Optimizing SSD Performance for Edge AI NVMe Dataset Management Insights

Name: David Yeh

Title: Product Marketing Manger

Silicon Motion Technology Corp.

Legal Notice and Disclaimer

- The content of this document including, but not limited to, concepts, ideas, figures and architectures is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Silicon Motion Inc. and its affiliates. Silicon Motion Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this document.
- Nothing in these materials is an offer to sell any of the components or devices referenced herein.
- Silicon Motion Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Silicon Motion, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.
-
- Silicon Motion, the Silicon Motion logo, MonTitan™, the MonTitan™ logo are trademarks or registered trademarks of Silicon Motion Inc.

Enhanced SSD Performance for AI Edge Devices

- Reduced Latency
  - Faster read/write speeds shorten AI inference and decision-making time
  - Real-time responsiveness improves user experience
- Increased Data Throughput
  - Supports rapid processing of large volumes of sensor and device data
  - Ensures high-speed data access for AI models, enhancing accuracy
- Lower Write Amplification Factor (WAF)
  - Extends SSD lifespan, reducing maintenance costs
  - Maintains long-term stability to ensure reliable AI service operation

Dataset Management (DSM) Concept

- The NVMe specification defines a DSM command attribute that can effectively improve SSD drive's efficiency and performance.
- By utilizing the "Access Frequency" defined in the specification, it is able to effectively identify hot and cold data, achieving a more efficient data placement.
  - The DSM command and FDP are more flexible compared to ZNS.

DSM Command (Dataset Management)	FDP (Fixable Data Placement)	ZNS (Zoned Namespaces)
Backward compatible		**Not** Backward compatible
Requires full L2P mapping table		Full L2P mapping table **not** required
Sequential, Random Write		**Only** sequential write
No specific implementation method	Reclaim Unit Base (Super Block)	Zone Base (Die Block)

Dataset Management (DSM) Command Definition

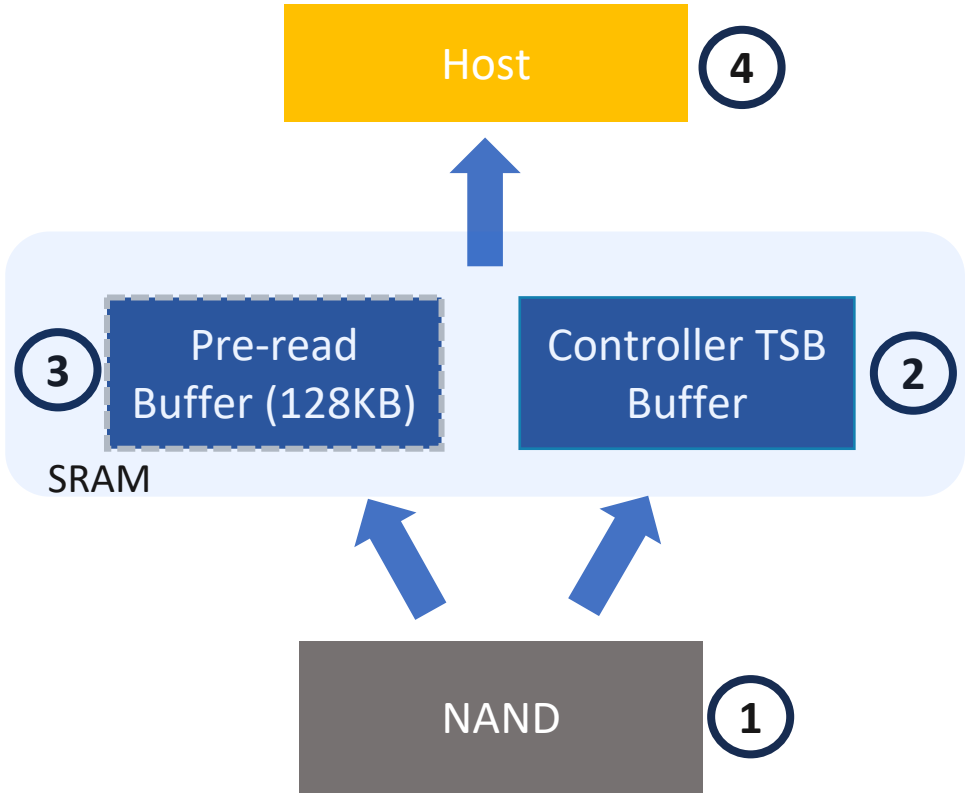
Figure 42: Dataset Management – Context Attributes

