SDC 18
SNIA EMEA
FEBRUARY 2018
TEL AVIV, ISRAEL
STORAGE DEVELOPER CONFERENCE

Achieving Predictable Latency for Solid State Drives

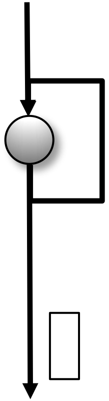
Mark Carlson
Toshiba

Agenda

- Hyperscaler Datacenter requirements on SSDs
- IO Determinism Concepts
- NVM Sets
- Endurance Groups
- Read Recovery Levels
- Namespace Management
- Predictable Latency
  - Mode Config/Window
  - Eventing
  - Windowing
- Host use case
- Futures

Tail Latency Impact at Scale

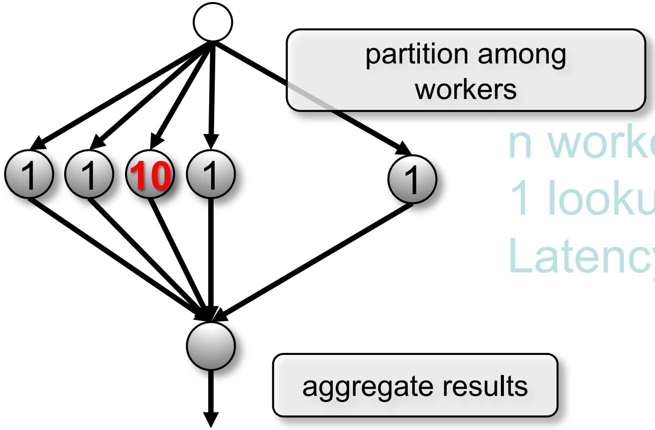
Legacy Mindset



1 worker
n iterations
Latency O(n)

Execution time ≈ *Sum of* lookup latency

Hyperscale Mindset



n workers
1 lookup each
Latency O(1)

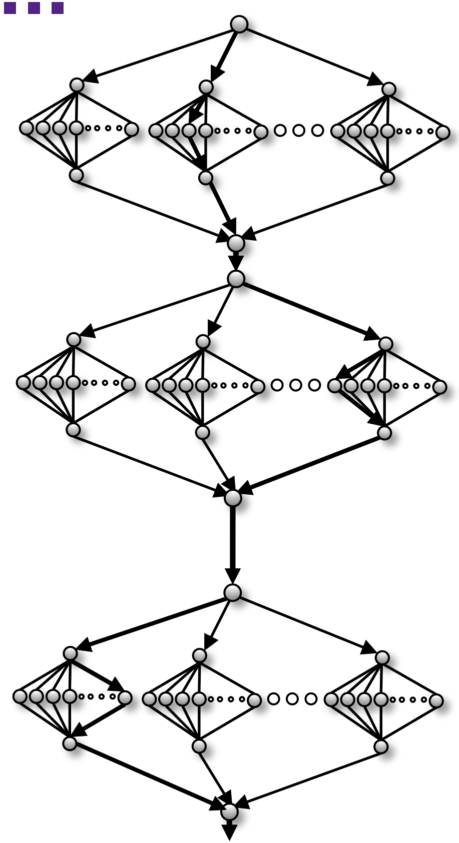
Execution time ≈ *Longest* lookup latency

Real Implementations are...

