OCP SUMMIT

March 20-21
2018
San Jose, CA

OPEN. FOR BUSINESS.

DENALI

The next-Generation High-Density Storage Interface

Laura Caulfield
Senior Software Engineer

Arie van der Hoeven
Principal Program Manager

Microsoft

OPEN. FOR BUSINESS.

OCP SUMMIT

Outline

- Technology Trends & Application Requirements

- Proof-of-Concept

- Host-Drive Specification

Outline

- Technology Trends & Application Requirements

- Proof-of-Concept
- Host-Drive Specification

Design Principles For Cloud Hardware

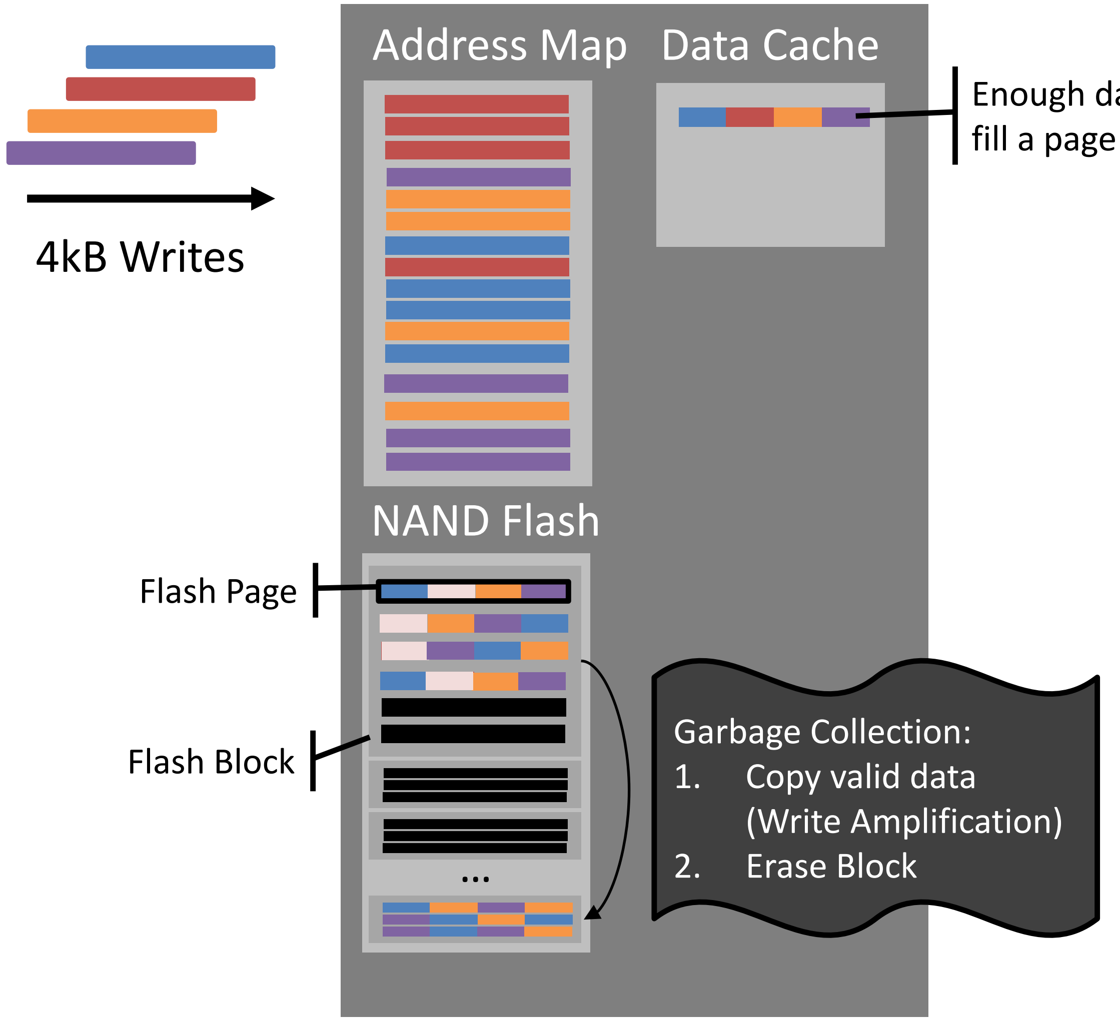
- Support a broad set of applications on shared hardware
  Azure (>600 services), Bing, Exchange, O365, others

- Scale requires vendor neutrality & supply chain diversity
  Azure operates in 38 regions globally, more than any other cloud provider