Attribute	Bits	Description
Command Access Size	31:24	Number of logical blocks expected to be transferred in a single Read or Write command from this dataset. A value of 0h indicates no Command Access Size is provided.
Reserved	23:11	Reserved
WP: Write Prepare	10	If set to '1', then the provided range is expected to be written in the near future.
1 SW: Sequential Write Range	09	If set to '1', then the dataset should be optimized for sequential write access. The host expects to perform operations on the dataset as a single object for writes.
SR: Sequential Read Range	08	If set to '1', then the dataset should be optimized for sequential read access. The host expects to perform operations on the dataset as a single object for reads.
Reserved	07:06	Reserved
2 AL: Access Latency	05:04	Value / Definition 00b: None. No latency information provided. 01b: Idle. Longer latency acceptable. 10b: Normal. Typical latency. 11b: Low. Smallest possible latency.
3 AF: Access Frequency	03:00	Value / Definition 0h: No frequency information provided. 1h: Typical number of reads and writes expected for this LBA range. 2h: Infrequent writes and infrequent reads to the LBA range indicated. 3h: Infrequent writes and frequent reads to the LBA range indicated. 4h: Frequent writes and infrequent reads to the LBA range indicated. 5h: Frequent writes and frequent reads to the LBA range indicated. 6h to Fh: Reserved

1. Using the Sequential command hint value provided by the host, the firmware can pre-fetch data, thereby achieving performance improvement.
2. Using the "Access Latency" information, low latency data can be separated and programmed together to minimize the latency.
3. By utilizing "Access Frequency", hot and cold data can be effectively allocated, thereby achieving better Write Amplification Factor (WAF).

Read Look Ahead Implementation

- Reserve a pre-read buffer within the TSB(Time Sharing Buffer) for storing read look-ahead data.

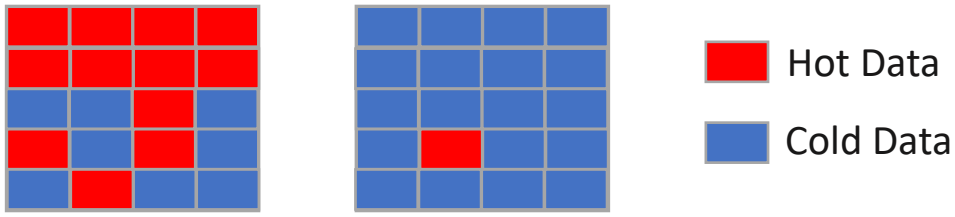


Command	LBA	Data Length	Path
#1	0~255	128KB	1->2->4
#2	256~511	128KB	1->2->4
#3	512~ 767	128KB	1->2->4
#4	768~1023	128KB	3->4

Trigger Pre-fetch data to pre-read buffer

Implementation by DSM Hint Value - 1

	Value	Definition	Score
Access Frequency	0h	No frequency information provided.	N/A
	1h	Typical number of reads and writes expected for this LBA range.	N/A
	2h	Infrequent writes and infrequent reads to the LBA range indicated.	+0
	3h	Infrequent writes and frequent reads to the LBA range indicated	+1
	4h	Frequent writes and infrequent reads to the LBA range indicated.	+2
	5h	Frequent writes and frequent reads to the LBA range indicated.	+3

