WINTECHCON

Enhancing eMMC with Multi-Stream

September 27, 2019

Sushma Vishwakarma
Samsung Semiconductor India R&D Center

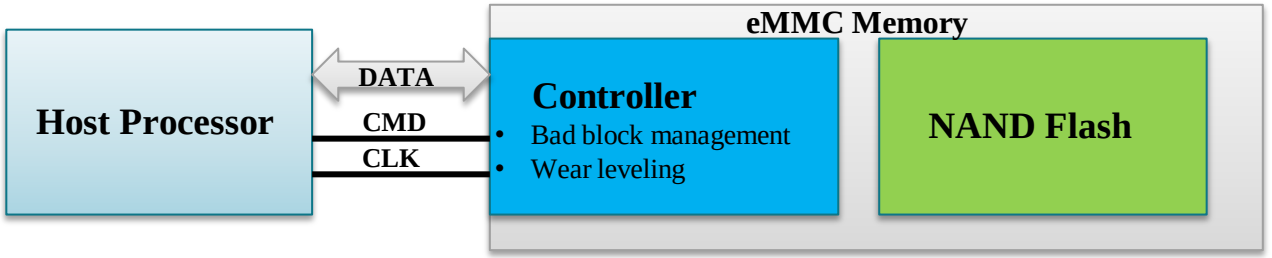
Presented at WinTechCon-2019, organized by IEEE CAS Bangalore Chapter, IEEE Bangalore Section, and IEEE WiE Council Bangalore

Contents

- Introduction
- Problem Statements
- Design Idea
- Challenges
- Our approach
- Test Environment
- Experimental Result
- Real Time Use cases
- Conclusion
- References

Introduction

- eMMC is an embedded storage solution with an MMC interface, Flash memory & Controller in a BGA package.

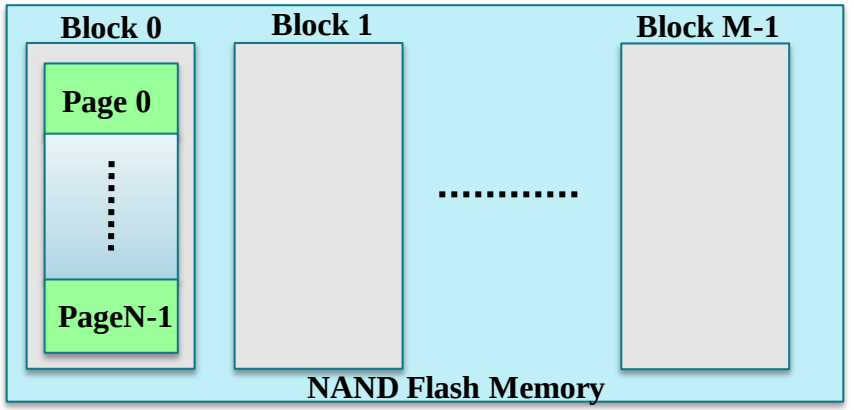


- NAND Flash based eMMC used for high performance applications.
  - Mobile phones, Smart phones, Tablet computers, Notebook computers etc.

- eMMC provides fast scalable performance with interface speed up to 400Mbps.

Problem Statement

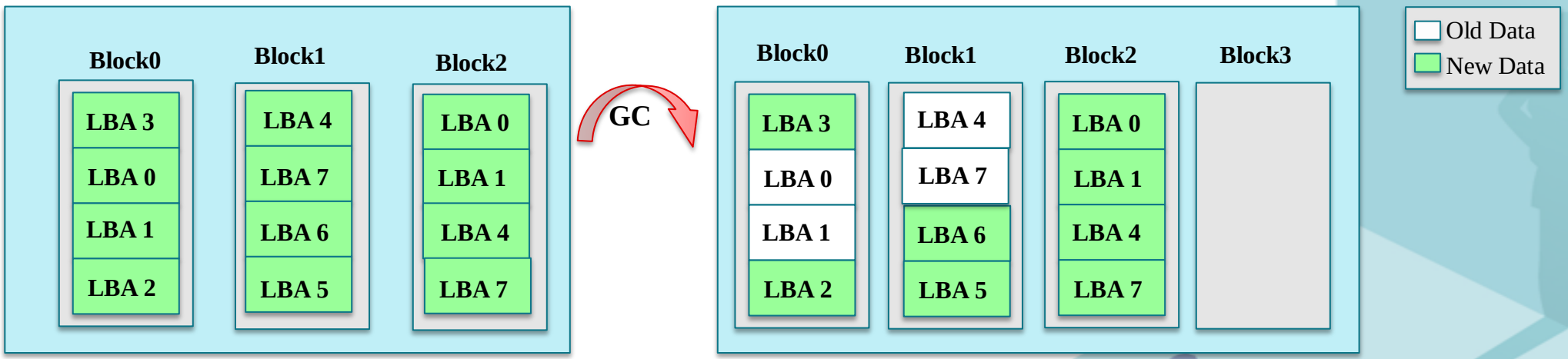
- eMMC chip is made up of NAND flash memory.
- NAND is architecturally partitioned into blocks and each block contains fixed number of pages.