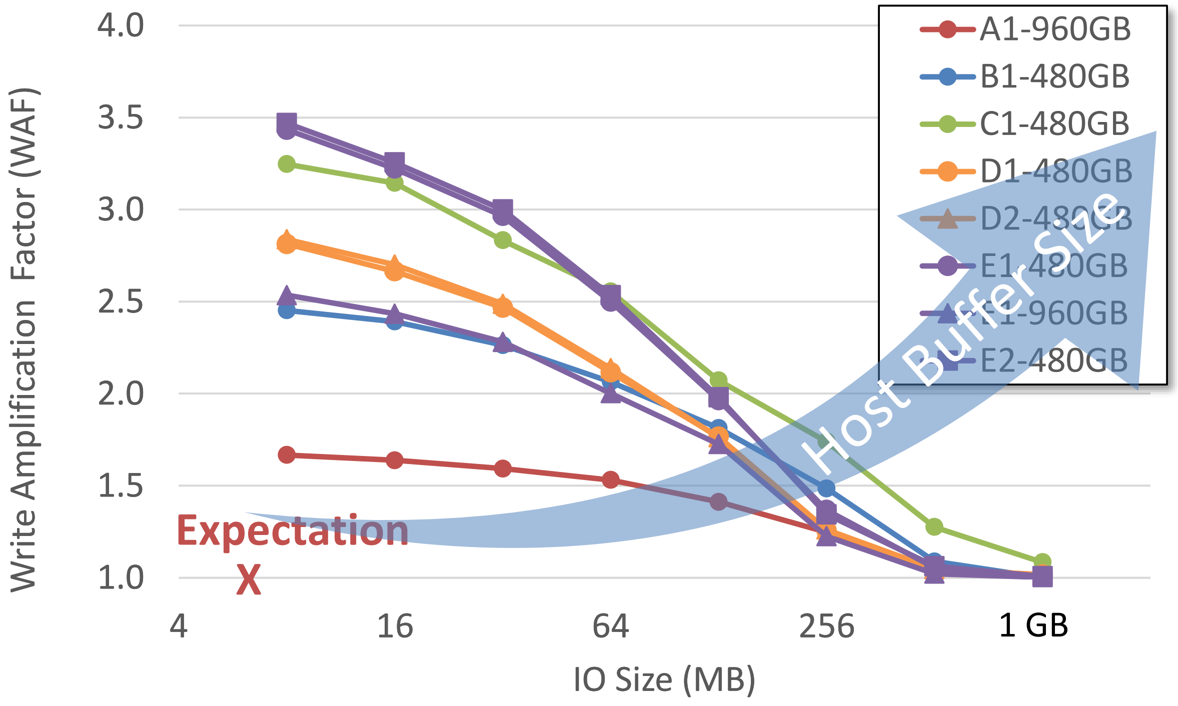
- Rapid enablement of new generations
  New NAND every 18 months, hours to precondition, hundreds of workloads

- Flexible enough for software to evolve faster than hardware
  SSDs rated for 3-5 years, heavy process for FW update, software updated daily

