An Integrated Approach for Managing the Lifetime of Flash-Based SSDs

Sungjin Lee, Taejin Kim, Ji-Sung Park, and Jihong Kim
Department of Computer Science and Engineering, Seoul National University, Korea
{*chamdoo, taejin1999, jspark, jihong*} *@davinci.snu.ac.kr*

***Abstract*—As the semiconductor process is scaled down, the endurance of NAND flash memory greatly deteriorates. To overcome such a poor endurance characteristic and to provide a reasonable storage lifetime, system-level endurance enhancement techniques are rapidly adopted in recent NAND flash-based storage devices like solid-state drives (SSDs). In this paper, we propose an integrated lifetime management approach for SSDs. The proposed lifetime management technique combines several lifetime-enhancement schemes, including lossless compression, deduplication, and performance throttling, in an integrated fashion so that the lifetime of SSDs can be maximally extended. By selectively disabling less effective lifetime-enhancement schemes, the proposed technique achieves both high performance and high energy efficiency while meeting the required lifetime. Our evaluation results show that the proposed technique, over the SSDs with no lifetime management schemes, improves write performance by up to 55% and reduces energy consumption by up to 43% while satisfying a 5-year lifetime warranty.**

I. Introduction

As the cell size of NAND flash memory is shrinking down to 20 nm (and below), both the capacity per NAND chip and the price per byte are quickly improving, thus opening many new market opportunities for NAND flash memories. However, a significant reduction in the number of program/erase (P/E) cycles with newer process generations is emerging as one of the main obstacles to the wider adoption of NAND flash-based solid-state drives (SSDs) in various computing environments.

This poor endurance characteristic of high-density NAND flash memory is mainly due to unreliable physical device properties. To overcome such physical limitations of the semiconductor substrate, various specialized system-level lifetime-enhancement techniques are proposed. Unlike conventional approaches (e.g., wear-leveling), these system-level approaches improve the SSD lifetime by employing specialized software algorithms which are often supported by hardware accelerators. For example, hardware-accelerated lossless compression [1] and deduplication techniques [2] belong to such system-level approaches. They reduce the amount of write traffic sent to flash memory by eliminating redundant bit patterns or by avoiding the repeated writes of duplicate data, thereby improving the lifetime of SSDs. Performance throttling [3] is another promising system-level approach for extending the storage lifetime. By intentionally limiting the write bandwidth of SSDs, it guarantees the predetermined SSD lifetime at the cost of increased write response times.

Although individual lifetime-enhancement techniques have been well-studied, little has been known about how these techniques interact with each other. To maximally extend the lifetime of SSDs, however, it would be desirable to combine several lifetime-enhancement techniques. The efficiency of each system-level technique on lifetime, performance, and energy consumption varies significantly depending on the characteristics of a particular workload. Therefore, it is important to integrate individual techniques in a systematic fashion so that the combined lifetime-enhancement techniques provide an optimal storage solution in response to a given workload. For example, if the compressibility of a certain workload is high, using lossless compression is a good choice because it reduces a large amount of write traffic sent to flash memory, having positive effects on performance, lifetime, and energy consumption. However, for a different workload whose compressibility is low, it would be better to disable compression to avoid useless energy consumption which is required for compressing incompressible data. Considering the decreasing endurance of recent high-density NAND flash memory, more advanced lifetime-enhancement techniques (e.g., delta compression and near deduplication) are expected to be employed in future SSDs. Thus, integrating different lifetime-enhancement techniques efficiently will become one of the most important design considerations in designing future SSDs.

In this paper, we propose an integrated lifetime management approach that guarantees the lifetime warranty of SSDs, while maximizing their performance with high energy efficiency. The proposed lifetime management technique prolongs the lifetime of SSDs with less performance/energy overhead by intelligently employing different combinations of lifetime-enhancement schemes. Our technique considers both the remaining SSD lifetime requirement and the workload characteristics in selecting the most appropriate combination of lifetime-enhancement schemes during runtime. In order to evaluate the proposed lifetime management technique, we implemented a trace-driven SSD simulator and performed preliminary comparative studies, using various real-world traces. Our evaluation results show that the proposed technique guarantees the target SSD lifetime while reducing the write response time and energy consumption by up to 55% and 43%, respectively, over the SSDs with no lifetime management schemes.

II. Target SSD Architecture

In this study, we primarily focus on three well-known lifetime-enhancement techniques, including lossless compression, data deduplication, and performance throttling because they are commonly employed in recent SSDs. However, the proposed lifetime management technique can be easily extended to deal with other lifetime-enhancement schemes.