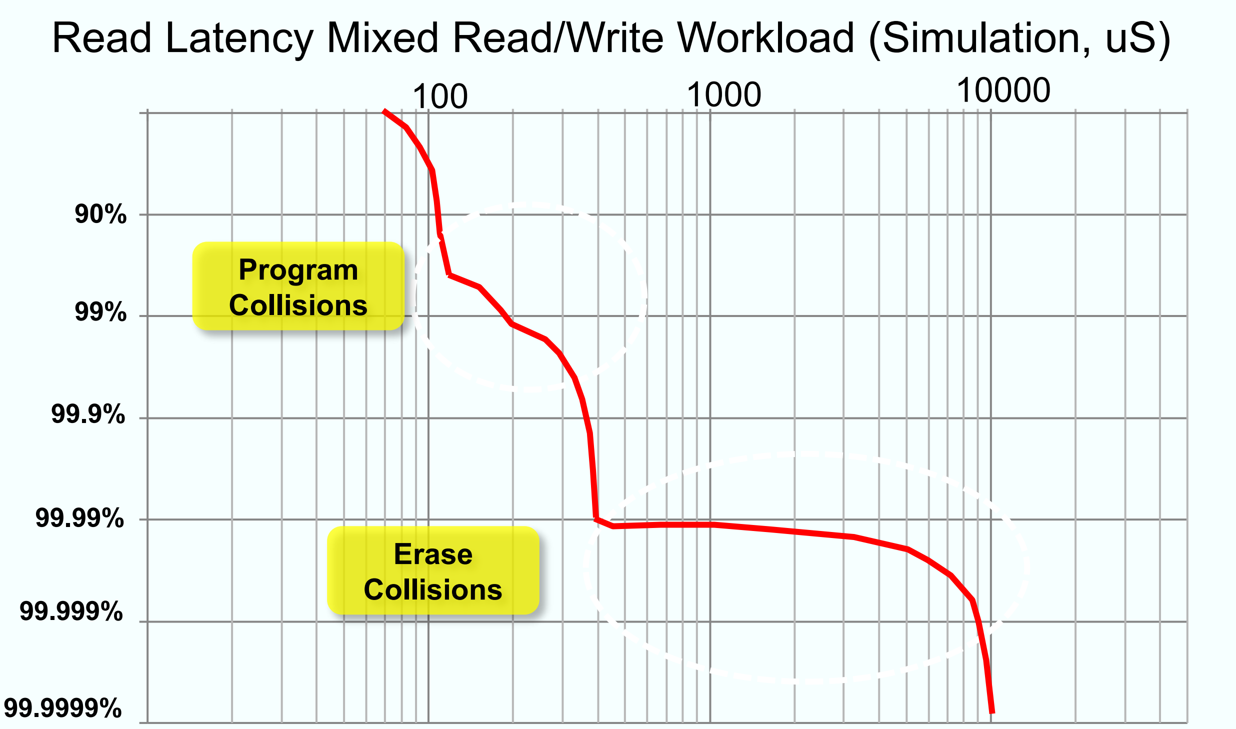
“In practice, a single user request may result in thousands of subqueries, with a critical path that
is dozens of subqueries long.”

“The fork/join structure of subqueries causes latency outliers to have a disproportionate effect on total latency, and the large number of subqueries would cause slowdowns or unavailability to quickly propagate…”

Challenges to Adopting Stronger Consistency at Scale
- Ajoux et. Al., (Facebook & USC)



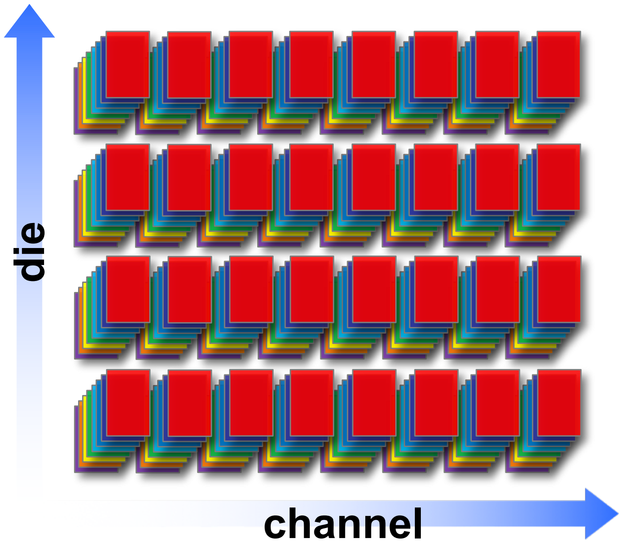
The Maintenance of an SSD

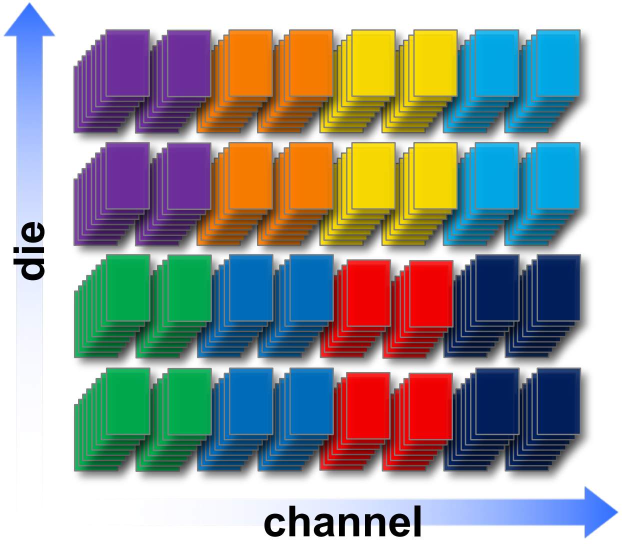


IO Determinism

- The idea is to limit and possibly eliminate long tails in Datacenter Scale Out Storage Systems
- Two major requirements
  1. Noisy neighbor interference with large capacity drives (> 1TB)
  2. Background tasks result in 10x or longer latency tails
- First concept: isolate raw capacity into individual NVM Sets (1.)
- Second concept: hold off background tasks during special read window (2.)
- Additionally: fail fast when recovery is necessary for deterministic reads
- The Host is expected to behave in a certain manner
  - Do mostly reads during the Deterministic Window (may accommodate writes)
  - Allow periodic Non-Deterministic Windows to do maintenance, and writes
  - Orchestrate the windowing across multiple drives if needed

