Improving the Sustainability of Solid-State Drives by Prolonging Lifetime

Zhaokang Ke[†], Dingyi Kang[*], Bo Yuan[‡], David Du[†], and Bingzhe Li[*]
[†]University of Minnesota, Twin Cities
[*]University of Texas at Dallas
[‡]Rutgers University
{ke000030, du}@umn.edu; bo.yuan@soe.rutgers.edu; {dingyi.kang, bingzhe.li}@utdallas.edu

2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) | 979-8-3503-5411-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISVLSI61997.2024.00096

***Abstract*—In the big data era, vast amounts of data are processed and stored in cloud data centers using Solid-State Drives (SSDs). While SSDs are crucial for efficient data handling as the fast-tier in tiered storage, they also contribute significantly to embodied carbon emissions. In this paper, we explore a method to enhance SSD sustainability by reducing the maximum threshold voltage of flash cells, extending lifespan and improving write performance. However, this leads to decreased read performance due to a high Raw Bit Error Rate (RBER). To improve the SSD performance while prolonging its lifespan, we partition the device into two regions. One region maintains standard voltage levels to efficiently manage read-intensive tasks. The other with a reduced threshold voltage absorbs write-intensive data, leveraging the benefits of extended lifespan and enhanced write performance provided by low-voltage operation. Additionally, we propose a dynamic caching policy that caches frequently accessed data to the DRAM in SSD. Finally, our experimental results demonstrate that the proposed design improves the SSD lifespan by 8.7% to 20.3% compared to some previously proposed schemes while achieving average latency reductions ranging from 19.7% to 69.8%. Our work demonstrates the potential for enhancing the sustainability of SSDs while ensuring high performance, contributing to the reduction of the carbon footprint associated with storage systems in the era of big data.**

***Index Terms*—Sustainability, Solid-State Drives (SSDs), NAND Flash Memory, Error Correction Codes (ECC)**

I. Introduction

In the era of big data, massive amounts of data generated at edge nodes are moved and processed in cloud data centers, involving a wide range of storage devices. This has placed a spotlight on the sustainability of IT infrastructures, propelled by a significant increase in data volume and the resultant carbon emissions from data storage and management activities. Among various storage options, NAND flash-based memory drives are particularly crucial, ranging from mobile devices to large data centers [1], [2]. These drives have largely replaced older magnetic storage due to their speed and compactness. As the technology evolves, there is a trend towards higher storage density in flash cells. Innovations have moved from multi-level cell (MLC) to denser triple-level cell (TLC) and quad-level cell (QLC), which store more bits per cell. While this increases economic efficiency, it compromises sustainability and performance by reducing the reliability and lifespan of flash memory, as seen in fewer Program-Erase (PE) cycles.

The lifespan and access times of NAND flash memory can be balanced by adjusting parameters such as threshold voltages and error correction code iterations. For example, lowering the threshold voltage extends lifespan but may increase write latency. Additionally, sophisticated Error Correction Codes (ECC), like Low-Density Parity-Check Codes (LDPC), help manage the higher Raw Bit Error Rates (RBERs) and extend the operational lifespan of flash memory systems, enhancing SSD sustainability. However, enhancing sustainability in SSDs is challenging due to the complex error correction processes like LDPC, which can slow data access, creating a trade-off between extending lifespan and maintaining performance. In data centers, where meeting Service-Level Agreements (SLAs) is mandatory, the balance between SSD sustainability and performance presents a significant design challenge.

In this paper, we propose a sustainability-aware flash translation layer (FTL) for SSDs that optimizes lifetime and performance. Our design divides the flash memory into two regions: one with a lower threshold voltage for write-intensive data to extend lifespan, and another with a higher threshold voltage for read-intensive data to ensure optimal read performance. We introduce an intelligent data allocation strategy that dynamically assigns data based on access patterns, minimizing wear. Additionally, we propose an adaptive caching policy to enhance both the lifetime and performance. By combining these techniques, our FTL effectively balances sustainability and performance trade-offs in SSDs.

