An Advanced TRIM Command for Extending Lifetime of TLC NAND Flash-based Storage

Kirock Kwon†‡, Dong Hyun Kang†, Jonggyu Park†, and Young Ik Eom†
†Sungkyunkwan University, Korea ‡Samsung Electronics, Korea
{krkwon,kkangsu,whdrb2722,yieom}@skku.edu

***Abstract*—In mobile market, the popularity of TLC-based flash storage is steadily growing due to its high capacity. In this paper, we study the relationship between TLC-based storage and log-structured file system in terms of lifetime of the storage device. We also propose a novel technique, called *Segment Trimming*, which helps to eliminate the unnecessary data migration inside TLC storage. Our experimental results clearly show that our technique reduces the unnecessary data migration by 83% on average, compared with the traditional approach.**

I. INTRODUCTION

Recently, mobile devices, such as smartphones and tablet PCs, are adopting triple-level cell (TLC)-based flash storage as their main storage. This is because TLC memory can provide high-capacity by storing three bits in a TLC cell. In addition, recent advanced technologies for TLC memory make it more useful in mobile environments. For example, the emerging 3-dimensional (3D) flash technology efficiently solves the *cell interference issue* of TLC memory, which is caused by reducing the cell size, through the vertical cell stacking structure [1]. Another optimization technique is to combine single-level-cell (SLC) flash memory with TLC memory [2]. This technique can hide slower access speed of TLC memory by using high-speed SLC memory as its inner write buffer, called *SLC buffer*.

Several studies have focused on optimizing the performance of mobile storage and proposed the flash-friendly file systems, such as F2FS [3]. Unfortunately, the file system may lose the performance gains of the *SLC buffer* because the file system was designed for high-performance flash storages such as SLC or multi-level cell (MLC) flash storage.

In this paper, we first investigate the relationship between TLC-based storage and log-structured file system (LFS) in terms of the lifetime of flash storage. Especially, we focus on the block allocation mechanism of LFS because it employs the *append-only policy* to reshape random write requests into sequential ones. However, this policy would be harmful in TLC-based storage because of two reasons. First, LFS does not send a hint of "page invalidation" to the underlying flash storage when a block in the file system is updated with the *append-only policy*. Second, the TLC storage always flushes all pages in the *SLC buffer* to TLC memory during idle time to reclaim free space. As a result, pages that have invalid blocks of the file system may unnecessarily be migrated from the SLC buffer to TLC memory. We propose a novel technique, called Segment Trimming to eliminate the unnecessary data migration and implement it on F2FS file system which is a log-structured file system designed for flash storages. We performed experiments with a real storage and our results show that Segment Trimming can reduce unnecessary data migration by 83% on average.

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2015M3C4A7065696). Young Ik Eom is the corresponding author of this paper.

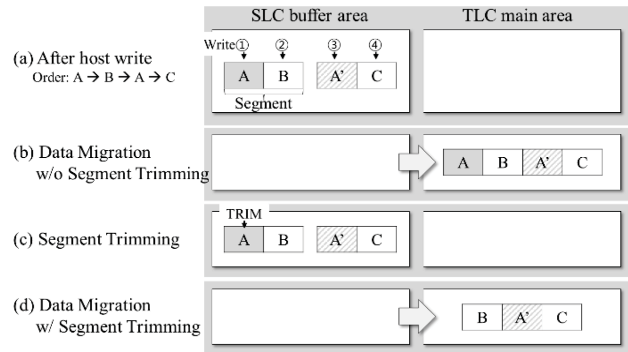

Fig. 1. SLC-buffer operation with and without Segment Trimming

II. BACKGROUND AND MOTIVATION

A. SLC buffer in TLC NAND-based Storage

NAND flash memories can be classified into three types relative to the number of bits that can be stored in a cell: SLC, MLC, and TLC. Each cell can store one, two, and three bits, respectively. As more bits are stored in a cell, flash memories offer higher density and lower-priced products. However, TLC NAND storage has several disadvantages, such as lower performance and endurance: The write performance of TLC is about one-fifth of that of SLC and the endurance of TLC is about one-hundredth of that of SLC. To overcome these limitations, TLC NAND-based storage adopts an SLC buffer as a high-performance write buffer. Thereby, it can provide similar write performance to that of SLC NAND-based storage: the data is first written to the high-speed SLC-buffer and then migrated to TLC memory during idle time.

B. Effect of LFS on SLC buffer

One of the important features of LFS is that all writes are performed sequentially to improve write performance [4] (Append-only write policy). For the append-only write policy, LFS sequentially allocates new blocks for write requests that overwrite existing data, and then marks the existing data as invalid. However, the append-only write policy can reduce the lifetime of TLC NAND-based storage that adopts an SLC buffer. Because LFS does not overwrite data, storage devices have no information about the validity of pages on the file system layer (The semantic gap between storage device and log-structured file system). Consequentially, it unnecessarily migrates invalid data from the SLC buffer to TLC memory during the migration process. Figure 1 illustrates the unnecessary migration caused by the semantic gap. In Figure 1(a), there are two segments and each segment consists of two pages. Here, the SLC buffer handles the write requests in the order of A, B, A', and C, where A' invalidates A on the file system layer. As shown in Figure 1(b), all pages are migrated to TLC memory although page A is invalid on the file system layer because of the semantic gap. This unnecessary migration increases the number of write operations, resulting in TLC lifetime reduction.