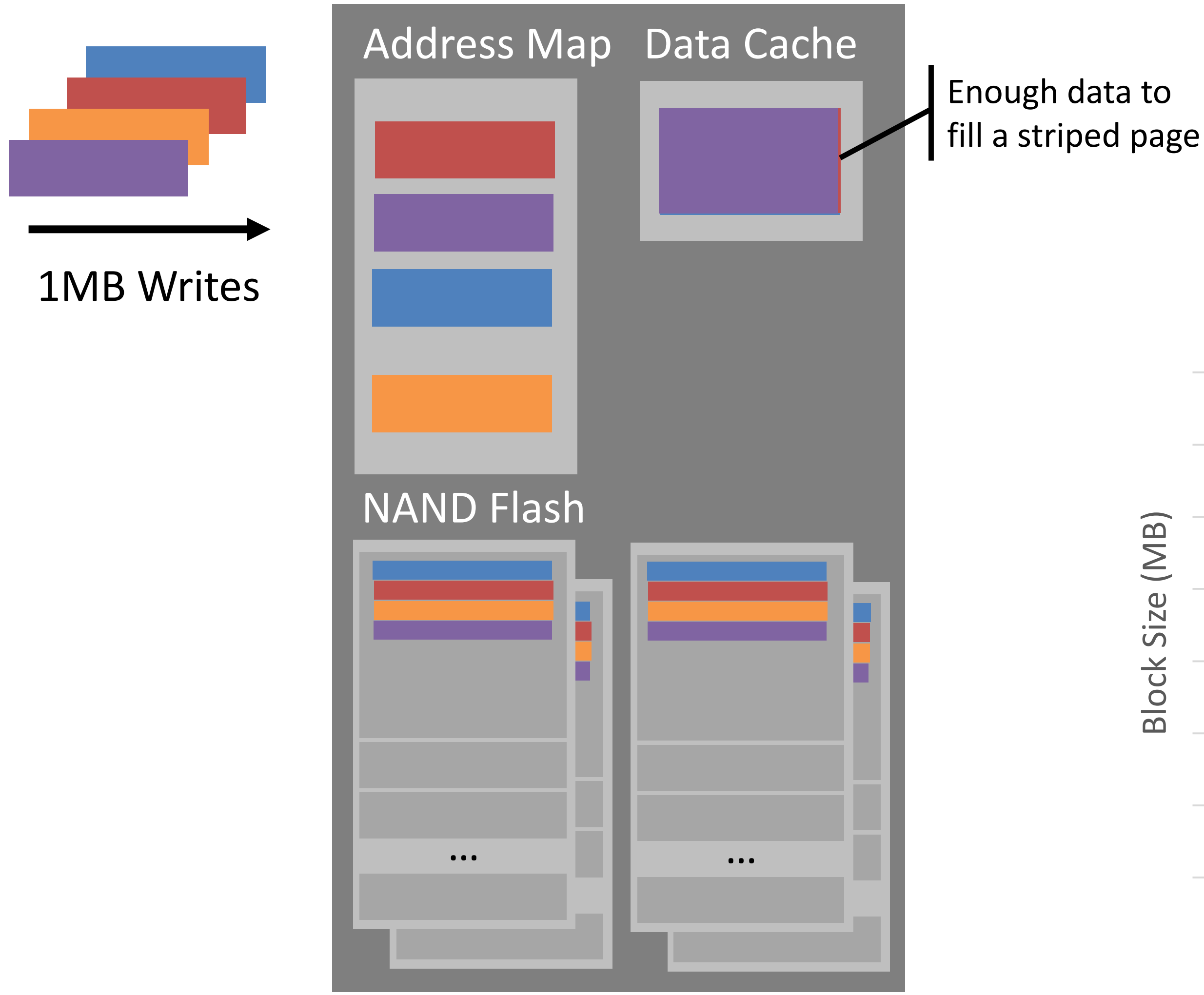
SSD Architecture



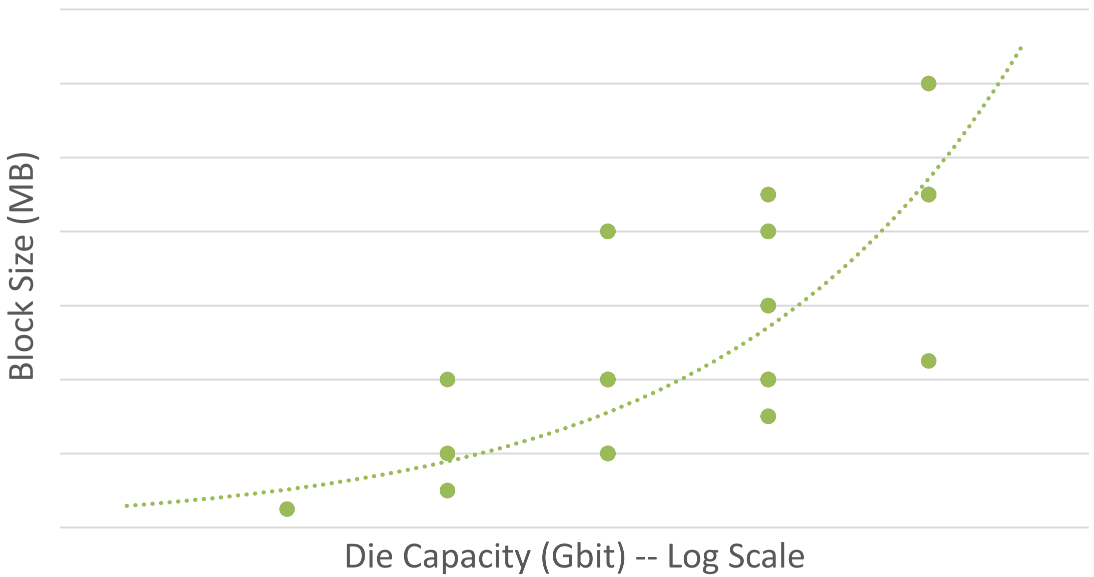
Attribute	Size
Flash Page	16kB
Flash Block	4MB - 9MB
Map Granularity	4kB



SSD Architecture



Attribute	Size
Flash Page	~~16kB~~ 1MB
Flash Block	~~4MB - 9MB~~ 1GB
Map Granularity	4kB