The rest of the paper is organized as follows. Section II describes the backgrounds of flash memory. The discussion of motivations is introduced in Section III. Section IV discusses the structure and algorithm of the proposed scheme. Section V shows the experimental results of EFM compared to the existing schemes. Finally, some conclusions are presented in Section VI.

II. Background

A. Background of Flash Memory

NAND flash memory encodes N bits within a cell by channeling electrons into the cell. The representation of N-bit data in NAND flash memory is achieved through 2^N distinct voltage levels. These levels exhibit a broad, Gaussian-like

2159-3477/24/$31.00 ©2024 IEEE
DOI 10.1109/ISVLSI61997.2024.00096

distribution as noted by [3] and can be closely approximated by the following Gaussian model:

$$P_e(x) = \frac{1}{\sigma_e\sqrt{2\pi}} e^{-\frac{(x-\mu_e)^2}{2\sigma_e^2}} \tag{1}$$

Here, μ_e and σ_e denote the mean and standard deviation of the threshold voltage for the erased state, respectively.

As flash memory technology evolves to store more bits per cell (e.g., from SLC to QLC), the number of distinct voltage levels increases, leading to a reduction in the noise margin between adjacent levels and a decrease in reliability [4]–[6].

B. Read and Write in Flash Memory

Writing data to a flash cell is accomplished using the Incremental Step Pulse Programming (ISPP) technique [4]. ISPP applies a series of voltage pulses to the cell, gradually increasing the voltage until the desired threshold voltage level is reached. The voltage step size used in ISPP, denoted as ΔV_{pp}, plays a crucial role in determining the write latency and raw bit error rate (RBER). The relationship between the program latency t_p, the ISPP step size ΔV_{pp}, and a fixed coefficient γ can be expressed as:

$$t_p \propto \gamma \times \frac{1}{\Delta V_{pp}} \tag{2}$$

A larger ISPP step size reduces the number of programming pulses required, thereby decreasing the write latency. However, this comes at the cost of a higher RBER due to the reduced noise margin between adjacent voltage levels [6].

Reading data from a flash cell involves comparing the cell's threshold voltage to a set of predefined reference voltages. The presence of errors, caused by factors such as wear, cell-to-cell interference, and retention loss, necessitates the use of error correction codes (ECC) like Bose-Chaudhuri-Hocquenghem (BCH) [7] and Low-Density Parity-Check (LDPC) codes [8]. The read latency is determined by the RBER and the number of ECC decoding iterations required [3], [5]. The relationship between the RBER and the voltage distribution of the k-th state can be modeled as:

$$RBER = \sum_k \left(\int_{-\infty}^{V_p^{(k)}} p^{(k)}(x)dx + \int_{V_p^{(k+1)}}^{+\infty} p^{(k)}(x)dx \right) \tag{3}$$

where $p^{(k)}(x)$ represents the probability density function of the threshold voltage distribution for the k-th state, and $V_p^{(k)}$ and $V_p^{(k+1)}$ are the lower and upper bounds of the voltage window for the k-th state, respectively. As the RBER increases, more ECC decoding iterations are needed, leading to a longer read latency.

C. Lifetime of Flash Memory

The characteristic of flash memory necessitates erasing a flash block before programming new data onto it. This entails a program operation for each page and an erase operation for the block, known as a P/E cycle. During programming, charges are inserted into the flash cells, and during erasing, these charges are removed by applying a high voltage, causing them to move through the oxide layer of the flash cell. This movement will damage the oxide layer, leading to the gradual wearing out of the flash memory as the number of P/E cycles increases. Therefore, the maximum number of P/E cycles refers to the lifetime of flash memory (i.e., SSD).

