Access Characteristic Guided Read and Write Regulation on Flash based Storage Systems

Qiao Li, Liang Shi, Congming Gao, Yejia Di, Chun Jason Xue

Abstract—NAND flash memory is now used in various storage systems, such as embedded systems, personal computers, and web servers. The developments in bit density and technology scaling have reduced its price, but worsen the reliability, leading to shortened lifetime and degraded access performance. This paper proposes to exploit access characteristics of workloads to improve flash performance and lifetime. The basic idea is to regulate the read and write operations based on the identified access characteristics. First, an access cost model is presented, which indicates a tradeoff between read and write time cost on NAND flash memory. Based on the access characteristics of workloads, read-only pages will be written with high cost so that they can be read with low cost, and write-only pages will be written with low cost. Second, the tradeoff between read cost and flash wearing is exploited for lifetime improvement. The write requests on write-only data are processed with reduced wearing by regulating the programm threshold voltage. Finally, as these approaches apply different write operations on write-only data for performance and lifetime improvement respectively, a combined approach is proposed to satisfy both goals. Simulation results show that the proposed approaches can improve performance and lifetime significantly with negligible overhead.

Index Terms—Access characteristics, LDPC, Access cost regulation, Threshold voltage regulation, NAND flash memory.

1 Introduction

NAND flash-based storage has been widely deployed as it outperforms most magnetic-based storage in several aspects, such as light weight, high performance, and small form factors [2] [3]. However, with the development of bit density and technology scaling, less charges can be stored in flash cells to represent data, which degrades the reliability [4] [5] [6]. There are two consequences for this trend. First, the lifetime of flash memory will be shortened as the supported program/erase (P/E) cycles will be reduced. Second, to provide stronger error correction capability from error correction codes (ECC), low-density parity-check codes (LDPC), are applied on NAND flash memory. LDPC needs long access latency to decode data with high raw bit error rates (RBER), which deteriorates flash access performance [7]. In this paper, we propose techniques to improve access performance and lifetime of NAND flash memory based storage systems.

There is a tradeoff between read cost and write cost. The incremental-step pulse programming (ISPP) [8] [9] scheme is applied for write operations in flash memory. ISPP is designed to iteratively increase the program voltage and use a small verifying voltage to reliably program flash cells to their specified voltages. In each iteration, the program voltage is increased by a predefined program step size. Increasing the program step size reduces the number of iterations and thus reduces write cost, but increases the RBER of the programmed data pages. On the other hand, read cost highly depends on the RBER of data pages and the deployed error correcting code (ECC). For a specific ECC, the higher the RBER is, the higher the read cost will be [7] [10] [11]. Many previous works have proposed to regulate the read and write access costs to exploit the tradeoff. Based on the characteristic that a high-cost write reduces the cost of the following reads to the same page [7] [10], several optimization approaches have been proposed. For example, Li et al. [12] proposed to apply low-cost writes when there are several queued requests, and high-cost writes otherwise. Wu et al. [13] proposed to apply high-cost writes for future low-cost reads on the data when, according to their estimation, the next access operation will not be delayed due to the increased cost. However, none of these works exploits the access characteristics of workloads, which show great potential for performance improvement through cost regulation.

There is a tradeoff between flash wearing and access cost. Flash wearing is proportional to the threshold voltages which define the voltage states of flash cells [14] [15] [16] [17]. Note that the threshold voltages for the above tradeoff between read and write costs are constant and thus the flash wearing is constant. The lower the threshold voltages are, the less the wearing incurred by each P/E cycling is, as well as the less the noise margins between the states of a flash cell are. The reduced noise margins in turn result in high RBER. Previous work [16] has proposed to slow down the program speeds to compensate the increased RBER from the reduced threshold voltages. Shi et al. [15] proposed to reduce the threshold voltages by exploiting the specific retention time requirement of the data to be programmed. In this work, an access characteristic guided lifetime improvement approach

- *Q. Li and C. J. Xue are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: qiaoli045@gmail.com, jasonxue@cityu.edu.hk.*
- *L. Shi, C. Gao, and Y. Di are with the College of Computer Science, Chongqing University, Chongqing, China. E-mail: {shi.liang.hk, albertgaocm, yejia.di.cqu}@gmail.com. Corresponding author: Liang Shi*

An earlier version of this work has been accepted in the Proceedings of the 14th File and Storage Technologies (FAST16) [1]. The conference paper covers only the access performance improvement, whereas the new manuscript presents the lifetime improvement approach.

will be presented.

This work is the first in exploiting the access characteristics of workloads for performance and lifetime improvement. First, three page access characteristics are defined in this work. If almost all the accesses to a data page are read (write) requests, the page is characterized as *read-only* (*write-only*). If the accesses to a data page are interleaved with reads and writes, the page is characterized as *interleaved-access*. The proposed approach is based on the observation that *most read requests from the host are performed on read-only data pages and most write requests are performed on write-only data pages.* The main idea is to guide regulation of read and write operations by utilizing the observed access characteristics as follows. First, to improve access performance, the read and write costs are regulated. For the read requests accessing read-only data, low-cost reads are preferred. For the write requests accessing write-only data, low-cost writes are preferred. Second, to improve flash lifetime, write requests are performed differently to regulate the threshold voltage. For the write requests accessing write-only data, writes with reduced threshold voltage, thus reduced wearing, are preferred. Third, to combine the performance and lifetime improvement, we propose to switch the write mode of write requests on write-only data based on the system status. The proposed approach chooses LPDC as the default ECC, which is the best candidate for current flash memory storage systems [10] [7]. The main contributions of this work are as follows:

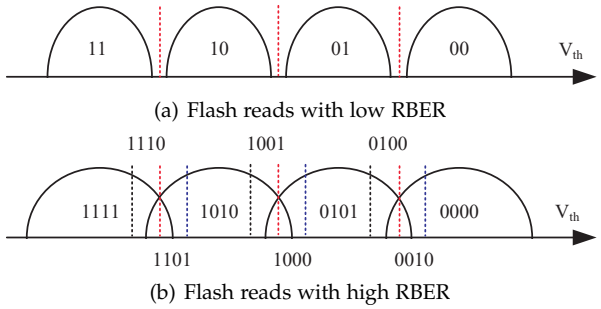
- We present the access characteristics of flash memory, which are the basis of the following two approaches;
- We propose an approach to improve access performance by regulating the cost of writes and reads;
- We propose an approach to improve flash lifetime by regulating the write voltage;
- We present an efficient implementation of the proposed approaches with negligible overhead.

The rest of the paper is organized as follows. Section 2 presents the background and related work. Section 3 presents the motivation of this work. Section 4 presents the proposed approaches. Experiments and analysis are presented in Section 5. Section 6 concludes this work.

2 Background and Related Work

2.1 Basics of Flash Memory

NAND flash memory stores data by trapping charges in the floating gate of memory cells. For the flash memory with n bits per cell, the data are represented by 2^n different voltage levels. Figure 1(a) illustrates an example for flash memory with 2 bits per cell, which is realized with 4 voltage states. To write data to flash memory, program operation is performed to inject certain amount of charges into the floating gate. To read data from flash memory, read operation is performed to sense the voltage state of flash cells. The distribution of each voltage state and noise margins between neighboring voltage states directly impact the reliability of data stored in flash memory. The increasing demands on large storage capacity have promoted the flash memory to develop in bit densities from 1 bit per cell to the latest 6 bits per cell [11] and technology scales from 65nm to the latest 10nm technology [18]. This is achieved at the sacrifice of reduced charges stored in flash cells, which narrows the noise margins. As a result, the reliability is degraded, which causes both lifetime and performance degradation. First, the degraded reliability shortens lifetime of flash memory. Second, to deal with the reliability issue, error correction codes (ECC) with strong error correction capability, such as low-density parity-check codes (LDPC), are applied to NAND flash memory [7] [19]. However, LDPC needs long access latency to decode data, especially for data with high RBER, which further deteriorates flash performance. In the following two subsections, detailed descriptions on flash performance and lifetime are presented.



(a) Flash reads with low RBER

(b) Flash reads with high RBER

Fig. 1. Illustration of flash reads with low and high RBER.

2.2 Read and Write in Flash Memory

Flash write operation is the process to inject charges to floating gates to increase the threshold voltage of each cell to one of 2^n non-overlapping voltage levels that are apart from each other with certain noise margin. Incremental-step pulse programming scheme (ISPP) is used for reliably programming a flash page, using an iterative program-verify algorithm. The iterative algorithm has two stages in each iteration step: it first programs a flash cell with an incremental program voltage, and then verifies the voltage of the cell. If the voltage is lower than the predefined threshold, the process continues. In each iteration, the program voltage increases by a step size, ΔV_{pp}. The step size ΔV_{pp} determines the write cost. To reach the same threshold voltage level, a larger step size indicates a smaller number of iterations, hence a lower write cost. However, programming with a larger step size will narrow the noise margin between two adjacent voltage states and result in higher raw bit error rates (RBER) in the programmed pages [20] [9] [21].

