a privately held start-up company
- Founded in 2013 by semiconductor industry veterans in Silicon Valley
- Funded by VC and investments from Fortune 500 companies in storage and networking
- Chartered to deliver innovative system solutions in the form of semiconductors and software
- First product is a highly differentiated NVMe SSD controller ASIC
- Currently shipping SDK’s; engaged with strategic customers and partners for mass production
SSD Controllers: Terminology and Core Functionality

Traditional SSD
- Logical Block Addressing (LBA) on Device
- FTL controlled by Device Firmware ("Black-Box")
- Fixed functionality & performance

Flash Translation Layer (FTL) for a typical NVMe SSD device

Where does it shine?
When is this not-so-good?

Hint:
Jeffrey Dean, Luiz André Barroso, "The Tale at Scale"
Key Drivers:
- Web-Scale Datacenters
- Hyper-converged Infrastructure
- Flash Array Products
- High-Performance Computing

Key Requirements
- Latency
  - Low & Deterministic
  - Versus Endurance and Throughput
- Power/Energy efficiency

LightNVM for Application-Defined-Storage

- Full host control of Physical data placement, I/O scheduling, and background operations
- FTL tailored for specific application types and workloads
- Low and predictable latency, DRAM-less controllers, and energy efficiency
LightNVM: Key Concepts

**Traditional SSD**
- Logical Block Addressing (LBA) on Device
- FTL controlled by Device Firmware ("Black-Box")
- Fixed functionality & performance

**Open-Channel SSD**
- Physical Page Addressing (PPA) Command Set
- Key FTL functions exposed to LightNVM on Host
- Flexible for application-specific performance

- **Host System with LightNVM**
  - Data Placement
  - I/O Scheduling
  - Background Operations

- **Legacy SSD Controller ASIC**
  - PCle/NVMe
  - Read/Write/Trim “Logical Blocks”
  - Wear-Leveling
  - Metadata State Management
  - Error Handling
  - XOR engines
  - ECC engines
  - SMART/health Management

- **Non-Volatile Media**

- **Open-Channel SSD Controller ASIC**
  - PCle/NVMe
  - Read/Write/Erase “Physical Pages”
  - Wear-leveling
  - Metadata State Management
  - Error Handling
  - XOR engines
  - ECC engines
  - SMART/health Management

- **Non-Volatile Media**

**Flash Memory Summit 2016, Santa Clara, CA**
LightNVM with Open-Channel SSD Hardware

NVMe compatible:
- Physical Page Addressing (PPA) Command Set
- Linux kernel 4.4+
- Managed using standardized nvme tools (nvme-cli)

Common Data Structures
- Append-only
- Key-Value

Provisioning Interface
- Reserve block
- Release block

NVMe Physical Page Address Command Set

Open-Channel SSD
- Metadata State Mgmt.
- ECC Engine
- XOR Engine

Device Responsibilities

Block Storage

Flash Memory Summit 2016, Santa Clara, CA
LightNVM Leverages NVMe for Minimal Disruption

- Use existing NVMe Admin and Queuing structure, and NVMe device driver
- Add I/O Commands for “Physical Page Addressing” (PPA)
  - Currently implemented as NVMe “vendor unique” commands;

**Open-Channel PPA I/O Commands:**

- **Read PPA:** “Read a PPA, in unit of a sector”
- **Write PPA:** “Write to a PPA, in unit of a sector”
- **Erase PPA:** “Erase an NVM block”
- **Identify Geometry:** “Get geometry of device & media”
Predictable Performance, Latency

- With Open-Channel SSDs, host FTL software can be tuned for workloads and application types
- Enables data placement by data “type” or “class”, to avoid mixing data within NAND flash blocks
- Reduced overprovisioning, reduced write-amplification, intelligent garbage collection…
- A qualitative example:
Read/Write Latency

Flash Memory Summit 2016, Santa Clara, CA
User Space FTL with LightNVM, liblightnvm

- Potential to collapse multiple layers of redundant mapping in application & filesystem
- Bypass Kernel processing, preserve low-latency characteristics of new/emerging NV Media types
Application Acceleration with LightNVM, liblightnvm

- Maps Flash blocks to RocksDB levels
  - Perfect Layout on SSD
  - No garbage collection
  - Reduced write amplification

- Maps flash blocks to large data blocks
  - No garbage collection necessary

- Metadata in RocksDB
  - Fast updates

Predictable throughput and latency
Summary

- Significant advantages to OpenChannel SSD with Host FTL
  - Performance, Latency, Power, Endurance, Application Specific Performance, …
  - De-couples FTL SW from SSD Controller Hardware (Development Cycles) – Enables Rapid Innovation
- Minimal disruption
  - Utilize existing NVMe; add I/O commands for Physical Page Addressing
- OpenChannel SSD and LightNVM FTL is a Growing ecosystem!
  - Participate at: https://github.com/OpenChannelSSD

See OpenChannel/LightNVM SSD demos at FMS:
  - Liteon: Booth 621
  - Micron: Booth 134
  - Radian: Booth 615

Thank-You!