- NAND flash memory constraints :
  - Data can not be modified in-place
  - Write and Erase asymmetry
    - Write performed on PAGE
    - Erase executes on BLOCK
  - Limited number of P/E cycles (Aging)

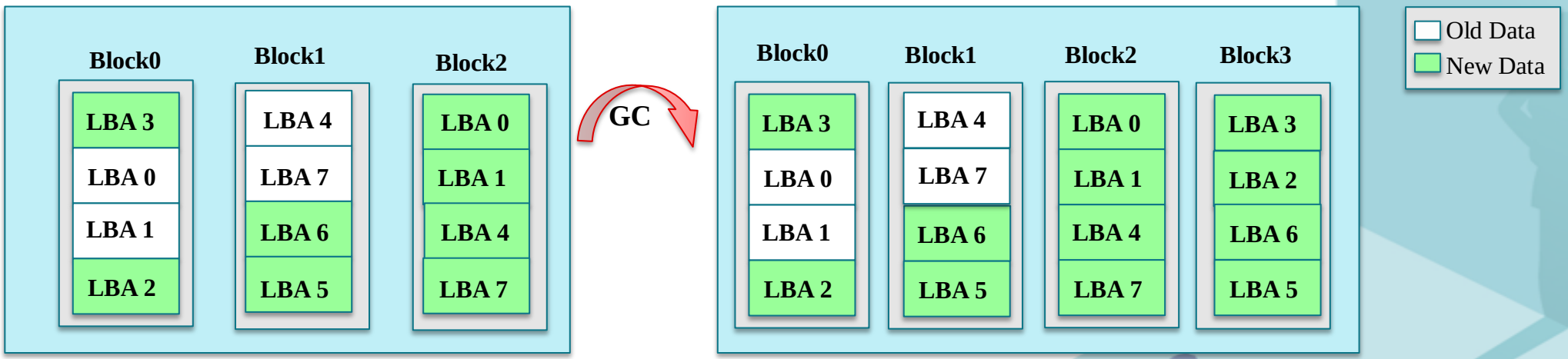
Cont….

- Write pattern (0-1-4-7) applied, as the result some data become invalid in Block0 and Block1



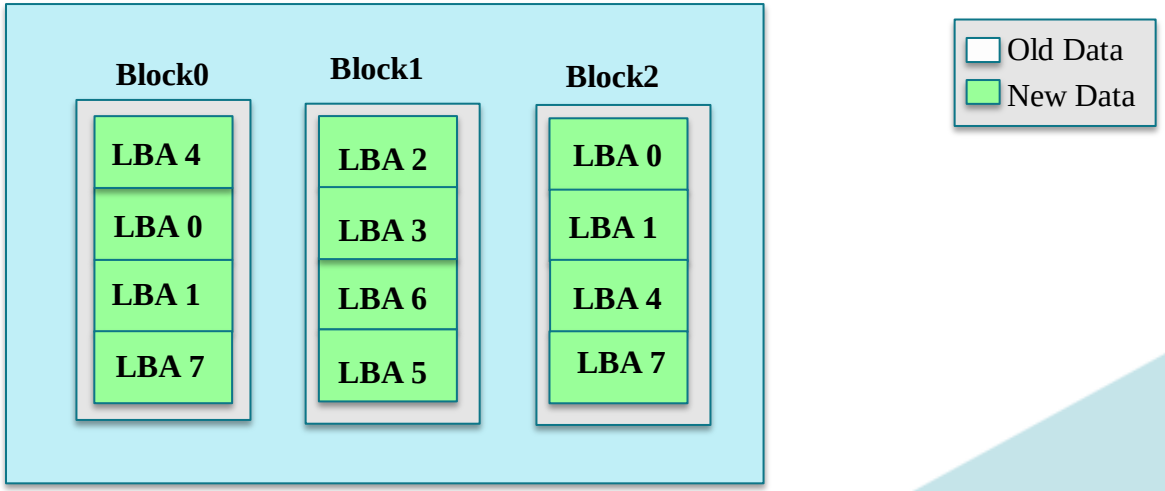
Cont….

- Write pattern (0-1-4-7) applied, as the result some data become invalid in Block0 and Block1



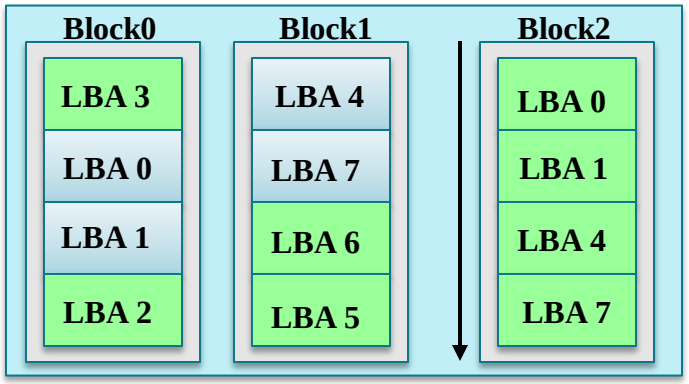
Cont….