Flash read operation is the process to correctly recognize the voltage levels of each cell in a flash page. Errors occur when the threshold voltage of a cell shifts to another voltage level. LDPC decoding scheme is based on the probability information of the read data. The error correction capability of LDPC depends on the accuracy of the input information. For the data with a high RBER, the adjacent voltage states have small noise margin or have been overlapped, as shown in Figure 1(b). In this case, LDPC requires fine-grained memory-cell sensing to gain the probability information in each region, which is realized by comparing a series of N reference voltages. The number of reference voltage N determines the read cost. A larger N required by the data

with a higher RBER indicates a longer on-chip memory sensing latency and longer memory-to-controller data transfer latency, hence a higher read cost.

Based on above descriptions, there is a strong relationship between read cost and write cost on flash memory. Programming with a large step size during the ISPP process will reduce the write cost but increase the RBER of the data, and further increase the read cost, and vice versa. This relationship is valid under the constraint of the same maximum retention time. It is because that the RBER of flash data will increase with the growing of retention time [20] [21]. In this work, we focus on read and write cost interaction by tuning the program step size, where the tuning is constrained by the maximum required retention time [22].

2.3 Flash Lifetime

Due to the out-of-place update characteristic, a flash block has to be erased before new data can be programmed. This process including one program operation on each page and one erase operation on the block is called a P/E cycle. During program operations, charges are injected into the floating gate of flash cells, by applying a high program voltage. During erase operations, charges are evicted from the floating gate by applying a high erase voltage. The charges travel through the oxide layer of flash cells during a P/E cycle, which will damage the oxide layer. The flash wearing is accumulated with the increasing of P/E cycles, which weakens the ability to store data. Therefore, the lifetime of flash memory can be represented by the maximal number of P/E cycles to the flash memory.

The damages caused by program and erase voltages are highly related with charges trapped in the tunneling oxide layer, which prevent the flash cell from being discharged to the erase state. To reduce the damages from the trapped charges, a common approach is to reduce the maximum threshold voltages of flash cells [14] [16]. With less charges, the program and erase voltages are reduced, causing reduced wearing to flash cells. However, reduced threshold voltages will result in narrow noise margin between adjacent voltage states of flash cells. This situation introduces increased RBER of stored data, which requires read operations performed with increased read latency. As a result, there exists a tradeoff between the flash wearing and read performance.

2.4 Related Work

This subsection presents the related works from two aspects: access performance and lifetime.

2.4.1 Access Performance Improvement

The performance of ECC have been widely studied. As the promising ECC in recent years, LDPC has drawn a lot of attention. There are many prior works focusing on optimizing LDPC decoding strategy. Zhang et al. [23] proposed to optimize the decoding strategy for flash memory, and thus improve flash performance. Dong et al. [10] and Li et al. [24] proposed to optimize the soft-decision memory sensing by exploiting the distribution characteristic of threshold voltage on multiple-level cell (MLC) flash memory. Zhao et al. [7] further proposed an approach to progressively increase the memory sensing precision level-by-level and retry LDPC decoding accordingly. All these works have improved the efficiency and performance of LDPC decoding, which promotes the adoption in flash memory.

Several strategies have been recently proposed to regulate read and write costs by exploiting the reliability characteristics to improve flash memory performance. The RBER of flash data are affected by several factors, mainly including P/E cycling, retention time, and the program step size in ISPP. There are several works proposed based on the relationship between RBER and retention time. The longer the retention time is, the higher the RBER is. Pan et al. [20] and Liu et al. [21] proposed to reduce write costs by relaxing the retention time requirement of programmed pages. To periodically refresh flash data, the data suffers less errors during retention time. Therefore, the data can tolerate more errors from programming with a larger step size in ISPP. Other strategies are proposed based on the tradeoff between read and write costs. Wu et al. [13] proposed to apply a high-cost write to reduce the cost of read requests performed on the same page. The high-cost write is performed on the premise that upcoming requests will not be delayed by the increased cost. The method can effectively improve read performance, but impact write performance meanwhile. Li et al. [12] proposed to apply low-cost writes when there are queued requests to reduce the queueing delay, and apply high-cost writes otherwise, allowing low-cost reads of the page. It is motivated by the observation that the conflict latency is the dominate part in access latency and has a significant influence on access latency.

However, these works are mostly based on the physical characteristics of flash memory. None of them exploited the access characteristics of a specific data page for cost regulation. Several types of workload access characteristics have been studied in previous works, including hot and cold accesses [25], inter-reference gaps [26], and temporal and spacial locality [27] [28] [29]. These access characteristics have been employed to guide the design of buffer caches [26] [27] [28], and flash translation layers (FTL) [25] [29]. However, these characteristics focus on access frequency, which cannot be exploited for read and write cost regulation because the cost regulation hints the cost relationship for a specific data page. This work proposes to regulate read and write costs based on the access characteristics of workloads.

2.4.2 Lifetime Improvement

As the endurance of recent high-density NAND flash memory is continuously decreasing, many works proposed approaches to improve the lifetime of flash-based storage systems. Other than wear leveling techniques [30] [31], several system-level techniques which exploited the physical characteristics of NAND flash memory have been proposed. Since flash blocks are worn out if the data have more errors than ECC's correction capability, many works focused on reducing different types of errors. Pan et al. [20] proposed to reduce the errors from retention time by refreshing data at the late age of flash memory. Cai et al. [32] proposed to recharge flash cells after charge leakage with the increasing of retention time. Lee et al. [33] proposed to improve lifetime by intentionally throttling write performance.

There are several works focused on the tradeoff of flash wearing and RBER of data through regulating the threshold voltage [14] [16] [17] [15]. Flash cells with reduced threshold voltage endure less wearing, but suffer more errors from the reduced noise margin between two adjacent voltage states. To guarantee the same reliability, Jeong et al. [14] [16] proposed to sacrifice write performance, as fine-grained programming operations cause reduced errors. Based on the observation that the updating time intervals of most data are much shorter than the promised retention time, Shi et al. [15] proposed to relax the retention time requirements for some data, which avoided the degradation of write performance in lifetime improvement. Peleato et al. [17] proposed to optimize the target level placement by achieving the tradeoff between flash lifetime and error rates. Different from these works, this paper exploits the access characteristics of flash memory to achieve tradeoff between read cost and flash wearing.

2.4.3 Performance and Lifetime Improvement

This subsection presents the related work on both performance and lifetime improvement [34] [35] [36]. When NAND flash memory is used as second level cache, traditional cache mechanisms need modifications considering the characteristics of flash. Cache replacement policies often maximize the hit ratio to achieve a better performance, during which a great number of small random writes are introduced and this impacts the lifetime of flash memory. Several works have been proposed to design flash friendly cache management schemes. Tang et al. [34] proposed RIPQ framework which efficiently approximated a priority queue on flash for performance improvement, and aggregated small random writes by lazily moving updated contents to reduce erase operations. Cheng et al. [35] proposed to exploit the tradeoff between the hit ratio and endurance, to improve endurance while maintaining hit ratio. Li et al. [36] introduced a container-based flash cache to manage compound blocks. The focus is also the tradeoff between the performance and lifespan. All of these works are cache management policies and implemented in the host, which are orthogonal to this work.

3 Motivation

This section presents the motivation of this work by analyzing two models on flash memory: read and write cost model, and flash wearing model.

3.1 Read and Write Cost Model

This subsection presents the read and write time cost model, the specific relationship between read cost and write cost. The model has been validated by simulation in previous studies [7] [12] [13] [37]. It consists of a write cost model and a read cost model. First, the write cost is inversely proportional to the program step size of ISPP [8] [20] [21]. Thus, the write cost model is as follows:

$$WC(\Delta V_{pp}) = \gamma \times \frac{1}{\Delta V_{pp}}$$
$$where \quad RBER(\Delta V_{pp}) < CBER_{LDPC}(N); \qquad (1)$$

$WC(\Delta V_{pp})$ denotes the write cost when the program step size is ΔV_{pp}, and γ is a constant. A coarser step size results in a lower write cost, but a higher RBER. In addition, the reduction of the write cost is limited by the condition that the resulting RBER should be within the error correction capability of the deployed LDPC code, $CBER_{LDPC}(N)$, with N reference voltages.

The read cost is composed of the time to sense the page, which is proportional to the number of reference voltages, N, and the time to transfer the data from the page to the controller, which is proportional to the size of the transferred information [7] [10] [11]. Thus, the read cost model is as follows:

$$RC(N) = \alpha \times N + \beta \times \lceil \log(N+1) \rceil$$
$$where \quad RBER(\Delta V_{pp}) < CBER_{LDPC}(N); \qquad (2)$$

where $RC(N)$ denotes the read cost with N reference voltages in the deployed LDPC code, and α and β are two constants. With a larger N, longer on-chip memory sensing time and longer flash-to-controller data transfer time are required, which lead to increased read cost.

