2017

# Enhancing SSDs with Multi-Stream: What? Why? How?

Janki Bhimani, *Northeastern University*
Jingpei Yang
Zhengyu Yang, *Northeastern University*
Ningfang Mi
N. H. V. Krishna Giri, et al.

# Enhancing SSDs with Multi-Stream: What? Why? How?

Janki Bhimani*, Jingpei Yang†, Zhengyu Yang*, Ningfang Mi*, N. H. V. Krishna Giri†,
Rajinikanth Pandurangan†, Changho Choi†, and Vijay Balakrishnan†
* Dept. of Electrical & Computer Engineering, Northeastern University, Boston, MA 02115
† Memory Solution Research Lab, Samsung Semiconductor Inc., San Jose, CA 95134

***Abstract*—The adoption of SSDs has become very prominent, but they still suffer from challenges to control write amplification. Traditional SSDs have single active append point where new data writes can be stored. Data of different lifetime stored together causes high write amplification. Recently, *multi-stream* SSDs are developed that allows multiple active append points. These multiple active append points can be used to store data of different lifetime in different locations within SSD. Such a data placement according to the lifetime of data would considerably reduce internal write amplification of SSD. For using multi-stream SSDs it is required to attach stream-id to each new incoming data writes. According to these stream-ids, the flash transition layer (FTL) of a multi-stream SSD then appends data to different erase blocks. Thus, multi-stream SSDs will help to reduce write amplification. But, to efficiently use this new multi-stream SSDs, it is important to properly identify stream-ids of data with respect to its lifetime. The lifetime of data is expected using different features that data exhibits like frequency, sequentiality etc. Stream-id identification using different features may have different impact on the final write amplification of multi-stream SSDs, depending on workload. Thus, it is required to quantify the impact of different data features that are used for stream-id identification on the resultant write amplification. Additionally, the combination of these data features may be used for stream-id identification, so it is also important to be able to study the impact of such different combinations. In order to address above challenges towards efficiently using multi-stream SSDs, here we propose a portable and adoptable framework to study the impact of stream-id identification using different workload data features and their combinations on write amplification of multi-stream SSDs. Our evaluation results show that use of appropriate features according to workload can considerably reduce the Write Amplification Factor (WAF) when compared to the legacy SSDs.***

***Keywords*—*Solid State Drives, Multi-Streaming, Write Amplification Factor (WAF), stream-id identification***

## I. Introduction

The performance and fast data access speed of SSDs have attracted users to shift from traditional HDDs to SSDs. Although the major concern regarding SSDs is the degradation of its performance over time. The internal write amplification of SSDs is one of the main reasons that impacts performance of SSDs over time. Storing the mixed data with different data-lifetime can cause high write amplification which reduces endurance and performance of SSDs [2]. The traditional SSDs (legacy SSDs) have single open erase block to append the new incoming data, which would easily lead to mixed data within SSDs. To address this issue, the storage industry recently developed a new multi-streaming technology [3], [4] that allows a host system to explicitly open different "streams" in SSD and allocate write requests to these streams according to the expected lifetime of data. A multi-stream SSD enables the device to maintain more than one open erase blocks (append points) to store data of different lifetime in different physical locations. An efficient stream assignment in multi-streaming flash drives can reduce write amplification, and improve endurance and performance of SSDs. In order to better utilize multi-stream SSDs, it is challenging to predict data lifetime and ensure that data with similar lifetime are assigned same stream-id. This work proposes a multi-streaming interface that guides data placement within SSD drives. We bring-forth the first feature-based lifetime expectation model which co-relates multiple workload features with stream-ids to ensure a good data placement. We explore the impact of different features on the lifetime of data and analyze the impact of their possible combinations on garbage collection and write amplification factor (WAF). In this paper, we present our idea of feature-based I/O stream identification, to packetize I/O data into different streams. Our technique provides a scalable method that is also capable to combine any number of application related features, to accurately capture data lifetime for construction of better streams. Our evaluation results show that usage of appropriate features according to workload can reduce the WAF up to 80% compared to legacy SSDs that do not contain streams. Moreover, we observe two very interesting behaviors that: (1) none of the features can be claimed as the best for all types of workloads, and (2) the benefit derived by using the combination of multiple features is not additive.

