AutoStream: Automatic Stream Management for Multi-stream SSDs in Big Data Era

Changho Choi, PhD
Principal Engineer
Memory Solutions Lab (San Jose, CA)
Samsung Semiconductor, Inc.
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.
Agenda

- Industry performance requests and initiatives
- Multi-stream for performance and latency improvement
- Autostream: Automatic stream management
  - Autostream implementation
  - Autostream algorithm
- Performance enhancement
- Summary
Industry Requests & Initiatives

- Deterministic IO and performance
  - Get deterministic latency and performance
  - Minimize/remove read tail latency spike
- IO Determinism initiative in NVMe TWG
  - IO and physical hardware(e.g., channel) isolation in NVM Sets
  - Control IOs with Deterministic/Non-deterministic mode
- Many researches to provide IO determinism
  - Open Channel, FPGA, etc.
- Multi-stream/AutoStream
Multi-stream: Better Performance and Lower Latency

- Store similar lifetime data into the same erase block and reduce GC overhead
- Provide better performance with lower latency
  - Application associates each write operation with a stream
  - All data associated with a stream is expected to be invalidated at the same time (e.g., updated, trimmed, unmapped, deallocated)
  - Align NAND block allocation based on application data characteristics (e.g., data lifetime)

Legacy: Without Stream
Data is written in order writes are processed

Multi-stream
Data is grouped according to stream
Multi-stream Operation

- Application maps data with different lifetime to different streams

- Provide hint about data lifetime
  - Hot data
  - Warm data
  - Cold data

- Place data with similar lifetime into the same erase unit

- SSD
  - NAND Flash Memory
    - Block
      - Stream ID = 1
        - Data1
        - Data3
        - Page
      - Stream ID = 2
        - Data2
        - Data7
        - Page
      - Stream ID = 3
        - Data10
        - Data12
        - Page

Application maps data with different lifetime to different streams.
World Transitioning To Micro-Services

Monolithic Legacy System

- Single Host
- Single Application
- MySQL or Cassandra

Relatively straightforward stream management by single application

Micro-Services Application System (e.g., Docker/Container)

- Single Host
- Multiple Applications

Non-obvious stream management and data placement
AutoStream: Automatic Stream Management

- **App. managed multi-stream delivers great benefit especially in single application systems**
  - Challenges in micro-service and multi-application systems (e.g., VM or Docker)

- **AutoStream**
  - Make stream detection independent of applications (e.g., in device driver)
  - Cluster data into streams according to data update frequency, recency and sequentiality
  - Minimize stream management overhead in application and systems

**Multi-stream**

Applications manage streams

- Application
- Filesystem
- Block Layer
- Device Driver

Automatic stream management

 SSD

Stream 1 Stream 2 Stream 3

2017 Storage Developer Conference. © Samsung Semiconductor. All Rights Reserved.
AutoStream Implementation

Application

OS kernel
- Filesystem
- Block Layer
- Device Driver

SSD

Device Driver

AutoStream module

Submission queue

Write \(<sLBA, sz>\)

AutoStream controller

Write \(<sLBA, sz, sID>\)

AS algorithm (table update)

Table:
- clD
- sID
- TL

1. \(<sLBA, sz>\)
2. \(<sLBA>\)
3. \(<sID>\)
4. \(\text{Write} <sLBA, sz, sID>\)
AutoStream Algorithm

- Divide a whole SSD space into the same size chunks
  - For example, 2MB chunk size
- Track statistics for each chunk
  - access time, access count, etc.
- Manage streams in chunk granularity
AutoStream Algorithm Leveraging Sequentiality, Frequency, Recency

AutoStream controller

<sLBA, sz>

Sequential write?

yes

no

sID := prev_sID

Get sID from stream table

Update prev_sID

Put sLBA to submission queue

Sequentiality

Stream table update

(Frequency, Recency)

Submission Q

SFR thread processes requests

Increase access_cnt

Calculate recency_weight := pow(2, (curr_time – last_access_time)/decay_period)

access_cnt := access_cnt/recency_weight

sID := log(access_cnt)
Cassandra Performance Measurement

- PM953 480GB
- Cassandra-stress
- 16KB – 10 million records
- 128 threads

More consistent ops

Graph showing performance metrics with labels: Legacy, App managed, SFR.
Performance Measurement System

**System**

**Hardware**
- Processor: 2 x Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
- DRAM: 256 GB

**Software**
- Ubuntu 16.04.2, v4.8.0 Kernel with NVMe driver
- AutoStream patch

**Device**
- Multi-Stream NVMe PM1725a 960GB SSD

**Database & Benchmark tool**

**MySQL TPC-C**
- MySQL: 5.7.12
- 1600 warehouses
- 60 connections

**Cassandra cassandra-stress**
- Cassandra 3.0
- 200M records
- 1KB record
- Workload: 50% read/50% Update
Performance Enhancement with AutoStream

- 40% MySQL Throughput enhancement
- 15% tail latency (95%, 99%) reduction

Throughput (TpmC)

Latency (ms)
Performance Enhancement with AutoStream

- Up to 40% tail latency reduction (99.9%) in Cassandra
- Better throughput
Algorithm Analysis

- **Resource requirements**
  - Memory consumption for a 480GB drive
  - CPU consumption – for background operation

<table>
<thead>
<tr>
<th>Chunk size</th>
<th>Size per chunk</th>
<th># of chunks</th>
<th>Total mem MB</th>
<th>CPU cycle</th>
<th>Binary size ~lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFR</td>
<td>2MB</td>
<td>16 bytes</td>
<td>245760</td>
<td>3.75*</td>
<td>~550</td>
</tr>
</tbody>
</table>

- **Algorithm overhead**
  - Latency: one table lookup
  - *Yet to be optimized size per chunk
  - Background operation: single thread
Performance & Latency

- IOD: IO Determinism
- MS: Multi-stream
- AS: AutoStream
- Legacy SSD

Effort & Complexity
- Low
- High

Performance

Latency Improvement
Summary

- **Application managed Multi-stream**
  - Better performance and latency especially in single application systems
  - Challenges in micro-services and multi-application systems

- **AutoStream: Automatic stream management**
  - Enhance SSD performance and tail latency
  - Great fit for multi-service and multi-application environments (e.g., Docker/container)
Multi-stream Ecosystem is Ready!

AutoStream enables easy multi-stream deployment!
changho.c@samsung.com