Nuances of FDP Implementation

Dan Helmick, PhD
Principal Architect

Outline

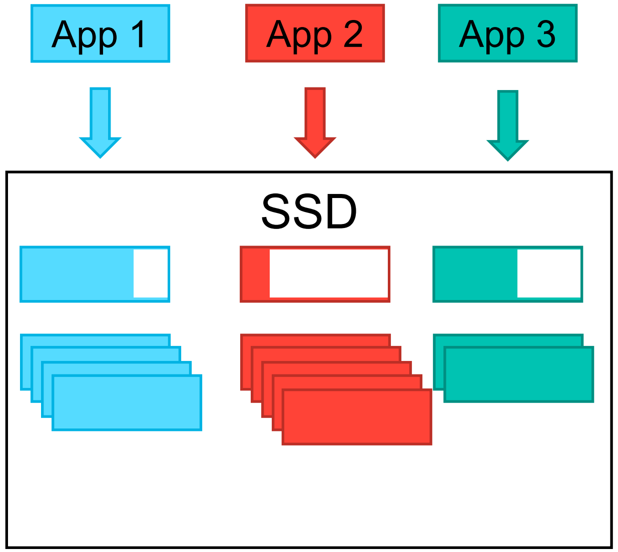
- Background
- Reclaim Group (RG) Configurations
- Reclaim Unit (RU) Sizing
- Reclaim Unit Handle (RUH) Count

Background

Flexible Data Placement (FDP) – Overview

- Apps can direct write data to be co-located in an SSD
  - Possible for a VMM to set-up defaults for legacy VMs
- Filling and deallocating appropriately can achieve WAF==1

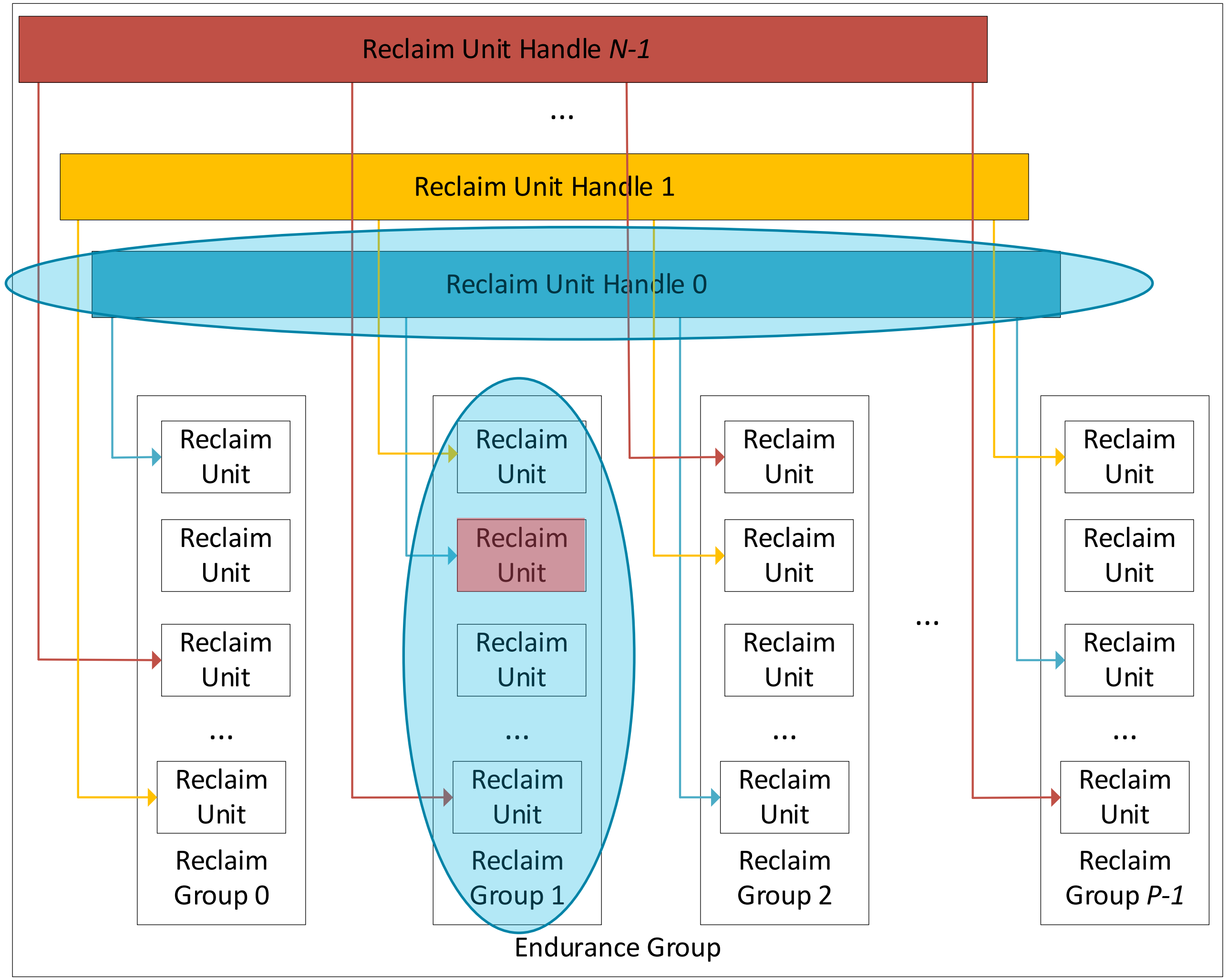
Logical View



Streams	Flexible Data Placement (FDP)	Zoned Namespaces (ZNS)
Open Loop WAF==1	Polling for WAF==1	WAF==1 or Error
Backwards Compatible	Backwards Compatible	Not Backwards Compatible
Streams Granularity Size (SGS)	Reclaim Unit (RU) Size	Zone Capacity <= Zone Size
Placement and LBA disconnect	Placement and LBA disconnect	Placement and LBA relationship
QD>1 allowed	QD>1 allowed	QD>1 requires Zone Append
Full FTL mapping required	Full FTL mapping required	Potential for compacted FTL Mapping