- Write pattern (0-1-4-7) applied, as the result invalidating all the data of Block 0 but none in block 1

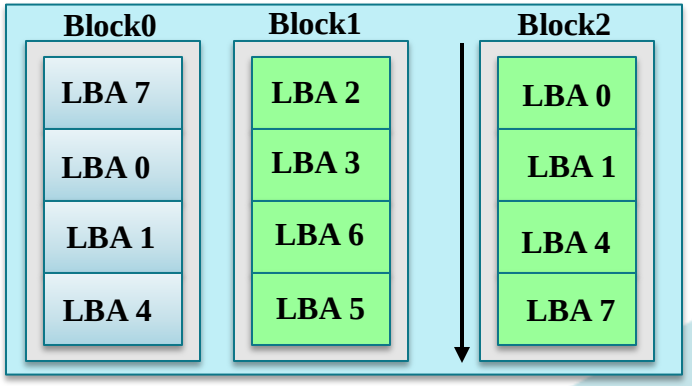


Design Idea

- Multi-Stream is a concept where we can group data stream into multiple streams.
- All the data belonging to particular stream will be having similar life time
- Example describe how this stream can address the eMMC aging problem:



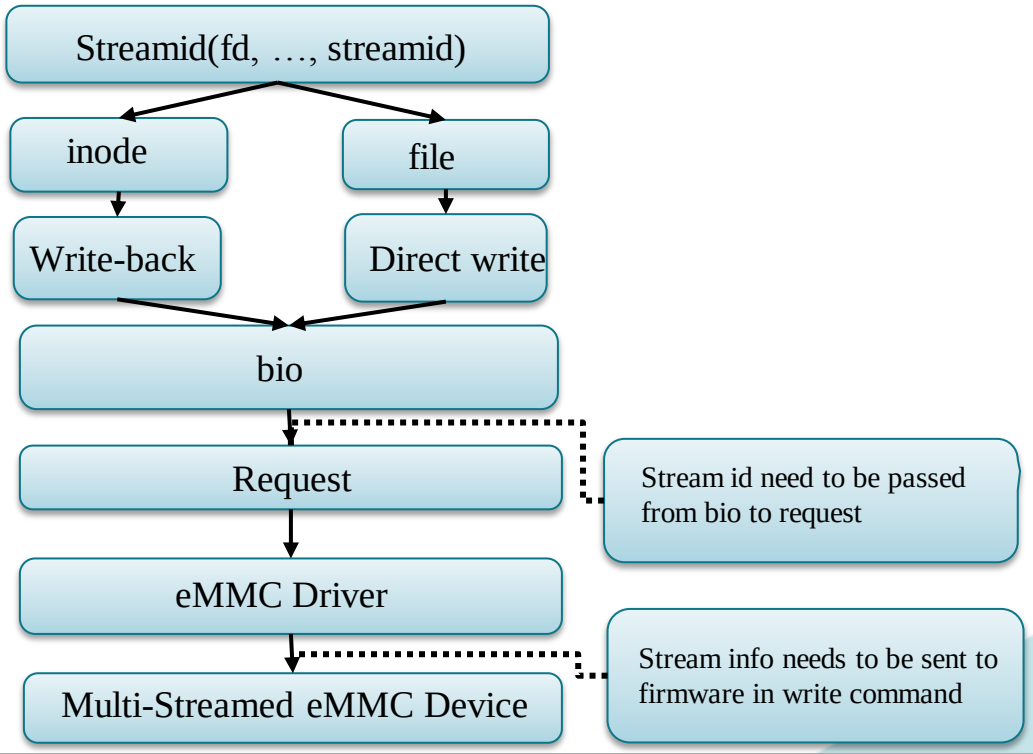
Write pattern (0-1-4-7) applied, as the result some data become invalid in Block0 and Block1



Write pattern (0-1-4-7) applied, invalidating all data in Block0 but none in Block1