Microsoft

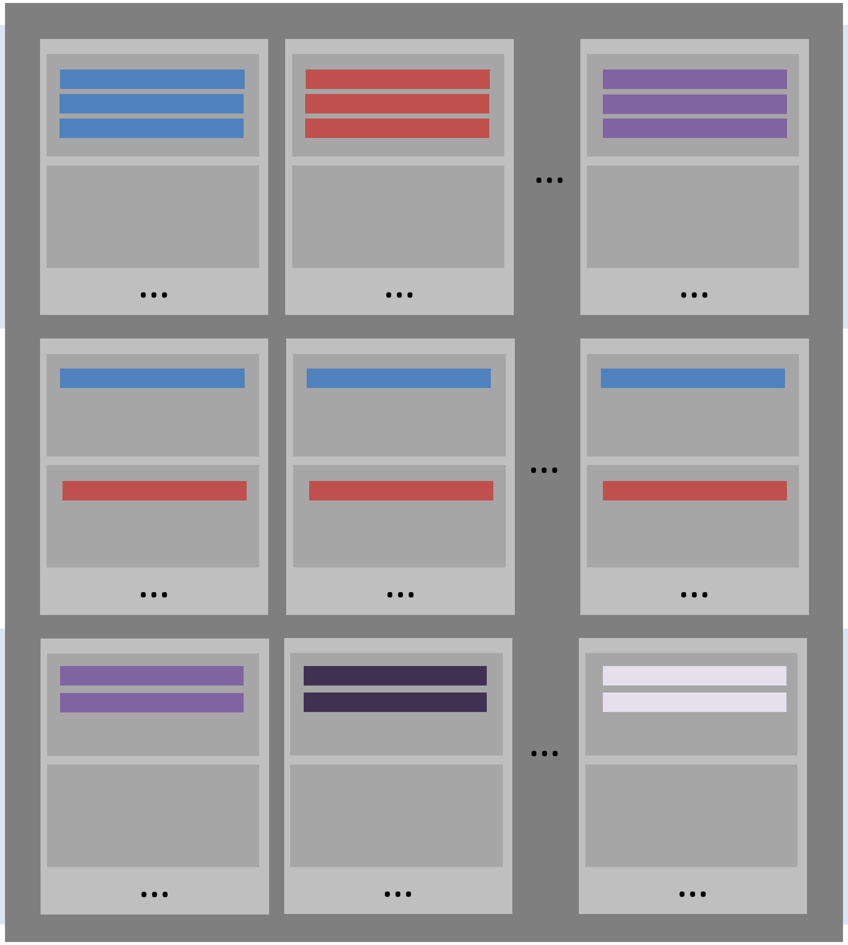
Cloud-Scale Workloads

What is the most efficient placement of their data in an SSD's NAND Flash Array?

Azure Storage Backend **(SOSP '11)**
- Lowest tier in hierarchy ("streaming")
- Write Perf. ↑, Stream Count ↑
- Read QoS via small reclaim unit

Application in Virtual Machine **(VM)**
- Small updates
- Unaligned Peak Traffic (Bursty)

New Application in VM
- Same resources as any VM guest
- Adaptable to flash sizes



Vertical Stripe
- High throughput through aggregation
- Smallest possible effective block size

Horizontal Stripe
- Each write receives peak performance
- Erase blocks when VM closes

Hybrid Stripe
- VM Host allocates horizontal stripe
- VM Guest partitions it further

Allow these and other stripe dimensions simultaneously in the same SSD

Outline

- Technology Trends & Application Requirements

- Proof-of-Concept

- Host-Drive Specification

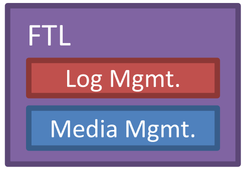
Denali SSD Architecture

Terminology

Open Channel SSD:
Drive exposes physical addresses such as channels

Denali SSD:
Drive exposes logical hierarchy of addresses that map to physical attributes

FTL (Flash Translation Layer):
Algorithms which allow SSD to replace conventional HDDs

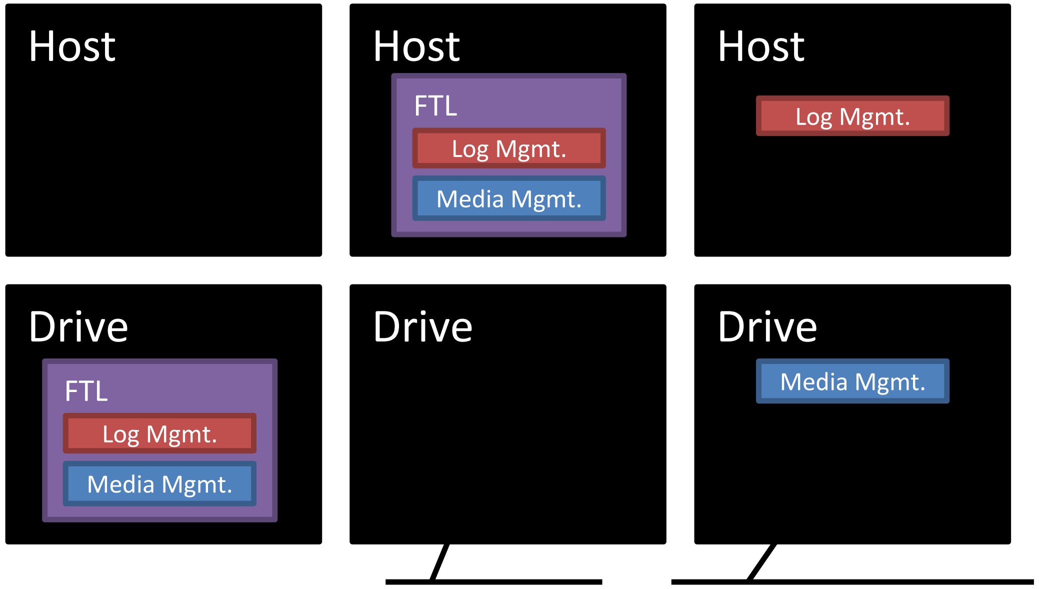


Log Manager:
Receives random writes
Transmits one or more streams of sequential writes
Maintains address map, performs garbage collection

Media Manager:
Written for a specific generation of media
Implements error correction such as ECC, RAID and read-retry
Prevents errors through scrubbing, mapping out bad blocks, etc.