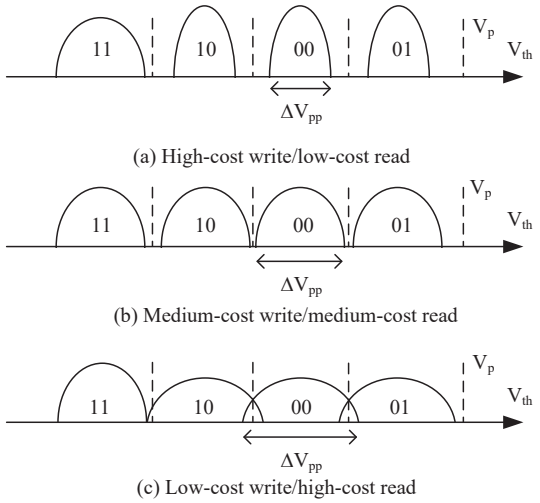
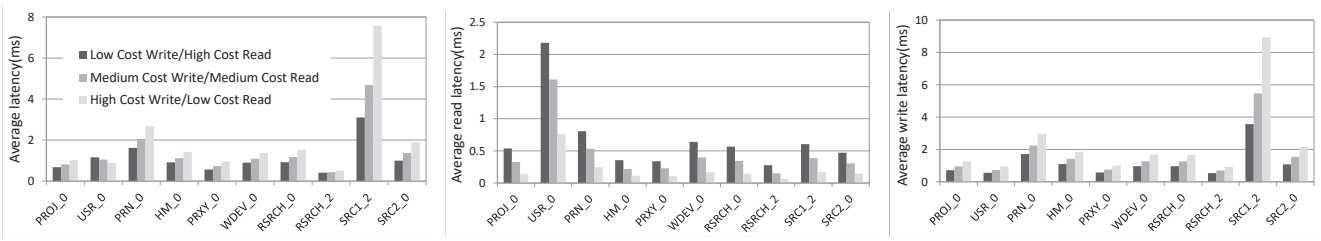


Fig. 2. The voltage state distribution with different read and write costs.

Figure 2 shows the different voltage state distributions for different read and write costs. Figure 2(b) is the normal case with medium step size ΔV_{pp}, where read and write operations are performed with medium cost, denoted as medium-cost write/medium-cost read (MCW/MCR). Figure 2(a) shows the voltage distribution with reduced ΔV_{pp} and therefore reduced RBER, where writes are performed with high cost and reads are performed with low cost, denoted as high-cost write/low-cost read (HCW/LCR). Figure 2(c) shows the voltage distribution with increased ΔV_{pp} and therefore increased RBER, where writes are performed with low cost and reads are performed with high cost, denoted as low-cost write/high-cost read (LCW/HCR). The maximum of threshold voltage stays the same for the three cases, and regulation of threshold voltage will be discussed in the following subsection. Based on this model, experiments are conducted to evaluate the performance difference between the three cases. The detailed settings are presented in Section 5.1.



(a) Overall average access latency (b) Average access latency for read requests (c) Average access latency for write requests

Fig. 3. I/O latency comparison for read and write requests with different access costs.

Figure 3 presents the comparison of access latency for 10 representative traces of the enterprise servers from Microsoft Research (MSR) Cambridge [38] and 4 traces we collected on flash memory systems, which will be introduced in Section 5.1. Compared to the default MCW/MCR, HCW/LCR improves read performance by 54%, and LCW/HCR improves write performance by 26% on average. The significant performance improvement comes from the reduction in access costs. Comparing LCW/HCR and HCW/LCR, the differences in their read and write latencies are 114% and 61%, respectively. The performance gap indicates that the read and write cost regulation should be applied carefully. While LCW/HCR is able to improve the overall performance, it introduces the worst read performance, as shown Figure 3(b). However, read operations are always in the critical path, which motivates this paper to regulate read and write costs carefully.

3.2 Access Cost and Flash Wearing Model

The wearing to NAND flash depends on the program and erase voltages applied. Based on previous work [16] [15], the wearing to a flash cell is proportional to its threshold voltage. Wearing for a flash page or block depends on the flash cell with the largest wearing. We use the maximum of threshold voltage (denoted as V_p) of flash cells to calculate the effective wearing w_e. The relationship is $w_e = \lambda \times V_p$, where λ is a constant, which depends on the physical characteristics of NAND flash cells. The effective wearing can be reduced when the maximum threshold voltage V_p is reduced. The reduction of V_p can be achieved by reducing the program step size ΔV_{pp}, which affects the write cost, or by reducing the noise margin of adjacent voltage states, which affects the read cost. In this work, we exploit the relationship between flash wearing and read cost, while maintaining the write cost as constant. Thus, the effective wearing to flash memory can be modeled as follows.

$$\begin{aligned} & w_e = \lambda \times V_p \\ where \quad & \frac{V_p}{\Delta V_{pp}} = C \\ & RBER(V_p, \Delta V_{pp}) < CBER_{LDPC}(N); \end{aligned} \quad (3)$$

The effective wearing is reduced as the reduction of V_p, on the condition that the write cost stays constant (C) and the resulting RBER ($RBER(V_p, \Delta V_{pp})$) is within the error correction capability of the deployed LDPC code, $CBER_{LDPC}(N)$, with N reference voltages. With the reduction of V_p, the noise margin between two adjacent voltage states will be reduced, leading to increased RBER of data. Therefore, the high-cost read needs to be performed to read the data.

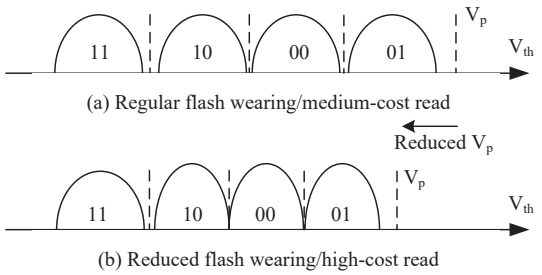


Fig. 4. The voltage state distribution for different read cost and flash wearing.

Figure 4 shows the comparison of voltage state distributions based on the wearing model. Figure 4(a) is the normal case with normal step size ΔV_{pp} and regular wearing, which is the default setting for flash memory, denoted as regular wearing/medium-cost read. Figure 4(b) shows the voltage distribution with reduced threshold voltage and reduced flash wearing but increased read cost. The step size ΔV_{pp} is reduced as well, but the write cost stays the same, denoted as reduced wearing/high-cost read. Based on the model, experiments are conducted to evaluate the read performance and effective wearing difference between the two cases. The detailed settings can be found in Section 5.1.

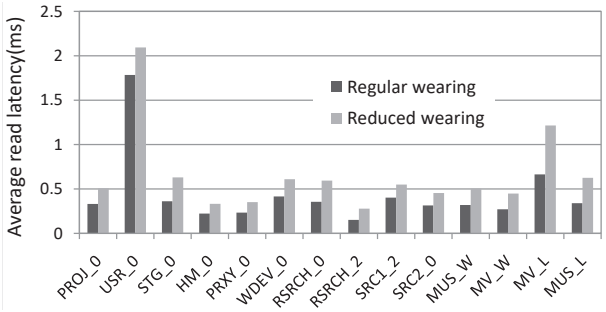
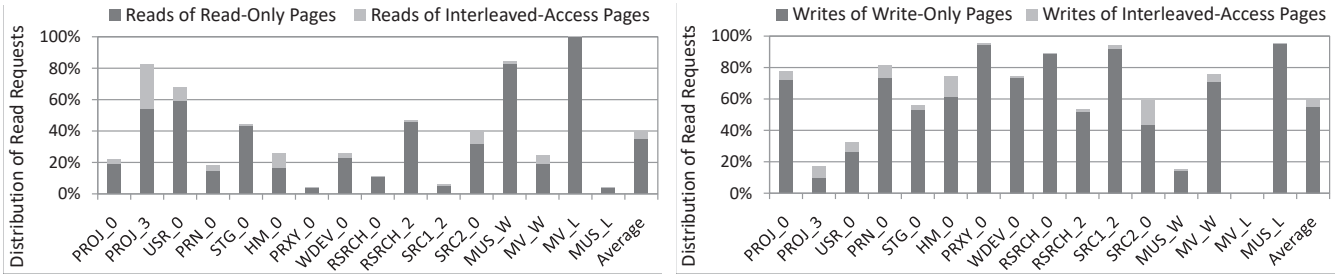


Fig. 5. The read access latency comparison for different flash wearing schemes.

Figure 5 presents the comparison of read latency. Compared to the regular wearing, the read latency of reduced wearing is increased by 52% on average, where the effective wearing is reduced to 80% of the regular wearing. The performance degradation of reduced wearing shows that the threshold voltage reduction approach for flash wearing reduction should be applied carefully. This work will exploit



(a) Distribution of read requests (b) Distribution of write requests

Fig. 6. Distribution of read and write requests on three access characteristics.

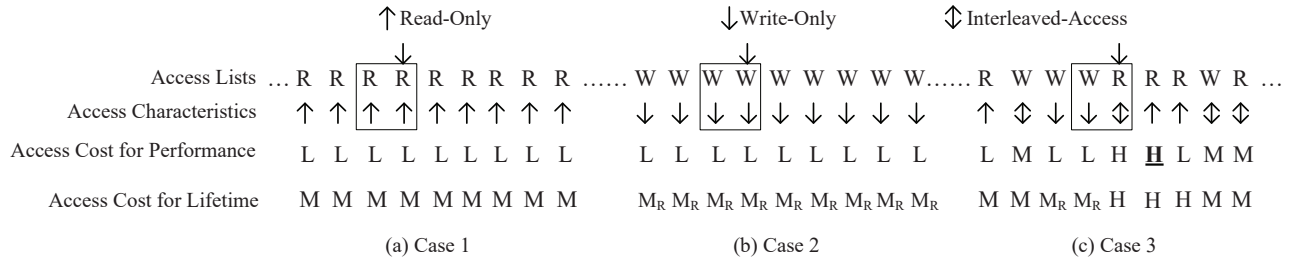


Fig. 7. Example on the access characteristic identification and cost regulation: access characteristic, ↑ – read-only, ↓ – write-only, and ↕ – interleaved-access; access cost, L – low-cost, M – medium-cost, H – high-cost, and M_R – medium-cost with reduced wearing.

the access characteristics on flash memory to regulate the threshold voltage for lifetime improvement.