NVM Sets Architecture – Achieving Isolation





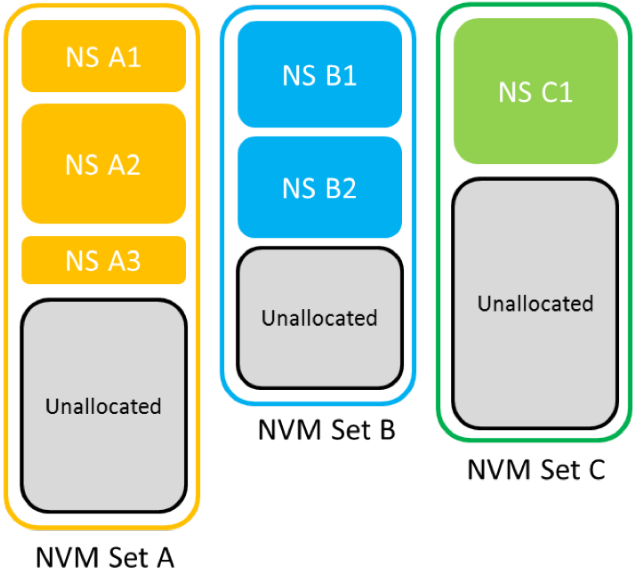
- Classic SSD architecture uses “bands” of devices on every channel to maximize bandwidth. Maintenance is also on every die on every channel
- New SSD array

NVM Sets

- An NVM Set is a collection of NVM that is separate (logically and potentially physically) from NVM in other NVM Sets.
- One or more namespaces may be created within an NVM Set and those namespaces inherit the attributes of the NVM Set.
- A namespace is wholly contained within a single NVM Set and shall not span more than one NVM Set.
- There is a subset of Admin commands that are NVM Set aware

Namespaces and NVM Sets

- Example:
  - NVM Set A contains three namespaces (NS A1, NS A2, and NS A3).
  - NVM Set B contains two namespaces (NS B1 and NS B2).
  - NVM Set C contains one namespace (NS C1).
- Each NVM Set shown also contains 'Unallocated' regions that consist of NVM that is not yet allocated to a namespace.
- All Namespaces in an NVM Set transition windows together for Predictable Latency



Namespaces in a NVM Set

- Each NVM Set is associated with exactly one Endurance Group (refer to later slides).
- The NVM Set with which a namespace is associated is reported in the Identify Namespace data structure.
  - When a host creates a namespace using the Namespace Management command, the host specifies the NVM Set Identifier of the NVM Set that the namespace is to be created in.
  - The namespace that is created inherits attributes from the NVM Set (e.g., the optimal write size to the NVM).