Storage Entities

- An FDP configuration consists of:
  - One or more Reclaim Units (RUs)
  - One or more Reclaim Groups (RGs)
  - One or more Reclaim Unit Handles (RUHs) that reference to a Reclaim Unit in each RG
- An Endurance Group that supports FDP
  - Supports one or more FDP configurations
- A Host enables specific FDP configuration in an Endurance Group
- Write commands allowed to specify and an Reclaim Group and Reclaim Unit Handle that indicates the Reclaim Unit to place the LBAs
- RUH references to an RU are modified by the Host
  - Referenced Reclaim Unit written to capacity
  - New I/O Management Send command
- RUH references to an RU may be modified by the controller:
  - Controller Level Reset
  - Sanitize operation

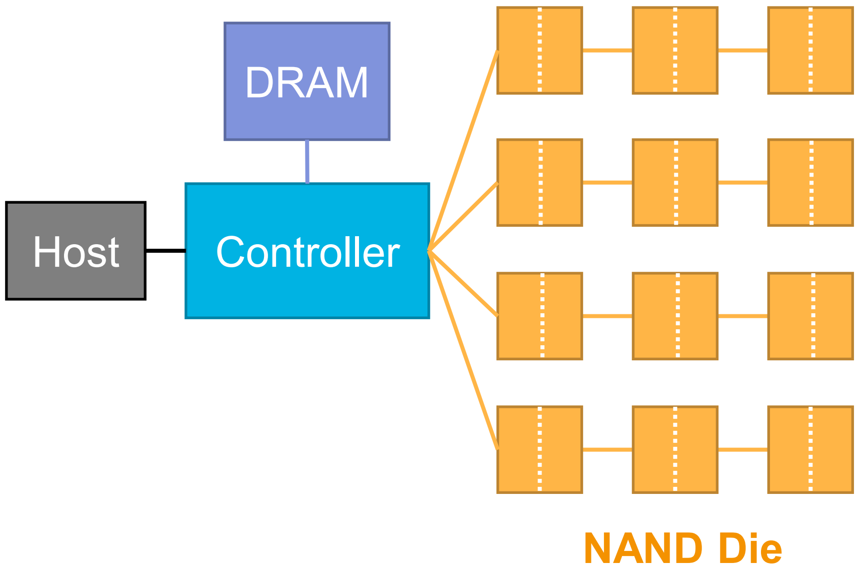


Credit: Mike Allison

Assumed Parameters and Configurations in this Presentation

No Vendor has these specs
But they're sufficiently representative

- Sizes
  - LBA size = 4096B
  - Page = 4x LBAs
  - Word Line (WL) = 3 TLC pages
  - WL per EB = 1024
  - Erase Block (EB) Size = 48MB
  - 2 Planes
  - EB per Plane = 684
  - Die Size = 512Gb
  - Channels = 16
  - Die per Channel = 16
  - SSD Physical Capacity = 16TB
  - SSD Logical Capacity = 15.36TB
- Performances
  - tRead = 50us
  - tProg (for all 3 pages) = 1.5ms
  - Latency to Program/Erase Suspend = 150us

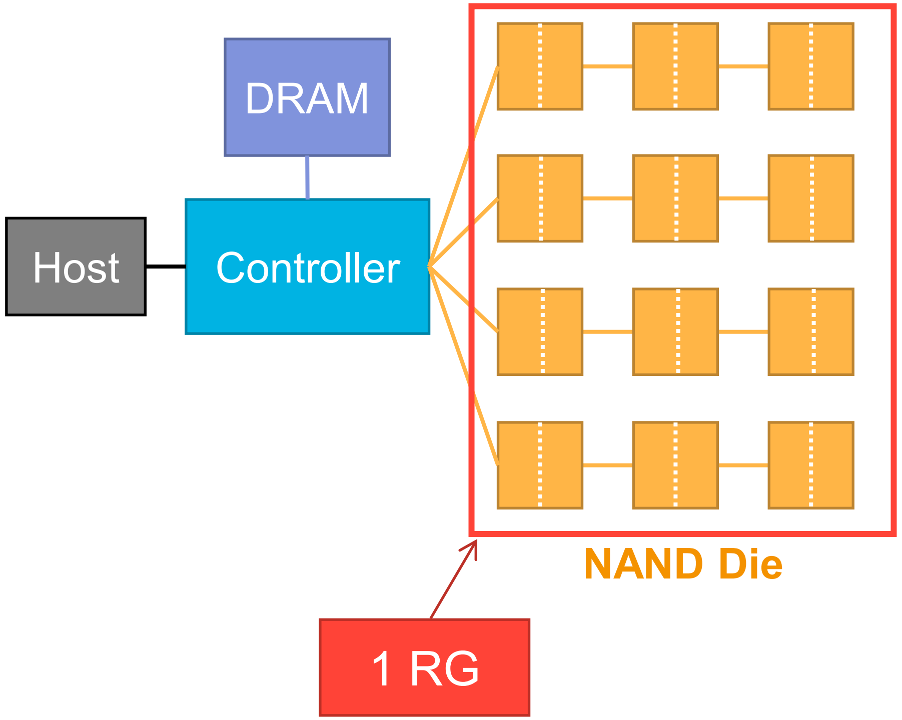


- **Illustrative** Example Above
  - Used for simple diagrams throughout
  - 4 Channels
  - 3 NAND Die each = 12 total
- FDP is Enterprise SSD focused feature
- Nominal Enterprise SSD configurations in the market
  - 8 – 16 channels
  - 4 – 16 NAND Die each = 32 – 256 total
- This presentation assumes high die count and channel count because this is where data placement is likely to assist the most

How many Reclaim Groups (RGs) should an FDP SSD have?