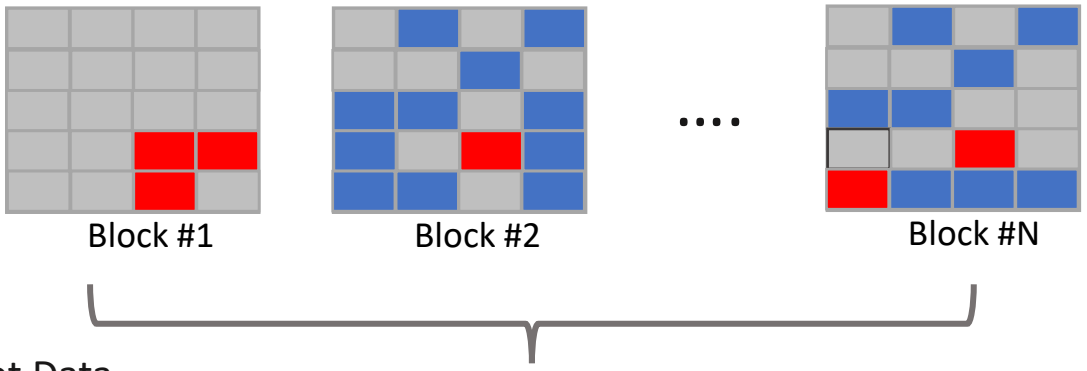


Block #1
Score: +22

Block #2
Score: +5

- Using the hint value of Access Frequency, when each command is written into a block, we can calculate the score of that block.
- For example:
  - **Block#1**: There are many frequent write LBAs (+2 or +3) written into the block, so this block can be considered to contain a lot of hot data.
  - **Block #2**: The LBAs written into this block are mostly infrequent writes, so the block primarily contains cold data.