Fig. 1 depicts an architectural overview of our target SSDs.

978-3-9815370-0-0/DATE13/©2013 EDAA

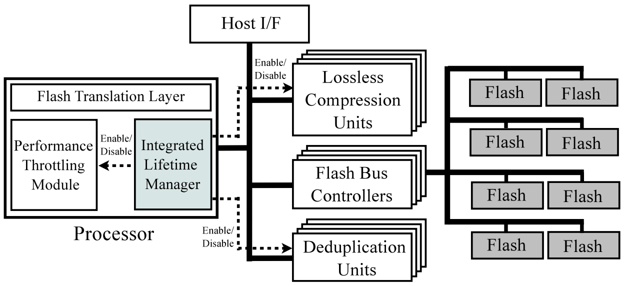


Fig. 1: An overall architecture of our target SSDs.

There are multiple flash buses, each of which holds several flash chips, and multiple flash bus controllers that can simultaneously handle several read/write requests. Our target SSDs employ hardware acceleration units for lossless compression and deduplication, respectively. A processor inside the SSD is responsible for executing software modules, including a flash translation layer (FTL) and a performance throttling module. Our integrated lifetime management technique is implemented as software and manages three lifetime-enhancement schemes.

III. Integrated Lifetime Management

In this section, we explain how lifetime management technique integrates and manages several lifetime-enhancement schemes. Note that we did not consider read operations in this work because their effects on lifetime, performance, and energy are insignificant in comparison with write operations.

A. Integration of Individual Lifetime-Enhancement Schemes

The primary goal of the proposed lifetime management technique is to guarantee a preset lifetime of SSDs, which is usually set to 3-5 years. Therefore, the proposed technique reconfigures (i.e., enable or disable) individual lifetime-enhancement techniques to achieve higher performance with less energy consumption while guaranteeing the required lifetime. To make a proper decision, the lifetime management technique needs to understand the respective effect of individual techniques on lifetime, performance, and energy consumption. For this reason, the proposed technique employs an interval-based approach, dividing the required SSD time into smaller epochs. For each epoch, it collects the information required for a decision. At the beginning of every epoch, the lifetime manager makes a decision on whether to enable or disable each lifetime-enhancement scheme by using the information collected from previous epochs. This epoch-based approach is effective in adapting to a changing workload.

If there is a write request from a host system (and deduplication is enabled), the lifetime manager computes a hash key for the requested data, which is a unique ID of the data, using the deduplication unit. The lifetime manager then sees if the same hash key exists in the hash table that contains a set of hash keys for the data previously written. If there exists a matched hash key, it means that the same data were already written to flash memory. Thus, the integrated manager updates the FTL's address mapping table only so that the logical page address of the requested data points to the physical page of the data previously written, instead of writing the data to flash memory. If there is no matching key, the requested data must be written to flash memory, and a new hash value is inserted into the hash table.

If the requested data cannot be eliminated by data deduplication (and lossless compression is enabled), the integrated lifetime manager compresses the requested data using the hardware-accelerated lossless compression unit. If lossless compression is conducted for an individual page, it causes a serious internal fragmentation problem [1]. That is, the size of compressed data is smaller than a page, wasting the rest of the page. Thus, the actual number of pages written to flash memory is not reduced. To resolve this problem, the integrated manager compresses several pages together, e.g., four pages, as a compression/decompression unit, and then writes them to flash memory at once. We call this compression/decompression unit as a compression chunk.

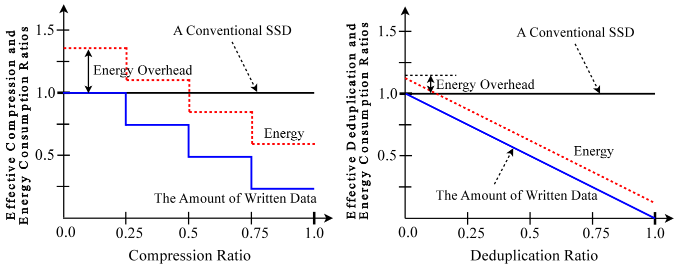
The lossless compression and data deduplication modules help to extend the lifetime of SSDs because they reduce the amount of data written to SSDs. In addition, they can also improve the performance of SSDs because both compression and deduplication effectively reduce the total number of write operations to flash memory. Since data compression and deduplication are supported by special hardware accelerated units, their impact on the write latency is negligible over slow flash access times. The main drawback of both data compression and deduplication may be their increased energy consumption, especially when data compression or/and duplication perform poorly with a workload.