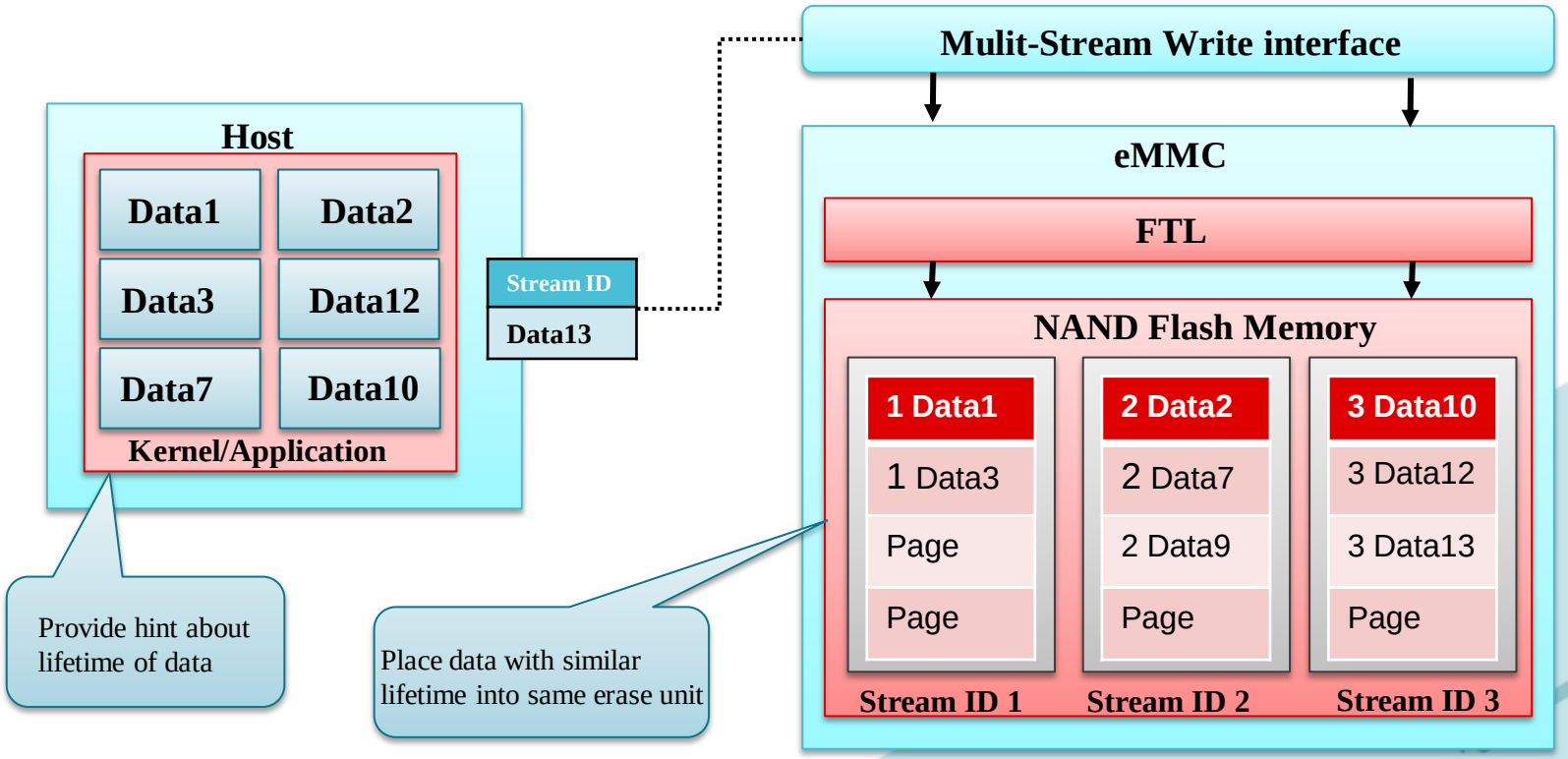
Cont…

- eMMC that implements the proposed multi-stream interface allows host system to specify the lifetime of data in the form of stream-id.
- Multi-streamed eMMC place data in a stream together and not mix data from different streams.



Cont…

- Multi-Streamed eMMC writes data into a related NAND block according to stream ID



