SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Providing Efficient Storage Operations for Both Data Centers and Hyperscale Applications

Bill Martin

Samsung

Disclaimer

This presentation and/or accompanying oral statements by Samsung representatives collectively, the "Presentation") is intended to provide information concerning the SSD and memory industry and Samsung Electronics Co., Ltd. and certain affiliates (collectively, "Samsung"). While Samsung strives to provide information that is accurate and up-to-date, this Presentation may nonetheless contain inaccuracies or omissions. As a consequence, Samsung does not in any way guarantee the accuracy or completeness of the information provided in this Presentation.

This Presentation may include forward-looking statements, including, but not limited to, statements about any matter that is not a historical fact; statements regarding Samsung's intentions, beliefs or current expectations concerning, among other things, market prospects, technological developments, growth, strategies, and the industry in which Samsung operates; and statements regarding products or features that are still in development. By their nature, forward-looking statements involve risks and uncertainties, because they relate to events and depend on circumstances that may or may not occur in the future. Samsung cautions you that forward looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements in this Presentation. In addition, even if such forward-looking statements are shown to be accurate, those developments may not be indicative of developments in future periods.

What is the requirement?

- Consistent latency
  - Eliminate the long tail
  - Prevent impact from background operations
    - Garbage Collection
    - Read Disturb – rewrite
  - Prevent impact from noisy neighbors

Alternate approaches

- Open Channel
- IO Determinism
- Other options

Open Channel Description

- Protocol that allows the host control over the placement of data
- Host discovers the device configuration
  - Channels
  - Die
  - other
- Specifies physical location in the data transfer phase
- Host manages background operation on the device
  - Read Disturb re-write
  - Garbage Collection
  - Component failure
  - Other
- Host prevents noisy neighbor
  - Manages what device(s) communicates with a specific resource at any given time

Open Channel Benefits

- Allows application to determine the applications best optimization of SSD
- Application is in control of when background operations are performed
- Application controls access to specific resources
- Application determines what resource conflict causes performance degradation

Open Channel Implications

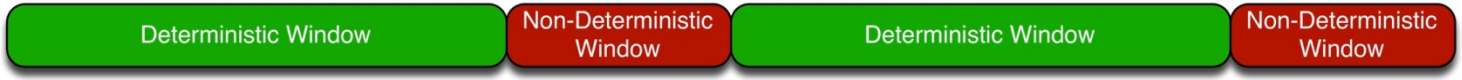
- Application must be enhanced for EVERY SSD
  - Technology
    - NAND Flash
    - 3D NAND Flash
    - 3D XPoint
    - Other
  - Configuration
    - Number of independent regions
    - Constraints on background operation
    - Other
- Application must maintain a lookup table in addition to any FTL on the device
- Device may have to do some activities in spite of management by application management
  - Negates some of the application management
- Host processing used for something that device has processor power to accomplish
- Application must be aware of all neighbors
- Application must change from current implementation

Device considerations for Open Channel

- What requirements does a particular device have for re-writing data
  - Frequency
  - Read/Write impact
- What requirements for garbage collection
  - Block Size
  - Block configuration
    - Erase block
    - What components are part of an erase block
- How are physical blocks accessed
  - What interaction between reads/writes are implied
    - Channel
    - Die
    - Other
  - How are these constraints communicated

IO Determinism
Description

- Define NVM Sets that provide isolation
- NVM Set provides:
  - Deterministic read periods
  - Non-deterministic period
  - Isolation of any operation in one set impacting reads in another set



Deterministic read period

- Remains in this period for:
  - Maximum number of reads
  - Maximum number of writes
  - Time before maintenance is required

Non-deterministic read period

- Remains in this period for:
  - Minimum maintenance time
- In this period the device (dependent on technology):
  - Writes any buffered data
  - Performs garbage collection
  - Performs re-writes required because of read-disturb