Required Features

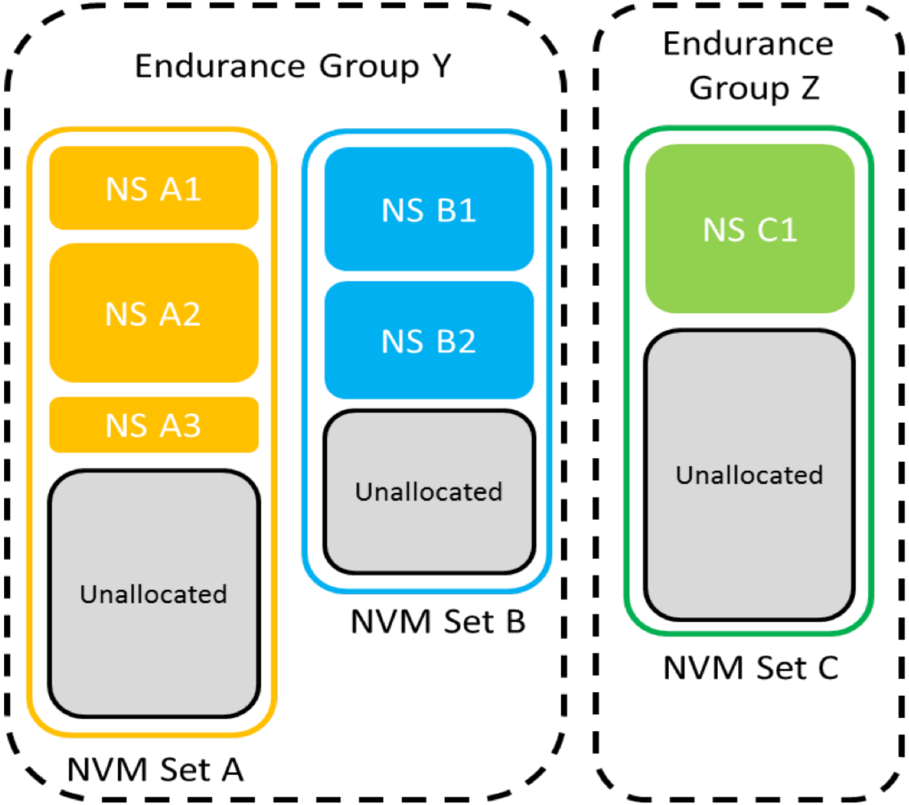
- If NVM Sets are supported, then the NVM subsystem and all controllers shall:
  - Indicate support for NVM Sets in the Controller Attributes field in the Identify Controller data structure;
  - Support the NVM Set Identifier in all commands that use the NVM Set Identifier;
  - Support the NVM Set List for the Identify command;
  - Indicate the NVM Set Identifier with which the namespace is associated in the Identify Namespace data structure;
  - Support Endurance Groups; and
  - For each NVM Set, indicate the associated Endurance Group as an attribute.

Endurance Groups

- Endurance may be managed within a single NVM Set or across a collection of NVM Sets.
  - Wear leveling across Sets is not required
- Each NVM Set is associated with an Endurance Group.
  - If two or more NVM Sets have the same Endurance Group Identifier, then endurance is managed by the NVM subsystem across that collection of NVM Sets.
  - If only one NVM Set is associated with a specific Endurance Group Identifier, then endurance is managed locally to that NVM Set. Thus the host needs to wear level across the NVM Sets if desired.
- The endurance information for an Endurance Group is specified in the Endurance Group Information log page

Example Endurance Groups

- In this example, the endurance of NVM Set A and NVM Set B are managed together as part of Endurance Group Y.
- The endurance of NVM Set C is managed only within NVM Set C as it is the only NVM Set that is part of Endurance Group Z.

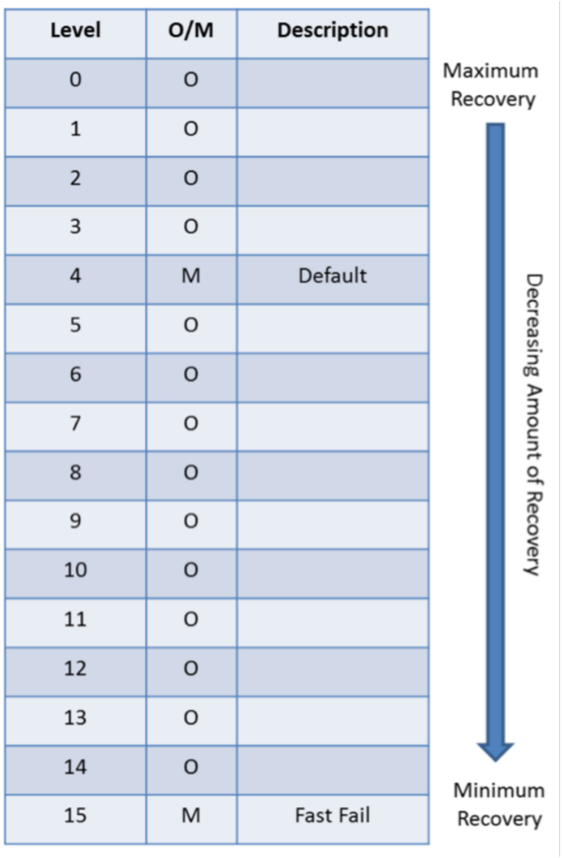


Read Recovery Level

- The Read Recovery Level (RRL) is a configurable attribute that balances the completion time for read commands and the amount of error recovery applied to those read commands.
  - The Read Recovery Level applies to an NVM Set with which it is associated.
  - A namespace created within an NVM Set inherits the Read Recovery Level of that NVM Set.
- The controller indicates support for Read Recovery Levels in the Controller Attributes field in the Identify Controller data structure.
  - If Read Recovery Levels are supported, then the specific levels supported are indicated in the Read Recovery Levels Supported field in the Identify Controller data structure.
  - There are 16 levels that may be supported.
- Level 0, if supported, provides the maximum amount of recovery.
- Level 4 is a mandatory level that provides a nominal amount of recovery and is the default level.
- Level 15 is a mandatory level that provides the minimum amount of recovery and is referred to as the ‘Fast Fail’ level.
- The levels are organized based on the amount of recovery supported, such that a higher numbered level provides less recovery than the preceding lower level.