Implementation by DSM Hint Value - 2

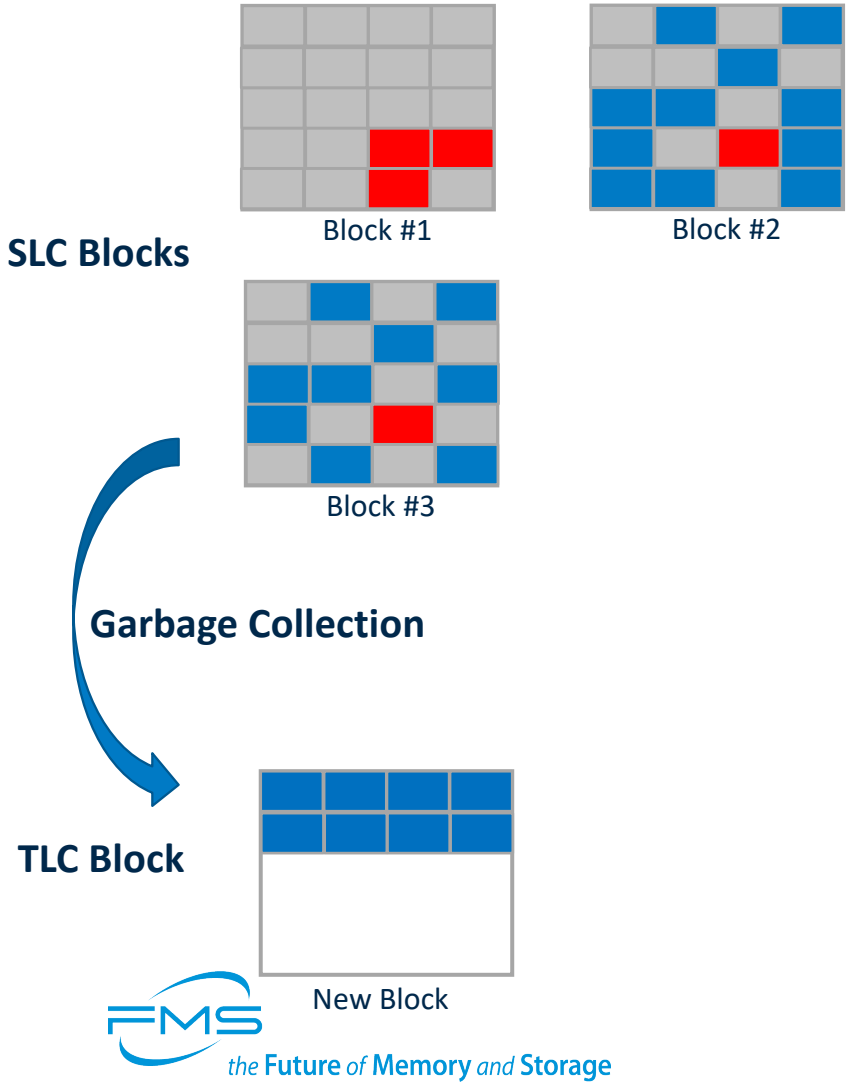


Block Scoring Table

Block	Total Score
#1	22
#2	5
#3	12
#4	8
⋮	⋮

- We will create a score table to record the score of each block. More frequent LBAs represent hot data, and the score will be higher.
- For more frequent LBA writes, we aim to avoid moving data within the same block. This way, it will be easier for the block to become directly invalidated and turn into an empty block.
- For example, since block #1 mainly contains hot data, it may be reclaimed without the need for data migration.

Performance and WAF Improvement



Block Scoring Table

Block	Total Score
#1	22
#2	5
#3	8
#4	12
⋮	⋮

Threshold

GC Source Blk Bitmap

Block	Bit
#1	1
#2	0
#3	0
#4	1
⋮	⋮

- The GC (Garbage Collection) policy can also leverage the score table to determine the criteria for selecting source blocks.
- In addition, we will create a source block bitmap to identify candidate blocks for GC. Blocks with scores below a defined threshold will be marked as potential source blocks.
- This approach ensures that only a small amount of data needs to be moved, allowing blocks to be reclaimed more efficiently.

Performance and WAF Improvement

- Based on our firmware implementation, we are able to achieve significant improvements in both performance and WAF.
  - Hot data is retained in SLC blocks, allowing host read requests to be served from high-speed SLC regions.
  - Furthermore, the garbage collection mechanism is optimized to prioritize source blocks with a low number of valid pages, thereby minimizing write amplification.

Summary of DSM (Dataset Management)

- Compared to other data placement methods, the DSM command offers greater flexibility in firmware implementation.
- With the host or driver providing the hint value within the command, performance and WAF (Write Amplification Factor) can be effectively improved through firmware implementation.
- Improving SSD performance enhances AI edge computing by reducing latency, accelerating inference, and ensuring long-term reliability.

SiliconMotion

Meet us at booth

#315

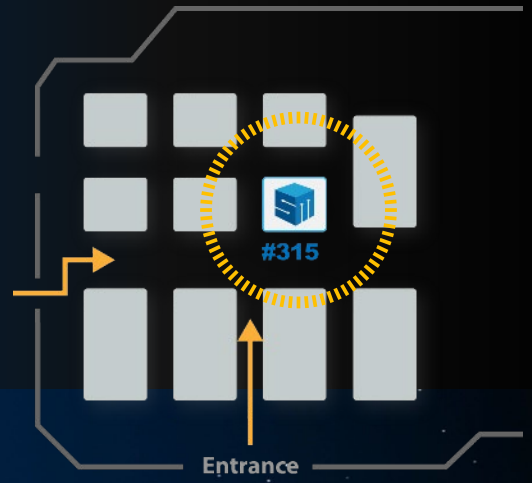
Scan to learn more!

Scan to learn more!

Meet us at booth

#315

SiliconMotion



Smart Storage in Motion: From Silicon Innovation to AI Transformation Across all Spectrums!