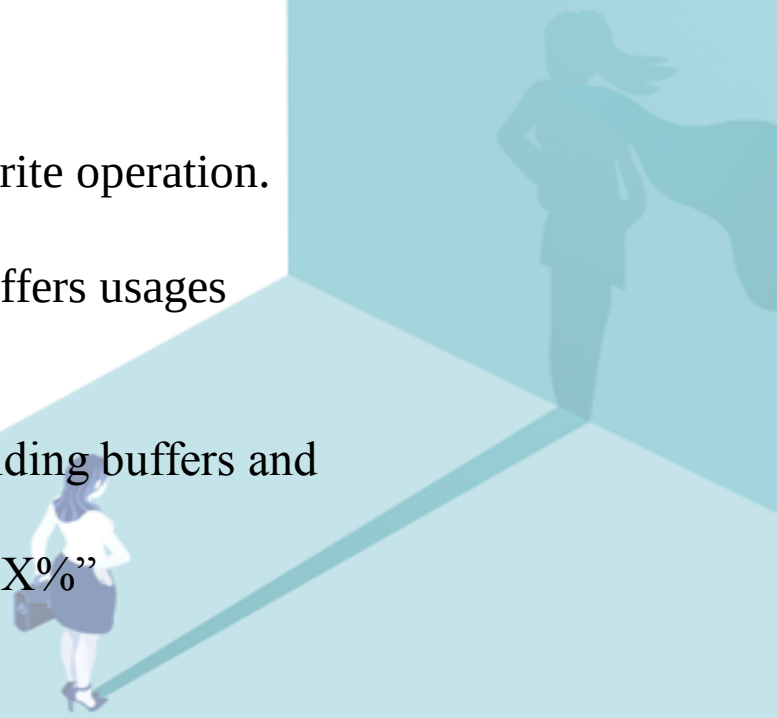
Challenges

- To associate Data with Stream-id JEDEC eMMC SPEC 5.1 does not have support.
  - We used context id field of command 23 for stream id information.
  - Which is 4 bit field, can support up to 15 streams.

- eMMC has limited computation resource and DRAM.
- Maintaining stream information increases Meta size
  - We limited maintenance of stream information only during host write operation.

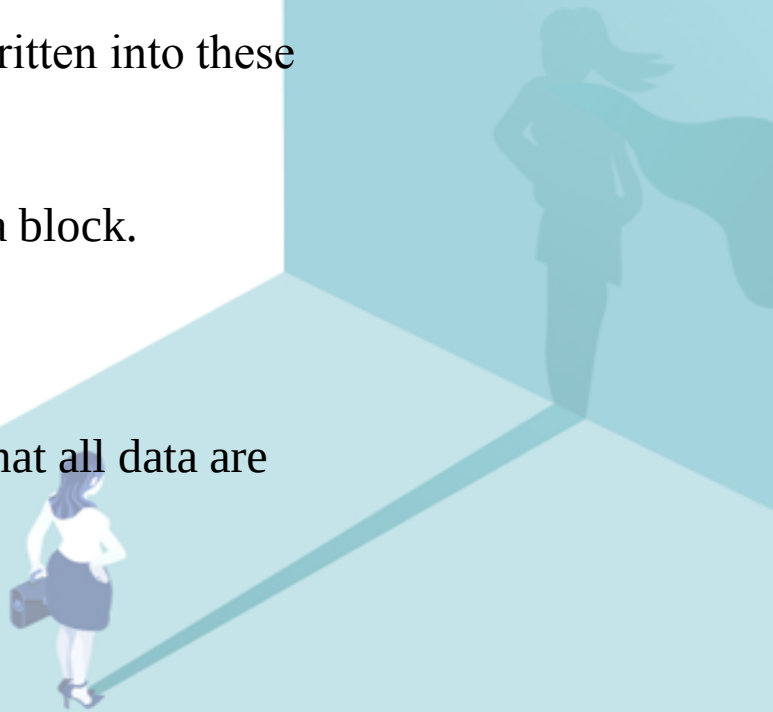
- Maintaining Stream information during host write increases transfer buffers usages
  - Our implementation supports limited streams.

- Performance depends on fill factor of device if device is “X%” full, holding buffers and active block for all stream ids can cause resource crunch
  - Our implementation disables stream support if fill factor reaches “X%”



Our Approach

- Stream id information will be shared by Application.
- Stream information maintains only for host write operation.
- Allocate “Active Block” for each stream separately, incoming data is written into these active blocks according to their stream id.
- Once Active Block completely written, we add it to list of common data block.
- During GC this stream-id information ignored.
- Since data with similar life time is residing in one block it is expected that all data are going to be erased almost at same time.