Recovery Level – Fast Fail

- What does “Fast Fail” Actually mean?
- It is NOT a timeout
- At the first indication of the need for recovery:
  - Don’t retry with different V_t
  - Don’t apply ECC correction
  - Don’t apply parity correction
- Fail the command immediately
- The host will use another copy or shard to complete the application request
- The host may set the level to 4 in order to tell the NVM Set that heroic recovery is needed before retrying the read command



Level	O/M	Description
0	O	
1	O	
2	O	
3	O	
4	M	Default
5	O	
6	O	
7	O	
8	O	
9	O	
10	O	
11	O	
12	O	
13	O	
14	O	
15	M	Fast Fail

Maximum Recovery

Decreasing Amount of Recovery

Minimum Recovery

Namespace Management

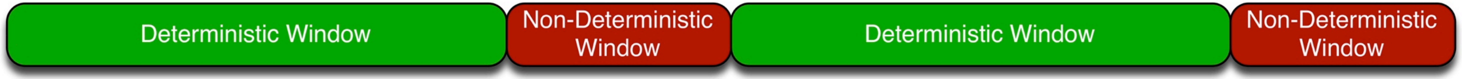
- For controllers that support NVM Sets, the total and unallocated NVM capacity for each NVM Set is reported as part of the NVM Set Attributes Entry.
  - For each namespace, the NVM Set Identifier that the namespace is allocated in is reported in the Identify Namespace data structure.
  - The NVM Set to be used for a namespace is based on the value in the NVM Set Identifier field in a create operation.
  - If the NVM Set Identifier field is cleared to 0h in a create operation, then the controller shall choose the NVM Set from which to allocate the namespace.

Predictable Latency

- Predictable Latency Mode is used to achieve predictable latency for read and write operations.
  - When configured to operate in this mode using the Predictable Latency Mode Config Feature, the namespaces in an NVM Set provide windows of operation for deterministic operation or non-deterministic operation.
- NVM Sets and their associated namespaces have quality of service attributes that are provided when Predictable Latency Mode is enabled.
  - These are vendor specific.
  - The quality of service properties are isolated between commands to namespaces on different NVM Sets when Predictable Latency Mode is enabled.
  - The quality of service attributes are specified from the NVM subsystem port to the NVM, and thus are not impacted by the PCIe or fabric connection to the NVM subsystem.
- Read Recovery Levels shall be supported when Predictable Latency Mode is supported.
  - The host configures the Read Recovery Level to specify the quality of service desired versus the amount of error recovery to apply for a particular NVM Set.

Windowing Controls Background Tasks

- An NVM Set that implements Predictable Latency supports two windows of activity from the host
  - Deterministic Window – background tasks are held off from running providing the most predictable read latency
  - Non-Deterministic Window – background tasks and periodic maintenance are allowed to run
- Log Page parameters are provided by the NVM Set to govern the switching of Windows
  - The Host may subscribe to notifications for parameter thresholds



Deterministic Window

Non-Deterministic Window

Deterministic Window

Non-Deterministic Window

The Windows

- The Deterministic Window (DTWIN) is the window of operation during which the NVM Set is able to provide deterministic latency for read and write operations.
- The Non-Deterministic Window (NDWIN) is the window of operation during which the NVM Set is not able to provide deterministic latency for read and write operations as a result of preparing for a subsequent Deterministic Window.
  - Examples of actions that may be performed in the Non-Deterministic Window include repair and background operations on the non-volatile media.
- The current window that an NVM Set is operating in is configured by the host using the Predictable Latency Mode Window Feature or by the controller as a result of an autonomous action.

Well Behaved Host

- To remain in the Deterministic Window, the host is required to follow operating rules ensuring that certain attributes remain above required thresholds.
  - If the attributes fall below the thresholds or a Deterministic Excursion occurs, then the associated NVM Set may autonomously transition to the Non-Deterministic Window.
  - A Deterministic Excursion is a rare excursion by the NVM subsystem that requires immediate action by the controller.