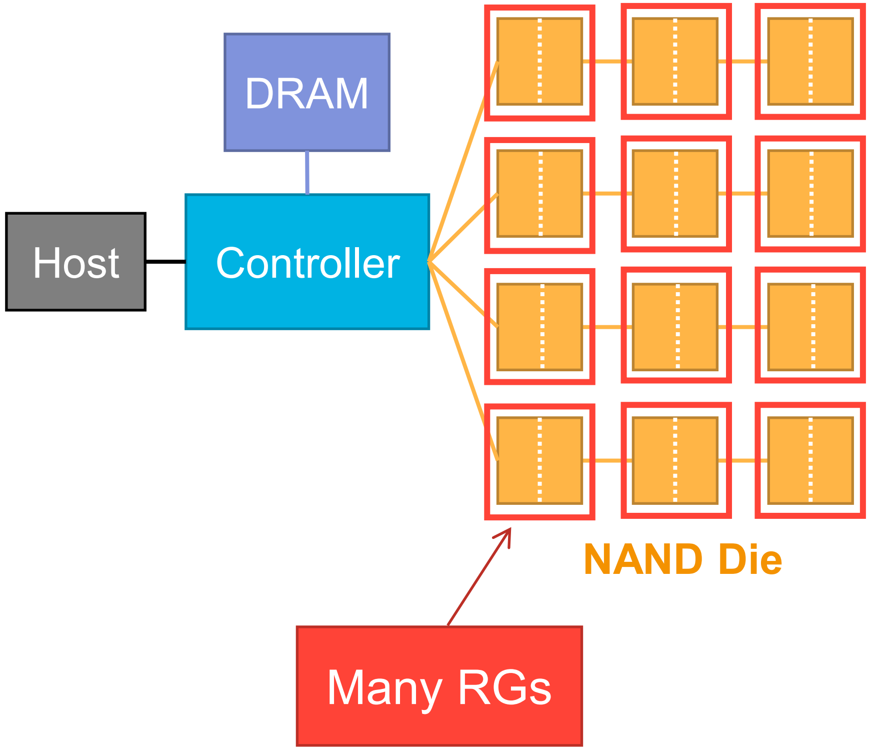
1 RG per SSD

- 1 RG will include all of the SSD's storage capacity
- Leverages existing NAND management algorithms
  - SSD can decide precise data placement within the RU that an RUH is filling
- Available Optimizations for an SSD
  - Performance – Route incoming data around concurrent traffic (Reads or other RUH programs)
  - Endurance – EBs composing an RU
- Similar to existing conventional SSDs



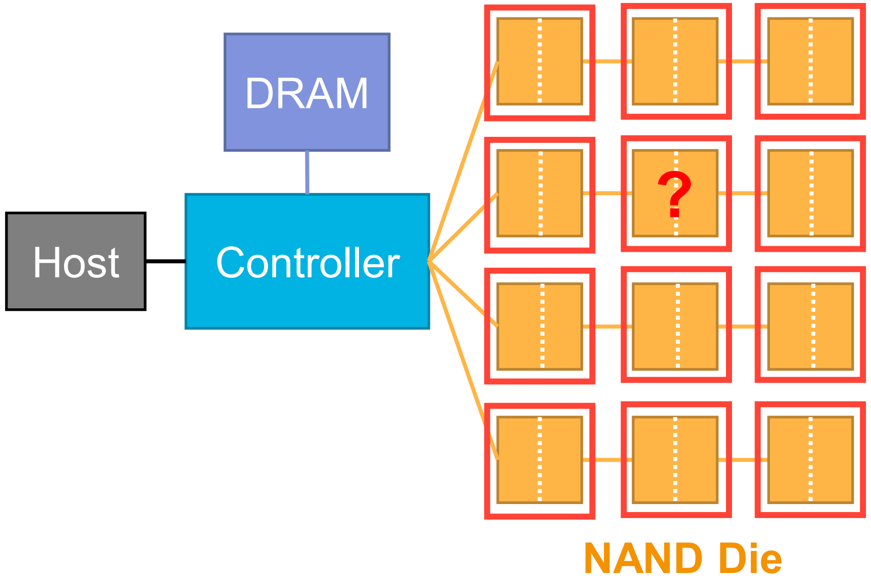
1 RG per Die

- SSD’s NAND management is restricted to be within each RG
  - Algorithms and tracking must be more granular
    - Risk of increasing memory and storage requirements
  - Confined decision making by the drive
  - OP can only be managed within each Die
- Increased Host burdens
  - Capacity – FDP has no guard rails for Hosts writing too much data to an RG
  - Performance – Channel and Die conflicts cannot be managed by the SSD
  - Endurance – Balancing traffic to each RG
- Increased challenges
  - Warrantees are at risk if a Host routes too much traffic to an RG
  - SSDs can make decisions when data overflows a NAND Die
    - Vendor specific behaviors on overflow
    - Does the Die overflow at filled logical capacity or physical capacity?
    - If the physical capacity is accessible, OP management and GC algorithms will likely impede data placement goals of Host



1RG per Die Configurations are Challenged to be Performant

- Latencies for a Host to make a decision
  - NAND Channel
  - 1us
- Latencies for a Drive to make a decision
  - NAND Channel
  - 1us
- Open Loop Feedforward Compensation?
  - Will need to differ per vendor, per generation, and per analog operation
  - Recommend: Accept the risk of a Program Suspend latency within the SSD



Some activity happens at the NAND
Example: Program failure

Yes. These latency approximations are imperfect.