4 Read and Write Regulation for Performance and Lifetime Improvement

As presented above, the cost model and wearing model indicate that the read and write operations can be regulated. However, simply regulating the read and write operations is not able to achieve performance and lifetime improvement. In this section, we propose to exploit the access characteristics of workloads for the operation regulation. In the following, the access characteristics of workloads are first studied. Then, based on the observations from the workload analysis, we proposed several approaches for performance and lifetime improvement.

4.1 Access Characteristics

This subsection first presents the study on the access characteristics of several workloads. Then, we present a high-accuracy access characteristic identification scheme.

4.1.1 Access Characteristics of Workloads

Several workloads are studied in respect of read and write requests. Figure 6 presents the statistical results for the workloads analyzed in our study. We collected access characteristics for all data pages at the host system and distinguished among three types of data accesses:

1) Read-only: If almost all the accesses (>95%) to a data page are read requests, we characterize this page as read-only. This is typical when accessing media files or other read-only files;
2) Write-only: If almost all the accesses (>95%) to a data page are write requests, we characterize this page as write-only. This is typical in log files and periodical data flushes from the memory system to storage for consistency maintenance;
3) Interleaved-access: If the accesses to a data page are interleaved with read and write requests, we characterize this page as interleaved-access.

Figure 6 shows the request distributions for these logical data pages with different access characteristics. The read requests will either access read-only pages or interleaved-access pages, and Figure 6(a) shows the distribution of read requests accessing these two type of pages. Similarly, the write requests will either access write-only pages or interleaved-access pages, and Figure 6(b) shows the distribution of write requests. For each trace, the sum of the read from Figure 6(a) and the write requests from 6(b) is near 100% (more than 99%). Three observations can be made from Figure 6:

- Observation 1 – Most read requests access read-only pages, more than 87% on average;
- Observation 2 – Most write requests access write-only pages, more than 92% on average;
- Observation 3 – Only a small part of all requests access interleaved-access pages.

These observations will guide the design of the performance and lifetime improvement approaches in this paper. In the following, a simple, yet accurate access characteristic identification method is presented.

4.1.2 Access Characteristic Identification

We identify the access characteristic of a data page based on its most recent requests and the upcoming request which are recorded using a history window. If the most recent requests and the upcoming request are all read requests/write requests, this page is characterized as read-only/write-only.

Otherwise, this page is characterized as interleaved-access. Essentially, the identification method looks for consecutive reads or writes to characterize the page.

Figure 7 shows an example of the identification method. The first line shows the access lists. Case 1 has read requests only, case 2 has write requests only, and case 3 has interleaved read and write requests. The length of the history window is set to 2 in the examples. Thus it includes the upcoming request (indicated by the arrow above the window) and the most recent request to this page. The arrows below the access lists represent the identified access type of the page. The identification is updated upon the arrival of each access to the page. If the access characteristic of the page is indeed read-only or write-only, the identification is accurate and stable, as shown in Figure 7(a) and 7(b). However, if the access characteristic of the page is interleaved, such as in Figure 7(c), the identification is unstable and may change from time to time.

This simple identification method works well, because, as observed above, most requests will access either read-only pages or write-only pages. The accuracy of the identification method will be further evaluated in Section 5, where we analyze the effects of the history window size.

4.2 Read and Write Regulation

This subsection exploits the access characteristics studied above for performance and lifetime improvement. Figure 8 shows the outline of the proposed approach. First, the approach for performance improvement by exploiting the tradeoff between read cost and write cost is presented. Second, the approach for lifetime improvement by exploiting the tradeoff between read cost and effective wearing is presented. Finally, we combine these two approaches for both performance and lifetime improvement.

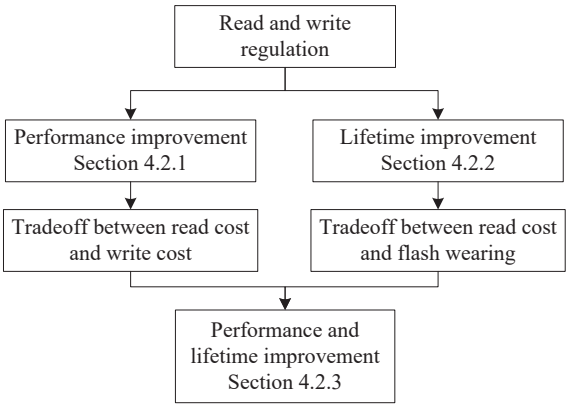


Fig. 8. The outline of the proposed approach.

4.2.1 Read and Write Cost Regulation for Performance Improvement

Based on the read and write cost model, the basic idea for performance improvement is to apply low-cost writes for write-only pages, and low-cost reads for read-only pages to improve access performance. This section does not consider the lifetime impact, and thus the maximum threshold voltage is constant. In the following, regulation for read and write cost is presented.

Read cost regulation. The read cost is determined by the cost of the last write operation performed on the data page. A low-cost read can only be processed if the page was written by a high-cost write. Thus, to ensure a low-cost read for read-only pages, a high-cost re-write operation will be done during idle time, if the page was not written with high cost previously. As shown in Figure 7, there are three cases of read cost regulation. Upon arrival of a read request, the page is characterized, and its cost is regulated as follows:

- For a *read-only* page with a low-cost read, it is the expected case and nothing should be done;
- For a *read-only* page with a high-cost read, a high-cost re-write operation is inserted to the re-write queue and performed during idle time to reduce the cost of upcoming reads (denoted with **H** in Figure 7(c));
- For an *interleaved-access* page, the read cost is not regulated.

The read performance improvement mainly comes from low-cost read for read-only pages, which mostly benefits from the high-cost write operation initially. For example, a media file is typical read-only data and will be read repeatedly after being written. The re-write operation will occur only on the situation a read-only page is updated, which seldom happens. Therefore, the re-write ratio of the proposed method should be very low, which will be further evaluated in the experiments.

Write cost regulation. The cost of a write will be regulated by adjusting the program step size in ISPP when it is issued. As shown in Figure 7, there are two cases for write cost regulation. Upon arrival of a write request, the page is characterized and the cost of the write request is regulated as follows:

- A *write-only* page will be written with a low-cost write to improve write performance which will rarely influence read performance;
- An *interleaved-access* page, as in Figure 7(c), will be written with a medium-cost write to balance read and write performance.

When a page is written for the first time, there is no history for access characteristic identification. In this case, a high-cost write is performed, which can benefit the following read requests especially when the page is a read-only page. However, if it is a write-only page, the write performance will be impacted only for this first time.

In this paper, we only regulate the access cost of read-only and write-only pages, while the interleaved-access pages are not regulated. It comes from the consideration for balance of read and write costs. The observation made in Section 4.1.1 shows that only a small part of requests access interleaved-access pages. Thus, the read and write performance will be rarely influenced by the write cost regulation or read cost regulation, which guarantees the performance improvement of the proposed approach.

4.2.2 Threshold Voltage Regulation for Lifetime Improvement

In this subsection, the access characteristics are exploited to reduce flash wearing for lifetime improvement. Two write types are used in the lifetime improvement scheme:

regular write with normal threshold voltage distribution and reduced wearing write with reduced threshold voltage. Note that the write costs for the two write types are the same based on the wearing model in Section 3.2. Data with regular write has much smaller RBER than the data with reduced wearing write. In this case, the data with reduced wearing write will have a high read cost based on the cost model presented in Section 3.1.

As read operations will not cause wearing to flash cells and the voltage of write operations determines flash wearing, we change the threshold voltage during the issuing of the upcoming write request after the identification of access characteristics. We define two write types: regular write with normal threshold voltage distribution and low-wearing write with reduced threshold voltage distribution. Note that the write cost remains the same (medium-cost write) for these two types. Data being written by regular write type will be read with medium cost, while data written by low-wearing write type has higher RBER, which needs higher cost to read. Upon arrival of a write request, the page is characterized and the the write operation is regulated as follows:

- For an *interleaved-access* page, regular write type is applied;
- For a *write-only* page, write operation with reduced wearing is applied to reduce the maximum threshold voltage.

Based on the access characteristic that most write requests access write-only data pages, effective wearing to NAND flash will be significantly reduced.

Figure 7 shows an example on the lifetime improvement process. At the bottom of the figure, if the data is identified as write-only data, it is programmed with a reduced wearing write operation (M_R). Otherwise, it is programmed by regular write operation. If the data written with reduced wearing write operation is read, a high-cost read is required (H). However, considering the access characteristics presented above, we believe that this case rarely happens.

4.2.3 Performance and Lifetime Improvement

The above two approaches benefit access performance and lifetime of NAND flash memory respectively. A write request on write-only data will be processed either by low-cost write with regular wearing for write performance improvement or medium-cost write with reduced wearing for lifetime improvement. However, the demands for optimizing both aspects cannot be satisfied. This subsection presents an approach to improve both performance and lifetime. The basic idea is to selectively apply these two types of writes for the write requests on write-only data based on the system status. If the system is busy, a low-cost write is a better choice. Otherwise, a medium-cost write with reduced wearing is selected. In the following, we discuss the details for the above idea.