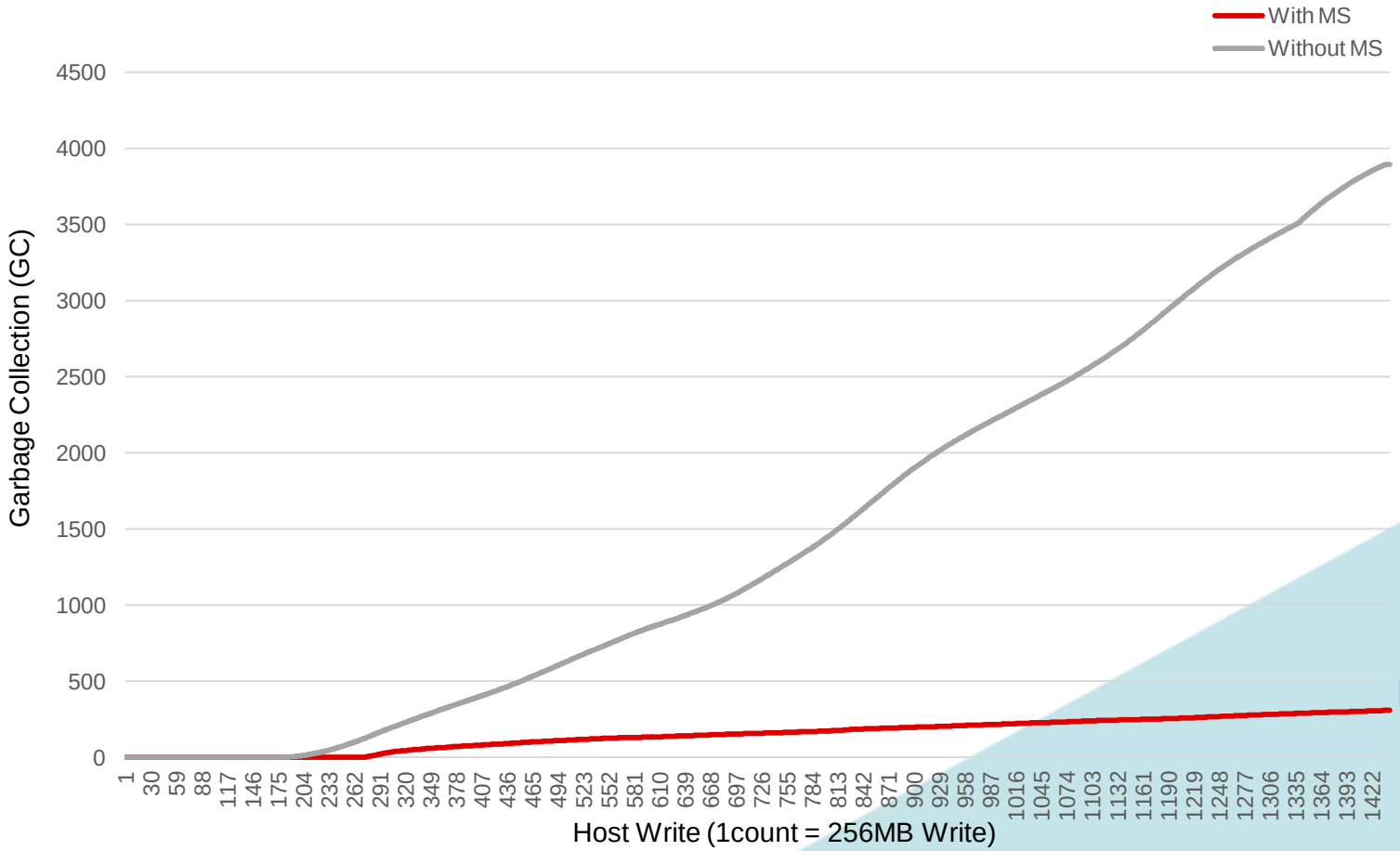


Test Environment

- System Configuration
  - Hardware
    - V310 SMDK
    - eMMC sample 64 GB
  - Software
    - Linux kernel : Linaro 3.18.57
    - Preconditioning : Complete Erase
- FIO Configuration
  - I/O Work load
    - 0% Read
    - 100% Write
  - Job Details
    - 3 parallel jobs with different life time