To improve the lifetime of SSDs, a strategy involves lowering the maximum threshold voltages of flash cells. By doing so, lower voltage will lessen the wear out on the cells. The relationship between threshold voltage and P/E cycle can be modeled as [9]:

$$PE \propto V_{threshold} \times r \tag{4}$$

where r represents a threshold voltage reduction coefficient ($0 \leq r \leq 1$). However, this reduction in threshold voltages leads to a decreased noise margin between the voltage states of adjacent flash cells, resulting in a higher Raw Bit Error Rate (RBER) of the stored data. This will increase the read latency and establish a trade-off between minimizing flash wearing-out and maintaining read performance.

III. Motivation

A. Sustainability Improvement of SSD

As of 2021, computing and networking devices had accounted for approximately 2% of total carbon emissions [10]. During the lifecycles of these devices, manufacturing and operations are the primary contributors to their total carbon emissions. While operations largely drive the emissions of server systems, for devices like workstations, desktops, laptops, and mobiles, manufacturing is the predominant source of carbon emissions [11]. In particular, as an integral part of all computing systems, the SSD manufacturing process contributes to a significant fraction of the total carbon emissions [11]. For instance, the SSD in a Fujitsu workstation, with a capacity of 512 GB, is responsible for about 38% of the total manufacturing emissions, which is the highest among all components [11]. Hence, reducing the carbon footprint associated with SSD manufacturing can therefore significantly lower overall emissions from computing systems.

The carbon emissions of SSDs originate from three main phases: manufacturing, operations, and disposal. The relationship between them and the total carbon emissions of SSDs can be modeled as:

$$Tot_{carbon} = \sum_{i=1}^{T}(C_{man,i} + C_{op,i}) + \sum_{i=1}^{K} C_{disp,i} \tag{5}$$

where Tot_{carbon} denotes the total carbon emissions over X years, T corresponds to the total number of drives used in these X years, and K is the number of drives that are disposed of during this period. $C_{man,i}$ indicates the manufacturing emissions for the i-th drive, $C_{op,i}$ represents the operational emissions from the i-th drive within the X years, and $C_{disp,i}$

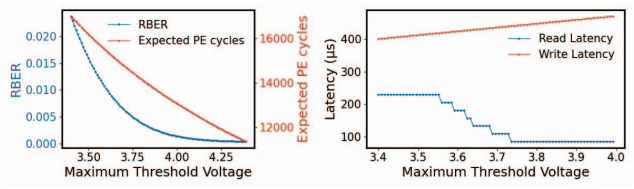

Fig. 1. Impact of Maximum Threshold Voltage on RBER, Lifetime, and Read/Write Latencies

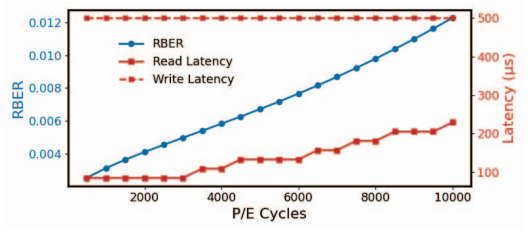

Fig. 2. Impact of Maximum Threshold Voltage on RBER and Latencies

captures the disposal emissions from the i-th drive during this period.

$$K \propto X/lifetime_{SSD} \tag{6}$$

$$T \propto X/lifetime_{SSD} \tag{7}$$

Assuming the total data migration processed over X years remains constant, equations (6) and (7) imply that extending SSD lifespans reduces both the total amount of drives to be disposed during these X years (i.e., K) and the total number of drives involved during these X years (i.e., T). Consequently, with manufacturing and disposal emissions per SSD constant, the total emissions from these stages decrease as the lifetime of SSDs gets lengthened. As for the operational emissions of SSDs during this period of time, given the consistent workloads of data migration and energy consumption within the X years, its amount remain unchanged. Therefore, lengthening the lifespan of SSDs can effectively decrease the total carbon emissions associated with these devices.