Evolution of the Architecture

Standard SSD | Open Channel SSD | Denali SSD

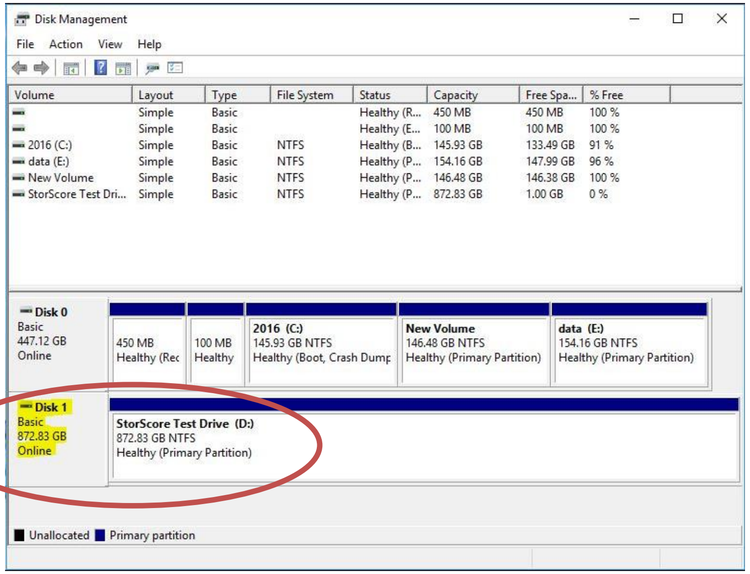


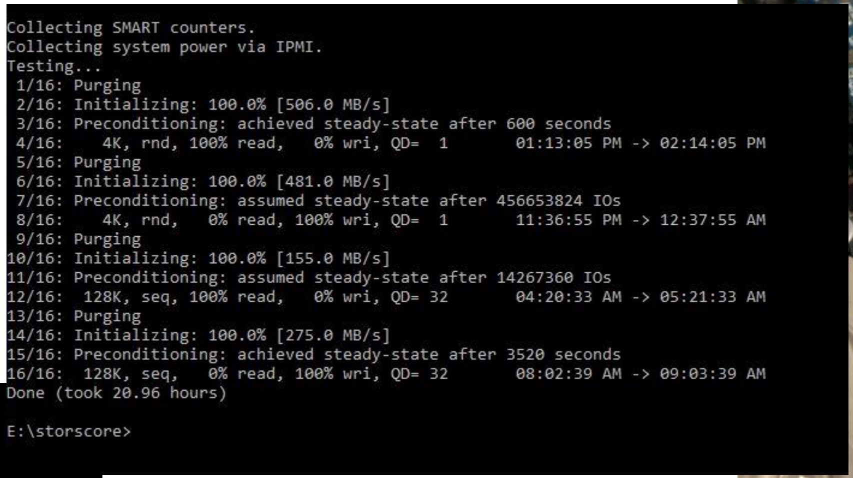
Prototype OCSSD 1.2

Production Ready OCSSD 2.0

✓ *POC Goal*
Migrate FTL
to Azure's kernel

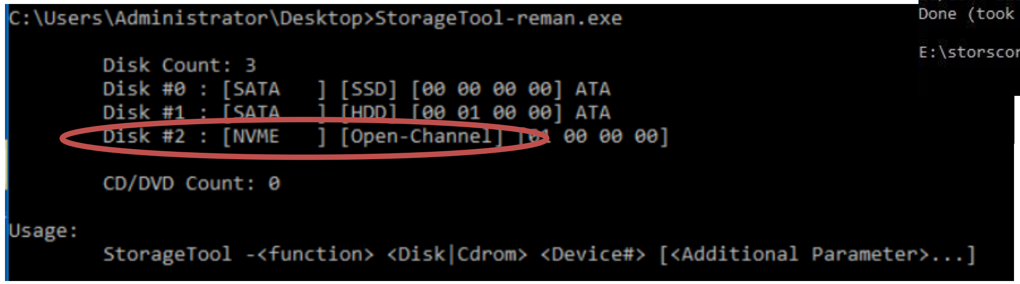
POC Test Configuration





```
Collecting SMART counters.
Collecting system power via IPMI.
Testing...
 1/16: Purging
 2/16: Initializing: 100.0% [506.0 MB/s]
 3/16: Preconditioning: achieved steady-state after 600 seconds
 4/16:     4K, rnd, 100% read,    0% wri, QD=  1        01:13:05 PM -> 02:14:05 PM
 5/16: Purging
 6/16: Initializing: 100.0% [481.0 MB/s]
 7/16: Preconditioning: assumed steady-state after 456653824 IOs
 8/16:     4K, rnd,   0% read, 100% wri, QD=  1        11:36:55 PM -> 12:37:55 AM
 9/16: Purging
10/16: Initializing: 100.0% [155.0 MB/s]
11/16: Preconditioning: assumed steady-state after 14267360 IOs
12/16:   128K, seq, 100% read,    0% wri, QD= 32        04:20:33 AM -> 05:21:33 AM
13/16: Purging
14/16: Initializing: 100.0% [275.0 MB/s]
15/16: Preconditioning: achieved steady-state after 3520 seconds
16/16:   128K, seq,   0% read, 100% wri, QD= 32        08:02:39 AM -> 09:03:39 AM
Done (took 20.96 hours)

E:\storscore>
```



```
C:\Users\Administrator\Desktop>StorageTool-reman.exe

        Disk Count: 3
        Disk #0 : [SATA    ] [SSD] [00 00 00 00] ATA
        Disk #1 : [SATA    ] [HDD] [00 01 00 00] ATA
        Disk #2 : [NVME    ] [Open-Channel] [01 00 00 00]

        CD/DVD Count: 0

Usage:
        StorageTool -<function> <Disk|Cdrom> <Device#> [<Additional Parameter>...]
```