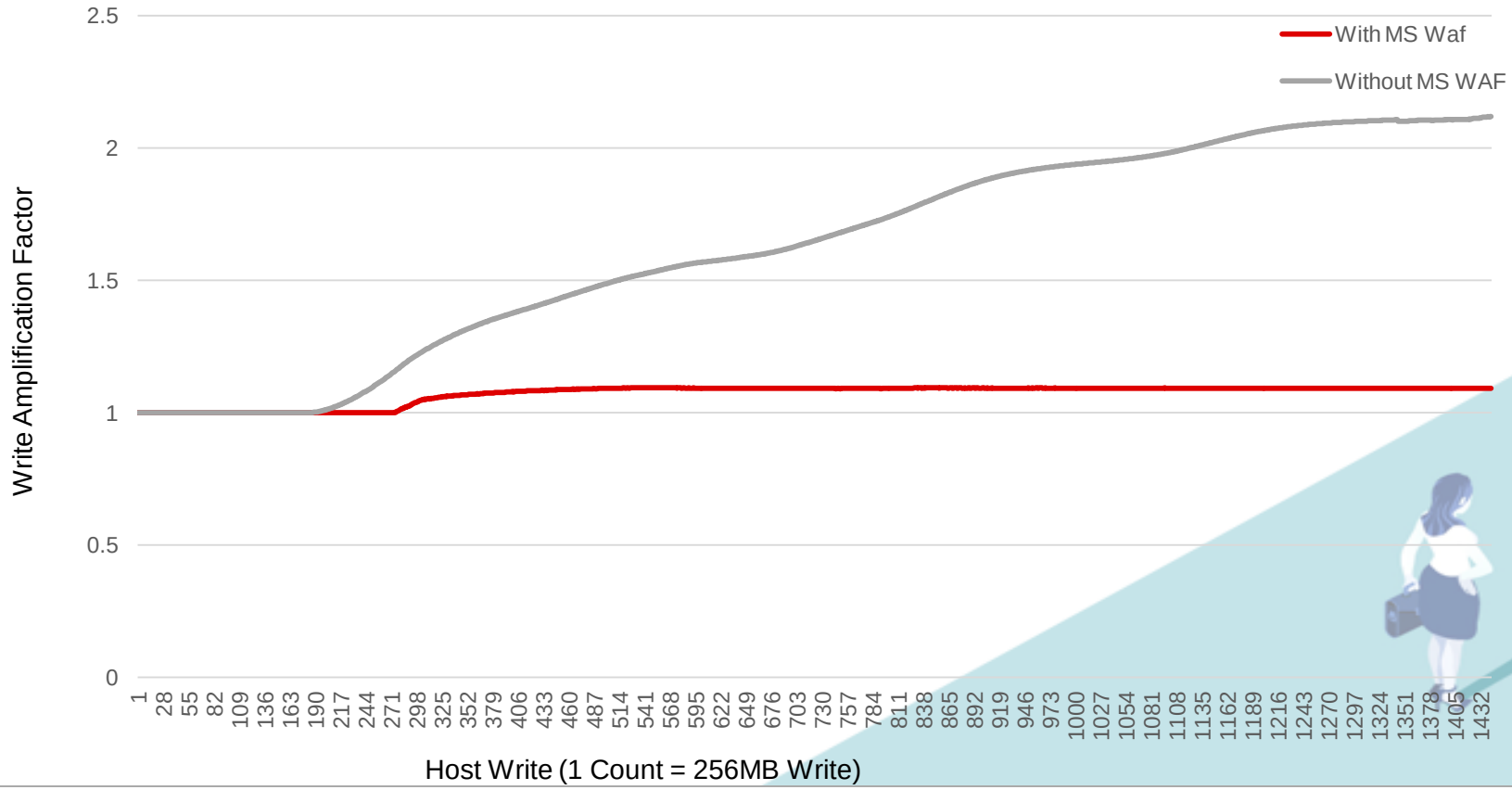
Experimental Results

- Garbage Collection(GC) copy valid existing pages from victim block to destination block.



Experimental Results

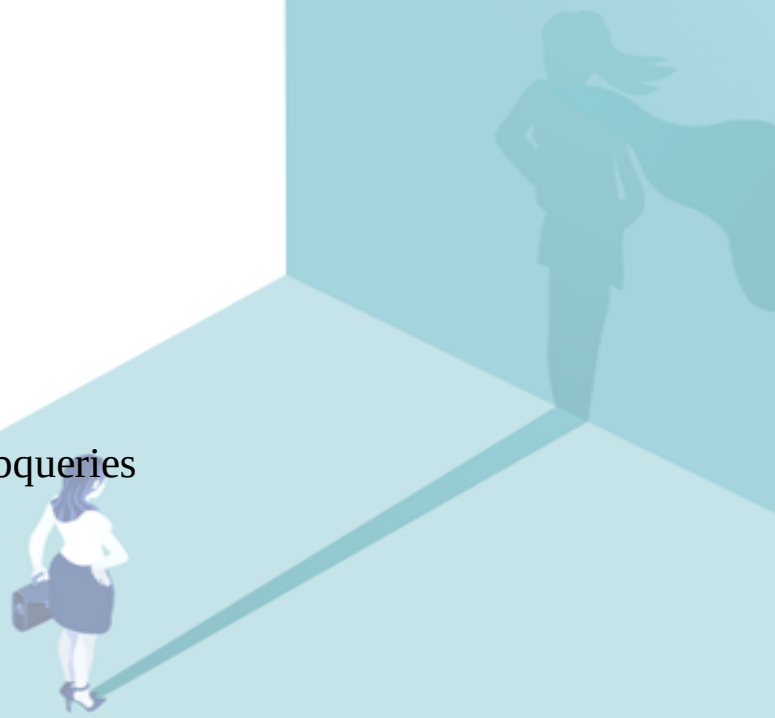
- Write Amplification Factor (WAF) = $\frac{\text{Actual NAND Write}}{\text{Host Write}}$



Real Time Use cases

- Flash Friendly File System (F2FS)
  - F2FS supports hot and cold data separation in
  - Runtime F2FS manages six active logs inside main area
    - Hot/Warm/Cold for Node
    - Hot/Warm/Cold for Data

- SQLite Data Base
  - SQLite is one of the most used data base application.
  - It uses different types files Which is having different lifetime
    - Rollback Journal/ Temp Database
    - Master Journal/ WAL files/ Shared Memory Files
    - Statement Journal/ Transient Indies/ Materialization of views and subqueries