B. Balancing Lifespan Extension and Performance

Based on the background discussion, there is a trade-off between the performance and lifetime of SSDs. Reducing the threshold voltage can prolong the P/E cycle and extend the SSD lifespan, but it comes at the cost of increased Raw Bit Error Rate (RBER), necessitating more LDPC iterations and consequently longer read latency.

Figure 1 illustrates this trade-off, showing that decreasing the threshold voltage improves the P/E cycle (i.e., lifetime) of the SSD but degrades the read performance by increasing the RBER and read latency, based on the SSD characteristic model [3]. The write latency, however, slightly decreases with lower threshold voltages.

As SSDs age and undergo more P/E cycles, their read performance continuously degrades. Figure 2 shows that the RBER increases with the number of P/E cycles, leading to a corresponding increase in read latency, while the write latency remains relatively constant. This trend indicates that the negative impact of reduced threshold voltage on read latency becomes more pronounced as the SSD ages, making it increasingly challenging to meet the read performance requirements specified in Service-Level Agreements (SLAs).

The combination of these factors presents a complex challenge in SSD design. Lowering the maximum threshold voltage offers the benefit of an extended lifespan, which is crucial for sustainability, but it also introduces performance overheads, particularly in terms of read latency. On the other hand, increasing the threshold voltage improves read performance but at the cost of reduced lifespan and slightly higher write latency.

To address this challenge and achieve a balance between performance and sustainability, it motivates us to design a novel flash translation layer (FTL). The goal is to leverage the advantages of reduced threshold voltage for lifetime improvement while mitigating its negative impact on read performance. By carefully managing this trade-off, we aim to enhance the overall sustainability of SSDs without compromising on their performance characteristics.

IV. Design

A. Overall Architecture

The proposed access pattern-aware data allocation architecture for SSDs consists of several key components. The overall architecture is shown in Figure 3. The SSD storage space is partitioned into two regions with different maximum threshold voltages. The Access Pattern Sensor monitors the read and write access patterns of each data page, providing statistical information to the Data Allocator for dynamic data placement. The DRAM cache prioritizes storing the most frequently accessed data from Region 2 that exhibits both high read and write intensity. This helps to reduce the impact of the increased read latency in Region 2 due to its lower threshold voltage, while leveraging the improved write performance and extended lifespan. The Wearout Sensor monitors the wear-leveling balance between the regions and triggers data migration when necessary. These components work collaboratively to optimize data allocation, improve performance, and enhance the overall sustainability of the SSD.

B. Page Management and FTL Design

1) Page Metadata: In the Flash Translation Layer (FTL), each page entry contains several essential components. Beyond standard elements such as the Logical Address of the page and the Physical Address where the page is stored, we also include:

- **Write Counter**: W_count records the historical number of writes, with an initial value of 0.
- **Read Counter**: R_count records the historical number of reads, with an initial value of 0.
- **Last Access Time**: The timestamp of the last read or write operation.

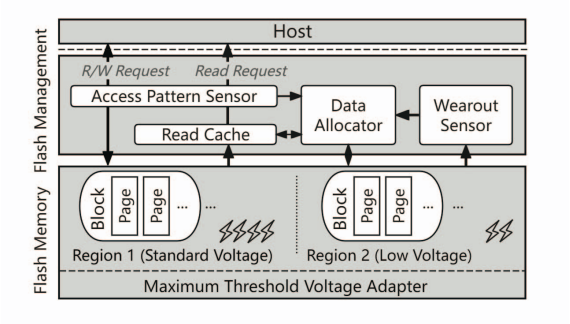

Fig. 3. Overall Architecture

TABLE I
SSD REGIONS CHARACTERISTICS ANALYSIS

Characteristic	**SSD Region**	
	Region 1	***Region 2***
Threshold Voltage	Standard	Low
P/E Cycles Endurance	Low	High
RBER	Low	High
Read Latency	Low	High
Write Latency	High	Low
Optimization	Read	Lifetime/Write
Data Type	Read-Focused	Write-Focused

Note: P/E Cycles Endurance reflects the memory cell endurance level.