First, upon arrival of a write request on write-only data, the write request is processed as follows:

- If there is no write request in the request queue, medium-cost write with reduced wearing is applied for wearing reduction. In this case, the lifetime will be improved without causing additional queueing time;
- If there are requests appending in the queue, low-cost write with regular wearing is applied for write performance improvement. In this case, the queueing time for appending write requests will be relieved, which leads to better performance.

Note that if a read request accesses the write-only data, high-cost read is required. Based on the observation presented in Section 4.1.1, this case rarely happens.

Figure 9 presents the write cost and write voltage regulation for a write request. Medium-cost with regular wearing is applied for the write requests of interleaved-access data. The cost and write voltage of write requests on write-only data are selected. The benefits of this approach will vary for different applications or different periods. It satisfies the more urgent requirement of flash memory, i.e., performance or lifetime.

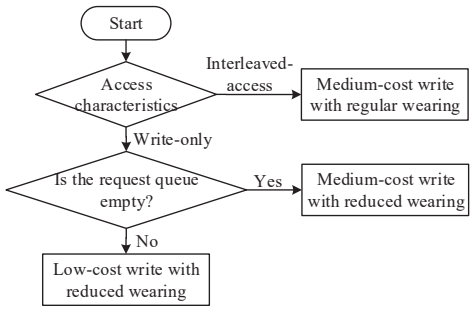


Fig. 9. Write cost regulation and write voltage regulation for performance and lifetime improvement.

4.3 Implementation and Overheads

Figure 10 shows the implementation of the proposed approach in the flash memory controller. Three components are added: Access Characteristic Identification, Access Regulator and Re-Write Queue. In addition, each mapping entry in the FTL is extended with two fields, as shown in Figure 11. The first is the access history, and the second is a 1-bit high-cost write tag. When the high-cost write tag is set for a read-only page, this indicates that the page is written with high-cost, which can be accessed with low-cost read and doesn't need to be re-written.

When an I/O request is issued, the page is characterized. First, for a read request accessing read-only data, high-cost write tag is checked. If it is unset, the logical address of the data page is added to the re-write queue. During idle time, re-write operations are triggered to program the data in the queue with a high cost, which guarantees low-cost reads on these read-only pages. After re-write operation, the high-cost write tag is set. The re-write overhead is evaluated in the experiments. Second, for a write request accessing write-only data, low-cost write with regular wearing will be performed when there are requests waiting in the queue, and medium-cost write with reduced wearing will be performed when there are no queueing requests. Third, for the read and write requests accessing interleaved-access data, medium-cost operations with regular wearing will be performed.

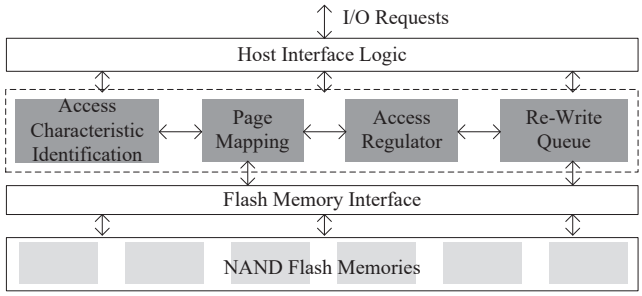


Fig. 10. The implementation of the proposed approach.

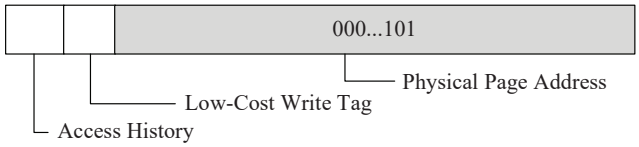


Fig. 11. An example on the FTL mapping entry.

This implementation incurs three types of overheads: storage, hardware and firmware overhead. The storage overhead includes, for each page, a number of bits for access history and one bit for the high-cost write. Assuming access history is set to one bit, which records only the most recent request to the page, the storage overhead is 16Mb for a 64GB flash memory with 4KB pages. The hardware overhead comes from the multiple voltage thresholds needed to support the reads and writes with different costs. Since the flash controller already supports multiple levels of read latency for different errors, the only change is using more than one programming step size ΔV_{pp}, which has been studied and verified in previous work [9] [20] [21] [39] [40]. To support the function, the device needs to provide multiple program voltages, each of which corresponds to one programming step size ΔV_{pp}. This overhead is negligible according to previous studies [9] [20] [21] [39] [40]. The firmware overhead includes the processes involved in access characterization, read and write regulation and rewrite queue. The overhead of these simple processes is negligible. The proposed approach does not introduce reliability issues thanks to the constraints in the read and write cost model and the constraints in wearing model. In addition, the energy consumption of the additional components is negligible.

5 Evaluation

This section first presents the experimental setup. Then, experiment results are presented, followed by the analysis of the performance and lifetime improvement.

5.1 Experimental Setup

To verify the effectiveness of the proposed technique, we have implemented the proposed scheme on a trace-driven simulator SSDSim [41] and conducted comprehensive experiments. Ten workloads from MSR [38] and four workloads collected on flash memory based systems are used for evaluation. Two of the four workloads are collected on Windows system with NTFS file system and the other two are collected on Linux system with EXT4 file system. MUS_W and MV_W are collected on Windows when playing online music and offline movie respectively. MUS_L and MV_L are collected on Linux when playing online music and offline movie respectively. From our studies, we found that all workloads have similar access characteristics. The SSD in the simulation has eight channels, with four chips per channel and four planes per chip. Each chip has 2048 blocks and each block has 64 4KB pages. Default page mapping based FTL, greedy garbage collection, and wear leveling are implemented in the simulator, representing state-of-the-art storage systems [42]. For fair comparison, we use the parameters for the cost regulator from previous work [12] [14] [20]. Table 1 shows the access costs and corresponding effective wearing. The cost of the erase operation is constant.

TABLE 1
The access cost and effective wearing configurations.

Read Cost (μs)	Low	Medium	High	High
	70	**170**	**310**	**310**
Write Cost (μs)	High	Medium	Medium	Low
	800	**600**	**600**	**450**
Effective Wearing	Regular	Regular	Reduced	Regular
	1	1	0.8	1

One of the most important parameters is the size of history window in the access characteristic identification method. It affects the storage overhead as well as the identification accuracy. Figure 12 shows the identification accuracy for three window sizes. The result shows that the identification approach can achieve great accuracy and a larger window results in a higher accuracy. However, the

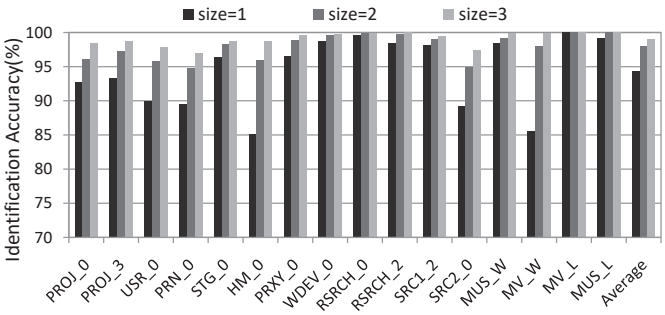


Fig. 12. History window size impact on identification accuracy.

increase in accuracy decreases as the size increases. In the figure, the increase from the first column to the second column is significant. While the increase of accuracy from the second column to the third column is reduced. In the following experiments, the size of history window is set as 2, containing the most recent request and the upcoming request (the second column in Figure 12) for identification to trade-off the storage overhead and identification accuracy.

5.2 Experimental Results

In this section, the results of five schemes are compared: Traditional, Li et al. [12], R+W, R+L and R+W+L. Traditional scheme is the traditional approach without cost regulation

and all requests are performed with medium cost and regular wearing. Li et al. applied low-cost reads or writes when there is queueing delay, where the flash wearing is constant as regular. The other three schemes are proposed in this work. Read and write performance improvement (R+W) represents the approach in Section 4.2.1, where read and write costs are regulated with constant flash wearing. Read performance and lifetime improvement (R+L) represents the approach in Section 4.2.2, where the read requests of read-only data are accessed with low cost and write requests of write-only data are processed with medium-cost and regular-wearing write. Read, write performance and lifetime improvement (R+W+L) represents the approach in Section 4.2.3, where the read requests of read-only data are accessed with low cost and write requests of write-only data are processed by one of the two write types. We first discuss the performance improvement of the proposed approaches, with analysis of the overhead of re-write operations. Second, the flash wearing is evaluated. Finally, the sensitivity study is presented.

5.2.1 Performance Improvement

Figures 13(a) and 13(b) present the normalized read and write latency. As shown in Figure 13(a), read performance is improved in all three proposed schemes, about 48% over Li et al. on average. This indicates that the read performance is hardly impacted by the processing of write requests of write-only data. As shown in Figure 13(b), the proposed schemes achieve better performance over Li et al. R+W achieves the best write performance, 20% on average, R+L has no write performance benefit since the write requests are explored to improve lifetime, and R+W+L achieves a trade-off between the two schemes. In conclusion, the proposed schemes can improve access performance significantly over the state-of-the-art work.