If the required lifetime cannot be guaranteed with deduplication and compression, the integrated manager enables performance throttling. The performance throttling module determines a throttling delay and regulates the write speed of SSDs by delaying write operations. Performance throttling is the most effective way of guaranteeing the required lifetime. Moreover, its decision logic is relatively simple, so extra energy consumption is negligible. However, since it regulates the write bandwidth of SSDs, a relatively high write performance penalty cannot be avoided.

B. Integrated Lifetime Management Technique

In this subsection, we first describe important parameters used for estimating the benefits of the lifetime-enhancement schemes. We then explain our lifetime management algorithm.

1) Estimation of epoch write demand and epoch capacity: On the start of each epoch e, the lifetime manager estimates how many bytes *will be written* during the epoch e and decides how many bytes *are allowed to be written* to flash memory until the epoch e ends. The former is called the *epoch write demand* W_e and the latter is called the *epoch capacity* C_e. The epoch write demand W_e can be estimated in many different ways. We predict W_e as a weighted average of previous 10 epochs' write demands. Deciding the epoch capacity C_e is more complicated. We first estimate the remaining P/E cycles before all blocks are worn out. The remaining P/E cycles are then translated into the total number of remaining bytes that can be written to flash memory. C_e is obtained by dividing the total number of remaining bytes by the remaining number of epochs. For all the epochs, if the proposed technique reduces



(a) Lossless Compression (b) Deduplication

Fig. 2: The amount of written data and energy consumption, over a conventional SSD, depending on compression and deduplication ratios. In lossless compression, we assume that one compression chunk is composed of four pages.

W_e so that it becomes smaller than C_e, the required storage lifetime can be guaranteed.

2) Estimation of benefits of lifetime-enhancement schemes: Once W_e and C_e are decided, the lifetime manager estimates the benefits of compression and deduplication for the epoch e, over a conventional SSD, in terms of lifetime, performance, and energy. These benefits are estimated using the compression and deduplication ratios of the epoch e. Since the compression/deduplication ratios are unknown, the lifetime manager predicts them using the history of the previous epochs.

Fig. 2 shows how the amount of data written to flash memory and the energy consumption are affected as compression and deduplication ratios change. Here, the Y-axis indicates the effective compression or deduplication ratio and the energy consumption ratio over a conventional SSD. The effective compression or deduplication ratio indicates the ratio of the amount of data written to flash memory over a case when no lifetime-enhancement scheme is used. As the amount of data written to flash memory is reduced, the benefits in lifetime and performance are increased. The smaller the energy consumption ratio is, the better the energy benefit is. In this work, the effective compression and energy consumption ratios of lossless compression are denoted by D_{comp} and E_{comp}, respectively. Similarly, the effective deduplication and energy consumption ratios of deduplication are denoted by D_{dedup} and E_{dedup}, respectively.

In the case of lossless compression, which is depicted in Fig. 2(a), the lifetime manager compresses several pages together into one compression chunk, and then writes them to flash memory at once. The effective compression and energy consumption ratios are reduced whenever the number of physical pages that compose a compression chunk is reduced. Due to relatively high base energy consumption, lossless compression requires a high compression ratio to reduce the energy consumption. For example, in Fig. 2(a), we can benefit from lossless compression in terms of both lifetime and energy consumption when a compression ratio is higher than 0.5. Unlike compression, deduplication does not have the fragmentation problem. Thus, as shown in Fig. 2(b), the amount of written data and the energy consumption of deduplication decrease linearly in proportion to a deduplication ratio.

3) Lifetime management algorithm: Algorithm 1 summarizes the lifetime management algorithm which determines whether to enable or disable each lifetime-enhancement scheme using the estimated benefits of each scheme.