But Hosts are certainly starting at a disadvantage.

RG Sizings Summarized

- Recommending: 1 RG per SSD
  - Enables maximized leverage of a HW optimized SSD
  - NAND management for performance and endurance remains responsibility of SSD vendors
- Generalization: Increasing RGs per SSD
  - Improves specificity of physical data placement
  - Increases responsibilities of the Host to maintain performance and endurance
  - NAND specific knowledge (different per vender and generation) are required to approach performance parity
  - SSDs are likely to have HW limitations on the maximum RG count per drive
- Different RG configurations are possible
  - Not discussed today since there has not been industry interest

What is the right size for a Reclaim Unit (RU)?

Example Potential RU Configurations

- Assuming 1 RG per SSD – allows the most RU size choices
  - Note: that the most common implementation is likely to match the RU size to span the RG
- RUs must align to Erase Block boundaries in order to gain the WAF by erasing the entire RU entity together.
- Some Example RUs
  - Superblock (SB)
    - 1 EB per Plane from every Die
    - Example Drive: 24GiB

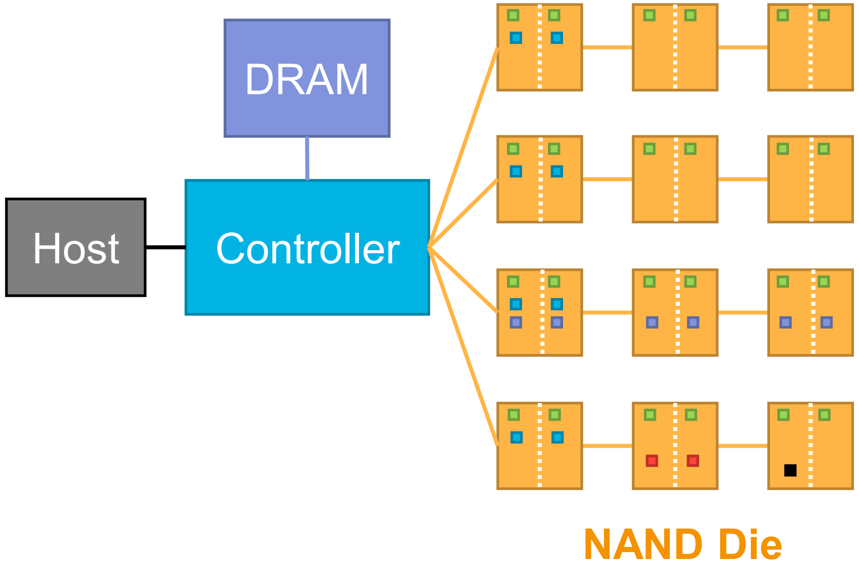
  - Way Stripe – Across Channel Stripe
    - 1 EB per Plane from every Die equally distant to the Controller
    - Example Drive: 1.5GiB

  - Down Channel Stripe
    - 1 EB per Plane on each Die in a Channel
    - Example Drive: 1.5GiB

  - Die Stripe
    - 1 EB per Plane
    - Example Drive: 96MiB

  - EB Size
    - 1 EB
    - Example Drive: 48MiB

  - Wrapping
    - An RU could potentially include more than 1 EB per plane – The RU could “Wrap”
    - Not discussed because there are no recognized advantages
- Conclusions
  - There are many options for RU sizes



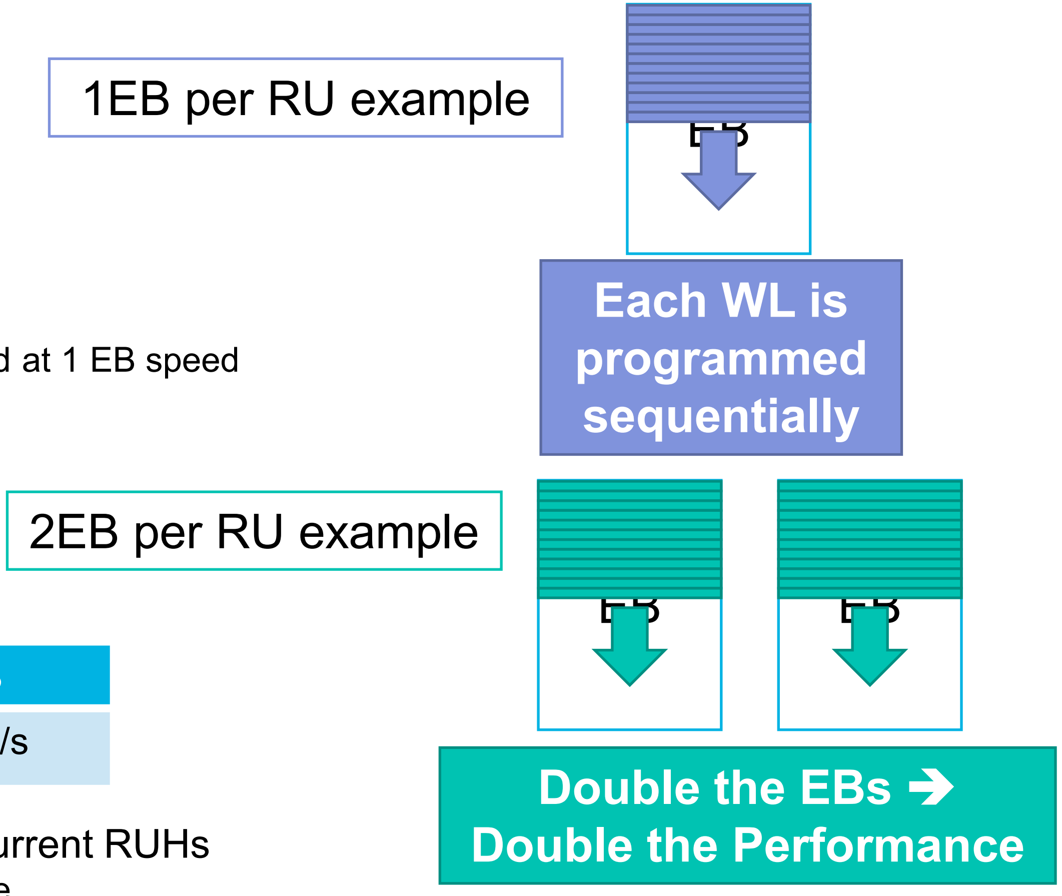
Down Channel Stripes have channel contention concerns, but otherwise can be similar to Way Stripes. Not a focus in this presentation.