This work was initiated during Janki Bhimani's internship at Samsung Semiconductor Inc. [1]. This project was partially supported by the NSF Career Award CNS-1452751.

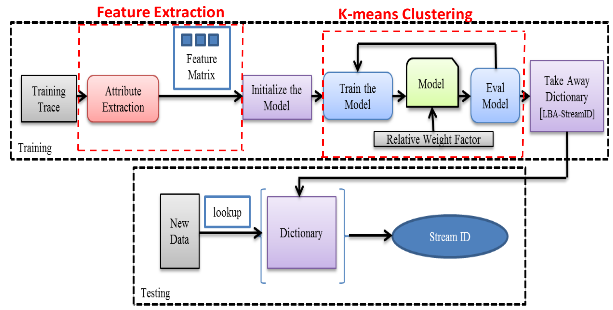
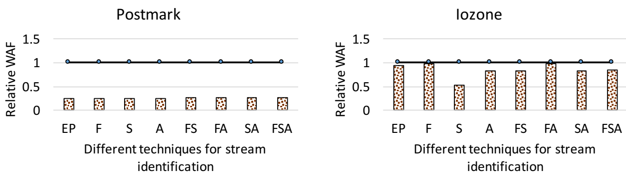



Fig. 1. Block diagram of our stream identification framework.

## II. Framework Design

The block diagram of our technique is shown in Figure 1. It consists of two main phases: *training* and *testing*. Training is the pre-processing step for stream-id identification which needs to be performed only once. While testing represents the actual runtime phase of applications during which stream-id assignment is performed. Training and testing phases are performed in a cyclic pattern to capture the runtime workload changes. Here, we use a single cycle of these two phases to explain operation of our technique. It first uses `blktrace` and `blkparse` commands to obtain a real I/O block trace from the application platform as a training trace. This trace is then used by the training phase to extract the required features such as frequency, adjacent access and sequentiality. The captured




(a) (b)

Fig. 2. Relative write amplification factor (WAF) w.r.t. legacy device by using Postmark, IOzone for Equal Partition (`EP`), and our framework with dictionary framed from different features and their combinations. (F - Frequency, S - Sequentiality, A - Adjacent access

features are enclosed in form of feature matrix that consists of $n \times m$ cells, where $m$ is the number of features analyzed for stream detection and $n$ is the total number of sector_chunks in the storage volume. The storage volume may include a single SSD or multiple SSDs. Each sector_chunk comprises of several sectors. Here, we consider 64 sectors on disk as one sector_chunk considering the storage overhead of stream-id computation and its efficiency. The sector_chunks in rows are arranged in an incremental numerological order such that the product of row number and each sector_chunk's size gives the sector_chunk address. Each cell in the feature matrix gives data quanta which reflects the importance of the $i^{th}$ sector_chunk with respect to the $j^{th}$ feature.

To obtain high computational efficiency for feature extraction, we use our improved multi-threaded K-means clustering algorithm [5] to cluster sector_chunks into K streams in parallel. In particular, each feature makes one dimension of clustering inputs and a relative weight factor can be used as an optional input to emphasis the relative importance of each feature in deciding stream-ids. By default, all features are considered to be equally important with the same weights. The number of clusters (i.e., $K$) of K-means algorithm maps to the number of streams supported by the SSD drive (e.g., 16 streams in the latest SSD drives). K-means algorithm is used to group all data points in the feature matrix into the given number of streams, and the results are stored in the dictionary (i.e., pairs of sector_chunks and stream-ids).