Algorithm 1: Lifetime Management Algorithm

1: $W_e \leftarrow$ The number of bytes to be written during the epoch e;
2: $C_e \leftarrow$ The number of bytes allowed to be written during the epoch e;
3: $(D_{comp}, E_{comp}) \leftarrow$ The effective compression and energy consumption ratios;
4: $(D_{dedup}, E_{dedup}) \leftarrow$ The effective deduplication and energy consumption ratios;
5: $W_e' \leftarrow W_e$;
6: **if** $(D_{dedup} < 1.0$ **and** $E_{dedup} < 1.0)$ {
7: $W_e' \leftarrow W_e' \times D_{dedup}$;
8: is_dedup_enabled $\leftarrow$ **true**;
9: }
10: **if** $(D_{comp} < 1.0$ **and** $E_{comp} < 1.0)$ {
11: $W_e' \leftarrow W_e' \times D_{comp}$;
12: is_compr_enabled $\leftarrow$ **true**;
13: }
14: **if** $(W_e' < C_e)$ {
15: /* the decision process is finished */
16: } **else** {
17: $W_e'' \leftarrow W_e'$;
18: **if** (is_compr_enabled is **false and** $(1.0 - D_{comp}) \geq (E_{comp} - 1.0)$) {
19: $W_e'' \leftarrow W_e'' \times D_{comp}$;
20: is_compr_enabled $\leftarrow$ **true**;
21: }
22: **if** (is_dedup_enabled is **false and** $(1.0 - D_{dedup}) \geq (E_{dedup} - 1.0)$) {
23: $W_e'' \leftarrow W_e'' \times D_{dedup}$;
24: is_dedup_enabled $\leftarrow$ **true**;
25: }
26: **if** $(W_e'' > C_e)$ {
27: Calculate a throttling delay using W_e'' and C_e;
28: is_throttling_enabled $\leftarrow$ **true**;
29: /* the decision process is finished */
30: }
31: }

Algorithm 1 is invoked at the beginning of every epoch. All the lifetime-enhancement schemes are initially disabled. The lifetime manager enables the lifetime-enhancement schemes when the effective compression (or deduplication) ratio is smaller than 1.0 and the energy consumption ratio is also smaller than 1.0 (lines 7-8 and 11-12 of Algorithm 1). In this case, lifetime, performance, and energy consumption are all improved. The write traffic W_e is also reduced to W_e'.

If W_e' is smaller than C_e (line 14), it is not necessary to further reduce write traffic using additional lifetime-enhancement schemes. However, if W_e' is larger than C_e (lines 17-32), it is difficult to guarantee the required lifetime because write traffic is too heavy. Therefore, the lifetime manager attempts to reduce write traffic further by employing more lifetime-enhancement schemes. We first check whether lossless compression is already enabled or not. If it is disabled, lossless compression is enabled to reduce the number of writes to flash memory. It must be noted that if a compression ratio is very small, its effect on improving the lifetime is negligible even though a considerable amount of energy is consumed for data compression (e.g., the compression ratio is smaller than 0.25 in Fig. 2(a)). Thus, our lifetime management technique enables lossless compression only when the improvement of lifetime and performance is larger than an increase in energy consumption. In other words, data compression is enabled if $(1.0 - D_{comp}) \geq (E_{comp} - 1.0)$. The same procedure is conducted for data deduplication. As shown in Algorithm 1, the reduced write traffic by using additional lifetime-enhancement schemes is denoted by W_e''.

Finally, if W_e'' is smaller than C_e, our decision process is finished. However, if W_e'' is larger than C_e (lines 27-29), it is necessary to enable performance throttling. The lifetime manager decides a throttling delay so that the amount of data

	Baseline	*Throttling*	*Compression*	*Dedup*	*Integrated*
WKSN	6.8	6.8	20.9	8.3	23.7
Media	3.8	5.01	3.85	3.81	5.01
DVMT	3.04	5.01	3.65	2.92	5.04

TABLE I: SSD lifetimes (years) for five configurations.

written to the SSDs during the epoch e is smaller than C_e.

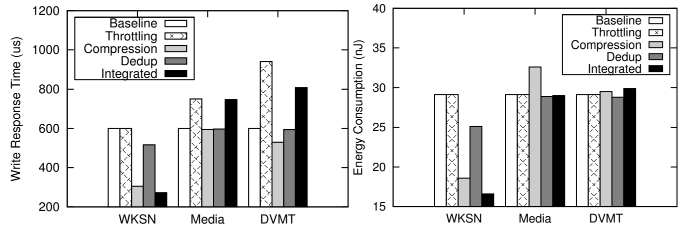
IV. Preliminary Results

In order to evaluate the proposed lifetime management technique, we implemented a trace-driven SSD simulator that supports three lifetime-enhancement schemes. 20 nm 2-bit MLC flash chips are assumed to be used in the target SSD. Each block is composed of 4 KB 128 pages and supports 3K P/E cycles. The SSD capacity is 64 GB and the target lifetime is set to 5 years. The compression chunk is composed of 4 pages and the epoch length is set to 5 minutes. The time taken for compressing 4 KB page data is 64 μs and 60 mW is consumed. For deduplication, the required time for hash computation is 8.6 μs and the power of 49 mW is required. A flash controller takes 600 μs consuming 48.5 mW to write a page to flash memory. The performance values are measured in our custom FPGA-based SSD prototype. The power values are estimated using an FPGA power estimation tool.

