Open-Channel SSDs Offer the Flexibility Required by Hyperscale Infrastructure

Matias Bjørling
CNEX Labs
Public and Private Cloud Providers
Workloads and Applications
NAND Capacity Grows Bigger

Performance – Endurance – DRAM overheads
Read Latency with 0% Writes

Random Read 4K

Percentiles

2017 Storage Developer Conference. © CNEX Labs. All Rights Reserved.
Read Latency with 20% Writes

Random Read 4K + Random Write 4K

Significant outliers! Worst-case 30X

4ms!
Requirements

1. Flexible enough for software to evolve faster than hardware
2. Strong QoS Guarantees
3. Continuous access – No maintenance windows
4. Support a broad set of applications on shared hardware
5. Rapid enablement of new NAND generations / media agnostic
6. Vendor neutrality & supply chain diversity
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 I/O 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.
Rebalance the Storage Interface

- Expose internal lanes to host
  - Each lane is a parallel unit
  - Logical or Physical
  - Performance characteristics
- Building Block
  - Similar to HDD SMR interface
  - Extension to NVMe
  - Can implement I/O Determinism, streams, and other new data management schemes
- Open specification
Rebalance the Storage Interface

- Expose geometry
  - Logical/Physical geometry
  - Performance characteristics
  - Controller functionalities
  - Similar to HDD SMR interface
- Exposes parallel units to the host
- Enable efficient access to the parallel units

# Channels, # Parallel Units, # Chunk, Chunk Size, Min. Write size, Optimal Write size, …

Encode parallel units into the address space

Up to the SSD vendor
Internals of an SSD

Solid-State Drive

- Host Interface
- Media Controller
  - Parallel Units
  - Channel X
  - Channel Y

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

Transforms 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!
Move Data Placement & I/O Scheduling to the Host

Solid-State Drive

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

Host Interface

LUNs

Media Controller

Channel X

Channel Y

 NAND

Read (50-100us)

Write (1-10ms)

Erase (3-15ms)
Move Data Placement & I/O Scheduling to the Host

Solid-State Drive

- Flash Translation Layer
- Responsibilities
  - Media Error Handling
  - Media Retention Management
- Host Interface
- Channel X
- LUNs
- Channel Y
- Media Controller
  - NAND
    - Read (50-100us)
    - Write (1-10ms)
    - Erase (3-15ms)
Unsupervised Parking Lot
Supervised Parking Lot
Open-Channel SSD 2.0 Specification

- Expose a description of parallelism within SSD
  - Expose #Channels, #LUNs (PUs), #Chunks, Write sizes, …
- Enhanced specification from customer requests and SSD vendors
  - Simplified
    - Log-structured. Write sequentially within a chunk
    - Minimal write size, optimal write size.
  - Media-agnostic
    - Work on NAND as well as PCM and other non-volatile memories
  - Exposed through similar constructs as Zones for SMR hard-drives
Drive Model - Chunks

- A chunk defines the area where writes must be sequential
- Reads
  - Sector size
- Writes
  - Minimum write size granularity
  - Synchronous – May fail – An error only marks write bad, and not the full SSD
- Reset Chunk (Erase)
  - Synchronous – May fail – An error only marks chunk bad, and not the full SSD

<table>
<thead>
<tr>
<th>Chunk</th>
<th>0</th>
<th>1</th>
<th>...</th>
<th>Chunk - 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBA</td>
<td>0</td>
<td>1</td>
<td></td>
<td>LBA -1</td>
</tr>
</tbody>
</table>
Drive Model - Organization

- Parallelism exposed over
  - Channels (Shared bus)
  - LUNs – Parallel Units
- Logical or Physical representation

<table>
<thead>
<tr>
<th>Channel</th>
<th>0</th>
<th>1</th>
<th>...</th>
<th>Channel - 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUN</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>LUN - 1</td>
</tr>
<tr>
<td>Chunk</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>Chunk - 1</td>
</tr>
<tr>
<td>LBA</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>LBA - 1</td>
</tr>
</tbody>
</table>
LightNVM Architecture

NVMe Device Driver
- Detection of OCSSD
- Implements OCSSD interface

LightNVM Subsystem
- Core functionality – "Partition manager"
- Target management (e.g., pblk)
- Sysfs integration

High-level I/O Interface
- Block device using pblk
- Application integration with liblightnvm
Host-side Flash Translation Layer - pblk

- Multi-target support - I/O isolation
- Fully associative L2P table (4KB mapping)
- Host-side write buffer to guarantee read and writes
- Cost-based garbage collector, using valid sector count as metric.
- Capacity based rate limiter. Designed as a function of user and GC I/O present in write buffer.
- Scan-based L2P recovery. Scan metadata in closed lines and OOB on open lines.
- sysfs interface for statistics and FTL tuning.
Raw Performance

### Read/Write Throughput

- **Write (MB/s)**
- **Read (MB/s)**

*Not final performance numbers*
Predictable Latency

- 4K reads during 64K concurrent writes
- Consistent low latency at 99.99, 99.999, 99.9999
Demo using Open-Channel SSD

4K Reads

256K Writes

Solid State Drive

Open-Channel SSD (Instance-based)
CNEX LABS
Welcomes a new world of data
Scalability

Latency (microseconds)

Percentiles

3 writes + 1 read
7 writes + 1 read
15 writes + 1 read
60 writes + 1 Read
120 Writes + 1 Read

0 10 20 30 40 50 60 70 80 90 100

0 50 100 150 200 250 300 350

2017 Storage Developer Conference. © CNEX Labs. All Rights Reserved.
liblightnvm

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

- Active community
  - Multiple drives in development by commercial SSD vendors
  - Multiple papers on Open-Channel SSDs

- Growing software stack
  - LightNVM subsystem since Linux kernel 4.4.
  - User-space library (liblightnvm) support from Linux kernel 4.11.
  - pblk available from Linux kernel 4.12.
Conclusion

- New storage interface that solves the software-defined storage challenges for hyperscalers
  - I/O Predictability
  - I/O Isolation
  - Application-level control of data placement and I/O scheduling
- Draft specification is open and available for implementers
- Formal standards initiative underway, target spec release 1H 2018
- More information available at: [http://lightnvm.io](http://lightnvm.io)