RU Size to Performance

- NAND Implications
  - EBs must be filled sequentially
  - Concurrent programming 1 EB per plane enables max performance
- FDP Implications
  - Data order is unrestricted within the RU
  - But each RU must be filled before opening the next RU
- Each RUH's performance is determined by the RU size
  - If an RU is sized to be 1 EB, then all of the data for the RU must be programed at 1 EB speed before moving to the next RU
  - *** **Except** ***
    - Programming Speed or Buffering contentions
    - Optimizing trickery
    - …
- Example performances for different RU sizes:

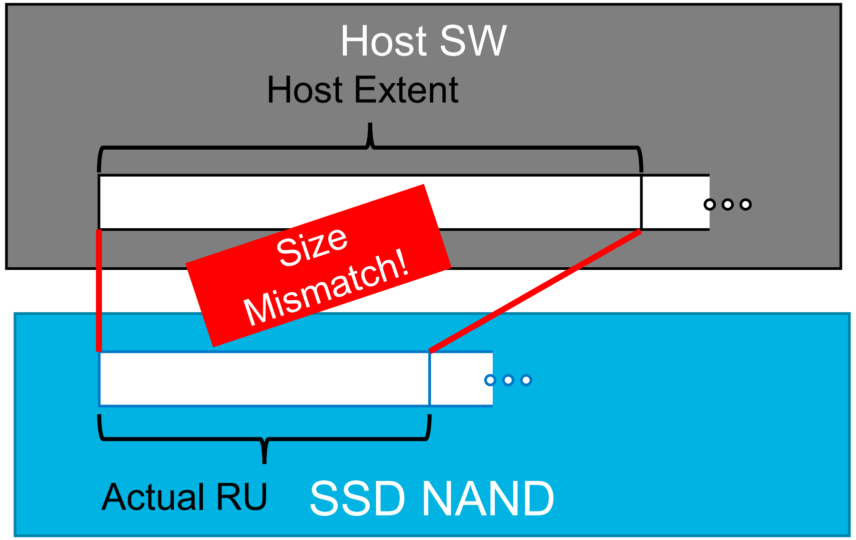
	1 EB	Die Stripe	Way Stripe	SB
Performance	32MB/s	64MB/s	512MB/s	8GB/s

- Achieving performance parity will require the Host to manage more concurrent RUHs
  - Example: 2 active RUHs of EB size required to match 1 RUH of Die Stripe size.
- Implied takeaway is that your data is likely in a rough layout as below
  - *** **Except** ***
    - Rerouting for access contentions
    - Various Race Conditions from Host to NAND
    - …
  - So, don't depend on this ordering



