Open-Channel SSDs
Then. Now. And Beyond.

Matias Bjørling, March 22, 2017
Introduction

▪ What is an Open-Channel SSD?
▪ Then
  - Physical Page Addressing v1.2
  - LightNVM Subsystem
  - Developing for an Open-Channel SSD
▪ Now
  - pblk – Host-side Flash Translation Layer
  - liblightnvm – User-space library
▪ And Beyond
  - Physical Page Addressing v2.0
  - Zones Support
  - Feedback loop
Solid-State Drives and its Non-Volatile Media

Solid-State Drive

- Responsibilities
  - Flash Translation Layer
  - Media Error Handling
  - Media Retention Management

- Host Interface

- Parallel Units
  - Channel X
  - Channel Y

- Media Controller

- Transform R/W/E to R/W

- Manage Media Constraints
  - ECC, RAID, Retention

- Read/Write

- Read/Write/Erase

- NAND
  - Read (50-100us)
  - Write (1-10ms)
  - Erase (3-15ms)

- Tens of Parallel Units
Mixed I/O Workloads

0% writes and latency is consistent

20% writes makes big impact on read latency

50% writes can make SSDs as slow as spinning drives...

Larger outliers on increased writes
Indirection and Read/Write I/O Interface

Even if Writes and Reads does not collide from application Indirection and loss of information due to the narrow Read/Write I/O interface

Log-on-Log

FTL-like implementation at multiple layers

Not able to align data on media = Write amplification increase + extra GC

Data placement + Buffering = Best Effort

SSD state is hidden due to the narrow I/O Interface

Writes decoupled from Reads

Writes

Reads
Open-Channel SSDs

I/O Isolation
Enable I/O isolation between tenants by allocating your SSD into separate parallel units.

Predictable Latency
No more guessing when an IO completes. You know which parallel unit is accessed on disk.

Data Placement & I/O Scheduling
Manage your non-volatile memory as a block device, through a file-system or inside your application.
Predictable Latency

- 4K reads during 64K concurrent writes
- Traditional NVMe device and an Open-Channel SSD
1. Physical Page Addressing Specification
   - Released 2H2016
   - Based on feedback from Hyper-scalers, AFAs, and HPC requirements
   - Implemented as a vendor specific command set

2. LightNVM Subsystem
   - Kernel integration of the above interface
   - Administration of drive instances

3. Writing to an Open-Channel SSD
Physical Page Addressing (PPA) Interface v1.2

- **Expose geometry**
  - Logical/Physical geometry
  - Performance
  - Controller functionalities

- **Hierarchical Address Space**
  - Encode geometry into the address space

- **Vector I/Os**
  - Read/Write/Erase

---

- # Channels, # Parallel Units, # Blocks, Block Size, Min. Write size, Optimal Write size, ...
- Encode parallel units into the address space
- Logical Block Address (LBA)
  - Sector
- Physical Page Address (Geometry encoded)
  - Channel, LUN, Chunk, Sector

---

Copyright © 2017 CNEX Labs
1. NVMe Device Driver
   - Detection of OCSSD
   - Implements PPA interface

2. LightNVM Subsystem
   - Generic layer
   - Core functionality
   - Target management (e.g., pblk)

3. High-level I/O Interface
   - Block device using pblk
   - Application integration with liblightnvm
Writing to an Open-Channel SSD (1)

▪ What’s different compared to a traditional SSD?
  - Geometry?
    • Presents a set of blocks (also known as zones/chunks)
    • Timings
  - Reads?
    • Sector-sized reads
    • Bonus: Vector interface defines extra error codes such as reading a non-written page, and hints if a page has high ECC
What’s different?

- Writes?
  - Requirement: write sequential within a block
  - Can fail. An error does not mean the hole disk is bad. Just means the write unit is bad.
  - Write data to next page within the block.