We compare the performance, lifetime, and energy consumption of five SSD configurations: *Baseline*, *Throttling*, *Compression*, *Dudup*, and *Integrated*. *Baseline* is a baseline SSD configuration without any lifetime-enhancement schemes. *Throttling*, *Compression*, and *Dedup* employ performance throttling, lossless compression, and deduplication techniques, respectively. *Integrated* is the proposed integrated lifetime management technique. We use three traces, WKSN, Media, and DVMT, for the evaluation. WKSN records I/O activities of a high-end PC and Media captures I/O activities while uploading and downloading multimedia files. DVMT is collected from a server used for software/hardware development.

Table I compares the lifetime of SSDs with different configurations. The WKSN trace exhibits relatively low write traffic, so the 5-year lifetime can be guaranteed for all five configurations. WKSN also exhibits very high compression and deduplication ratios. *Compression*, *Dedup*, and *Integrated* thus achieve a longer SSD lifetime than other configurations by reducing the amount of data written to flash memory. The Media trace exhibits heavy write traffic with very low compression and deduplication ratios. Therefore, only *Throttling* and *Integrated* that employ performance throttling can guarantee the 5-year SSD lifetime. The DVMT trace is also a write intensive workload. Even though the compression ratio of DVMT is relatively high (e.g., 10%), it is not so high to guarantee 5-year lifetime. For this reason, *Throttling* and *Integrated* can satisfy the required lifetime.

Fig. 3 shows the write response time and the energy consumption required to write a single page. For WKSN, *Integrated* achieves the best performance and consumes the least energy among all the configurations: it reduces the write response time and the energy consumption by 55% and 43%, respectively, over *Baseline*. These benefits are due to the fact that *Integrated* eliminates lots of write operations by exploiting both the high compression ratio and the high deduplication ratio.



(a) Write response time (μs) (b) Energy consumption (nJ)

Fig. 3: Performance and energy results.

For Media, *Integrated* does not perform lossless compression to avoid useless energy consumption because the compression ratio is very low. *Integrated* enables data deduplication because the hardware unit consumes a small amount of energy for hash computation. However, due to a quite low deduplication ratio, the benefit of using deduplication is trivial. As a result, 5-year lifetime can be satisfied by throttling write performance. Unlike *Integrated*, *Compression* performs lossless compression for incompressible data, consuming more energy uselessly without performance and lifetime benefits.

In DVMT, *Integrated* uses all three lifetime-enhancement techniques. The energy consumption of SSDs is increased by 3% because of the energy overhead caused by lossless compression. However, it reduces a large amount of write traffic sent to flash memory by using lossless compression, and thus the required lifetime can be ensured with less performance throttling. *Throttling* also guarantees the 5-year lifetime warranty, but the write response time is much longer than *Integrated*. This is because *Throttling* applies a longer throttling delay to guarantee the SSD lifetime.

V. Conclusion

In this paper, we proposed the integrated lifetime management technique that guarantees a target lifetime with higher performance and lower energy consumption. The proposed lifetime management technique combines several lifetime-enhancement techniques and then dynamically reconfigures them at runtime to realize a lifetime-, performance-, and energy-optimized storage device. Our evaluation results showed that the proposed technique achieves 55% higher performance and 43% less energy consumption while ensuring a 5-year SSD lifetime.

Acknowledgment

This research was supported by WCU (World Class University) program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (MEST) (No. R33-2012-000-10095-0). This work was also supported by NRF funded by MEST (No. 2012-0006417). The ICT at Seoul National University and IDEC provided research facilities for this study.

References

[1] Y. Park and J.-S. Kim, “Compression Support for Flash Translation Layer,” in *Proceedings of the International Workshop on Software Support for Portable Storage*, 2010.

[2] F. Chen, T. Luo, and X. Zhang, “CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory Based Solid State Drives,” in *Proceedings of the USENIX Conference on File and Storage Technologies*, 2011.

[3] S. Lee, T. Kim, K. Kim, and J. Kim, “Lifetime Management of Flash-Based SSDs Using Recovery-Aware Dynamic Throttling,” in *Proceedings of the USENIX Conference on File and Storage Technologies*, 2012.