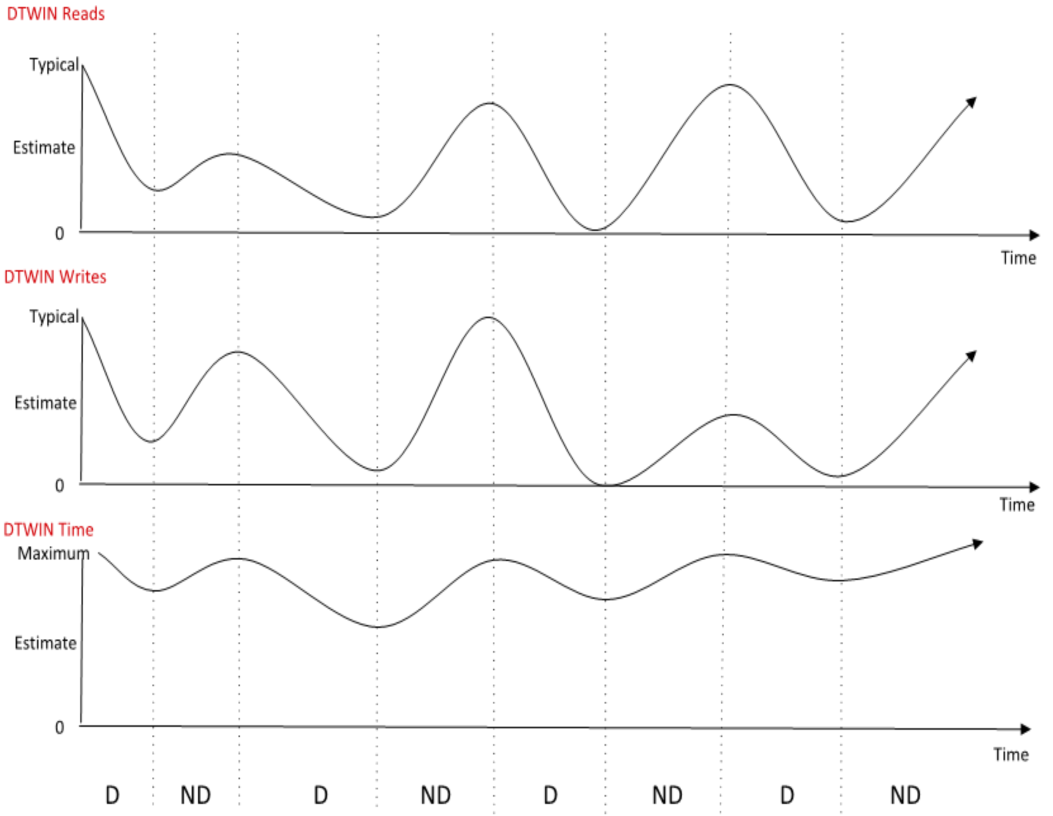
- The host may configure events to be triggered when thresholds fall below certain levels using the Predictable Latency Mode Config feature.
  - The events enable the host to manage window changes and avoid unnecessary autonomous transitions.

Operating Rules for “Well Behaved”

- An NVM Set remains in the Deterministic Window while attributes are above required thresholds and there is not a Deterministic Excursion.
  - The attributes specified in this specification are the number of random 4KB reads, the number of writes in Optimal Write Size, and time in the Deterministic Window. Additional attributes are vendor specific.
- For reads, writes, and time in the Deterministic Window, two values are provided in the Predictable Latency Mode log page:
  - A typical amount of that attribute that the host may consume during any given DTWIN.
  - A reliable estimate of the amount of that attribute that remains to be consumed during the current DTWIN.

How does the FW produce these values?

- The reliable estimates provided shall have the following properties when in the Deterministic Window:
  - The estimates shall be monotonically decreasing towards 0h for the entirety of the DTWIN, depending on the attribute.
  - For example, DTWIN Reads Reliable Estimate is monotonically decreasing and thus does not increase without transitioning from the DTWIN to the NDWIN.
- The estimates shall not change abruptly unless operating conditions have changed abruptly.
  - The estimate should be based on averaging or smoothing of data collected over some period of time.



Controlling the Windowing

- The host configures the current window to be either DTWIN or NDWIN using Set Features with the Predictable Latency Mode Window Feature.
- The host may use the reliable estimates provided in the Predictable Latency Mode log page to ensure the host transitions the NVM Set to the NDWIN prior to any reliable estimates crossing one of the thresholds (e.g., DTWIN Reads Estimate = 0).
- When using the NVM Set in Predictable Latency Mode, the host should transition the controller to NDWIN for periodic maintenance. The maintenance is required in order for the NVM subsystem to reliably provide the amount of time indicated for Deterministic Windows.
- The DTWIN Time Estimate may be used by the host when a Deterministic Excursion has occurred. This estimate allows the host to re-synchronize an NVM Set with other NVM Sets operating in Predictable Latency Mode, if applicable.