The system runtime is considered as the testing phase where we have actual I/O operations (e.g., writes/updates) performed on storage devices. For a read, the operation of legacy and multi-stream SSDs remains the same. For a write or an updates multi-stream SSD allows multiple append points. Thus, to decide data placement on a multi-stream SSD using our technique, a stream-id is assigned to each sector_chunk through a quick look-up in the `LBA-StreamID` (same as `Sector_chunk-StreamID`) dictionary. Thereafter, the assigned stream-ID is penetrated through the I/O stack until the data is actually written to that stream. For an erase, both legacy and multi-stream SSDs again work in the same way by searching all finalized blocks for a GC candidate, and then copying out valid pages from that candidate.

## III. Evaluation

In this section, we present our experimental results to demonstrate the reduction of write amplification using different stream identification features. *Write Amplification Factor* (WAF), is the quantitative ratio of physical writes (in bytes) on SSD to logical writes (in bytes) by applications running on the host. Lower the WAF, better the lifetime and performance of the flash device. For our evaluation environment, we adopt the flash volume structure, which consists of a logical volume of 3TB with full stripping of 128KB. We evaluate our framework by using two application workloads namely Postmark and IOzone. Figure 2 (a) and (b) shows the WAF reduction for these two benchmark traces. In detail, Figure 2 shows the relative WAF of multi-stream normalized with the WAF obtained by legacy. The x-axis of Figure 2 represents different techniques used for stream assignment in a multi-stream device. Here, `EP` is naive stream-identification technique that performs stream-id assignment by equally partitioning the available disk space. The remaining bars represent the features used to construct the dictionary for our framework. For example, *FSA* stands for a dictionary made by using a combination of *frequency*, *sequentiality* and *adjacent access*. Because the results are normalized to the WAF of legacy, the horizontal line at 1 represents the WAF of legacy with respect to itself. Thus, smaller the relative WAF, better the endurance of a multi-stream SSD compared to a legacy SSD.

In overall, WAF considerably reduced by using multi-stream SSDs when compared to the legacy SSDs. We first observe that if the workload is write intensive with a majority of random writes, then always multi-streaming gives a good reduction in WAF (e.g., Postmark which is 100% write/update workload with 98% of them as random writes). We also find that there exists a strong relation between I/O size and sequentiality. Both of these captures I/O writes happening for a long duration. For example, the average size of writes in IOzone is larger. Consequently, for IOzone the best feature to obtain the most WAF reduction is sequential (see "S" bar in Figure 2 (b)). However, we note that none of these features or combinations can be the best for both applications. Different workloads have different dominating attributes/characteristics. Different features thus have varying impacts on WAF under different workloads. Therefore we say that it is very important to have an effective approach to find a good combination of features.

## IV. Conclusion

In this paper, we build a framework to investigate the impact of stream-id assignment using different workload features, on write amplification of multi-stream SSDs. Our framework is scalable to incorporate any number of features. Our experimental results show that compared to legacy, multi-stream SSDs achieves at least 20% decrease in WAF across different workloads, which is a good qualitative improvement. We also found that different features have varying impacts on WAF of different workloads. In the future, we plan to explore better features to capture data lifetime. We would also like to study workloads of different I/O characteristics to suggest some features for better stream-identification.

## References

[1] J. S. Bhimani, J. Yang, C. Choi, and J. Huo, "Smart I/O stream detection based on multiple attributes," Nov. 4 2016, uS Patent App. 15/344,422.

[2] P. Desnoyers, "Analytic modeling of SSD write performance," in *Proceedings of the 5th Annual International Systems and Storage Conference*. ACM, 2012, p. 12.

[3] J.-U. Kang, J. Hyun, H. Maeng, and S. Cho, "The multi-streamed solid-state drive." in *HotStorage*, 2014.

[4] "Multi-Stream Technology," http://www.samsung.com/semiconductor/insights/article/25465/multistream.

[5] J. Bhimani, M. Leeser, and N. Mi, "Accelerating K-Means clustering with parallel implementations and GPU computing," in *High Performance Extreme Computing Conference*. IEEE, 2015.