✓ *POC Goal*

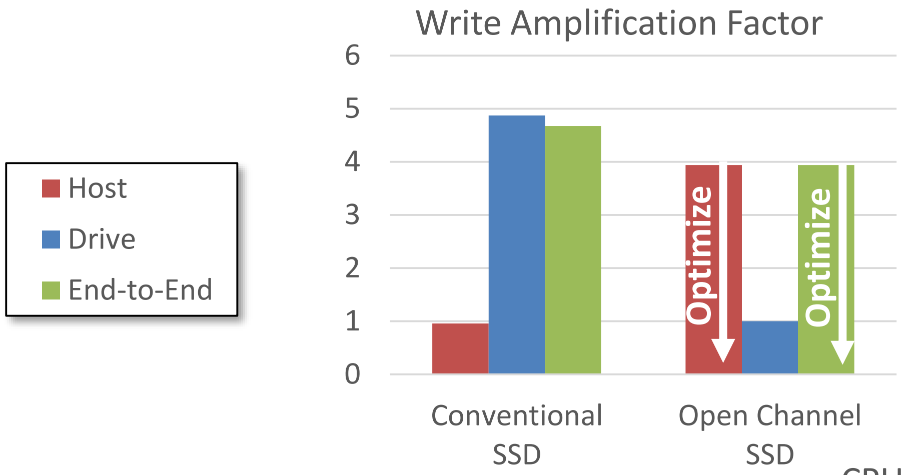
Quantify opportunity for optimization of resources

Results: Optimizing System’s Overheads

FW-based algorithms overheads are static, the host has information and flexibility to reduce them dynamically

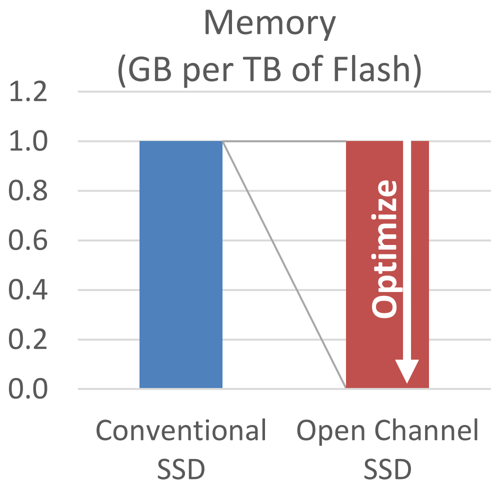
Write Amplification (4k Random Writes)

- Better end-to-end WAF – logic in FTL library is efficient
- Optimize host-side WAF using workload information



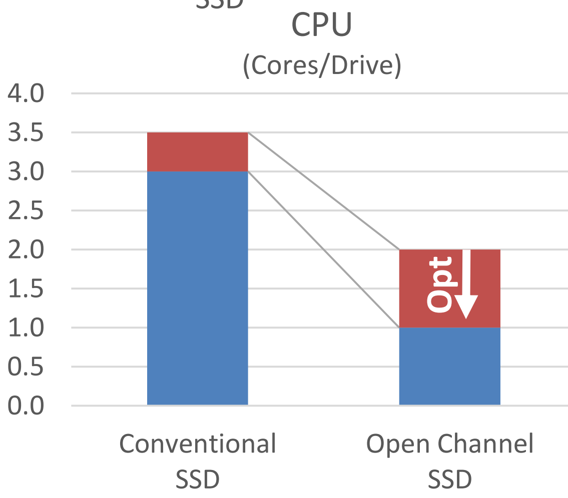
Memory

- 1GB of DRAM / TB of flash for address map
- Optimize map: sparse, granularity, dynamic allocation



CPU

- Implementation Specific Overheads in prototype
- Further optimization through end-to-end WAF reductions