C. Trim commands

NAND-based storages need garbage collection to reclaim free space because it does not support in-place update. However, the semantic gap between file system layer and storage layer exacerbates the overhead of garbage collection since pages that are invalid on the file system layer are unnecessarily copied to other pages during garbage collection. To reduce the semantic gap, file systems employ TRIM command, which informs NAND-based storage of invalid pages [5]. Using the information of invalid pages given by TRIM commands, NAND-based storage can eliminate unnecessary copy operations. These observations strongly motivate us to develop a technique that improves TRIM commands in order to eliminate the migration of invalid pages from SLC buffer to TLC memory.

III. Design of Segment Trimming

In this paper, our goal is to eliminate the migration of invalid pages from SLC buffer to TLC memory without any performance degradation. To reach this goal, we propose a novel technique, called Segment Trimming, on LFS. As mentioned in Section 2, data that are written into the SLC buffer is migrated to TLC memory during idle time. Segment Trimming on LFS notifies the underlying storage of invalid pages before the migration, by issuing TRIM commands. After the Segment Trimming operations, the storage can selectively migrate valid pages in the SLC buffer to TLC memory. To prevent performance degradation, Segment Trimming is triggered when there is no IO requests in the request queue.

Overall, the steps of Segment Trimming are as follows: (1) Segment Trimming performs checkpointing to guarantee the data consistency in case of failure, (2) It examines the segment information recorded in the segment summary block to get information on which pages are invalid in the segment, and (3) It issues the TRIM command for the invalid pages.

For example, as shown in Figure 1(c), the TRIM command is issued for invalid pages by a Segment Trimming operation. After the Segment Trimming operation, three valid pages (B, C, and A') are migrated from the SLC-buffer to TLC memory, as shown in Figure 1(d). As compared with Figure 1(b), Segment Trimming can eliminate unnecessary data migration (page-A in Figure 1(c)).

IV. Evaluation

In order to evaluate the performance of our scheme, we implemented Segment Trimming on F2FS. We conducted the evaluation on a PC with 3.40GHz Intel Core i7-2600 CPU with 8GB RAM. The system ran Ubuntu 14.04, 64-bit with Linux kernel version 4.5.4, and we used 120GB Samsung 850 EVO SSD, which is based on TLC V-NAND and 3GB SLC buffer. To examine the effectiveness of Segment Trimming more clearly, we conducted the evaluation using synthetic workloads based on sequential writes by varying the ratio of overwrites.

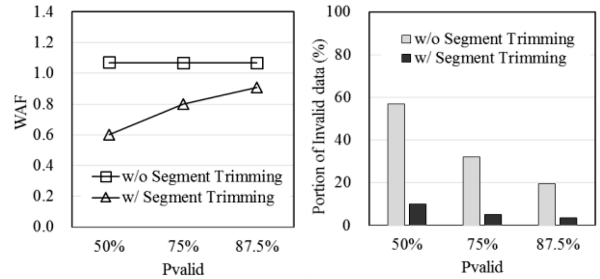

Fig. 2. Experimental results: (a) Write amplification factor, (b) Portion of invalid data migration from SLC-buffer to TLC memory

We examined the effectiveness of Segment Trimming by measuring the write amplification factor (WAF). Let P_{valid} denote the ratio of valid data of our synthetic workload. It can be expressed as $P_{valid} = \frac{C_{total} - C_{overwrite}}{C_{total}}$, where C_{total} and $C_{overwrite}$ denotes the amount of total writes and overwrites, respectively. As illustrated in Figure 2(a), we measured WAFs by varying the ratio of valid data (P_{valid} = 50, 75, 87.5%). Without Segment Trimming, WAFs are 1.07 regardless of P_{valid}. This shows that all of the written data (both valid and invalid data) are migrated from the SLC buffer to TLC memory. With Segment Trimming, WAFs are much lower than 1.0 and proportional to P_{valid}. This shows that the underlying storage can selectively migrate valid pages after Segment Trimming operation. From these WAF values, we calculated the portion of invalid pages that are migrated from the SLC buffer to TLC memory. As shown in Figure 2(b), Segment Trimming can reduce 83% of unnecessary data migration on average, compared with the traditional approach.

V. Conclusion

In this paper, we studied the write amplification caused by the semantic gap between TLC NAND-based storage and log-structured file system. We also introduced Segment Trimming for log-structured file systems to extend the lifetime of TLC NAND-based storages. Our results show that Segment Trimming can significantly eliminate the migration of invalid pages from the SLC buffer to TLC memory.

Reference

[1] K.-T. Park, *et al.*, "Three-dimensional 128Gb MLC Vertical NAND Flash-Memory with 24-WL Stacked Layers and 50MB/s High-speed Programming," In *Proc. of ISSCC*, 2014.

[2] Samsung, Samsung Solid State Drive TurboWrite Technology White Paper, *Samsung Electronics Co, Tech. Rep*., 2013.

[3] C. Lee, D. Sim, J.-Y. Hwang, and S. Cho. "F2FS: A New File System for Flash Storage." In *Proc. of the 13th USENIX FAST*, 2015.

[4] M. Rosenblum and J. Ousterhout, "The Design and Implementation of a Log-structured File System," *ACM Trans. on Computer Systems*, vol. 10, no. 1, pp. 26-52, 1992.

[5] G. Kim and D. Shin, "Performance Analysis of SSD Write using TRIM in NTFS and EXT4," In *Proc. of ICCIT*, 2011.