2) Decay Function Design: A decay function $D(t)$ is introduced to address the temporal locality of data access patterns and to prevent the counters from increasing indefinitely. The decay function gradually reduces the value of the counters over time, representing the "forgetting" effect of the access history. This is important because recent access patterns are more likely to be indicative of future access patterns than older ones. By applying the decay function, we ensure that the access counters give more weight to recent access patterns, allowing the system to adapt more quickly to changes in data access behavior. Additionally, the decay function prevents the counters from growing too large, which could lead to overflow or reduced sensitivity to new access patterns. The interval t represents the time elapsed since the last access to the current time.A simple form of the decay function could be exponential: $D(t) = e^{(-\lambda t)}$, where λ is the decay rate, determining the speed of forgetting.

3) Counter Update Rule: Each time a page is accessed, the counters R_count and W_count are updated first based on the time interval t and the decay function $D(t)$:

- If it is a read access: $R\_count = R\_count \times D(t) + 1$, $W\_count = W\_count \times D(t)$.
- If it is a write access: $W\_count = W\_count \times D(t) + 1$, $R\_count = R\_count \times D(t)$.

Then, $Last\_access\_time$ is updated to the current time.

C. Data Allocation Algorithm

1) Initialization: The SSD is divided into two physical regions: Region 1 and Region 2. Region 2 is larger than Region 1, as it is designed to accommodate the majority of the data initially. The specific characteristics of these regions, such as their threshold voltage levels, P/E cycle endurance, read and write latencies, and optimization goals, are shown in Table I.

First, when the specific access patterns of the workload are unknown, all data is initially written to Region 2 due to its superior write performance and durability compared to Region 1, thereby making it suitable for initial data writes. As the system continues to operate and the access patterns of the data become apparent, the data allocation algorithm dynamically migrates the data between the regions based on the observed access patterns, ensuring optimal placement for improved performance and endurance.

2) Access Pattern Classification: Based on the dynamic access counters and decay function, we classify the pages into four categories according to their access patterns:

- Pages with $R\_count < Read\_Threshold$ and $W\_count < Write\_Threshold$ are historically infrequently accessed and should remain in **Region 2**.
- Pages with $R\_count < Read\_Threshold$ and $W\_count > Write\_Threshold$ are infrequently read but frequently written and should keep in **Region 2**.
- Pages with $R\_count > Read\_Threshold$ and $W\_count > Write\_Threshold$ are frequently accessed and should be cached in SSD **DRAM** to reduce latency and avoid frequent accesses to **Region 2** due to its higher read cost/latency.
- Pages with $R\_count > Read\_Threshold$ and $W\_count < Write\_Threshold$ are frequently read but infrequently written, and should be migrated to **Region 1** for lower read latency. If DRAM is not full, we will cache pages with the highest R_count into it.

The $Read\_Threshold$ and $Write\_Threshold$ are established based on the characteristics of Region 1, Region 2 and DRAM, and can be dynamically adjusted in response to the system's workload and observed performance metrics.

D. DRAM Cache Strategy

1) DRAM Cache Entry Strategy: When DRAM cache is not full: Directly cache the pages that meet the condition into DRAM. When DRAM cache is full: Compare the R_count values of all pages in the DRAM. Prioritize evicting the page with the lowest R_count value, those that are least frequently read, to make space for new candidate pages.

2) DRAM Cache Exit Strategy: Passive Eviction: When a new page needs to be cached and the DRAM is full, pages with lower R_count need to be evicted. The evicted pages will be migrated based on the current wearing level of Region 1 and Region 2: If the wear level of Region 1 is lower, consider migrating the page to Region 1 to achieve some level of wear leveling. If Region 1 is already more worn, keep the evicted page in Region 2 to avoid exacerbating the wear in Region 1.