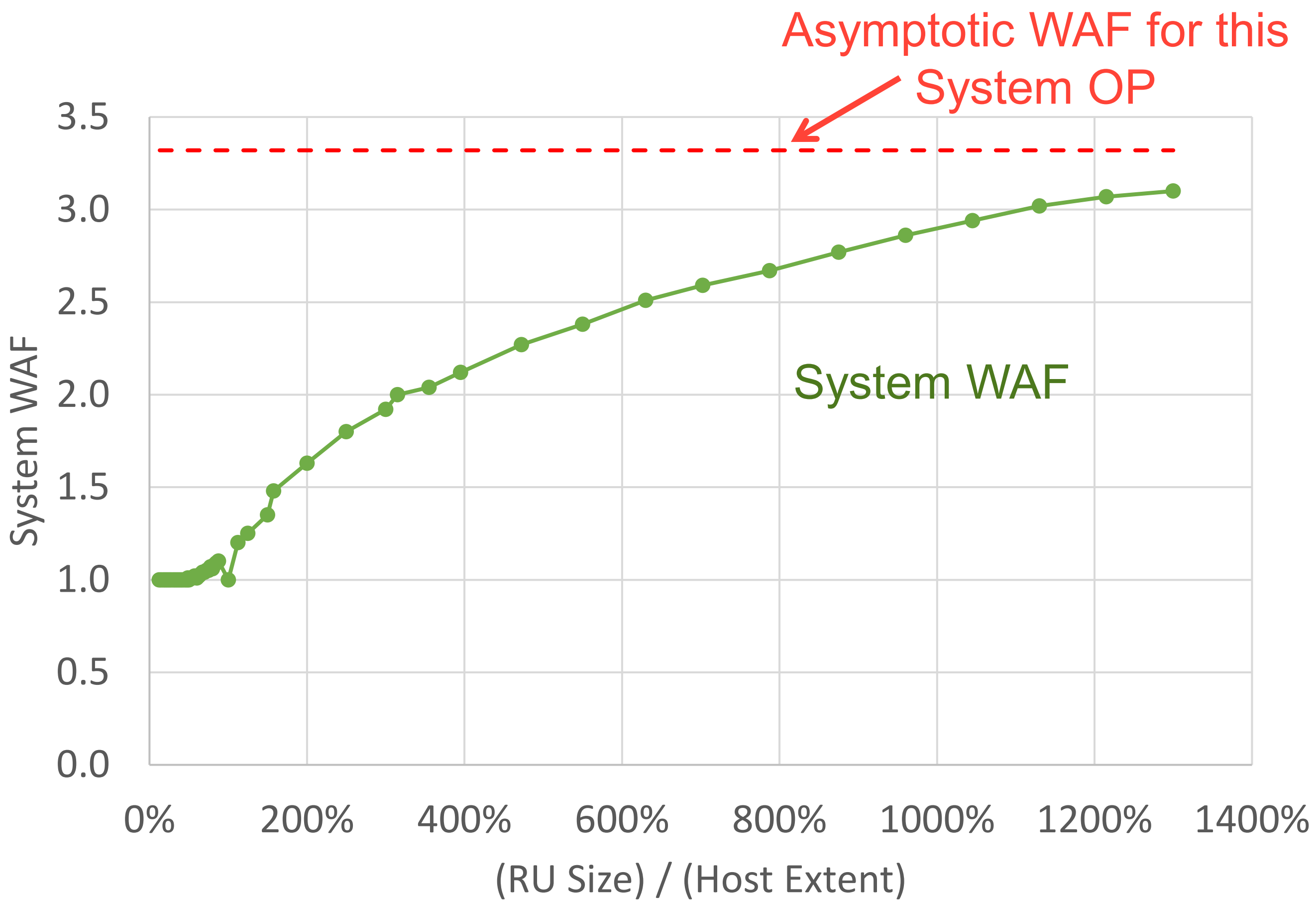
Additional RU Size Impacts

- File Systems built with Host Extents rather than RU matching
  - Host Extent is unlikely to match SSD RU size
- Reasons Host Extent may not match SSD RU
  - Vendor-to-Vendor mismatch
  - Generation over Generation SSD RU changes
  - SW developed separate from SSDs



WAF Impacts: RU Size to Host Extent Size

- Model – **Worst Case Scenario**
  - Host Extents filling randomly
  - Host Extents randomly deleted
  - Measured the combined WAF of the Host+SSD
- WAF is **always** better while using FDP with separated RUHs
  - Larger Host Extents and smaller RUs are always beneficial
- Very small RUs can enable WAF=1
  - But NAND's EBs are increasing generation over generation



Conclusions on RU Size Impacts

- Larger RUs
  - Maximizes performance per RUH
  - Permit an SSD to route traffic around temporary contentions
  - Always improves system WAF

- Smaller RUs
  - Require more active RUHs to reach similar performance
    - If a FDP SSD configuration pairs small RUs with small RGs (standard expectation), then more concurrent Host decisions are required on more RUHs/RGs.
    - Host is at a latency disadvantage for each of these decisions
  - Can reach better System WAFs

What is the correct number of Reclaim Unit Handles (RUHs)? Should they be Persistently Isolated?

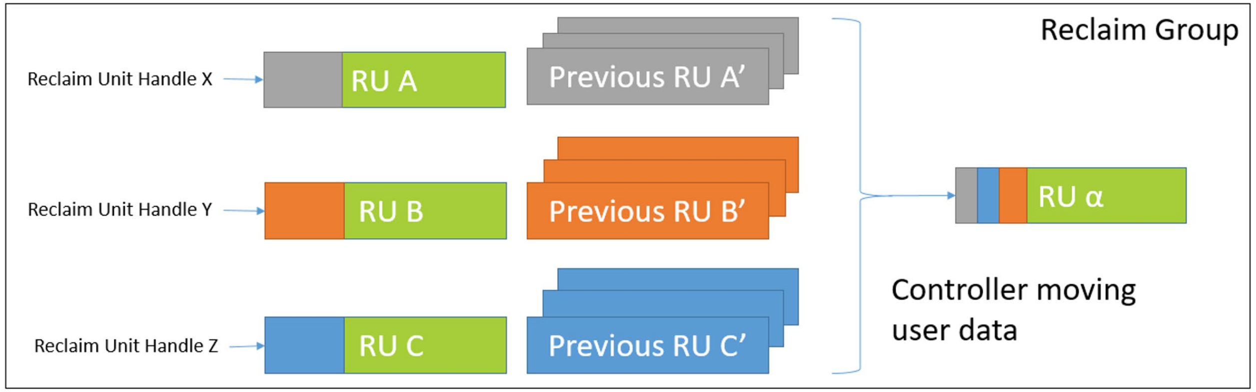
Requirements per RUH

- Every RUH can require
  - NAND append point
    - Consumes some OP
    - Risk: Open Block Timer for each EB that is partially programmed
  - Buffers for inflight data
    - Buffers sized to performance of the RU
  - Capacitors to power fail protect those buffers