To understand the performance variations, Figure 14 shows the distributions of operations of different costs, including low-cost, medium-cost and high-cost reads and writes, and medium-cost write with regular wearing and reduced wearing. The operations in the figure refer to the read and write access on the data pages. In the traditional case, all reads and writes are performed with medium cost and regular wearing, while Li et al. applied low-cost reads or writes when there is queueing delay, which are also performed with regular wearing. In the proposed schemes, the access costs are regulated based on the access characteristics identified and write types are decided based on the system status. If there is no history for the accessed page, a high-cost write is used for write requests and a low-cost read is used for read requests. The distribution in the figure matches the read and write performance in Figure 13. More low-cost reads and writes means better read and write performance. Comparing to Li et al.'s work, the proposed schemes issue considerably more low-cost reads and writes, thus improve read and write performance significantly.

Figure 15 presents the overall performance. Compared to Li et al., the proposed approach achieves the best overall performance improvement. Among three approaches proposed in this work, R+W has the best performance, 15% over Li's work on average, and R+L has the worst. This is because R+W is used to improve read and write performance while R+L is applied to improve read performance and flash lifetime.

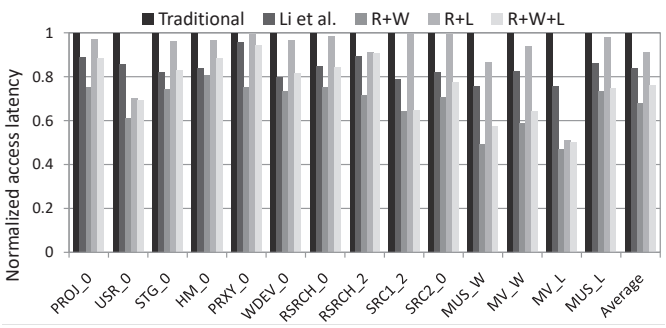


Fig. 15. Overall performance comparison.

Re-Write Overhead: Figure 16 shows the ratio of re-write operations of Li et al.'s work and the proposed approaches, which equals to the number of host I/O divided by the number of re-write operations. Since the re-write ration of the three schemes proposed in this work have the same policy for read requests, where the re-write operations are introduced, only R+W+L is compared in the figure. From the results, the proposed approach introduces negligible re-write operations, no more than 1%, while the state-of-the-art work is much higher. The result benefits from the strategy to apply high-cost write for the data pages that are first written to flash, which ensures most of the read-only pages will be read with low cost. The other reason for the low re-write ratio is that the requests for interleaved-access pages only constitute a small portion of total requests. Most read operations occur on read-only pages. The percentage of re-write operations presents the measures to avoid bad read performance on read-only pages. However, the re-write operations will influence the processing of host requests, especially write requests. In summary, more re-write operations will benefit read performance within a specific range, but too many will worsen the write performance.

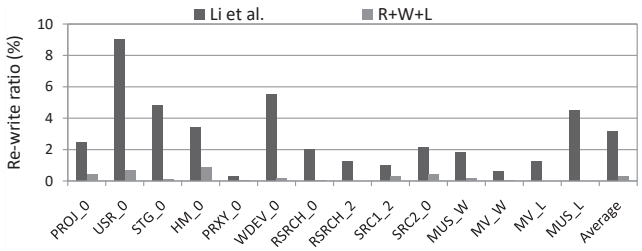
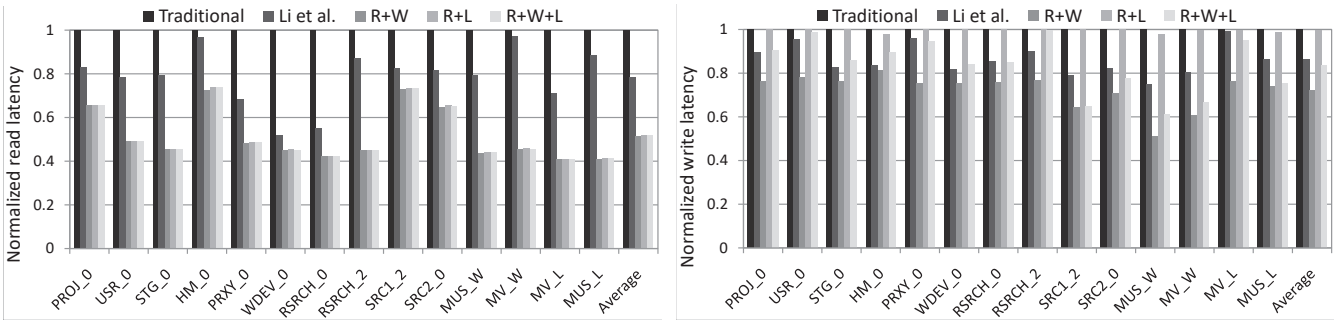


Fig. 16. Re-write ratio comparison of the state-of-the-art work and this work.

5.2.2 Lifetime Improvement

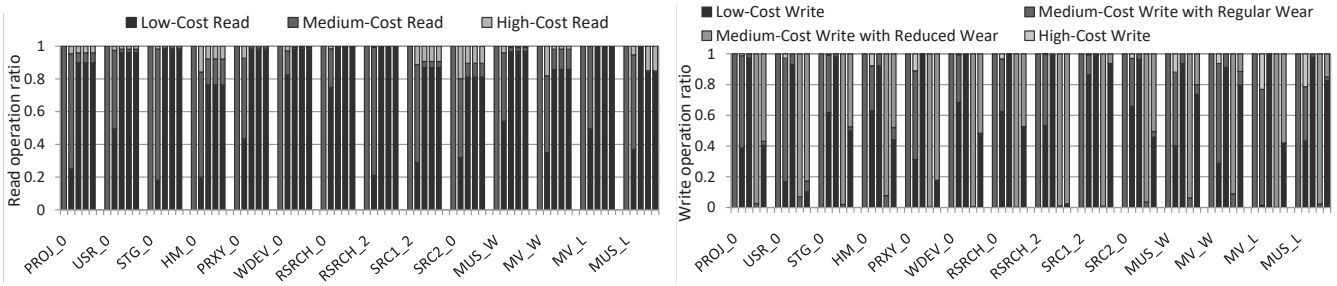
Figure 17 shows the comparison of the effective wearing for NAND flash memory. The traditional approach, Li et al., and R+W have no wearing reductions. R+L can achieve near 20% lifetime improvement since all the write requests of write-only data are processed with reduced wearing. The wearing reduction of R+W+L corresponds to the write performance improvement in Figure 13(b). For some traces, like SRC1_2, the write performance improvement is significant,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2018.2839671, IEEE Transactions on Computers



(a) I/O latency for read requests (b) I/O latency for write requests

Fig. 13. Normalized read and write latency compared with traditional case and Li et al. [12].



(a) Operation Ratios for read requests (b) Operation Ratios for read requests

Fig. 14. Distributions of different cost operations. For each trace, the five bars from the left to the right represent Traditional, Li et al., R+W, R+L and R+W+L respectively.

and the wearing reduction is small. This is because these traces are intensive and most of the time there are requests waiting in the queue. On the other hand, for some traces, such as USR_0 and RSRCH_2, the write performance improvement is small, and the wearing reduction is significant, near 20%. This is because more write requests are performed with reduced-wearing writes for lifetime improvement.

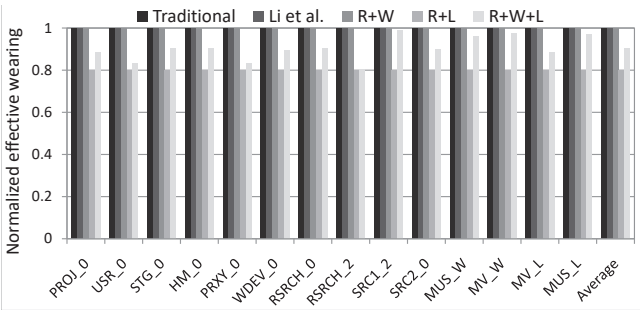


Fig. 17. Comparison of effective wearing for NAND flash memory.

5.2.3 Performance and Lifetime Improvement

In this section, a new metric WPLI is introduced to combine write performance improvement and lifetime improvement into a single number for comparison purpose.

$$WPLI = w * (WP_T - WP_P)/WP_T + (1-w) * (EW_T - EW_P)/EW_T \quad (4)$$

where WP_T and WP_P are the write latency of the traditional approach and the proposed approach respectively, and EW_T and EW_P are the effective wearing of the traditional approach and the proposed approach respectively. w ($0 < w < 1$) is a weight to specify the relative importance between the write performance improvement and the lifetime improvement. A larger WPLI means a greater improvement based on a specific w.

Figure 18 shows the comparison of WPLI when w equals to 0.2 and 0.8. Two conclusions can be made from the results. First, R+L achieves the highest WPLI among all the approaches when w is 0.2, and R+W achieves the highest WPLI when w is 0.8. This is consistent with the approaches since they are designed for lifetime improvement and write performance improvement respectively. Second, WPLI has a higher average when w is 0.8, which indicates that the write performance improvement is more significant than lifetime improvement with the proposed approach.