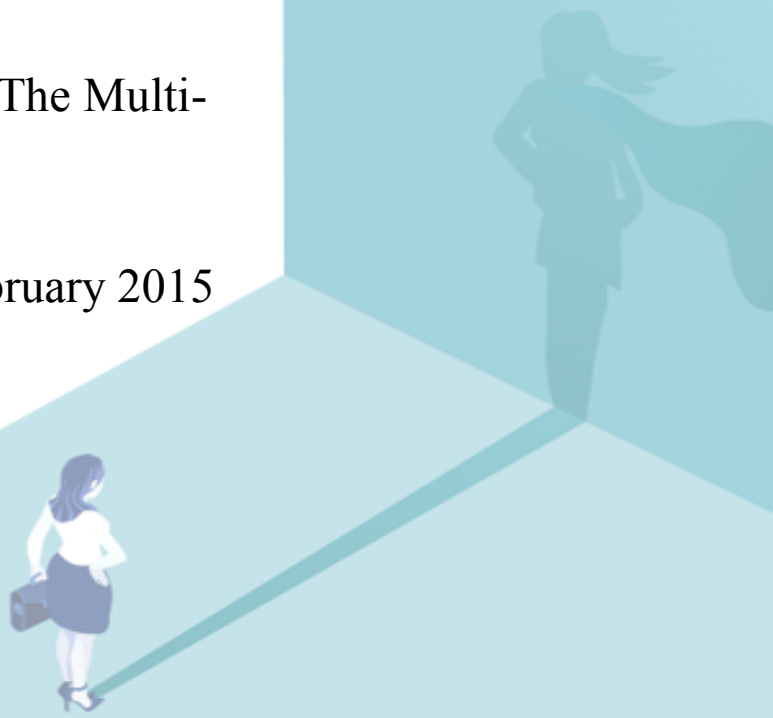


Conclusion

- Intuitive data to stream mapping can lead to large benefits in throughput, consistent latency and NAND flash life time.
- There is significant improvement in endurance as WAF for multi-streamed eMMC reduced by ~50%
- Other Applications can get similar gain from multi-streamed eMMC if host provides appropriate lifetime information of the incoming data.

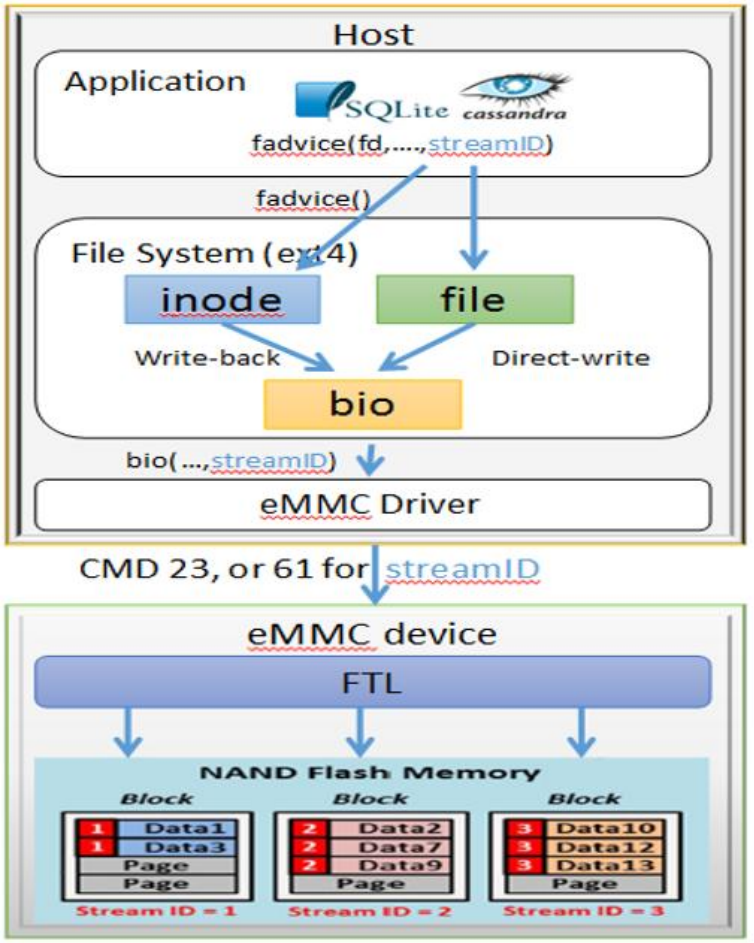
References

1. Feng Chen, David A. Kaufaty, and Xiaodong Zhang, “Understanding Intrinsic Charactristics and System Implications of Flash Memory based Solid State Drives”.
2. Jeong-Uk Kang, Jeeseok Hyun, Hyunjo Maeng, and Sangyeum cho “The Multi-streamed Solid-state Drive”
3. JEDEC eMMC Electrical Standard 5.1 “https://www.jedec.org/sites/default/files/docs/JESD84-B51.pdf” February 2015



Thank You

Multi-Streaming Architecture:



Multi-Streamed eMMC approach