Control of deterministic/non-deterministic periods

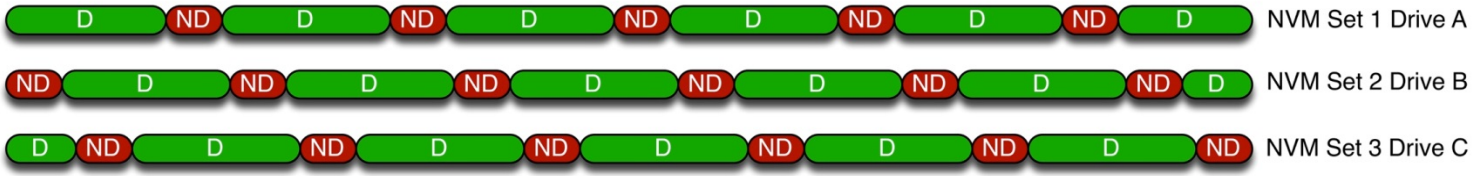
- Host retrieves configuration from the device
- Host puts the device in the deterministic window
  - Host does not perform more than the specified number of reads or writes
  - Host does not leave the device in this window for greater than the specified maximum time
- Host takes device out of the deterministic window
- Device may transition from the deterministic window to the non-deterministic window for extraordinary events
- Host does not transition device out of the non-deterministic window until the minimum required time has expired

Device responsibilities in deterministic period

- Provide deterministic read latency by:
  - Holding off background tasks
  - Perform up to the maximum specified writes without impacting read latency
  - Optionally fail fast for reads with read recovery errors

How does a host use this?

- The host orchestrates the timing of reads and writes for objects to multiple drives for redundancy and availability
- The scheduling is time based with exceptions driven by the drive
- To write data, each drive is put into ND for one or more writes and then back into D for reads
- Reads may be scheduled for multiple drives in D to increase performance



IO Determinism Benefits

- Allows SSD vendors to add value due to knowledge of technology/configuration
  - As technology changes device vendors know the implications of those changes
  - Device vendors can tune performance as device characteristics change
  - Technology does not have to report unnecessary characteristics
- Uses compute power already present on device to manage device

IO Determinism Implications

- Device must communicate additional characteristics to the application
  - These are generic not technology/configuration specific
- Application must identify data associations to avoid performance implications of Reads vs Writes
- Application must be re-written to take advantage of the potential performance improvements

Device considerations for IO Determinism

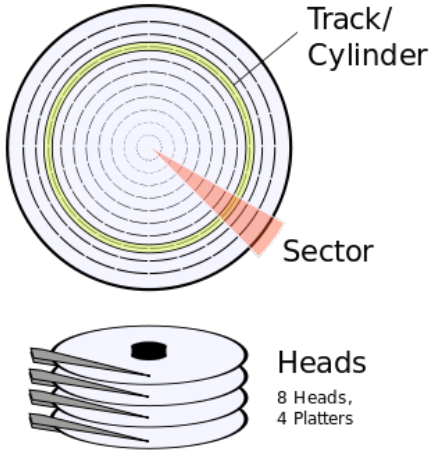
- What does the device communicate to the application
  - Generalized to be technology/configuration agnostic
- How do you report a generalized requirement by the application for specific class of latency
  - Per read/write?
- How do you communicate a device requirement to perform tasks that may impact latency
  - Interrupt
  - Polling
  - Time based

Alternative approaches

- Reduced latency technologies
  - Inherently without Tail Latency
  - With short enough latency that Tail Latency is within the requirements of the application
- “Tiny-Tail Flash: Near-Perfect Elimination of Garbage Collection Tail Latencies in NAND SSDs”
  - Shiqin Yan, et.al., University of Chicago, Fast17 proceedings

Historical perspective

- HDD industry started with physical addressing
  - Cylinder/Head/Sector
  - As media density grew, devices reported a logical geometry to increase addressability
  - Eventually abstracted to Logical Blocks
- Don’t repeat the mistakes of the past
  - Don’t require devices to stay with a configuration to avoid software changes
  - Don’t require devices to report false information to account for advancements in device technology



Call to action

- NVMe technical committee is currently developing IO Determinism
  - Participate in defining how applications pick the right solution
- Allow devices to provide the latency that applications require
  - Do not re-write code for every change in SSD products

Thank You