Active Eviction (regular checks): Regularly inspect the pages in the DRAM cache to assess their R_count values. For pages with an R_count below $Read\_Threshold$, evict them from the DRAM cache, as their frequency of being

TABLE II
CONFIGURATIONS OF TRACES

Trace	Number of IOs (Millions)		Total request size (GB)	
	Write	Read	Write	Read
PROJ_1	2.50	21.14	25.58	750.36
USR_1	3.86	41.43	56.13	2079.23
WEB_2	0.04	5.14	0.78	262.82
HM_0	2.58	1.42	20.48	9.96
PROJ_0	3.70	0.53	144.27	8.97
PRXY_0	12.14	0.38	53.80	3.05

read has decreased and they no longer require high-speed caching. Since this data originally belongs to Region 2, no extra migration operation is needed upon eviction; the data still remains in Region 2.

E. SSD Wear Leveling Strategy

1) Monitoring Wear: We continue to monitor the wear condition of both Region 1 and Region 2 through the counters Region 1_Wear and Region 2_Wear. For example, if Region 1 has 10000 PE cycles while Region 2 has 17000. We maintain a wear ratio threshold range between 1:1.5 and 1:2.

2) Real-time Calculation of Wear Ratio: The Current_Wear_Ratio = Region 1_Wear / Region 2_Wear is calculated periodically to monitor whether the wear distribution is within the acceptable range.

3) Data Migration Strategy: When the Current_Wear_Ratio falls below 1:1.5, it indicates that Region 2 is wearing out more rapidly. At this point, we may migrate some data from Region 2 to Region 1 to balance the wear.

V. EVALUATION

A. Experimental Setup

Our assessment utilized various algorithms through the SSDSim simulator [12], augmented to feature diverse read and write latencies across two distinct physical areas. These latency figures were derived from the device architecture outlined in [3]. Within this simulation, the flash memory was configured with a capacity of 256GB and a page size of 4KB, with each block holding 128 pages. For our experimental data, we collected six MSR Cambridge traces [13] as detailed in Table II [14]. To evaluate our proposed scheme, we compared it against three established frameworks: SSDSim without cache [12], SSDSim with read cache, and EFM [15], serving as our reference models.

B. Sustainability Improvement

To evaluate the sustainability improvement of our design, we analyze the normalized SSD sustainability under different workloads, which is proportional to the SSD lifetime as discussed in Section III. As shown in Figure 4, our design significantly enhances the SSD sustainability compared to the baseline SSDSim with Cache and SSDSim without Cache by an average of 20.3% across all traces. This improvement is attributed to the lower threshold voltage used in Region 2, which reduces the wear on flash cells during write operations. Compared to EFM, our approach improves sustainability by an

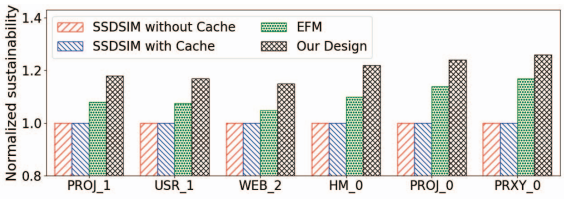

Fig. 4. Normalized sustainability improvement

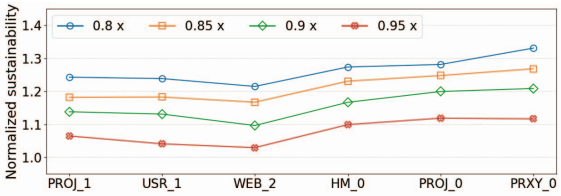

Fig. 5. Normalized sustainability improvement

average of 8.7%, due to our more precise access pattern classification and cache policy, minimizing unnecessary rewrites and associated wear by lowering the frequency of data reallocation.