- Erases?
  - Can fail. An error does not mean the hole disk is bad. Just means the block is bad and data should be rewritten to a new page.
Now

pblk

Linux Kernel

File System

File System

LightNVM Subsystem

NVMe Device Driver

Hardware

Open-Channel SSD

User-space

liblightnvm

I/O Apps

Kernel-space

LightNVM Subsystem

NVMe Device Driver

Open-Channel SSD
Host-side Flash Translation Layer - pblk

- Mapping table
  - Sector-granularity
- Write buffering
  - Lockless circular buffer
  - Multiple producers
  - Single consumer (Write Thread)
- Error Handling
  - Media write/erase errors
- Garbage Collection
  - Rewrite blocks
- Recovery of metadata
But does it blend?

- CNEX Labs Open-Channel SSD
  - NVMe
  - PCIe Gen3x8
  - 2TB MLC NAND

- Geometry
  - 16 channels
  - 8 PUs per channel (Total: 128 PUs)

- Parallel Unit Characteristics
  - Read Size: 4K
  - Write size: 16K + 64B user OOB
  - Blocks: 1.067, Block Size: 16MB

- Performance:
  - Write: Single PU 47MB/s
  - Read: Single 108MB/s, 280MB/s (64K)

- Base Performance
- Limit # Active Parallel Write Units
- Multi-tenancy using I/O Isolation
Base Performance using Vector I/O

Throughput & Latency

Grows with parallelism

RR slightly lower due to scheduling conflicts

Request I/O Size

Throughput (GB/s)

Latency (ms)
Limit # Active Writers

- A priori knowledge of workload. E.g., limit to 400MB/s Write
- Limit number of Active PU Writers, and achieve better read latency

- Single Read and Write Perf.
- Mixed Read/Write

- 256K Write QD1 256K Read QD16

- Write throughput 400MB/s
- Write latency increases, and read latency reduces
Multi-Tenant Workloads

2 Tenants (1W/1R)

4 Tenants (3W/1R)

8 Tenants (7W/1R)

NVMe SSD

OCSSD
liblightnvm

- User-space interface to interact with vectoring using Open-Channel SSDs
  - Easy to use interface for geometry layout and I/O accesses

- Interface
  - struct nvm_dev
  - struct nvm_geo
  - struct nvm_addr
  - struct nvm_vblk
  - ...

- Terrific set of CLI tools

- Great tutorial available
  - [http://lightnvm.io/liblightnvm/tutorial/](http://lightnvm.io/liblightnvm/tutorial/)
And Beyond.

- Physical Page Addressing v2.0
- Zones Support
- Media Feedback Channel
Physical Page Addressing 2.0

- Adapted to customer requests
  - Simplified
    - Log-structured. Write sequentially within block
    - Minimal write size, optimal write size.
  - Media-agnostic
    - Work on NAND as well as PCM and other next generation memories
  - Able to be exposed through the Zones interface introduced for SMR drives
    - Prototype patches available.
And Beyond. – Media Feedback

- Why a feedback channel? How?

![Diagram showing Host, NVMe Device Driver, Open-Channel SSD, Retention Policy, Read Retry, High ECC, Wear-leveling, and NVMe AER.]
Conclusion

- New interface that provides
  - I/O Predictability
  - I/O Isolation
  - Puts the host in front seat of data placement and I/O scheduling

- PPA Specification is open and available for implementors

- Active community using OCSSDs both for production and research
  - Multiple drives in development within SSD vendors
  - Multiple papers already on Open-Channel SSDs that shows how this interface can improve workloads

- Fundamental building blocks are available:
  - Initial release in Linux kernel 4.4.
  - User-space library (liblightnvm) support with Linux kernel 4.11.
  - Pblk will be upstream with Linux kernel 4.12.

- The right time to dive into Open-Channel SSDs
  - More information available at: http://lightnvm.io
CNEX Labs, Inc.

Teaming with NAND Flash manufacturers and industry leaders in storage and networking to deliver the next big innovation for solid-state-storage.