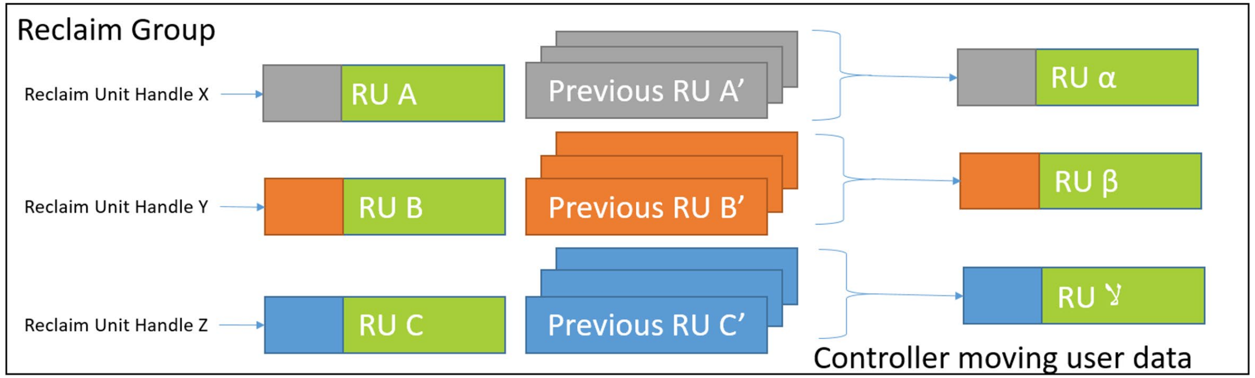
- Persistently Isolated RUHs may additionally require
  - GC Append points – Multi-threading the GC
  - SSD processing
  - Tracking – Consumes OP
  - Development and Validation time

- Increasing RUH counts can additionally risk Die contentions
  - Reminder: More active RUHs are needed for a Host to match performance on small RU size configuration