To explore how the maximum threshold voltage affects SSD sustainability, we compare the normalized sustainability of our design using different scaling factors for the standard voltage: 0.95, 0.9, 0.85, and 0.8. The results in Figure 5 show that reducing the maximum threshold voltage enhances sustainability improvements for all traces. However, excessively lowering the maximum threshold voltage may exceed the error correction limits of the LDPC codes used in SSDs, potentially reducing reliability. Stronger LDPC codes with higher error correction capabilities are typically employed to manage the increased error rates. In this study, we assume that the employed LDPC codes can correct errors from lowered voltages. Future work will investigate different LDPC codes to address errors caused by significantly reduced threshold voltages.

C. Performance Improvement

To evaluate the performance of our proposed design, we focus on the average latency metric across different traces and compare it with three other designs. The results are presented in Figure 6. Our design demonstrates significant improvements in average latency compared to all three designs, with average reductions of 69.8%, 29.6%, and 19.7% compared to SSDSim without Cache, SSDSim with Cache, and EFM, respectively. These improvements can be attributed to our efficient cache management, optimized data allocation strategy, and the use of a lower threshold voltage for write operations. It is noteworthy that while our design outperforms EFM on average, the performance advantage is more pronounced for read-intensive traces due to our efficient cache management policy. For write-intensive traces, the performance difference between our design and EFM is less significant, as both designs benefit from the reduced write latency achieved by lowering the maximum threshold voltage.

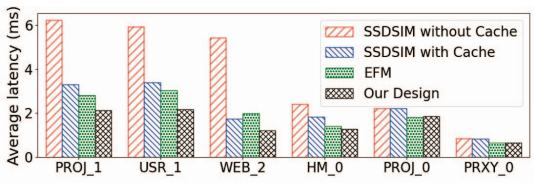

Fig. 6. Overall performance comparison

TABLE III
LATENCIES OF READ AND WRITE WITH DIFFERENT REDUCED EFFECTIVE WEARING FOR DIFFERENT PE CYCLES

PE cycle		Stage-1	Stage-2	Stage-3	Stage-4
Region-1	Read (us)	90	120	160	200
	Write (us)	500			
Region-2	Read (us)	130	180	250	350
	Write (us)	400			

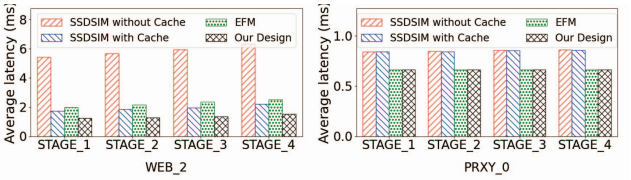

Fig. 7. Performance comparison in the context of flash memory degradation

D. Performance Analysis under Flash Memory Degradation

As discussed previously, flash memory wear leads to degraded read performance due to increased Raw Bit Error Rates (RBERs) and longer decoding times. This subsection examines the average latency across two traces at various SSD lifespan stages. As shown in Table III, the SSD lifespan is divided into four stages based on the number of PE cycles, with each stage corresponding to a different set of read and write latencies for Region 1 and Region 2.

As shown in Figure 7, our design consistently achieves the lowest average latency across all stages for the read-intensive trace WEB_2, with the performance gap becoming more pronounced as the SSD ages. This can be attributed to our efficient cache management and data allocation strategy, which effectively mitigates the impact of increasing read latencies in Region 2. For the write-intensive trace PRXY_0, our design maintains a stable average latency across all stages, comparable to EFM, as both designs benefit from the reduced write latency achieved by lowering the threshold voltage. In contrast, SSDSim with Cache and SSDSim without Cache exhibit higher and increasing average latencies due to their higher threshold voltage, resulting in higher write latency.