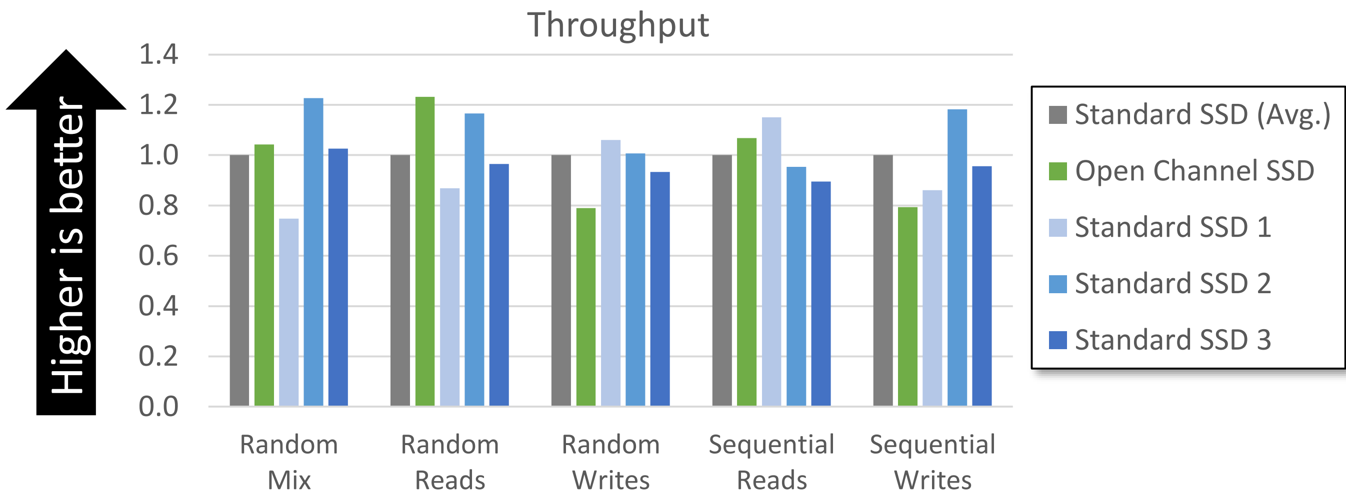


Results: Performance Parity

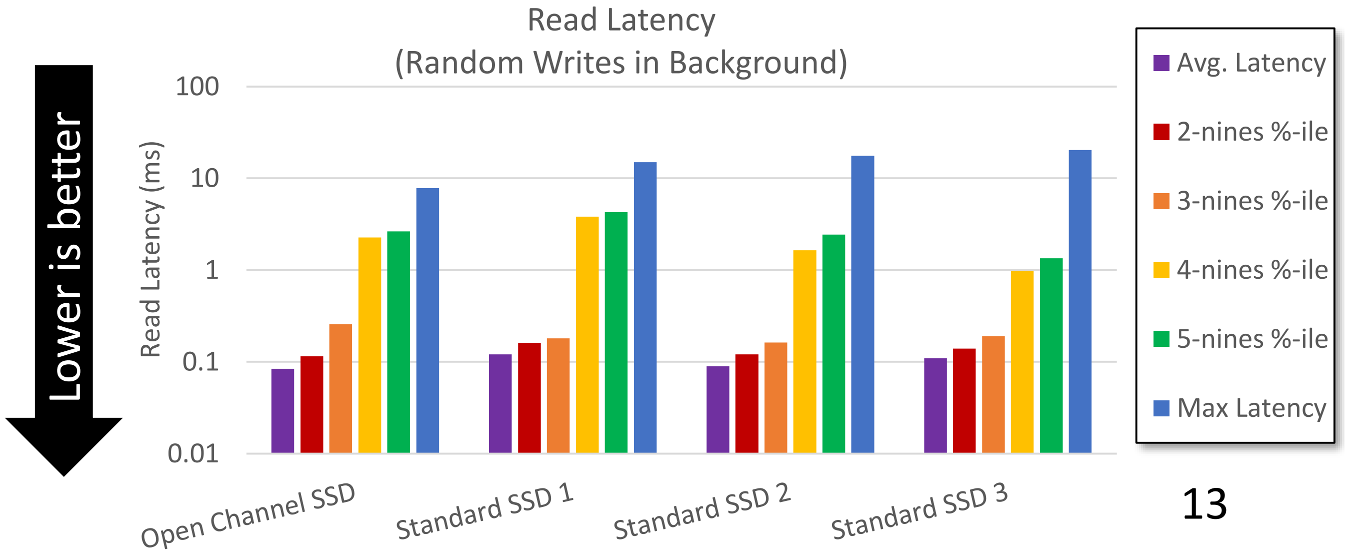
✓ *POC Goal*

Remain competitive with conventional SSDs' performance

- Workloads:
  - Seq: 4 threads, QD 32, 128kB
  - Mix: 4 threads, QD 4, 4kB
  - Rand: 4 threads, QD 32, 4kB, 70/30
- Read Perf.: Top in Class
- Write Perf.: Pending Typical Optimizations



- Workload:
  - Measured: 4kB random reads, QD 1
  - Background: 256kB random writes
- Top-in-class



Outline

- Technology Trends & Application Requirements
- Proof-of-Concept
- Host-Drive Specification

“Open-Channel Solid State Drives NVMe Specification” Matias Bjørling, Oct 2016 https://aka.ms/i9my03

Logical Hierarchical Addressing

- Each field maps to logical part of architecture
  - Flexibility in HW to manage NAND (such as mapping out bad blocks)
  - System can implement 2-part wear leveling*
  - Overheads significantly lower than conventional SSDs

- Host IO Requirements
  - Allocate a fresh a chunk before writing any sectors
  - Write sectors within the chunk sequentially
  - Some new elements to abstract NAND management, for example, the cache minimum write size

Address Format:

MSB ... LSB

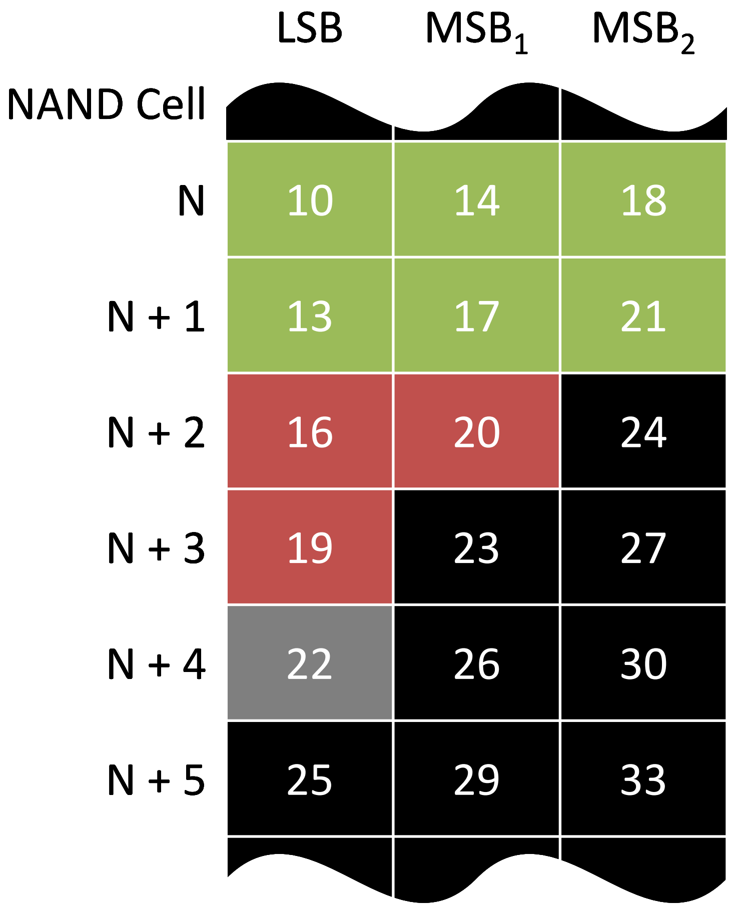
Group	Parallel Unit	Chunk	Sector

Group: SSD Channel
Parallel Unit (PU): NAND Die
Chunk: multi-plane block
Sector: 512B or 4k region of NAND page

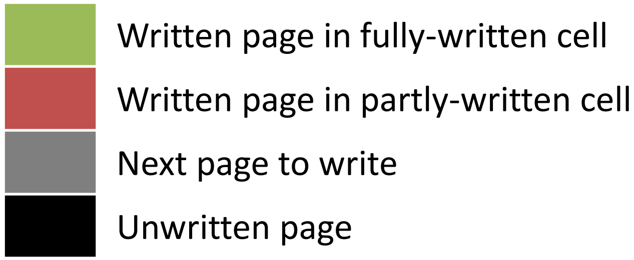
* “FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs” Huang et al, USENIX-FAST 2017

Cache Minimum Write Size (CMWS)

Defining a logical abstraction for an idiosyncrasy of NAND flash physics



- Open NAND cells susceptible to read disturb
- Example: Cache the last 3-5 pages written to any write point
- Host-Device Contract:
  - CMWS = max kB in open cells
  - CMWS = 0kB if drive caches to mitigate the effect
  - Host queries for CMWS
  - Drive fails reads to CMWS region



"Vulnerabilities in MLC NAND Flash Memory Programming." Yu Cai et al. HPCA 2017

Reliability and QoS

This is the same challenge that the IO Determinism community is working to solve.

- RAID and isolation are at odds

  (Small tenant == high RAID overheads)

- Mechanism must enable spectrum of users
  - Many tenants use cross server replication, don’t require RAID
  - Some require standard reliability

- Solution: IO Determinism’s Read Recovery Levels

Microsoft

Conclusions

- Let's architect the new storage interface for the long term
  - Correct division of responsibilities between Host and SSD
  - Control to define heterogeneous block stripes
  - HyperScale: Hundreds or thousands of workers per TB
- Successful proof-of-concept
  - System overheads: as expected & ready for optimization
  - Performance parity on standard microbenchmarks
  - Next step: complete interface for warrantable Open-Channel SSD
- Final solution must include expertise from community
  - Currently working through the division between host and SSD
  - Contact us to discuss
  - Read more in our FAST 2017 paper: FlashBlox

OCP SUMMIT

Microsoft

References

- Azure Storage Backend (SOSP '11)
  Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency
- FlashBlox (FAST '17)
  FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs
- LightNVM (FAST '17)
  LightNVM: The Linux Open-Channel SSD Subsystem
- Read Determinism (SDC '16)
  Standards for improving SSD performance and endurance
- Software-Defined Flash (ASPLOS '14)
  SDF: Software-Defined Flash for Web-Scale Internet Storage Systems
- Multi-Streamed SSD (HotStor '14)
  The Multi-streamed Solid-State Drive
- De-Indirection (FAST '12)
  De-Indirection for Flash-based SSDs with Nameless Writes
- Programmable Flash (ADMS '11)
  Fast, Energy Efficient Scan inside Flash Memory SSDs