Initially Isolated Reclaim Unit Handles

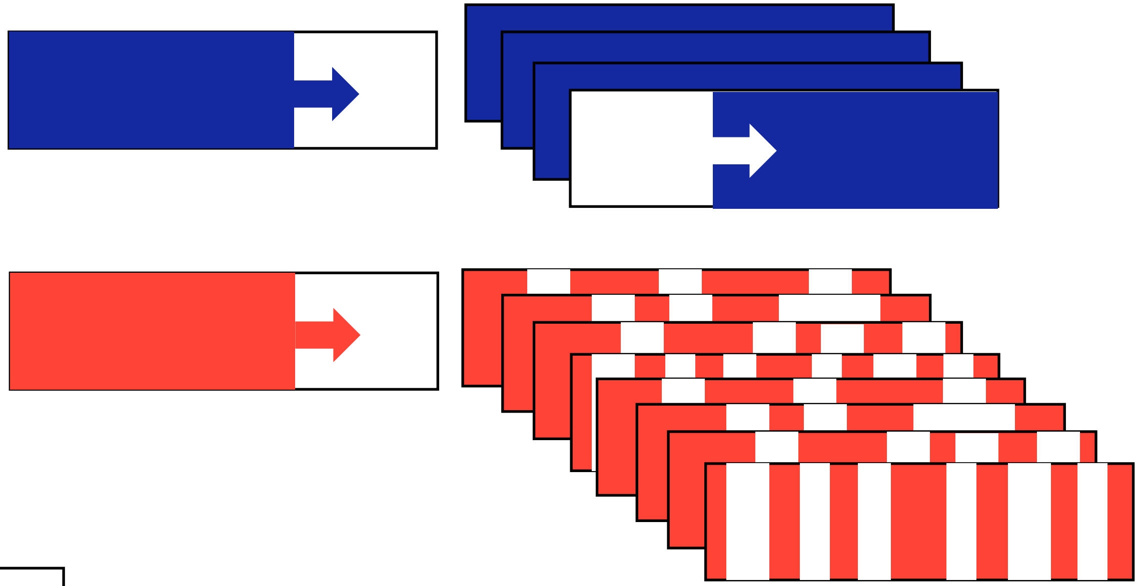


Persistently Isolated Reclaim Unit Handle

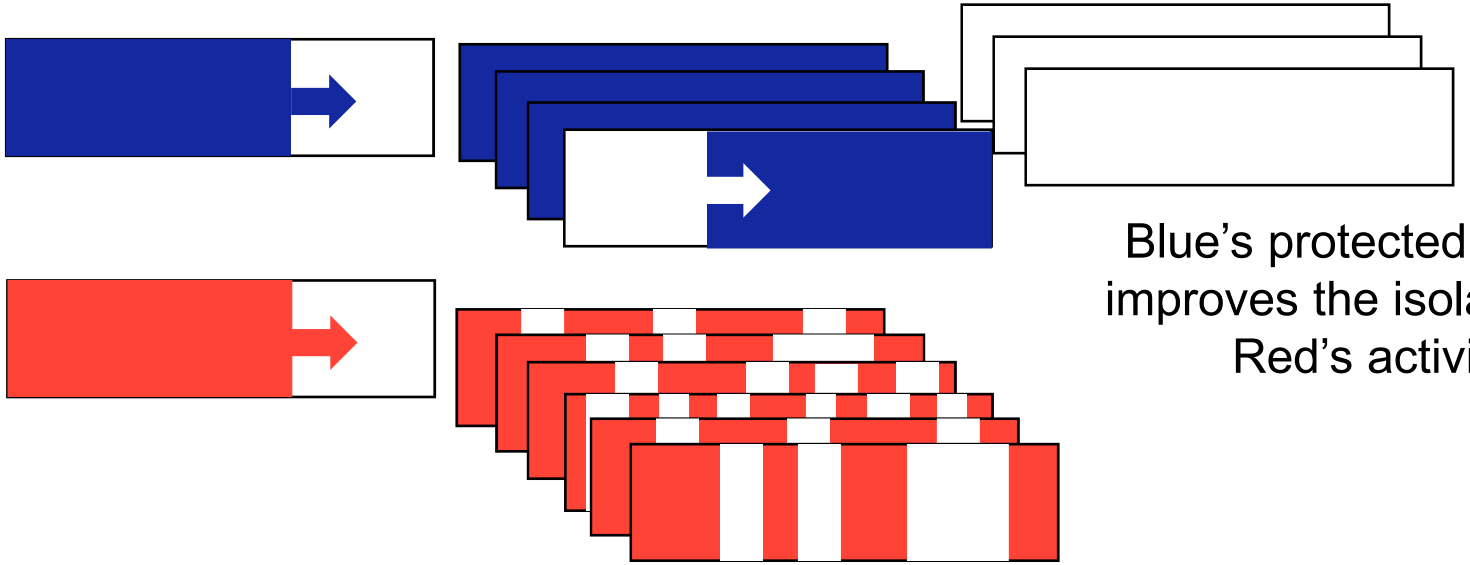


Garbage Collection (GC) and Over Provisioning (OP) Decisions for RUHs

- FDP does not define the OP sharing or GC trigger rules
- Shared OP pools
  - Poor behaving RUHs automatically consume OP and the WAF is reduced to the best extent possible
- Restricted OP pools
  - Improve isolation per RUH
  - Poorly behaved RUHs can consume SSD endurance prematurely



Red consumes the shared OP Pool for a net reduction in WAF



Blue’s protected OP Pool improves the isolation from Red’s activities

GC with Persistently Isolated RUHs

- PI RUHs may have additional information available
  - WAF Estimates
  - Logical Capacity
  - Physical Capacity

- Samsung is examining the tradeoffs for making OP a function of the incoming workloads
  - Interested in engaging with customers to better understand their workloads!

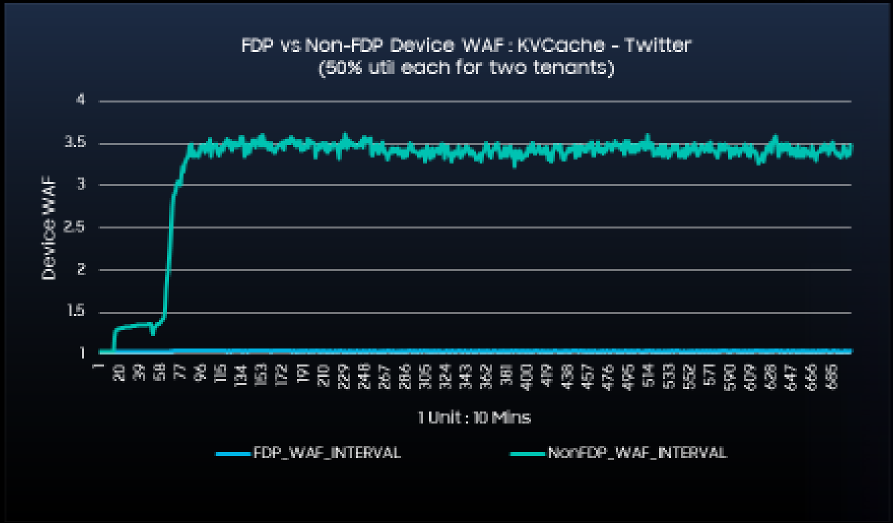
WAF=1 Achievable with Initially Isolated RUHs

- Measured data from Samsung PM9D3 with Initially Isolated RUHs
- Increasingly complex and diverse workloads are demonstrating WAF~=1
- Industry Risk: The advantages of Persistently Isolated RUHs may be over idealized.
  - Pending more data and industry experience
- Reference: “FDP Integration in CacheLib” by Arun George at the Future of Memory and Storage Conference 2024

Scaling to two heterogeneous tenants – KVCache +Twitter

SAMSUNG

- KVCache instance using 50% SSD, Twitter workload uses the other 50%
- Both workloads achieve WAF of ~1 in FDP mode as single tenants
- FDP reduces the Device WAF from 3.5 to ~1 in this experiment
- CacheLib metrics like Throughput, Hit rate, App-WAF are unaffected
- FDP provides tenant isolation for varied workloads also.



Metrics	Non-FDP		FDP	
	KVCache	Twitter	KVCache	Twitter
Write Throughput, SET/s	43 K	20 K	46 K	20 K
Read Throughput, GET/s	344 K	5 K	372 K	5 K
P999 Write latency, usec	1294 us	1464 us	140 us	92 us
P999 Read latency, usec	542 us	653 us	367 us	397 us

10

Conclusions

- Join SNIA’s Storage Data Placement TWG on Tuesdays!
- RG
  - Recommending 1 RG per SSD for all customers
- RU Sizing
  - Larger RUs are generally beneficial for SSDs
  - Smaller RUs can make sense in some Host use-cases
- RUHs
  - Inflated RUH counts and Persistently Isolated can have hidden negative impacts
    - Moderate RUH counts are encouraged
    - Initially Isolated RUHs can be very effective
  - OP management and GC triggers will likely differentiate vendors