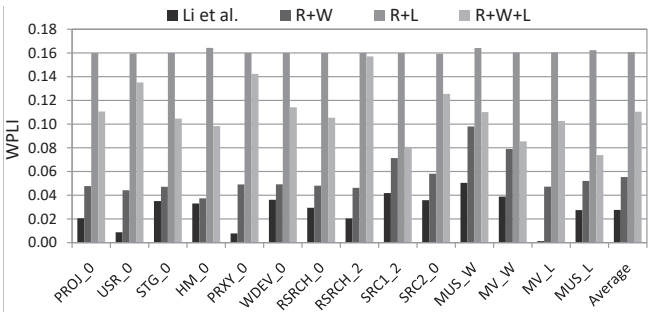
5.3 Sensitivity Study

5.3.1 History Window Size

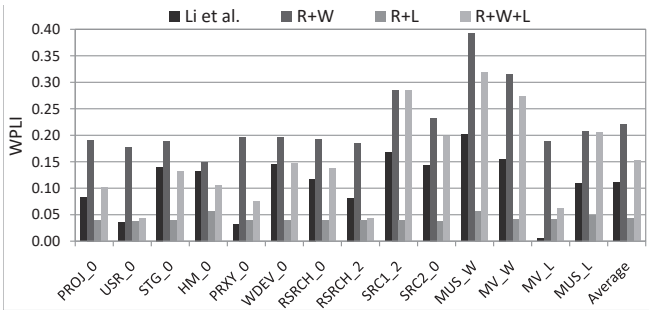
This section evaluates the sensitiveness of the proposed approaches by varying the size of history window, from 1 to 3. Size equaling to 1 means that the history window only contains the upcoming request, 2 means that the history window contains the upcoming request and one most recent accessing request, and 3 means that the history window contains the upcoming request and two most recent accessing requests. Read and write performance together with effective wearing is evaluated.

Figure 19 shows the normalized read and write latency and Figure 20 shows the normalized effective flash wearing. Two observations can be made from the results. First, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2018.2839671, IEEE Transactions on Computers

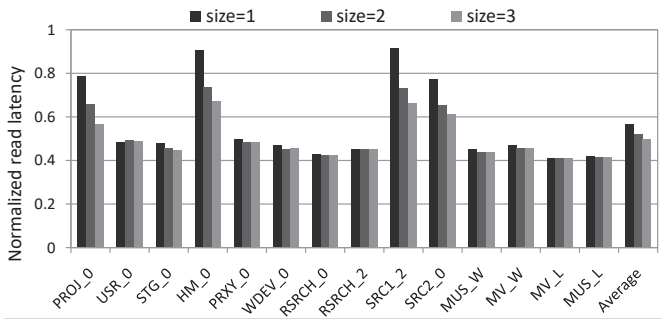


(a) WPLI (w = 0.2)

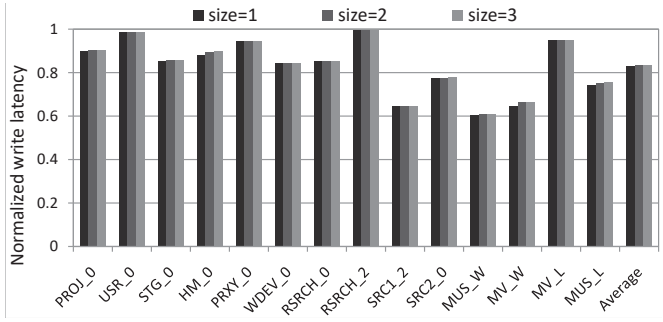


(b) WPLI (w = 0.8)

Fig. 18. Write performance and lifetime improvement (WPLI) comparison.



(a) Normalized read latency



(b) Normalized write latency

Fig. 19. Sensitivity study of history window size on performance.

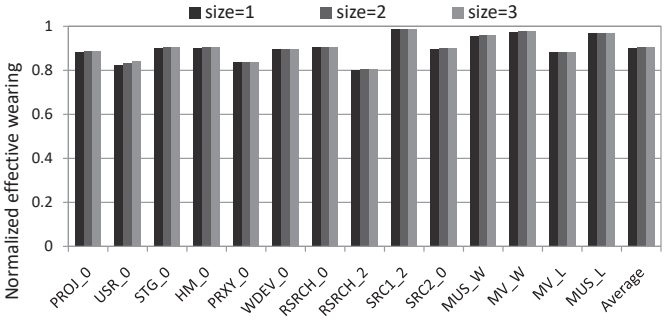


Fig. 20. Sensitivity study of history window size on lifetime.

read performance increases with the size of history window. This is because smaller window size will lead to more re-write operations, which can affect the foreground access performance, especially when the size equals to 1. In addition, too many data waiting for idle time to be re-written can delay the re-write of real read-only data, thus affecting the read performance. Second, the size of history window has little influence on the write performance and lifetime improvement.

5.3.2 Degree of Parallelism

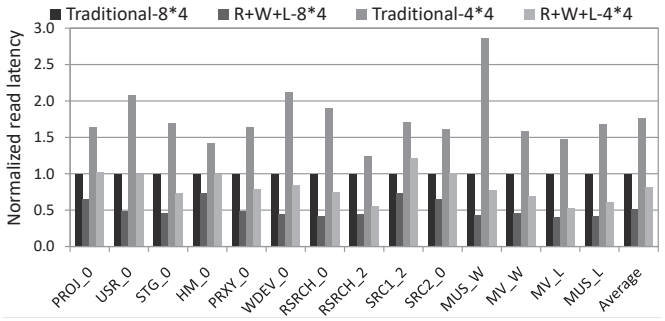
This section evaluates the proposed approaches by varying the degree of parallelism, from 8 channels to 4 channels, where each channel has 4 chips. Figure 21 shows the normalized read and write latency and Figure 22 shows and the effective flash wearing, respectively. In both figures, the results are normalized to the results of the traditional approach with 8 channels. Two observations can be made from the results. First, when the degree of parallelism is reduced, the performance becomes worse, but the improvement from the proposed approach increases. Second, the lifetime improvement is smaller with the reduction of parallelism. The reason is that with the reduction of the degree of parallelism, there will be more requests waiting to be issued, which leads to the result that more write requests on write-only data will be issued with low-cost writes with normal wearing.

6 Conclusions

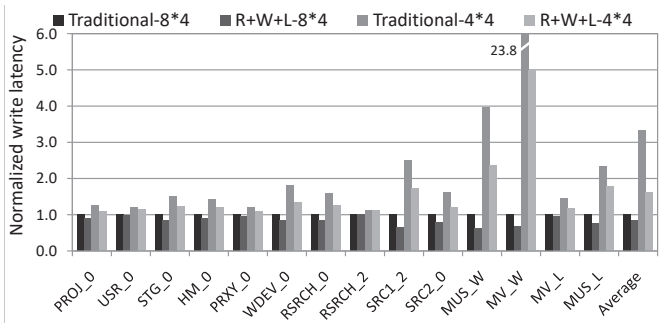
This paper conducted a study on the access characteristics of flash memory, which reveals that most read (write) requests access read-only (write-only) data. These access characteristics are exploited for performance and lifetime improvement. For performance improvement, the tradeoff between read and write time costs is combined with the access characteristics, where read requests of read-only data are processed with low-cost read and write requests of write-only data are processed with low-cost write. For lifetime improvement, the tradeoff between the read cost and effective wearing is combined with the access characteristics, where the write requests of write-only data are performed with reduced wearing. To achieve both performance and lifetime improvement, a strategy is proposed to switch the write mode based on the system status. Simulation results show that the proposed approaches are effective in reducing I/O latency and reducing effective wearing on flash memory.

Acknowledgements

This work is supported by the NSFC (61772092 and 61572411). The first author, Qiao Li, of the paper finished part of the work during her master period at Chongqing University. She is now a first year PhD candidate at City University of Hong Kong.



(a) Normalized read latency



(b) Normalized write latency

Fig. 21. Sensitivity study of degree of parallelism on performance.

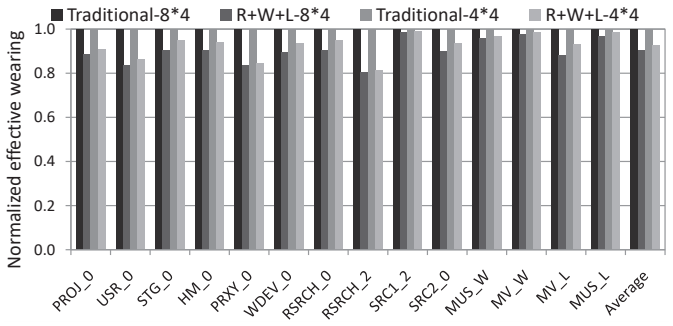


Fig. 22. Sensitivity study of degree of parallelism on lifetime.

REFERENCES

[1] Q. Li, L. Shi, C. J. Xue, K. Wu, C. Ji, Q. Zhuge, and E. H.-M. Sha, "Access characteristic guided read and write cost regulation for performance improvement on flash memory." in *FAST*, 2016, pp. 125–132.

[2] C. Lee, D. Sim, J. Hwang, and S. Cho, "F2FS: A new file system for flash storage," in *Proceedings of the USENIX Conference on File and Storage Technologies*, 2015, pp. 273–286.

[3] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. S. Manasse, and R. Panigrahy, "Design tradeoffs for SSD performance," in *Proceedings of USENIX Annual Technical Conference*, 2008, pp. 226–229.

[4] Y. Cai, S. Ghose, Y. Luo, K. Mai, O. Mutlu, and E. F. Haratsch, "Vulnerabilities in mlc nand flash memory programming: experimental analysis, exploits, and mitigation techniques," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2017, pp. 49–60.