VI. CONCLUSION

In this paper, we propose a novel SSD architecture that enhances sustainability by extending the lifespan of SSDs while maintaining high performance. By leveraging the trade-off between threshold voltage, lifespan, and read/write latencies, our design divides the SSD into two regions with different threshold voltages and utilizes a DRAM cache and an intelligent data allocation algorithm. This approach improves the SSD lifespan by 8.7% to 20.3% compared to current designs, while achieving latency reductions ranging from 19.7% to 69.8%. Our work demonstrates the potential for enhancing the sustainability of SSDs while ensuring high performance, contributing to the reduction of the carbon footprint associated with storage systems in the era of big data.

VII. ACKNOWLEDGEMENT

This work was partially supported by NSF 2204656, 2343863, and 2413520. Any opinions, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] N. R. Mielke, R. E. Frickey, I. Kalastirsky, M. Quan, D. Ustinov, and V. J. Vasudevan, "Reliability of solid-state drives based on nand flash memory," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1725–1750, 2017.

[2] B. Li, C. Deng, J. Yang, D. Lilja, B. Yuan, and D. Du, "Haml-ssd: A hardware accelerated hotness-aware machine learning based ssd management," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.

[3] Y. Pan, G. Dong, and T. Zhang, "Exploiting memory device wear-out dynamics to improve nand flash memory system performance," in *Proceedings of the 9th USENIX Conference on File and Stroage Technologies (FAST 11)*. USENIX Association, 2011, pp. 1–14.

[4] K.-D. Suh, B.-H. Suh, Y.-H. Lim, J.-K. Kim, Y.-J. Choi, Y.-N. Koh, S.-S. Lee, S.-C. Kwon, B.-S. Choi, J.-S. Yum *et al.*, "A 3.3 v 32 mb nand flash memory with incremental step pulse programming scheme," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, 1995.

[5] W. Liu, F. Wu, M. Zhang, Y. Wang, Z. Lu, X. Lu, and C. Xie, "Characterizing the reliability and threshold voltage shifting of 3d charge trap nand flash," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, pp. 312–315.

[6] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, "Errors in flash-memory-based solid-state drives: Analysis, mitigation, and recovery," *arXiv preprint arXiv:1711.11427*, 2017.

[7] G. Dong, N. Xie, and T. Zhang, "Enabling nand flash memory use soft-decision error correction codes at minimal read latency overhead," *IEEE Transactions on Circuits and Systems I: regular papers*, vol. 60, no. 9, pp. 2412–2421, 2013.

[8] Y. Du, D. Zou, Q. Li, L. Shi, H. Jin, and C. J. Xue, "Laldpc: Latency-aware ldpc for read performance improvement of solid state drives," in *2017 33rd International Conference on Massive Storage Systems and Technology (MSST)*, 2017, pp. 1–13.

[9] J. Jeong, S. S. Hahn, S. Lee, and J. Kim, "Lifetime improvement of nand flash-based storage systems using dynamic program and erase scaling," in *Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST 14)*, 2014, pp. 61–74.

[10] K. W. B. K. G. S. B. C. Freitag, M. Berners-Lee and A. Friday, "The real climate and transformative impact of ict: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, p. 100340, 2021.

[11] S. Tannu and P. Nair, "The dirty secret of ssds: Embodied carbon," *ACM SIGENERGY Energy Informatics Review*, vol. 3, no. 3, pp. 4–9, 2023.

[12] Y. Hu, H. Jiang, D. Feng, L. Tian, H. Luo, and C. Ren, "Exploring and exploiting the multilevel parallelism inside ssds for improved performance and endurance," *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1141–1155, 2013.

[13] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," *ACM Transactions on Storage (TOS)*, vol. 4, no. 3, p. 10, 2008.

[14] "Snia," http://iotta.snia.org/traces/block-io/388.

[15] B. Li, B. Yuan, and D. Du, "Efm: Elastic flash management to enhance performance of hybrid flash memory," in *2021 IEEE 39th International Conference on Computer Design (ICCD)*. IEEE, 2021, pp. 162–169.