Time based parameters

- There are three static time based parameters reported in the Predictable Latency Per NVM Set log page that may be used by the host to configure periodic windows.
  - NDWIN Time Minimum Low is the minimum time the controller remains in the Non-Deterministic Window. The controller may delay completion of a Set Features requesting a transition to the Deterministic Window until this time is completed.
  - NDWIN Time Minimum High is the minimum time that the host should allow the NVM Set to remain in the Non-Deterministic Window after the NVM Set remained in the previous Deterministic Window for DTWIN Time Maximum. This time does *not* account for additional host activity in the Non-Deterministic Window.
  - DTWIN Time Maximum is the maximum time that the NVM Set is able to stay in a Deterministic Window.
- The DTWIN Time Maximum and NDWIN Time Minimum High may provide a ratio of the amount of maintenance that needs to be performed based on the time that the NVM Set remains in the DTWIN.

Transitions

- An NVM Set may transition autonomously to the NDWIN if since entry to the current DTWIN:
  - the number of reads is greater than the value indicated in the DTWIN Reads Typical field;
  - the number of writes is greater than the value indicated in the DTWIN Writes Typical field;
  - the amount of time indicated in the DTWIN Time Maximum field has passed; or
  - a Deterministic Excursion occurs.
- The host may configure events to be triggered when thresholds fall below certain levels or when autonomous transitions occur using the Predictable Latency Mode Feature.
  - The host submits a Set Feature for the particular NVM Set and configures the specific event(s) and threshold(s) values that shall trigger an event to the host.