[5] L. M. Grupp, J. D. Davis, and S. Swanson, "The bleak future of nand flash memory," in *Proceedings of the 10th USENIX conference on File and Storage Technologies*. USENIX Association, 2012, pp. 2–2.

[6] X. Jimenez, D. Novo, and P. Ienne, "Phoenix: reviving MLC blocks as SLC to extend NAND flash devices lifetime," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2013, pp. 226–229.

[7] K. Zhao, W. Zhao, H. Sun, T. Zhang, X. Zhang, and N. Zheng, "LDPC-in-SSD: Making advanced error correction codes work effectively in solid state drives," in *Proceedings of the USENIX Conference on File and Storage Technologies*, 2013, pp. 244–256.

[8] 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, J.-H. Choi, J.-R. Kim, and H.-K. Lim, "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.

[9] Y. Pan, G. Dong, and T. Zhang, "Exploiting memory device wear-out dynamics to improve NAND flash memory system performance," in *Proceedings of the USENIX conference on File and Storage Technologies*, 2011, pp. 1–14.

[10] 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.

[11] K.-C. Ho, P.-C. Fang, H.-P. Li, C.-Y. Wang, and H.-C. Chang, "A 45nm 6b/cell charge-trapping flash memory using LDPC-based ECC and drift-immune soft-sensing engine," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 222–223.

[12] Q. Li, L. Shi, C. Gao, K. Wu, C. J. Xue, Q. Zhuge, and E. H.-M. Sha, "Maximizing IO performance via conflict reduction for flash memory storage systems," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 904–907.

[13] G. Wu, X. He, N. Xie, and T. Zhang, "Exploiting workload dynamics to improve ssd read latency via differentiated error correction codes," *ACM Transactions on Design Automation of Electronic Systems*, vol. 18, no. 4, pp. 55:1–55:22, 2013.

[14] J. Jeong, S. S. Hahn, S. Lee, and J. Kim, "Improving nand endurance by dynamic program and erase scaling." in *HotStorage*, 2013.

[15] L. Shi, K. Wu, M. Zhao, C. J. Xue, and H. Edwin, "Retention trimming for wear reduction of flash memory storage systems," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2014, pp. 1–6.

[16] 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 *FAST*, 2014, pp. 61–74.

[17] B. Peleato and R. Agarwal, "Maximizing mlc nand lifetime and reliability in the presence of write noise," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 3752–3756.

[18] S. E. C. Ltd., "High-Performance 128-gigabit 3-bit Multi-level-cell NAND Flash Memory," https://memorylink.samsung.com/.../detail.do?newsId=12761, 2013.

[19] 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." MSST, 2017.

[20] Y. Pan, G. Dong, Q. Wu, and T. Zhang, "Quasi-nonvolatile SSD: Trading flash memory nonvolatility to improve storage system performance for enterprise applications," in *Proceedings of the High Performance Computer Architecture*, 2012, pp. 1–10.

[21] R.-S. Liu, C.-L. Yang, and W. Wu, "Optimizing NAND flash-based SSDs via retention relaxation," in *Proceedings of the USENIX conference on File and Storage Technologies*, 2012, pp. 1–11.

[22] H. Park, J. Kim, J. Choi, D. Lee, and S. H. Noh, "Incremental redundancy to reduce data retention errors in flash-based SSDs," in *Proceedings of Mass Storage Systems and Technologies*, 2015, pp. 1–13.

[23] M. Zhang, F. Wu, X. He, P. Huang, S. Wang, and C. Xie, "Real: A retention error aware ldpc decoding scheme to improve nand flash read performance," in *2016 32nd Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 2016, pp. 1–13.

[24] Q. Li, L. Shi, C. J. Xue, Q. Zhuge, and E. H.-M. Sha, "Improving ldpc performance via asymmetric sensing level placement on flash memory," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2017, pp. 560–565.

[25] D. Park and D. H. Du, "Hot data identification for flash-based storage systems using multiple bloom filters," in *IEEE Symposium on Mass Storage Systems and Technologies*, 2011, pp. 1–11.

[26] G. Wu, B. Eckart, and X. He, "BPAC: An adaptive write buffer management scheme for flash-based Solid State Drives," in *IEEE Symposium on Mass Storage Systems and Technologies*, 2010, pp. 1–6.

[27] L. Shi, J. Li, C. J. Xue, C. Yang, and X. Zhou, "ExLRU: a unified write buffer cache management for flash memory," in *Proceedings*

of ACM international conference on Embedded software, 2011, pp. 339–348.
[28] H. Kim and S. Ahn, "BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage," in *Proceedings of the USENIX Conference on File and Storage Technologies*, 2008, pp. 1–14.
[29] S. Lee, D. Shin, Y.-J. Kim, and J. Kim, "LAST: locality-aware sector translation for NAND flash memory-based storage systems," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 6, pp. 36–42, 2008.
[30] Y. Pan, G. Dong, and T. Zhang, "Error rate-based wear-leveling for nand flash memory at highly scaled technology nodes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 7, pp. 1350–1354, 2013.
[31] Y.-J. Woo and J.-S. Kim, "Diversifying wear index for mlc nand flash memory to extend the lifetime of ssds," in *2013 Proceedings of the International Conference on Embedded Software (EMSOFT)*. IEEE, 2013, pp. 1–10.
[32] Y. Cai, G. Yalcin, O. Mutlu, E. F. Haratsch, A. Cristal, O. S. Unsal, and K. Mai, "Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime," in *2012 IEEE 30th International Conference on Computer Design (ICCD)*. IEEE, 2012, pp. 94–101.
[33] S. Lee, T. Kim, K. Kim, and J. Kim, "Lifetime management of flash-based ssds using recovery-aware dynamic throttling." in *FAST*, 2012, p. 26.
[34] L. Tang, Q. Huang, W. Lloyd, S. Kumar, and K. Li, "Ripq: Advanced photo caching on flash for facebook." in *Proceedings of the 13th USENIX conference on File and Storage Technologies*, 2015, pp. 373–386.
[35] Y. Cheng, F. Douglis, P. Shilane, G. Wallace, P. Desnoyers, and K. Li, "Erasing belady's limitations: In search of flash cache offline optimality." in *USENIX Annual Technical Conference*, 2016, pp. 379–392.
[36] C. Li, P. Shilane, F. Douglis, and G. Wallace, "Pannier: Design and analysis of a container-based flash cache for compound objects," *ACM Transactions on Storage (TOS)*, vol. 13, no. 3, p. 24, 2017.
[37] K. Zhao, K. S. Venkataraman, X. Zhang, J. Li, N. Zheng, and T. Zhang, "Over-clocked SSD: Safely running beyond flash memory chip I/O clock specs," in *Proceedings of High Performance Computer Architecture*, 2014, pp. 536–545.
[38] D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, and A. Rowstron, "Migrating server storage to SSDs: analysis of tradeoffs," in *Proceedings of ACM European conference on Computer systems*, 2009, pp. 145–158.
[39] S. Lee and J. Kim, "Effective lifetime-aware dynamic throttling for nand flash-based ssds," *IEEE Transactions on Computers*, vol. 65, no. 4, pp. 1075–1089, 2016.
[40] J. Jeong, Y. Song, S. S. Hahn, S. Lee, and J. Kim, "Dynamic erase voltage and time scaling for extending lifetime of nand flash-based ssds," *IEEE Transactions on Computers*, vol. 66, no. 4, pp. 616–630, 2017.
[41] 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.
[42] Y. Hu, H. Jiang, D. Feng, L. Tian, H. Luo, and S. Zhang, "Performance impact and interplay of SSD parallelism through advanced commands, allocation strategy and data granularity," in *Proceedings of the international conference on Supercomputing*, 2011, pp. 96–107.

Qiao Li received the B.S. and M.S. degrees in College of Computer Science from Chongqing University in China in 2014 and 2017 respectively. She is now a first year PhD candidate in Department of Computer Science, City University of Hong Kong. Her research interests include NAND flash memory, embedded systems and computer architecture.

Liang Shi received the B.S. degrees in Computer Science from Xi'an University of Post & Telecommunication, Xi'an, Shanxi, China, in July, 2008, Ph.D. degree from University of Science and Technology of China, Hefei, China, in June, 2013. He is now an associate professor in the College of Computer Science at the Chongqing University. His research interests include flash memory, embedded systems, and emerging non-volatile memory technologies.

Congming Gao received the B.S. degree in College of Computer Science from Chongqing University, Chongqing, China, in 2014. He is now a Ph.D candidate in the College of Computer Science, Chongqing University. His research interests include embedded and real-time systems, nonvolatile memory, and architecture optimizations.

Yejia Di received BS degree in College of Computer Science from Chongqing University in China in 2014. She is now a Ph.D candidate in the College of Computer Science, Chongqing University. Her research interests include embedded and real-time systems, flash memory and system optimizations.

Chun Jason Xue received the B.S. degree in computer science and engineering from the University of Texas at Arlington, Arlington, TX, USA, in 1997, and the M.S. and Ph.D. degrees in computer science from the University of Texas at Dallas, Richardson, TX, USA, in 2002 and 2007, respectively. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include memory and parallelism optimization for embedded systems, software/hardware co-design, real-time systems, and computer security.