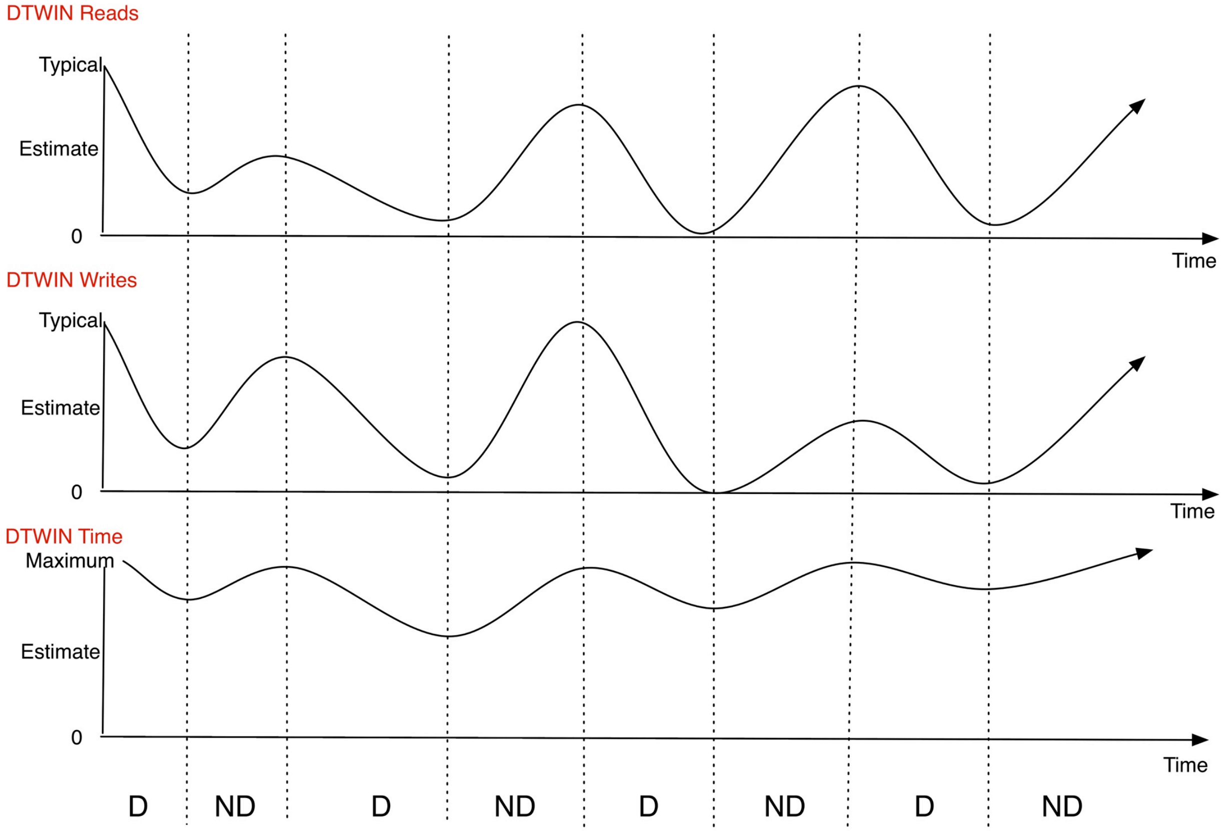
Reliable Estimates

Predictable Latency Per NVM Set Log

Bytes	Description
135:128	**DTWIN Reads Estimate:** Indicates a reliable estimate of the number of 4KB random reads remaining in the current Deterministic Window, if applicable. This value decrements from DTWIN Reads Typical to zero based on host read activity and operating conditions.
143:136	**DTWIN Writes Estimate:** Indicates a reliable estimate of the number of writes in units of the Optimal Write Size remaining in the current Deterministic Window, if applicable. This value decrements from DTWIN Writes Typical to zero based on host write activity and operating conditions.
151:144	**DTWIN Time Estimate:** Indicates a reliable estimate of the time in milliseconds remaining in the current Deterministic Window, if applicable.
511:152	Reserved

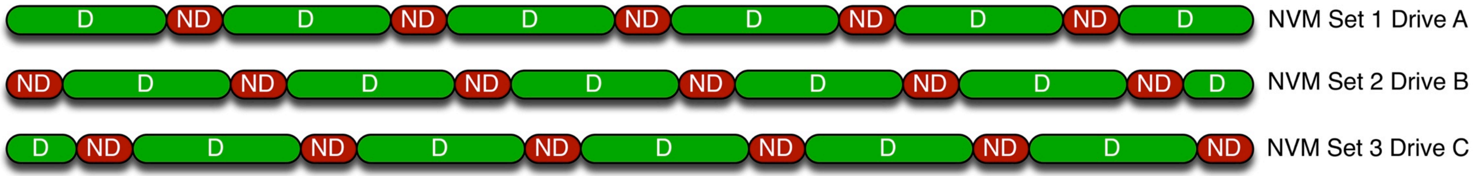
Predictable Latency Parameters

- The three parameters that affect maintenance work from the drive are: Reads, Writes, and Time.
- Host is given Typical or Maximum numbers for each of these
- Estimates allow the host to remain in synchronization with the drive



Orchestrating the Windows

- The host can orchestrate the timing of reads and writes for objects to multiple drives for redundancy and availability
- The scheduling can be primarily time based with exceptions driven by the drive (i.e. Fast Fail)
- To propagate a new object, each drive in turn is put into ND for the writes, then back into D
- Reads may be scheduled for multiple drives in D to increase performance


D ND D ND D ND D ND D ND D NVM Set 1 Drive A
ND D ND D ND D ND D ND D ND D NVM Set 2 Drive B
D ND D ND D ND D ND D ND D ND NVM Set 3 Drive C

Sharding and Erasure Coding

- Using 3 full copies of the data is expensive at scale so many organizations use a scheme to reduce the redundancy to less than 2 for a local copy.
- For example (from BackBlaze) shard the data into 17 pieces and add 3 parity pieces for a stripe across 20 NVM Sets. Other choices are possible and could be tuned to the type of data.
- For a read to complete deterministically, only 17 of the 20 NVM Sets would need to be in DTWIN at the same time.
- 3 NVM Sets at a time could be in NDWIN as the data is fully propagated in write operations without affecting read latency for the other shards.
- Any of the 3 could be quickly moved back to DTWIN if one of the 17 fails fast.

Host Expectations

- Datacenter Applications are:
  - Always Reading
  - Always Writing
  - Multiple Producers and Consumers on the same SSD
- With the right software/infrastructure changes, datacenter customers should be able to almost eliminate long tails of latency for SSDs
- Reduces the need to build their own solutions (bypassing the SSD value)
- Should be flexible enough to cover newer media types (PM)

IOD Futures

- NVM Set Management
  - Deferred to a future specification
  - This means we need some way to create NVM Sets
    - Factory (private interface) – but different configurations for different customers is problematic
    - Field (published interface?) – if we make it a logical (but vendor specific) extension to NVMe, we can push through standardization later
  - Need research on limitations (size, granularity) for our different architectures
- IOD interface currently aimed at devices
  - System Vendors will want to control tail latencies as well, relying on this interface at the back end to control drives

Thank you

mark@carlson.net